kubernetes-python-client Documentation

Kubernetes

May 12, 2018

Contents

Contents:

CHAPTER 1

Kubernetes Python Client

Python client for the kubernetes API.

1.1 Installation

From source:

```
git clone --recursive https://github.com/kubernetes-client/python.git
cd python
python setup.py install
```

From PyPi directly:

```
pip install kubernetes
```

1.2 Example

List all pods:

```
from kubernetes import client, config

# Configs can be set in Configuration class directly or using helper utility
config.load_kube_config()

v1 = client.CoreV1Api()
print("Listing pods with their IPs:")
ret = v1.list_pod_for_all_namespaces(watch=False)
for i in ret.items:
    print("%s\t%s\t%s" % (i.status.pod_ip, i.metadata.namespace, i.metadata.name))
```

Watch on namespace object:

```
from kubernetes import client, config, watch

# Configs can be set in Configuration class directly or using helper utility
config.load_kube_config()

v1 = client.CoreV1Api()
count = 10
w = watch.Watch()
for event in w.stream(v1.list_namespace, _request_timeout=60):
    print("Event: %s %s" % (event['type'], event['object'].metadata.name))
    count -= 1
    if not count:
        w.stop()

print("Ended.")
```

More examples can be found in examples folder. To run examples, run this command:

(replace example1 with the example base filename)

1.3 Documentation

All APIs and Models' documentation can be found at the Generated client's README file.

1.4 Compatibility

`client-python` follows semver, so until the major version of client-python gets increased, your code will continue to work with explicitly supported versions of Kubernetes clusters.

Table 1: Compatibility Matrix

	Kubernetes 1.4	Kubernetes 1.5	Kubernetes 1.6	Kubernetes 1.7	Kubernetes 1.8	Kubernetes 1.9	Kubernetes 1.10
client-python 1.0	•	✓	•	•	•	•	
client-python 2.0	•	•	✓	•	•	•	
client-python 3.0	•	•	•	✓	•	•	
client-python 4.0	•	•	•	•	✓	•	
client-python 5.0	•	•	•	•	•	✓	
client-python 6.0	•	•	•	•	•	•	✓
client-python HEAD	•	•	•	•	•	•	✓

Key:

- ✓ Exactly the same features / API objects in both client-python and the Kubernetes version.
- + client-python has features or api objects that may not be present in the Kubernetes cluster, but everything they have in common will work.
- - The Kubernetes cluster has features the client-python library can't use (additional API objects, etc).

See the CHANGELOG for a detailed description of changes between client-python versions.

Client version	Canonical source for OpenAPI spec	Maintenance status
1.0 Alpha/Beta	Kubernetes main repo, 1.5 branch	
1.0.x	Kubernetes main repo, 1.5 branch	✓
2.0 Alpha/Beta	Kubernetes main repo, 1.6 branch	
2.0.x	Kubernetes main repo, 1.6 branch	✓
3.0 Alpha/Beta	Kubernetes main repo, 1.7 branch	
3.0	Kubernetes main repo, 1.7 branch	✓
4.0 Alpha/Beta	Kubernetes main repo, 1.8 branch	
4.0	Kubernetes main repo, 1.8 branch	✓
5.0 Alpha/Beta	Kubernetes main repo, 1.9 branch	
5.0	Kubernetes main repo, 1.9 branch	✓
6.0 Alpha/Beta	Kubernetes main repo, 1.10 branch	
6.0	Kubernetes main repo, 1.10 branch	✓

Key:

- ✓ Changes in main Kubernetes repo are manually should be automated) published to client-python when they are available.

- No longer maintained; please upgrade.

Note: There would be no maintenance for alpha/beta releases except the latest one.

1.5 Community, Support, Discussion

You can reach the maintainers of this project at SIG API Machinery. If you have any problem with the package or any suggestions, please file an issue.

1.5.1 Code of Conduct

Participation in the Kubernetes community is governed by the CNCF Code of Conduct.

1.6 Kubernetes Incubator

This is a **'Kubernetes Incubator project <https://github.com/kubernetes/community/blob/master/incubator.md'_**.

- **'SIG: sig-api-machinery <https://github.com/kubernetes/community/tree/master/sig-api-machinery'_**

1.7 Troubleshooting

1.7.1 SSLError on macOS

If you get an SSLError, you likely need to update your version of python. The version that ships with macOS may not be supported.

Install the latest version of python with brew:

```
brew install python
```

Once installed, you can query the version of OpenSSL like so:

```
python -c "import ssl; print ssl.OPENSSL_VERSION"
```

You'll need a version with OpenSSL version 1.0.0 or later.

1.7.2 Hostname doesn't match

If you get an `ssl.CertificateError` complaining about hostname match, your installed packages does not meet version requirements. Specifically check `ipaddress` and `urllib3` package versions to make sure they met requirements in requirements.txt file.

1.7.3 Why Exec/Attach calls doesn't work

Starting from 4.0 release, we do not support directly calling exec or attach calls. you should use stream module to call them. so instead of `resp = api.connect_get_namespaced_pod_exec(name, ...` you should call `resp = stream(api.connect_get_namespaced_pod_exec, name, ...`. See more at exec example.

CHAPTER 2

Installation

At the command line:

```
$ pip install kubernetes
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv kubernetes
$ pip install kubernetes
```

CHAPTER 3

Usage

To use kubernetes-python-client in a project:

```
import kubernetes
```

CHAPTER 4

kubernetes

4.1 kubernetes package

4.1.1 Subpackages

kubernetes.client package

Subpackages

kubernetes.client.apis package

Submodules

kubernetes.client.apis.apis_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.apis_api.**ApisApi**(*api_client=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_versions(***kwargs*)

get available API versions This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_versions(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroupList

If the method is called asynchronously, returns the request thread.

get_api_versions_with_http_info(***kwargs*)
get available API versions This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_versions_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroupList

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.apps_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.apps_api.**AppsApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.apps_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.apps_v1beta1_api.**AppsV1beta1Api**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_namespaced_controller_revision(*namespace*, *body*, ***kwargs*)

create a ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_controller_revision(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ControllerRevision body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ControllerRevision

If the method is called asynchronously, returns the request thread.

create_namespaced_controller_revision_with_http_info(*namespace*, *body*, ***kwargs*)

create a ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_controller_revision_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ControllerRevision body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ControllerRevision

If the method is called asynchronously, returns the request thread.

create_namespaced_deployment(*namespace*, *body*, ***kwargs*)

create a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_deployment(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

create_namespaced_deployment_rollback(*name*, *namespace*, *body*, ***kwargs*)

create rollback of a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_deployment_rollback(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DeploymentRollback (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1DeploymentRollback body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1DeploymentRollback

If the method is called asynchronously, returns the request thread.

create_namespaced_deployment_rollback_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

create rollback of a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_deployment_rollback_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DeploymentRollback (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1DeploymentRollback body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1DeploymentRollback

If the method is called asynchronously, returns the request thread.

create_namespaced_deployment_with_http_info(*namespace*, *body*, ***kwargs*)
create a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_deployment_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

create_namespaced_stateful_set(*namespace*, *body*, ***kwargs*)
create a StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_stateful_set(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1StatefulSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

create_namespaced_stateful_set_with_http_info(*namespace*, *body*, ***kwargs*)
create a StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_stateful_set_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1StatefulSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_controller_revision(*namespace*, ***kwargs*)
delete collection of ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_controller_revision(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field

is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_controller_revision_with_http_info(*namespace*, ***kwargs*)

delete collection of ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_controller_revision_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_deployment(*namespace*, ***kwargs*)

delete collection of Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_deployment(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_deployment_with_http_info(*namespace*, ***kwargs*)

delete collection of Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_deployment_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is

true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_stateful_set(*namespace*, ***kwargs*)

delete collection of StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_stateful_set(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of

a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_stateful_set_with_http_info(*namespace*, ***kwargs*)

delete collection of StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_stateful_set_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_controller_revision(*name*, *namespace*, *body*, ***kwargs*)

delete a ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.delete_namespaced_controller_revision(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ControllerRevision (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_controller_revision_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_controller_revision_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ControllerRevision (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_deployment(*name*, *namespace*, *body*, ***kwargs*)

delete a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_deployment(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param

str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_deployment_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_deployment_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_stateful_set(*name*, *namespace*, *body*, ***kwargs*)

delete a StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_stateful_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_stateful_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_stateful_set_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_controller_revision_for_all_namespaces(***kwargs*)
list or watch objects of kind ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_controller_revision_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects

returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1ControllerRevisionList

If the method is called asynchronously, returns the request thread.

list_controller_revision_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_controller_revision_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1ControllerRevisionList

If the method is called asynchronously, returns the request thread.

list_deployment_for_all_namespaces(***kwargs*)

list or watch objects of kind Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_deployment_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: AppsV1beta1DeploymentList

If the method is called asynchronously, returns the request thread.

list_deployment_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_deployment_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum

number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: AppsV1beta1DeploymentList

If the method is called asynchronously, returns the request thread.

list_namespaced_controller_revision(*namespace*, ***kwargs*)

list or watch objects of kind ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_controller_revision(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version

of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1ControllerRevisionList

If the method is called asynchronously, returns the request thread.

list_namespaced_controller_revision_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_controller_revision_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1ControllerRevisionList

If the method is called asynchronously, returns the request thread.

list_namespaced_deployment(*namespace*, ***kwargs*)

list or watch objects of kind Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_deployment(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: AppsV1beta1DeploymentList

If the method is called asynchronously, returns the request thread.

list_namespaced_deployment_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_deployment_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that

can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: AppsV1beta1DeploymentList

If the method is called asynchronously, returns the request thread.

list_namespaced_stateful_set(*namespace*, ***kwargs*)

list or watch objects of kind StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_stateful_set(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return

what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1StatefulSetList

If the method is called asynchronously, returns the request thread.

list_namespaced_stateful_set_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_stateful_set_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1StatefulSetList

If the method is called asynchronously, returns the request thread.

list_stateful_set_for_all_namespaces(***kwargs*)

list or watch objects of kind StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_stateful_set_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject

a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1StatefulSetList

If the method is called asynchronously, returns the request thread.

list_stateful_set_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_stateful_set_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects

returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1StatefulSetList

If the method is called asynchronously, returns the request thread.

patch_namespaced_controller_revision(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_controller_revision(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ControllerRevision (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ControllerRevision

If the method is called asynchronously, returns the request thread.

patch_namespaced_controller_revision_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_controller_revision_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ControllerRevision (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ControllerRevision

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_scale(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

patch_namespaced_stateful_set(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_stateful_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_stateful_set_scale(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_stateful_set_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_stateful_set_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_stateful_set_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_stateful_set_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_stateful_set_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_stateful_set_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_stateful_set_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_stateful_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_stateful_set_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

read_namespaced_controller_revision(*name*, *namespace*, ***kwargs*)

read the specified ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_controller_revision(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ControllerRevision (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1ControllerRevision

If the method is called asynchronously, returns the request thread.

read_namespaced_controller_revision_with_http_info(*name*, *namespace*, ***kwargs*)

read the specified ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_controller_revision_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ControllerRevision (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1ControllerRevision

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment(*name*, *namespace*, ***kwargs*)

read the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_scale(*name*, *namespace*, ***kwargs*)

read scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_scale(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_scale_with_http_info(*name*, *namespace*, ***kwargs*)

read scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_scale_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_status(*name*, *namespace*, ***kwargs*)

read status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_status_with_http_info(*name*, *namespace*, ***kwargs*)

read status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_with_http_info(*name*, *namespace*, ***kwargs*)

read the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

read_namespaced_stateful_set(*name*, *namespace*, ***kwargs*)

read the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_stateful_set(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

read_namespaced_stateful_set_scale(*name*, *namespace*, ***kwargs*)

read scale of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_stateful_set_scale(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_stateful_set_scale_with_http_info(*name*, *namespace*, ***kwargs*)
read scale of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_stateful_set_scale_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_stateful_set_status(*name*, *namespace*, ***kwargs*)
read status of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_stateful_set_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

read_namespaced_stateful_set_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_stateful_set_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

read_namespaced_stateful_set_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_stateful_set_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_controller_revision(*name*, *namespace*, *body*, ***kwargs*)
replace the specified ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_controller_revision(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ControllerRevision (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ControllerRevision

body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ControllerRevision

If the method is called asynchronously, returns the request thread.

replace_namespaced_controller_revision_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified ControllerRevision This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_controller_revision_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ControllerRevision (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ControllerRevision body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ControllerRevision

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment(*name*, *namespace*, *body*, ***kwargs*)

replace the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_scale(*name*, *namespace*, *body*, ***kwargs*)

replace scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_status(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

replace_namespaced_stateful_set(*name*, *namespace*, *body*, ***kwargs*)

replace the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_stateful_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1StatefulSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_stateful_set_scale(*name*, *namespace*, *body*, ***kwargs*)

replace scale of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_stateful_set_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_stateful_set_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace scale of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_stateful_set_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param AppsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: AppsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_stateful_set_status(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_stateful_set_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1StatefulSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_stateful_set_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_stateful_set_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1StatefulSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_stateful_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified StatefulSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_stateful_set_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StatefulSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1StatefulSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StatefulSet

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.authentication_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.authentication_api.**AuthenticationApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.authentication_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.authentication_v1beta1_api.**AuthenticationV1beta1Api**(*api_client=N*
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_token_review(*body*, ***kwargs*)
create a TokenReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_token_review(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1TokenReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1TokenReview

If the method is called asynchronously, returns the request thread.

create_token_review_with_http_info(*body*, ***kwargs*)
create a TokenReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_token_review_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1TokenReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1TokenReview

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.authorization_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.authorization_api.**AuthorizationApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.authorization_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.authorization_v1beta1_api.**AuthorizationV1beta1Api**(*api_client=None*
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_namespaced_local_subject_access_review(*namespace*, *body*, ***kwargs*)

create a LocalSubjectAccessReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_local_subject_access_review(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1LocalSubjectAccessReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1LocalSubjectAccessReview

If the method is called asynchronously, returns the request thread.

create_namespaced_local_subject_access_review_with_http_info(*namespace*, *body*, ***kwargs*)

create a LocalSubjectAccessReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_local_subject_access_review_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1LocalSubjectAccessReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1LocalSubjectAccessReview

If the method is called asynchronously, returns the request thread.

create_self_subject_access_review(*body*, ***kwargs*)

create a SelfSubjectAccessReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_self_subject_access_review(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1SelfSubjectAccessReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1SelfSubjectAccessReview

If the method is called asynchronously, returns the request thread.

create_self_subject_access_review_with_http_info(*body*, ***kwargs*)

create a SelfSubjectAccessReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_self_subject_access_review_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1SelfSubjectAccessReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1SelfSubjectAccessReview

If the method is called asynchronously, returns the request thread.

create_self_subject_rules_review(*body*, ***kwargs*)

create a SelfSubjectRulesReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_self_subject_rules_review(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1SelfSubjectRulesReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1SelfSubjectRulesReview

If the method is called asynchronously, returns the request thread.

create_self_subject_rules_review_with_http_info(*body*, ***kwargs*)

create a SelfSubjectRulesReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_self_subject_rules_review_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1SelfSubjectRulesReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1SelfSubjectRulesReview

If the method is called asynchronously, returns the request thread.

create_subject_access_review(*body*, ***kwargs*)
create a SubjectAccessReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_subject_access_review(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1SubjectAccessReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1SubjectAccessReview

If the method is called asynchronously, returns the request thread.

create_subject_access_review_with_http_info(*body*, ***kwargs*)
create a SubjectAccessReview This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_subject_access_review_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1SubjectAccessReview body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1SubjectAccessReview

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.autoscaling_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.autoscaling_api.**AutoscalingApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an

asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.autoscaling_v1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.autoscaling_v1_api.**AutoscalingV1Api**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_namespaced_horizontal_pod_autoscaler(*namespace*, *body*, ***kwargs*)
create a HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_horizontal_pod_autoscaler(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1HorizontalPodAutoscaler body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

create_namespaced_horizontal_pod_autoscaler_with_http_info(*namespace*, *body*, ***kwargs*)
create a HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_horizontal_pod_autoscaler_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1HorizontalPodAutoscaler body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_horizontal_pod_autoscaler(*namespace*, ***kwargs*)
delete collection of HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread

= api.delete_collection_namespaced_horizontal_pod_autoscaler(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_horizontal_pod_autoscaler_with_http_info(*namespace*, ***kwargs*)

delete collection of HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_horizontal_pod_autoscaler_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are

included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_horizontal_pod_autoscaler(*name*, *namespace*, *body*, ***kwargs*)

delete a HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_horizontal_pod_autoscaler(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_horizontal_pod_autoscaler_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_horizontal_pod_autoscaler_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body:

(required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_horizontal_pod_autoscaler_for_all_namespaces(***kwargs*)

list or watch objects of kind HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_horizontal_pod_autoscaler_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent

continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1HorizontalPodAutoscalerList

If the method is called asynchronously, returns the request thread.

list_horizontal_pod_autoscaler_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_horizontal_pod_autoscaler_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1HorizontalPodAutoscalerList

If the method is called asynchronously, returns the request thread.

list_namespaced_horizontal_pod_autoscaler(*namespace*, ***kwargs*)

list or watch objects of kind HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_horizontal_pod_autoscaler(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1HorizontalPodAutoscalerList

If the method is called asynchronously, returns the request thread.

list_namespaced_horizontal_pod_autoscaler_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_horizontal_pod_autoscaler_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is

true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1HorizontalPodAutoscalerList

If the method is called asynchronously, returns the request thread.

patch_namespaced_horizontal_pod_autoscaler(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_horizontal_pod_autoscaler(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

patch_namespaced_horizontal_pod_autoscaler_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_horizontal_pod_autoscaler_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

patch_namespaced_horizontal_pod_autoscaler_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_horizontal_pod_autoscaler_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

patch_namespaced_horizontal_pod_autoscaler_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_horizontal_pod_autoscaler_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

read_namespaced_horizontal_pod_autoscaler(*name*, *namespace*, ***kwargs*)

read the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_horizontal_pod_autoscaler(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

read_namespaced_horizontal_pod_autoscaler_status(*name*, *namespace*, ***kwargs*)

read status of the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_horizontal_pod_autoscaler_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

read_namespaced_horizontal_pod_autoscaler_status_with_http_info(*name*, *namespace*, ***kwargs*)

read status of the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_horizontal_pod_autoscaler_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true',

then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

read_namespaced_horizontal_pod_autoscaler_with_http_info(*name*, *namespace*, ***kwargs*)

read the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_horizontal_pod_autoscaler_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

replace_namespaced_horizontal_pod_autoscaler(*name*, *namespace*, *body*, ***kwargs*)

replace the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_horizontal_pod_autoscaler(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1HorizontalPodAutoscaler body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

replace_namespaced_horizontal_pod_autoscaler_status(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_horizontal_pod_autoscaler_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1HorizontalPodAutoscaler body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

replace_namespaced_horizontal_pod_autoscaler_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_horizontal_pod_autoscaler_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1HorizontalPodAutoscaler body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

replace_namespaced_horizontal_pod_autoscaler_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified HorizontalPodAutoscaler This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_horizontal_pod_autoscaler_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the HorizontalPodAutoscaler (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1HorizontalPodAutoscaler body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1HorizontalPodAutoscaler

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.batch_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.batch_api.**BatchApi**(*api_client=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)

get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)

get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.batch_v1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.batch_v1_api.**BatchV1Api**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_namespaced_job(*namespace*, *body*, ***kwargs*)
create a Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_job(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Job body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

create_namespaced_job_with_http_info(*namespace*, *body*, ***kwargs*)
create a Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_job_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Job body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_job(*namespace*, ***kwargs*)
delete collection of Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_job(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str

resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_job_with_http_info(*namespace*, ***kwargs*)

delete collection of Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_job_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_job(*name*, *namespace*, *body*, ***kwargs*)

delete a Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_job(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param

str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_job_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_job_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_job_for_all_namespaces(***kwargs*)

list or watch objects of kind Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_job_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1JobList

If the method is called asynchronously, returns the request thread.

list_job_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_job_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of

the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1JobList

If the method is called asynchronously, returns the request thread.

list_namespaced_job(*namespace*, ***kwargs*)

list or watch objects of kind Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_job(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1JobList

If the method is called asynchronously, returns the request thread.

list_namespaced_job_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_job_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1JobList

If the method is called asynchronously, returns the request thread.

patch_namespaced_job(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_job(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

patch_namespaced_job_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_job_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

patch_namespaced_job_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update status of the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_job_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

patch_namespaced_job_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_job_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

read_namespaced_job(*name*, *namespace*, ***kwargs*)
read the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_job(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Job

If the method is called asynchronously, returns the request thread.

read_namespaced_job_status(*name*, *namespace*, ***kwargs*)
read status of the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_job_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

read_namespaced_job_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_job_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty

printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

read_namespaced_job_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_job_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Job

If the method is called asynchronously, returns the request thread.

replace_namespaced_job(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_job(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Job body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

replace_namespaced_job_status(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_job_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Job body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

replace_namespaced_job_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_job_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Job body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

replace_namespaced_job_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Job This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_job_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Job (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Job body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Job

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.batch_v2alpha1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.batch_v2alpha1_api.**BatchV2alpha1Api**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_namespaced_cron_job(*namespace*, *body*, ***kwargs*)
create a CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_cron_job(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V2alpha1CronJob body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

create_namespaced_cron_job_with_http_info(*namespace*, *body*, ***kwargs*)
create a CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_cron_job_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V2alpha1CronJob body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_cron_job(*namespace*, ***kwargs*)
delete collection of CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_cron_job(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available.

Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_cron_job_with_http_info(*namespace*, ***kwargs*)

delete collection of CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_cron_job_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return

them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_cron_job(*name*, *namespace*, *body*, ***kwargs*)
delete a CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_cron_job(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_cron_job_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
delete a CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_cron_job_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_cron_job_for_all_namespaces(***kwargs*)

list or watch objects of kind CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_cron_job_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V2alpha1CronJobList

If the method is called asynchronously, returns the request thread.

list_cron_job_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_cron_job_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to

expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V2alpha1CronJobList

If the method is called asynchronously, returns the request thread.

list_namespaced_cron_job(*namespace*, ***kwargs*)

list or watch objects of kind CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_cron_job(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field

is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V2alpha1CronJobList

If the method is called asynchronously, returns the request thread.

list_namespaced_cron_job_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_cron_job_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V2alpha1CronJobList

If the method is called asynchronously, returns the request thread.

patch_namespaced_cron_job(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_cron_job(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

patch_namespaced_cron_job_status(*name*, *namespace*, *body*, ***kwargs*)
partially update status of the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_cron_job_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

patch_namespaced_cron_job_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update status of the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_cron_job_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

patch_namespaced_cron_job_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_cron_job_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

read_namespaced_cron_job(*name*, *namespace*, ***kwargs*)
read the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_cron_job(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

read_namespaced_cron_job_status(*name*, *namespace*, ***kwargs*)
read status of the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_cron_job_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

read_namespaced_cron_job_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_cron_job_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

read_namespaced_cron_job_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_cron_job_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

replace_namespaced_cron_job(*name*, *namespace*, *body*, ***kwargs*)
replace the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_cron_job(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V2alpha1CronJob body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

replace_namespaced_cron_job_status(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_cron_job_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V2alpha1CronJob body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

replace_namespaced_cron_job_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.replace_namespaced_cron_job_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V2alpha1CronJob body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

replace_namespaced_cron_job_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified CronJob This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_cron_job_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the CronJob (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V2alpha1CronJob body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V2alpha1CronJob

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.certificates_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.certificates_api.**CertificatesApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.certificates_v1alpha1_api module

kubernetes.client.apis.core_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.core_api.**CoreApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_versions(***kwargs*)
get available API versions This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_versions(async=True) >>> result = thread.get()

:param async bool :return: V1APIVersions

If the method is called asynchronously, returns the request thread.

get_api_versions_with_http_info(***kwargs*)
get available API versions This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_versions_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIVersions

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.core_v1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.core_v1_api.**CoreV1Api**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

connect_delete_namespaced_pod_proxy(*name*, *namespace*, ***kwargs*)
connect DELETE requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_namespaced_pod_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_namespaced_pod_proxy_with_http_info(*name*, *namespace*, ***kwargs*)
connect DELETE requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_namespaced_pod_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_namespaced_pod_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect DELETE requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_namespaced_pod_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_namespaced_pod_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect DELETE requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_namespaced_pod_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_namespaced_service_proxy(*name*, *namespace*, ***kwargs*)

connect DELETE requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_namespaced_service_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_namespaced_service_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect DELETE requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_namespaced_service_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_namespaced_service_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect DELETE requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_namespaced_service_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_namespaced_service_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect DELETE requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_namespaced_service_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_node_proxy(*name*, ***kwargs*)

connect DELETE requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_node_proxy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_node_proxy_with_http_info(*name*, ***kwargs*)

connect DELETE requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_node_proxy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_node_proxy_with_path(*name*, *path*, ***kwargs*)

connect DELETE requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_node_proxy_with_path(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_delete_node_proxy_with_path_with_http_info(*name*, *path*, ***kwargs*)

connect DELETE requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_delete_node_proxy_with_path_with_http_info(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_attach(*name*, *namespace*, ***kwargs*)

connect GET requests to attach of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_attach(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str container: The container in which to execute the command. Defaults to only container if there is only one container in the pod. :param bool stderr: Stderr if true indicates that stderr is to be redirected for the attach call. Defaults to true. :param bool stdin: Stdin if true, redirects the standard input stream of the pod for this call. Defaults to false. :param bool stdout: Stdout if true indicates that stdout is to be redirected for the attach call. Defaults to true. :param bool tty: TTY if true indicates that a tty will be allocated for the attach call. This is passed through the container runtime so the tty is allocated on the worker node by the container runtime. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_attach_with_http_info(*name*, *namespace*, ***kwargs*)

connect GET requests to attach of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_attach_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str container: The container in which to execute the command. Defaults to only container if there is only one container in the pod. :param bool stderr: Stderr if true indicates that stderr is to be redirected for the attach call. Defaults to true. :param bool stdin: Stdin if true, redirects the standard input stream of the pod for this call. Defaults to false. :param bool stdout: Stdout if true indicates that stdout is to be redirected for the attach call. Defaults to true. :param bool tty: TTY if true indicates that a tty will be allocated for the attach call. This is passed through the container runtime so the tty is allocated on the worker node by the container runtime. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_exec(*name*, *namespace*, ***kwargs*)

connect GET requests to exec of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_exec(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str command: Command is the remote command to execute. argv array. Not executed within a shell. :param str container: Container in which to execute the command. Defaults to only container if there is only one container in the pod. :param bool

stderr: Redirect the standard error stream of the pod for this call. Defaults to true. :param bool stdin: Redirect the standard input stream of the pod for this call. Defaults to false. :param bool stdout: Redirect the standard output stream of the pod for this call. Defaults to true. :param bool tty: TTY if true indicates that a tty will be allocated for the exec call. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_exec_with_http_info(*name*, *namespace*, ***kwargs*)

connect GET requests to exec of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_exec_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str command: Command is the remote command to execute. argv array. Not executed within a shell. :param str container: Container in which to execute the command. Defaults to only container if there is only one container in the pod. :param bool stderr: Redirect the standard error stream of the pod for this call. Defaults to true. :param bool stdin: Redirect the standard input stream of the pod for this call. Defaults to false. :param bool stdout: Redirect the standard output stream of the pod for this call. Defaults to true. :param bool tty: TTY if true indicates that a tty will be allocated for the exec call. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_portforward(*name*, *namespace*, ***kwargs*)

connect GET requests to portforward of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_portforward(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param int ports: List of ports to forward Required when using WebSockets :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_portforward_with_http_info(*name*, *namespace*, ***kwargs*)

connect GET requests to portforward of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_portforward_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param int ports: List of ports to forward Required when using WebSockets :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_proxy(*name*, *namespace*, ***kwargs*)

connect GET requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect GET requests to proxy of Pod This method makes a synchronous HTTP request by

default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect GET requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_pod_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect GET requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_pod_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_service_proxy(*name*, *namespace*, ***kwargs*)

connect GET requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_service_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_service_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect GET requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_service_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_service_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect GET requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_service_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_namespaced_service_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect GET requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_namespaced_service_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_node_proxy(*name*, ***kwargs*)

connect GET requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_node_proxy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_node_proxy_with_http_info(*name*, ***kwargs*)

connect GET requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_node_proxy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_node_proxy_with_path(*name*, *path*, ***kwargs*)

connect GET requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_node_proxy_with_path(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_get_node_proxy_with_path_with_http_info(*name*, *path*, ***kwargs*)

connect GET requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_get_node_proxy_with_path_with_http_info(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_namespaced_pod_proxy(*name*, *namespace*, ***kwargs*)

connect HEAD requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_namespaced_pod_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_namespaced_pod_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect HEAD requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_namespaced_pod_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_namespaced_pod_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect HEAD requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_namespaced_pod_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_namespaced_pod_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect HEAD requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_namespaced_pod_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_namespaced_service_proxy(*name*, *namespace*, ***kwargs*)

connect HEAD requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_namespaced_service_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_namespaced_service_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect HEAD requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_namespaced_service_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_namespaced_service_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect HEAD requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_namespaced_service_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_namespaced_service_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect HEAD requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_namespaced_service_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http:

//localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_node_proxy(*name*, ***kwargs*)
connect HEAD requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_node_proxy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_node_proxy_with_http_info(*name*, ***kwargs*)
connect HEAD requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_node_proxy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_node_proxy_with_path(*name*, *path*, ***kwargs*)
connect HEAD requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_node_proxy_with_path(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_head_node_proxy_with_path_with_http_info(*name*, *path*, ***kwargs*)
connect HEAD requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_head_node_proxy_with_path_with_http_info(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_namespaced_pod_proxy(*name*, *namespace*, ***kwargs*)
connect OPTIONS requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_namespaced_pod_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_namespaced_pod_proxy_with_http_info(*name*, *namespace*, ***kwargs*)
connect OPTIONS requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_namespaced_pod_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_namespaced_pod_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect OPTIONS requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_namespaced_pod_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_namespaced_pod_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect OPTIONS requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_namespaced_pod_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_namespaced_service_proxy(*name*, *namespace*, ***kwargs*)

connect OPTIONS requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_namespaced_service_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_namespaced_service_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect OPTIONS requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_namespaced_service_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_namespaced_service_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect OPTIONS requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_namespaced_service_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_namespaced_service_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect OPTIONS requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_namespaced_service_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_node_proxy(*name*, ***kwargs*)

connect OPTIONS requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_node_proxy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_node_proxy_with_http_info(*name*, ***kwargs*)

connect OPTIONS requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_node_proxy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_node_proxy_with_path(*name*, *path*, ***kwargs*)

connect OPTIONS requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_node_proxy_with_path(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_options_node_proxy_with_path_with_http_info(*name*, *path*, ***kwargs*)
connect OPTIONS requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_options_node_proxy_with_path_with_http_info(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_namespaced_pod_proxy(*name*, *namespace*, ***kwargs*)
connect PATCH requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_namespaced_pod_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_namespaced_pod_proxy_with_http_info(*name*, *namespace*, ***kwargs*)
connect PATCH requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_namespaced_pod_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_namespaced_pod_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)
connect PATCH requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_namespaced_pod_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_namespaced_pod_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)
connect PATCH requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_namespaced_pod_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_namespaced_service_proxy(*name*, *namespace*, ***kwargs*)

connect PATCH requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_namespaced_service_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_namespaced_service_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect PATCH requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_namespaced_service_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_namespaced_service_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect PATCH requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_namespaced_service_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_namespaced_service_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect PATCH requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_namespaced_service_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http:

//localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_node_proxy(*name*, ***kwargs*)
connect PATCH requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_node_proxy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_node_proxy_with_http_info(*name*, ***kwargs*)
connect PATCH requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_node_proxy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_node_proxy_with_path(*name*, *path*, ***kwargs*)
connect PATCH requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_node_proxy_with_path(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_patch_node_proxy_with_path_with_http_info(*name*, *path*, ***kwargs*)
connect PATCH requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_patch_node_proxy_with_path_with_http_info(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_attach(*name*, *namespace*, ***kwargs*)
connect POST requests to attach of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_attach(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str container: The container in which to execute the command. Defaults to only container if there is only one container in the pod. :param bool stderr: Stderr if true indicates that stderr is to be redirected for the attach call. Defaults to true. :param bool stdin: Stdin if true, redirects the standard input stream of the pod for this call. Defaults to false. :param bool stdout: Stdout if true indicates that stdout is to be redirected for the attach call. Defaults to true. :param bool tty: TTY if true indicates that a tty will be allocated for the attach call. This is passed through the container runtime so the tty is allocated on the worker node by the container runtime. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_attach_with_http_info(*name*, *namespace*, ***kwargs*)
connect POST requests to attach of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_attach_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str container: The container in which to execute the command. Defaults to only container if there is only one container in the pod. :param bool stderr: Stderr if true indicates that stderr is to be redirected for the attach call. Defaults to true. :param bool stdin: Stdin if true, redirects the standard input stream of the pod for this call. Defaults to false. :param bool stdout: Stdout if true indicates that stdout is to be redirected for the attach call. Defaults to true. :param bool tty: TTY if true indicates that a tty will be allocated for the attach call. This is passed through the container runtime so the tty is allocated on the worker node by the container runtime. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_exec(*name*, *namespace*, ***kwargs*)
connect POST requests to exec of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_exec(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str command: Command is the remote command to execute. argv array. Not executed within a shell. :param str container: Container in which to execute the command. Defaults to only container if there is only one container in the pod. :param bool stderr: Redirect the standard error stream of the pod for this call. Defaults to true. :param bool stdin: Redirect the standard input stream of the pod for this call. Defaults to false. :param bool stdout: Redirect the standard output stream of the pod for this call. Defaults to true. :param bool tty: TTY if true indicates that a tty will be allocated for the exec call. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_exec_with_http_info(*name*, *namespace*, ***kwargs*)
connect POST requests to exec of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_exec_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str command: Command is the remote command to execute. argv array. Not executed within a shell. :param str container: Container in which to execute the command. Defaults to only container if there is only one container in the pod. :param bool stderr: Redirect the standard error stream of the pod for this call. Defaults to true. :param bool stdin: Redirect the standard input stream of the pod for this call. Defaults to false. :param bool stdout: Redirect the standard output stream of the pod for this call. Defaults to true. :param bool tty: TTY if true indicates that a tty will be allocated for the exec call. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_portforward(*name*, *namespace*, ***kwargs*)
connect POST requests to portforward of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_portforward(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param int ports: List of ports to forward Required when using WebSockets :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_portforward_with_http_info(*name*, *namespace*, ***kwargs*)

connect POST requests to portforward of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_portforward_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param int ports: List of ports to forward Required when using WebSockets :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_proxy(*name*, *namespace*, ***kwargs*)

connect POST requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect POST requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect POST requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_pod_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect POST requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_pod_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_service_proxy(*name*, *namespace*, ***kwargs*)

connect POST requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_service_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_service_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect POST requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_service_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_service_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect POST requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_service_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_namespaced_service_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect POST requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_namespaced_service_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http:

//localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_node_proxy(*name*, ***kwargs*)
connect POST requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_node_proxy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_node_proxy_with_http_info(*name*, ***kwargs*)
connect POST requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_node_proxy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_node_proxy_with_path(*name*, *path*, ***kwargs*)
connect POST requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_node_proxy_with_path(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_post_node_proxy_with_path_with_http_info(*name*, *path*, ***kwargs*)
connect POST requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_post_node_proxy_with_path_with_http_info(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_namespaced_pod_proxy(*name*, *namespace*, ***kwargs*)
connect PUT requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_namespaced_pod_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_namespaced_pod_proxy_with_http_info(*name*, *namespace*, ***kwargs*)
connect PUT requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_namespaced_pod_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_namespaced_pod_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect PUT requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_namespaced_pod_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_namespaced_pod_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect PUT requests to proxy of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_namespaced_pod_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to pod. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_namespaced_service_proxy(*name*, *namespace*, ***kwargs*)

connect PUT requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_namespaced_service_proxy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_namespaced_service_proxy_with_http_info(*name*, *namespace*, ***kwargs*)

connect PUT requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_namespaced_service_proxy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_namespaced_service_proxy_with_path(*name*, *namespace*, *path*, ***kwargs*)

connect PUT requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_namespaced_service_proxy_with_path(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_namespaced_service_proxy_with_path_with_http_info(*name*, *namespace*, *path*, ***kwargs*)

connect PUT requests to proxy of Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_namespaced_service_proxy_with_path_with_http_info(name, namespace, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str path: path to the resource (required) :param str path2: Path is the part of URLs that include service endpoints, suffixes, and parameters to use for the current proxy request to service. For example, the whole request URL is http://localhost/api/v1/namespaces/kube-system/services/elasticsearch-logging/_search?q=user:kimchy. Path is _search?q=user:kimchy. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_node_proxy(*name*, ***kwargs*)

connect PUT requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_node_proxy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_node_proxy_with_http_info(*name*, ***kwargs*)

connect PUT requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_node_proxy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_node_proxy_with_path(*name*, *path*, ***kwargs*)

connect PUT requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_node_proxy_with_path(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

connect_put_node_proxy_with_path_with_http_info(*name*, *path*, ***kwargs*)
connect PUT requests to proxy of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.connect_put_node_proxy_with_path_with_http_info(name, path, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str path: path to the resource (required) :param str path2: Path is the URL path to use for the current proxy request to node. :return: str

If the method is called asynchronously, returns the request thread.

create_namespace(*body*, ***kwargs*)
create a Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespace(body, async=True) >>> result = thread.get()

:param async bool :param V1Namespace body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

create_namespace_with_http_info(*body*, ***kwargs*)
create a Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespace_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1Namespace body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

create_namespaced_binding(*namespace*, *body*, ***kwargs*)
create a Binding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_binding(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Binding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Binding

If the method is called asynchronously, returns the request thread.

create_namespaced_binding_with_http_info(*namespace*, *body*, ***kwargs*)
create a Binding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_binding_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Binding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Binding

If the method is called asynchronously, returns the request thread.

create_namespaced_config_map(*namespace*, *body*, ***kwargs*)
create a ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_config_map(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ConfigMap body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ConfigMap

If the method is called asynchronously, returns the request thread.

create_namespaced_config_map_with_http_info(*namespace*, *body*, ***kwargs*)
create a ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_config_map_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ConfigMap body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ConfigMap

If the method is called asynchronously, returns the request thread.

create_namespaced_endpoints(*namespace*, *body*, ***kwargs*)
create Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_endpoints(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Endpoints body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Endpoints

If the method is called asynchronously, returns the request thread.

create_namespaced_endpoints_with_http_info(*namespace*, *body*, ***kwargs*)
create Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_endpoints_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Endpoints body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Endpoints

If the method is called asynchronously, returns the request thread.

create_namespaced_event(*namespace*, *body*, ***kwargs*)
create an Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_event(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Event body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Event

If the method is called asynchronously, returns the request thread.

create_namespaced_event_with_http_info(*namespace*, *body*, ***kwargs*)
create an Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_event_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Event body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Event

If the method is called asynchronously, returns the request thread.

create_namespaced_limit_range(*namespace*, *body*, ***kwargs*)
create a LimitRange This method makes a synchronous HTTP request by default.

To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_limit_range(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1LimitRange body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1LimitRange

If the method is called asynchronously, returns the request thread.

create_namespaced_limit_range_with_http_info(*namespace*, *body*, ***kwargs*)

create a LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_limit_range_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1LimitRange body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1LimitRange

If the method is called asynchronously, returns the request thread.

create_namespaced_persistent_volume_claim(*namespace*, *body*, ***kwargs*)

create a PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_persistent_volume_claim(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PersistentVolumeClaim body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

create_namespaced_persistent_volume_claim_with_http_info(*namespace*, *body*, ***kwargs*)

create a PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_persistent_volume_claim_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PersistentVolumeClaim body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

create_namespaced_pod(*namespace*, *body*, ***kwargs*)

create a Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Pod body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

create_namespaced_pod_binding(*name*, *namespace*, *body*, ***kwargs*)

create binding of a Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_binding(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Binding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Binding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Binding

If the method is called asynchronously, returns the request thread.

create_namespaced_pod_binding_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

create binding of a Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_binding_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Binding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Binding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Binding

If the method is called asynchronously, returns the request thread.

create_namespaced_pod_eviction(*name*, *namespace*, *body*, ***kwargs*)

create eviction of a Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_eviction(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Eviction (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1Eviction body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Eviction

If the method is called asynchronously, returns the request thread.

create_namespaced_pod_eviction_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

create eviction of a Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_eviction_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Eviction (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1Eviction body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Eviction

If the method is called asynchronously, returns the request thread.

create_namespaced_pod_template(*namespace*, *body*, ***kwargs*)

create a PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_template(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PodTemplate body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PodTemplate

If the method is called asynchronously, returns the request thread.

create_namespaced_pod_template_with_http_info(*namespace*, *body*, ***kwargs*)

create a PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_template_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PodTemplate body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PodTemplate

If the method is called asynchronously, returns the request thread.

create_namespaced_pod_with_http_info(*namespace*, *body*, ***kwargs*)
create a Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Pod body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

create_namespaced_replication_controller(*namespace*, *body*, ***kwargs*)
create a ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_replication_controller(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ReplicationController body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

create_namespaced_replication_controller_with_http_info(*namespace*, *body*, ***kwargs*)
create a ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_replication_controller_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ReplicationController body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

create_namespaced_resource_quota(*namespace*, *body*, ***kwargs*)
create a ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_resource_quota(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ResourceQuota body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

create_namespaced_resource_quota_with_http_info(*namespace*, *body*, ***kwargs*)
create a ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_resource_quota_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ResourceQuota body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

create_namespaced_secret(*namespace*, *body*, ***kwargs*)
create a Secret This method makes a synchronous HTTP request by default. To make an asynchronous

HTTP request, please pass async=True >>> thread = api.create_namespaced_secret(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Secret body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Secret

If the method is called asynchronously, returns the request thread.

create_namespaced_secret_with_http_info(*namespace*, *body*, ***kwargs*)

create a Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_secret_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Secret body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Secret

If the method is called asynchronously, returns the request thread.

create_namespaced_service(*namespace*, *body*, ***kwargs*)

create a Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_service(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Service body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

create_namespaced_service_account(*namespace*, *body*, ***kwargs*)

create a ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_service_account(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ServiceAccount body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ServiceAccount

If the method is called asynchronously, returns the request thread.

create_namespaced_service_account_with_http_info(*namespace*, *body*, ***kwargs*)

create a ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_service_account_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ServiceAccount body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ServiceAccount

If the method is called asynchronously, returns the request thread.

create_namespaced_service_with_http_info(*namespace*, *body*, ***kwargs*)

create a Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_service_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Service body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

create_node(*body*, ***kwargs*)
create a Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_node(body, async=True) >>> result = thread.get()

:param async bool :param V1Node body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

create_node_with_http_info(*body*, ***kwargs*)
create a Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_node_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1Node body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

create_persistent_volume(*body*, ***kwargs*)
create a PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_persistent_volume(body, async=True) >>> result = thread.get()

:param async bool :param V1PersistentVolume body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

create_persistent_volume_with_http_info(*body*, ***kwargs*)
create a PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_persistent_volume_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1PersistentVolume body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_config_map(*namespace*, ***kwargs*)
delete collection of ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_config_map(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that

can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_config_map_with_http_info(*namespace*, ***kwargs*)

delete collection of ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_config_map_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what

we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_endpoints(*namespace*, ***kwargs*)

delete collection of Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_endpoints(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_endpoints_with_http_info(*namespace*, ***kwargs*)

delete collection of Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_endpoints_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters

(except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_event (*namespace*, ***kwargs*)

delete collection of Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_event(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field

is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_event_with_http_info(*namespace*, ***kwargs*)

delete collection of Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_event_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_limit_range(*namespace*, ***kwargs*)

delete collection of LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_limit_range(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_limit_range_with_http_info(*namespace*, ***kwargs*)

delete collection of LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_limit_range_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss

any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_persistent_volume_claim(*namespace*, ***kwargs*)

delete collection of PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_persistent_volume_claim(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of

a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_persistent_volume_claim_with_http_info(*namespace*, ***kwargs*)

delete collection of PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_persistent_volume_claim_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_pod(*namespace*, ***kwargs*)

delete collection of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.delete_collection_namespaced_pod(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_pod_template(*namespace*, ***kwargs*)

delete collection of PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_pod_template(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that

can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_pod_template_with_http_info(*namespace*, ***kwargs*)

delete collection of PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_pod_template_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what

we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_pod_with_http_info(*namespace*, ***kwargs*)

delete collection of Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_pod_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_replication_controller(*namespace*, ***kwargs*)

delete collection of ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_replication_controller(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters

(except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_replication_controller_with_http_info(*namespace*, ***kwargs*)

delete collection of ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_replication_controller_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is

specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_resource_quota(*namespace*, ***kwargs*)

delete collection of ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_resource_quota(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_resource_quota_with_http_info(*namespace*, ***kwargs*)

delete collection of ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_resource_quota_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_secret(*namespace*, ***kwargs*)

delete collection of Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_secret(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss

any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_secret_with_http_info(*namespace*, ***kwargs*)

delete collection of Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_secret_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of

a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_service_account(*namespace*, ***kwargs*)

delete collection of ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_service_account(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_service_account_with_http_info(*namespace*, ***kwargs*)

delete collection of ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_service_account_with_http_info(namespace, async=True) >>> result

= thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_node(***kwargs*)

delete collection of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_node(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer

than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_node_with_http_info(***kwargs*)

delete collection of Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_node_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return

them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_persistent_volume(***kwargs*)

delete collection of PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_persistent_volume(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_persistent_volume_with_http_info(***kwargs*)

delete collection of PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_persistent_volume_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields.

Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespace(*name*, *body*, ***kwargs*)

delete a Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespace(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespace_with_http_info(*name*, *body*, ***kwargs*)

delete a Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespace_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7.

Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_config_map(*name*, *namespace*, *body*, ***kwargs*)
delete a ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_config_map(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ConfigMap (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_config_map_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
delete a ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_config_map_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ConfigMap (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_endpoints(*name*, *namespace*, *body*, ***kwargs*)
delete Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous

HTTP request, please pass async=True >>> thread = api.delete_namespaced_endpoints(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Endpoints (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_endpoints_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_endpoints_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Endpoints (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_event(*name*, *namespace*, *body*, ***kwargs*)

delete an Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_event(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Event (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy:

Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_event_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete an Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_event_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Event (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_limit_range(*name*, *namespace*, *body*, ***kwargs*)

delete a LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_limit_range(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the LimitRange (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_limit_range_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_limit_range_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the LimitRange (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_persistent_volume_claim(*name*, *namespace*, *body*, ***kwargs*)

delete a PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_persistent_volume_claim(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_persistent_volume_claim_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_persistent_volume_claim_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or Or-

phanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_pod(*name*, *namespace*, *body*, ***kwargs*)

delete a Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_pod(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_pod_template(*name*, *namespace*, *body*, ***kwargs*)

delete a PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_pod_template(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodTemplate (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_pod_template_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_pod_template_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodTemplate (required) :param str namespace: object

name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_pod_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_pod_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_replication_controller(*name*, *namespace*, *body*, ***kwargs*)

delete a ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_replication_controller(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background;

'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_replication_controller_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_replication_controller_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_resource_quota(*name*, *namespace*, *body*, ***kwargs*)

delete a ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_resource_quota(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_resource_quota_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_resource_quota_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required)

:param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_secret(*name*, *namespace*, *body*, ***kwargs*)

delete a Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_secret(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Secret (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_secret_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_secret_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Secret (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_service(*name*, *namespace*, *body*, ***kwargs*)

delete a Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_service(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_service_account(*name*, *namespace*, *body*, ***kwargs*)

delete a ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_service_account(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ServiceAccount (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_service_account_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_service_account_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ServiceAccount (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will

be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_service_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_service_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_node(*name*, *body*, ***kwargs*)

delete a Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_node(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_node_with_http_info(*name*, *body*, ***kwargs*)

delete a Node This method makes a synchronous HTTP request by default. To make an asyn-

chronous HTTP request, please pass async=True >>> thread = api.delete_node_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_persistent_volume(*name*, *body*, ***kwargs*)

delete a PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_persistent_volume(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_persistent_volume_with_http_info(*name*, *body*, ***kwargs*)

delete a PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_persistent_volume_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are:

'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_component_status(***kwargs*)

list objects of kind ComponentStatus This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_component_status(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call,

regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ComponentStatusList

If the method is called asynchronously, returns the request thread.

list_component_status_with_http_info(***kwargs*)

list objects of kind ComponentStatus This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_component_status_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ComponentStatusList

If the method is called asynchronously, returns the request thread.

list_config_map_for_all_namespaces(***kwargs*)

list or watch objects of kind ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_config_map_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This

field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ConfigMapList

If the method is called asynchronously, returns the request thread.

list_config_map_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_config_map_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that

is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ConfigMapList

If the method is called asynchronously, returns the request thread.

list_endpoints_for_all_namespaces(***kwargs*)

list or watch objects of kind Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_endpoints_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1EndpointsList

If the method is called asynchronously, returns the request thread.

list_endpoints_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Endpoints This method makes a synchronous HTTP request by

default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_endpoints_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1EndpointsList

If the method is called asynchronously, returns the request thread.

list_event_for_all_namespaces(***kwargs*)

list or watch objects of kind Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_event_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting

a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1EventList

If the method is called asynchronously, returns the request thread.

list_event_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_event_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless

of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1EventList

If the method is called asynchronously, returns the request thread.

list_limit_range_for_all_namespaces(***kwargs*)

list or watch objects of kind LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_limit_range_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1LimitRangeList

If the method is called asynchronously, returns the request thread.

list_limit_range_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_limit_range_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value

returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1LimitRangeList

If the method is called asynchronously, returns the request thread.

list_namespace(***kwargs*)

list or watch objects of kind Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespace(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of

a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1NamespaceList

If the method is called asynchronously, returns the request thread.

list_namespace_with_http_info(***kwargs*)

list or watch objects of kind Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespace_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1NamespaceList

If the method is called asynchronously, returns the request thread.

list_namespaced_config_map(*namespace*, ***kwargs*)

list or watch objects of kind ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.list_namespaced_config_map(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ConfigMapList

If the method is called asynchronously, returns the request thread.

list_namespaced_config_map_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_config_map_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return

for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ConfigMapList

If the method is called asynchronously, returns the request thread.

list_namespaced_endpoints(*namespace*, ***kwargs*)

list or watch objects of kind Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_endpoints(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then

the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1EndpointsList

If the method is called asynchronously, returns the request thread.

list_namespaced_endpoints_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_endpoints_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1EndpointsList

If the method is called asynchronously, returns the request thread.

list_namespaced_event(*namespace*, ***kwargs*)

list or watch objects of kind Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_event(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue

option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1EventList

If the method is called asynchronously, returns the request thread.

list_namespaced_event_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_event_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available.

Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1EventList

If the method is called asynchronously, returns the request thread.

list_namespaced_limit_range(*namespace*, ***kwargs*)

list or watch objects of kind LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_limit_range(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return:

V1LimitRangeList

If the method is called asynchronously, returns the request thread.

list_namespaced_limit_range_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_limit_range_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1LimitRangeList

If the method is called asynchronously, returns the request thread.

list_namespaced_persistent_volume_claim(*namespace*, ***kwargs*)

list or watch objects of kind PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_persistent_volume_claim(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating

the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PersistentVolumeClaimList

If the method is called asynchronously, returns the request thread.

list_namespaced_persistent_volume_claim_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_persistent_volume_claim_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will

be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PersistentVolumeClaimList

If the method is called asynchronously, returns the request thread.

list_namespaced_pod(*namespace*, ***kwargs*)

list or watch objects of kind Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_pod(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PodList

If the method is called asynchronously, returns the request thread.

list_namespaced_pod_template(*namespace*, ***kwargs*)

list or watch objects of kind PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_pod_template(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PodTemplateList

If the method is called asynchronously, returns the request thread.

list_namespaced_pod_template_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_pod_template_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their

fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PodTemplateList

If the method is called asynchronously, returns the request thread.

list_namespaced_pod_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_pod_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the

version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PodList

If the method is called asynchronously, returns the request thread.

list_namespaced_replication_controller(*namespace*, ***kwargs*)

list or watch objects of kind ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_replication_controller(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ReplicationControllerList

If the method is called asynchronously, returns the request thread.

list_namespaced_replication_controller_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_replication_controller_with_http_info(namespace, async=True) >>> result =

thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ReplicationControllerList

If the method is called asynchronously, returns the request thread.

list_namespaced_resource_quota(*namespace*, ***kwargs*)

list or watch objects of kind ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_resource_quota(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return

for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ResourceQuotaList

If the method is called asynchronously, returns the request thread.

list_namespaced_resource_quota_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_resource_quota_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then

the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ResourceQuotaList

If the method is called asynchronously, returns the request thread.

list_namespaced_secret(*namespace*, ***kwargs*)

list or watch objects of kind Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_secret(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1SecretList

If the method is called asynchronously, returns the request thread.

list_namespaced_secret_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_secret_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined,

clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1SecretList

If the method is called asynchronously, returns the request thread.

list_namespaced_service(*namespace*, ***kwargs*)

list or watch objects of kind Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_service(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is

specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ServiceList

If the method is called asynchronously, returns the request thread.

list_namespaced_service_account(*namespace*, ***kwargs*)

list or watch objects of kind ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_service_account(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return:

V1ServiceAccountList

If the method is called asynchronously, returns the request thread.

list_namespaced_service_account_with_http_info(*namespace*, ***kwargs*)
list or watch objects of kind ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_service_account_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ServiceAccountList

If the method is called asynchronously, returns the request thread.

list_namespaced_service_with_http_info(*namespace*, ***kwargs*)
list or watch objects of kind Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_service_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating

the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ServiceList

If the method is called asynchronously, returns the request thread.

list_node(***kwargs*)

list or watch objects of kind Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_node(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes

referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1NodeList

If the method is called asynchronously, returns the request thread.

list_node_with_http_info(***kwargs*)

list or watch objects of kind Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_node_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1NodeList

If the method is called asynchronously, returns the request thread.

list_persistent_volume(***kwargs*)

list or watch objects of kind PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_persistent_volume(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PersistentVolumeList

If the method is called asynchronously, returns the request thread.

list_persistent_volume_claim_for_all_namespaces(***kwargs*)

list or watch objects of kind PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_persistent_volume_claim_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether

more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PersistentVolumeClaimList

If the method is called asynchronously, returns the request thread.

list_persistent_volume_claim_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_persistent_volume_claim_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call,

regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PersistentVolumeClaimList

If the method is called asynchronously, returns the request thread.

list_persistent_volume_with_http_info(***kwargs*)

list or watch objects of kind PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_persistent_volume_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PersistentVolumeList

If the method is called asynchronously, returns the request thread.

list_pod_for_all_namespaces(***kwargs*)

list or watch objects of kind Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This

field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PodList

If the method is called asynchronously, returns the request thread.

list_pod_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If

objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PodList

If the method is called asynchronously, returns the request thread.

list_pod_template_for_all_namespaces(***kwargs*)

list or watch objects of kind PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_template_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PodTemplateList

If the method is called asynchronously, returns the request thread.

list_pod_template_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_template_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1PodTemplateList

If the method is called asynchronously, returns the request thread.

list_replication_controller_for_all_namespaces(***kwargs*)

list or watch objects of kind ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_replication_controller_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether

more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ReplicationControllerList

If the method is called asynchronously, returns the request thread.

list_replication_controller_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_replication_controller_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources

and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ReplicationControllerList

If the method is called asynchronously, returns the request thread.

list_resource_quota_for_all_namespaces(***kwargs*)

list or watch objects of kind ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_resource_quota_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ResourceQuotaList

If the method is called asynchronously, returns the request thread.

list_resource_quota_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_resource_quota_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value

returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ResourceQuotaList

If the method is called asynchronously, returns the request thread.

list_secret_for_all_namespaces(***kwargs*)

list or watch objects of kind Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_secret_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If

objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1SecretList

If the method is called asynchronously, returns the request thread.

list_secret_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_secret_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1SecretList

If the method is called asynchronously, returns the request thread.

list_service_account_for_all_namespaces(***kwargs*)

list or watch objects of kind ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_service_account_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results

from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ServiceAccountList

If the method is called asynchronously, returns the request thread.

list_service_account_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_service_account_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of

the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ServiceAccountList

If the method is called asynchronously, returns the request thread.

list_service_for_all_namespaces(***kwargs*)

list or watch objects of kind Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_service_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return:

V1ServiceList

If the method is called asynchronously, returns the request thread.

list_service_for_all_namespaces_with_http_info(***kwargs*)
list or watch objects of kind Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_service_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1ServiceList

If the method is called asynchronously, returns the request thread.

patch_namespace(*name*, *body*, ***kwargs*)
partially update the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespace(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

patch_namespace_status(*name*, *body*, ***kwargs*)
partially update status of the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespace_status(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

patch_namespace_status_with_http_info(*name*, *body*, ***kwargs*)
partially update status of the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespace_status_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

patch_namespace_with_http_info(*name*, *body*, ***kwargs*)
partially update the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespace_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

patch_namespaced_config_map(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_config_map(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ConfigMap (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ConfigMap

If the method is called asynchronously, returns the request thread.

patch_namespaced_config_map_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_config_map_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ConfigMap (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ConfigMap

If the method is called asynchronously, returns the request thread.

patch_namespaced_endpoints(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_endpoints(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Endpoints (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Endpoints

If the method is called asynchronously, returns the request thread.

patch_namespaced_endpoints_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.patch_namespaced_endpoints_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Endpoints (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Endpoints

If the method is called asynchronously, returns the request thread.

patch_namespaced_event(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_event(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Event (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Event

If the method is called asynchronously, returns the request thread.

patch_namespaced_event_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_event_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Event (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Event

If the method is called asynchronously, returns the request thread.

patch_namespaced_limit_range(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_limit_range(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the LimitRange (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1LimitRange

If the method is called asynchronously, returns the request thread.

patch_namespaced_limit_range_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_limit_range_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the LimitRange (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1LimitRange

If the method is called asynchronously, returns the request thread.

patch_namespaced_persistent_volume_claim(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_persistent_volume_claim(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

patch_namespaced_persistent_volume_claim_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_persistent_volume_claim_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

patch_namespaced_persistent_volume_claim_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_persistent_volume_claim_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

patch_namespaced_persistent_volume_claim_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_persistent_volume_claim_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified Pod This method makes a synchronous HTTP request

by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update status of the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_template(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_template(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodTemplate (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PodTemplate

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_template_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_template_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodTemplate (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PodTemplate

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

patch_namespaced_replication_controller(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replication_controller(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

patch_namespaced_replication_controller_scale(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replication_controller_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_replication_controller_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replication_controller_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_replication_controller_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replication_controller_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

patch_namespaced_replication_controller_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replication_controller_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

patch_namespaced_replication_controller_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replication_controller_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

patch_namespaced_resource_quota(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_resource_quota(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

patch_namespaced_resource_quota_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_resource_quota_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

patch_namespaced_resource_quota_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_resource_quota_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

patch_namespaced_resource_quota_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_resource_quota_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

patch_namespaced_secret(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_secret(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Secret (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Secret

If the method is called asynchronously, returns the request thread.

patch_namespaced_secret_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_secret_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Secret (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Secret

If the method is called asynchronously, returns the request thread.

patch_namespaced_service(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_service(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

patch_namespaced_service_account(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_service_account(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ServiceAccount (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ServiceAccount

If the method is called asynchronously, returns the request thread.

patch_namespaced_service_account_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_service_account_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ServiceAccount (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ServiceAccount

If the method is called asynchronously, returns the request thread.

patch_namespaced_service_status(*name*, *namespace*, *body*, ***kwargs*)
partially update status of the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_service_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

patch_namespaced_service_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_service_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

patch_namespaced_service_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_service_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

patch_node(*name*, *body*, ***kwargs*)

partially update the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_node(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

patch_node_status(*name*, *body*, ***kwargs*)

partially update status of the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_node_status(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

patch_node_status_with_http_info(*name*, *body*, ***kwargs*)

partially update status of the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_node_status_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

patch_node_with_http_info(*name*, *body*, ***kwargs*)
partially update the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_node_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

patch_persistent_volume(*name*, *body*, ***kwargs*)
partially update the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_persistent_volume(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

patch_persistent_volume_status(*name*, *body*, ***kwargs*)
partially update status of the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_persistent_volume_status(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

patch_persistent_volume_status_with_http_info(*name*, *body*, ***kwargs*)
partially update status of the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_persistent_volume_status_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

patch_persistent_volume_with_http_info(*name*, *body*, ***kwargs*)
partially update the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_persistent_volume_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

read_component_status(*name*, ***kwargs*)
read the specified ComponentStatus This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_component_status(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ComponentStatus (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ComponentStatus

If the method is called asynchronously, returns the request thread.

read_component_status_with_http_info(*name*, ***kwargs*)
read the specified ComponentStatus This method makes a synchronous HTTP request by de-

fault. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_component_status_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ComponentStatus (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ComponentStatus

If the method is called asynchronously, returns the request thread.

read_namespace(*name*, ***kwargs*)
read the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespace(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

read_namespace_status(*name*, ***kwargs*)
read status of the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespace_status(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

read_namespace_status_with_http_info(*name*, ***kwargs*)
read status of the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespace_status_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

read_namespace_with_http_info(*name*, ***kwargs*)
read the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespace_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

read_namespaced_config_map(*name*, *namespace*, ***kwargs*)
read the specified ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_config_map(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ConfigMap (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1ConfigMap

If the method is called asynchronously, returns the request thread.

read_namespaced_config_map_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_config_map_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ConfigMap (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1ConfigMap

If the method is called asynchronously, returns the request thread.

read_namespaced_endpoints(*name*, *namespace*, ***kwargs*)
read the specified Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_endpoints(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Endpoints (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Endpoints

If the method is called asynchronously, returns the request thread.

read_namespaced_endpoints_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_endpoints_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Endpoints (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Endpoints

If the method is called asynchronously, returns the request thread.

read_namespaced_event(*name*, *namespace*, ***kwargs*)
read the specified Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_event(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Event (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Event

If the method is called asynchronously, returns the request thread.

read_namespaced_event_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_event_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Event (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Event

If the method is called asynchronously, returns the request thread.

read_namespaced_limit_range(*name*, *namespace*, ***kwargs*)

read the specified LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_limit_range(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the LimitRange (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1LimitRange

If the method is called asynchronously, returns the request thread.

read_namespaced_limit_range_with_http_info(*name*, *namespace*, ***kwargs*)

read the specified LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_limit_range_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the LimitRange (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1LimitRange

If the method is called asynchronously, returns the request thread.

read_namespaced_persistent_volume_claim(*name*, *namespace*, ***kwargs*)

read the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_persistent_volume_claim(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

read_namespaced_persistent_volume_claim_status(*name*, *namespace*, ***kwargs*)

read status of the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_persistent_volume_claim_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

read_namespaced_persistent_volume_claim_status_with_http_info(*name*, *namespace*, ***kwargs*)

read status of the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_persistent_volume_claim_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

read_namespaced_persistent_volume_claim_with_http_info(*name*, *namespace*, ***kwargs*)

read the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_persistent_volume_claim_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

read_namespaced_pod(*name*, *namespace*, ***kwargs*)

read the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Pod

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_log(*name*, *namespace*, ***kwargs*)

read log of the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_log(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str container: The container for which to stream logs. Defaults to only container if there is one container in the pod. :param bool follow: Follow the log stream of the pod. Defaults to false. :param int limit_bytes: If set, the number of bytes to read from the server before terminating the log output. This may not display a complete final line of logging, and may return slightly more or slightly less than the specified limit. :param str pretty: If 'true', then the output is pretty printed. :param bool previous: Return previous terminated container logs. Defaults to false. :param int since_seconds: A relative time in seconds before the current time from which to show logs. If this value precedes the time a pod was started, only logs since the pod start will be returned. If this value is in the future, no logs will be returned. Only one of sinceSeconds or sinceTime may be specified. :param int tail_lines: If set, the number of lines from the end of the logs to show. If not specified, logs are shown from the creation of the container or sinceSeconds or sinceTime :param bool timestamps: If true,

add an RFC3339 or RFC3339Nano timestamp at the beginning of every line of log output. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_log_with_http_info(*name*, *namespace*, ***kwargs*)
read log of the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_log_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str container: The container for which to stream logs. Defaults to only container if there is one container in the pod. :param bool follow: Follow the log stream of the pod. Defaults to false. :param int limit_bytes: If set, the number of bytes to read from the server before terminating the log output. This may not display a complete final line of logging, and may return slightly more or slightly less than the specified limit. :param str pretty: If 'true', then the output is pretty printed. :param bool previous: Return previous terminated container logs. Defaults to false. :param int since_seconds: A relative time in seconds before the current time from which to show logs. If this value precedes the time a pod was started, only logs since the pod start will be returned. If this value is in the future, no logs will be returned. Only one of sinceSeconds or sinceTime may be specified. :param int tail_lines: If set, the number of lines from the end of the logs to show. If not specified, logs are shown from the creation of the container or sinceSeconds or sinceTime :param bool timestamps: If true, add an RFC3339 or RFC3339Nano timestamp at the beginning of every line of log output. Defaults to false. :return: str

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_status(*name*, *namespace*, ***kwargs*)
read status of the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_template(*name*, *namespace*, ***kwargs*)
read the specified PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_template(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodTemplate (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1PodTemplate

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_template_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_template_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodTemplate (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1PodTemplate

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Pod

If the method is called asynchronously, returns the request thread.

read_namespaced_replication_controller(*name*, *namespace*, ***kwargs*)
read the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replication_controller(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

read_namespaced_replication_controller_scale(*name*, *namespace*, ***kwargs*)
read scale of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replication_controller_scale(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_replication_controller_scale_with_http_info(*name*, *namespace*, ***kwargs*)
read scale of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replication_controller_scale_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_replication_controller_status(*name*, *namespace*, ***kwargs*)

read status of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replication_controller_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

read_namespaced_replication_controller_status_with_http_info(*name*, *namespace*, ***kwargs*)

read status of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replication_controller_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

read_namespaced_replication_controller_with_http_info(*name*, *namespace*, ***kwargs*)

read the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replication_controller_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

read_namespaced_resource_quota(*name*, *namespace*, ***kwargs*)

read the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_resource_quota(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

read_namespaced_resource_quota_status(*name*, *namespace*, ***kwargs*)
read status of the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_resource_quota_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

read_namespaced_resource_quota_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_resource_quota_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

read_namespaced_resource_quota_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_resource_quota_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

read_namespaced_secret(*name*, *namespace*, ***kwargs*)
read the specified Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_secret(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Secret (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Secret

If the method is called asynchronously, returns the request thread.

read_namespaced_secret_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_secret_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Secret (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Secret

If the method is called asynchronously, returns the request thread.

read_namespaced_service(*name*, *namespace*, ***kwargs*)
read the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_service(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Service

If the method is called asynchronously, returns the request thread.

read_namespaced_service_account(*name*, *namespace*, ***kwargs*)
read the specified ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_service_account(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ServiceAccount (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1ServiceAccount

If the method is called asynchronously, returns the request thread.

read_namespaced_service_account_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_service_account_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ServiceAccount (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1ServiceAccount

If the method is called asynchronously, returns the request thread.

read_namespaced_service_status(*name*, *namespace*, ***kwargs*)
read status of the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_service_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

read_namespaced_service_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_service_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is

pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

read_namespaced_service_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_service_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Service

If the method is called asynchronously, returns the request thread.

read_node(*name*, ***kwargs*)
read the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_node(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Node

If the method is called asynchronously, returns the request thread.

read_node_status(*name*, ***kwargs*)
read status of the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_node_status(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

read_node_status_with_http_info(*name*, ***kwargs*)
read status of the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_node_status_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

read_node_with_http_info(*name*, ***kwargs*)
read the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_node_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1Node

If the method is called asynchronously, returns the request thread.

read_persistent_volume(*name*, ***kwargs*)
read the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_persistent_volume(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

read_persistent_volume_status(*name*, ***kwargs*)
read status of the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_persistent_volume_status(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

read_persistent_volume_status_with_http_info(*name*, ***kwargs*)
read status of the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_persistent_volume_status_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

read_persistent_volume_with_http_info(*name*, ***kwargs*)
read the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_persistent_volume_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

replace_namespace(*name*, *body*, ***kwargs*)
replace the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespace(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param V1Namespace body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

replace_namespace_finalize(*name*, *body*, ***kwargs*)
replace finalize of the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespace_finalize(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param V1Namespace body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

replace_namespace_finalize_with_http_info(*name*, *body*, ***kwargs*)
replace finalize of the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespace_finalize_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param V1Namespace body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

replace_namespace_status(*name*, *body*, ***kwargs*)
replace status of the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespace_status(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param V1Namespace body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

replace_namespace_status_with_http_info(*name*, *body*, ***kwargs*)
replace status of the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespace_status_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param V1Namespace body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

replace_namespace_with_http_info(*name*, *body*, ***kwargs*)
replace the specified Namespace This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespace_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Namespace (required) :param V1Namespace body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Namespace

If the method is called asynchronously, returns the request thread.

replace_namespaced_config_map(*name*, *namespace*, *body*, ***kwargs*)
replace the specified ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_config_map(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ConfigMap (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ConfigMap body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ConfigMap

If the method is called asynchronously, returns the request thread.

replace_namespaced_config_map_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified ConfigMap This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_config_map_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ConfigMap (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ConfigMap body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ConfigMap

If the method is called asynchronously, returns the request thread.

replace_namespaced_endpoints(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_endpoints(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Endpoints (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Endpoints body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Endpoints

If the method is called asynchronously, returns the request thread.

replace_namespaced_endpoints_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Endpoints This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_endpoints_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Endpoints (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Endpoints body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Endpoints

If the method is called asynchronously, returns the request thread.

replace_namespaced_event(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_event(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Event (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Event body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Event

If the method is called asynchronously, returns the request thread.

replace_namespaced_event_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Event This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_event_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Event (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Event body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Event

If the method is called asynchronously, returns the request thread.

replace_namespaced_limit_range(*name*, *namespace*, *body*, ***kwargs*)
replace the specified LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_limit_range(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the LimitRange (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1LimitRange body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1LimitRange

If the method is called asynchronously, returns the request thread.

replace_namespaced_limit_range_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified LimitRange This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.replace_namespaced_limit_range_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the LimitRange (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1LimitRange body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1LimitRange

If the method is called asynchronously, returns the request thread.

replace_namespaced_persistent_volume_claim(*name*, *namespace*, *body*, ***kwargs*)

replace the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_persistent_volume_claim(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PersistentVolumeClaim body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

replace_namespaced_persistent_volume_claim_status(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_persistent_volume_claim_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PersistentVolumeClaim body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

replace_namespaced_persistent_volume_claim_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_persistent_volume_claim_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PersistentVolumeClaim body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

replace_namespaced_persistent_volume_claim_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified PersistentVolumeClaim This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_persistent_volume_claim_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolumeClaim (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PersistentVolumeClaim body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolumeClaim

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod(*name*, *namespace*, *body*, ***kwargs*)

replace the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Pod body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_status(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Pod body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Pod body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_template(*name*, *namespace*, *body*, ***kwargs*)

replace the specified PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_template(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodTemplate (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PodTemplate body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PodTemplate

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_template_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified PodTemplate This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_template_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodTemplate (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1PodTemplate body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PodTemplate

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Pod This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Pod (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Pod body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Pod

If the method is called asynchronously, returns the request thread.

replace_namespaced_replication_controller(*name*, *namespace*, *body*, ***kwargs*)
replace the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replication_controller(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ReplicationController body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

replace_namespaced_replication_controller_scale(*name*, *namespace*, *body*, ***kwargs*)
replace scale of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replication_controller_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_replication_controller_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace scale of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replication_controller_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_replication_controller_status(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.replace_namespaced_replication_controller_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ReplicationController body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

replace_namespaced_replication_controller_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replication_controller_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ReplicationController body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

replace_namespaced_replication_controller_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified ReplicationController This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replication_controller_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicationController (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ReplicationController body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ReplicationController

If the method is called asynchronously, returns the request thread.

replace_namespaced_resource_quota(*name*, *namespace*, *body*, ***kwargs*)

replace the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_resource_quota(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ResourceQuota body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

replace_namespaced_resource_quota_status(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_resource_quota_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ResourceQuota body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

replace_namespaced_resource_quota_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_resource_quota_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ResourceQuota body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

replace_namespaced_resource_quota_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified ResourceQuota This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_resource_quota_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ResourceQuota (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ResourceQuota body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ResourceQuota

If the method is called asynchronously, returns the request thread.

replace_namespaced_secret(*name*, *namespace*, *body*, ***kwargs*)

replace the specified Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_secret(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Secret (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Secret body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Secret

If the method is called asynchronously, returns the request thread.

replace_namespaced_secret_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified Secret This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_secret_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Secret (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Secret body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Secret

If the method is called asynchronously, returns the request thread.

replace_namespaced_service(*name*, *namespace*, *body*, ***kwargs*)

replace the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_service(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Service body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

replace_namespaced_service_account(*name*, *namespace*, *body*, ***kwargs*)
replace the specified ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_service_account(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ServiceAccount (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ServiceAccount body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ServiceAccount

If the method is called asynchronously, returns the request thread.

replace_namespaced_service_account_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified ServiceAccount This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_service_account_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ServiceAccount (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1ServiceAccount body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1ServiceAccount

If the method is called asynchronously, returns the request thread.

replace_namespaced_service_status(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_service_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Service body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

replace_namespaced_service_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_service_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Service body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

replace_namespaced_service_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Service This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_service_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Service (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1Service body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Service

If the method is called asynchronously, returns the request thread.

replace_node(*name*, *body*, ***kwargs*)
replace the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_node(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param V1Node body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

replace_node_status(*name*, *body*, ***kwargs*)
replace status of the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_node_status(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param V1Node body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

replace_node_status_with_http_info(*name*, *body*, ***kwargs*)
replace status of the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_node_status_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param V1Node body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

replace_node_with_http_info(*name*, *body*, ***kwargs*)
replace the specified Node This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_node_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Node (required) :param V1Node body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1Node

If the method is called asynchronously, returns the request thread.

replace_persistent_volume(*name*, *body*, ***kwargs*)
replace the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_persistent_volume(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param V1PersistentVolume body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

replace_persistent_volume_status(*name*, *body*, ***kwargs*)
replace status of the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_persistent_volume_status(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param V1PersistentVolume body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

replace_persistent_volume_status_with_http_info(*name*, *body*, ***kwargs*)
replace status of the specified PersistentVolume This method makes a synchronous HTTP request

by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_persistent_volume_status_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param V1PersistentVolume body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

replace_persistent_volume_with_http_info(*name*, *body*, ***kwargs*)
replace the specified PersistentVolume This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_persistent_volume_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PersistentVolume (required) :param V1PersistentVolume body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1PersistentVolume

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.extensions_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.extensions_api.**ExtensionsApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.extensions_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.extensions_v1beta1_api.**ExtensionsV1beta1Api**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_namespaced_daemon_set(*namespace*, *body*, ***kwargs*)
create a DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_daemon_set(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1DaemonSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

create_namespaced_daemon_set_with_http_info(*namespace*, *body*, ***kwargs*)
create a DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_daemon_set_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1DaemonSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

create_namespaced_deployment(*namespace*, *body*, ***kwargs*)
create a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_deployment(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

create_namespaced_deployment_rollback(*name*, *namespace*, *body*, ***kwargs*)
create rollback of a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_deployment_rollback(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DeploymentRollback (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1DeploymentRollback body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1DeploymentRollback

If the method is called asynchronously, returns the request thread.

create_namespaced_deployment_rollback_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
create rollback of a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_deployment_rollback_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DeploymentRollback (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param Exten-

sionsV1beta1DeploymentRollback body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1DeploymentRollback

If the method is called asynchronously, returns the request thread.

create_namespaced_deployment_with_http_info (*namespace*, *body*, ***kwargs*)
create a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_deployment_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

create_namespaced_ingress (*namespace*, *body*, ***kwargs*)
create an Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_ingress(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1Ingress body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

create_namespaced_ingress_with_http_info (*namespace*, *body*, ***kwargs*)
create an Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_ingress_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1Ingress body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

create_namespaced_network_policy (*namespace*, *body*, ***kwargs*)
create a NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_network_policy(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1NetworkPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1NetworkPolicy

If the method is called asynchronously, returns the request thread.

create_namespaced_network_policy_with_http_info (*namespace*, *body*, ***kwargs*)
create a NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_network_policy_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1NetworkPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1NetworkPolicy

If the method is called asynchronously, returns the request thread.

create_namespaced_replica_set(*namespace*, *body*, ***kwargs*)

create a ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_replica_set(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ReplicaSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

create_namespaced_replica_set_with_http_info(*namespace*, *body*, ***kwargs*)

create a ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_replica_set_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ReplicaSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

create_pod_security_policy(*body*, ***kwargs*)

create a PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_pod_security_policy(body, async=True) >>> result = thread.get()

:param async bool :param ExtensionsV1beta1PodSecurityPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

create_pod_security_policy_with_http_info(*body*, ***kwargs*)

create a PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_pod_security_policy_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param ExtensionsV1beta1PodSecurityPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_daemon_set(*namespace*, ***kwargs*)

delete collection of DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_daemon_set(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by

their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_daemon_set_with_http_info(*namespace*, ***kwargs*)

delete collection of DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_daemon_set_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version

of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_deployment(*namespace*, ***kwargs*)

delete collection of Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_deployment(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_deployment_with_http_info(*namespace*, ***kwargs*)

delete collection of Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_deployment_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects

(required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_ingress(*namespace*, ***kwargs*)

delete collection of Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_ingress(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and

clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_ingress_with_http_info(*namespace*, ***kwargs*)

delete collection of Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_ingress_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless

of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_network_policy(*namespace*, ***kwargs*)

delete collection of NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_network_policy(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_network_policy_with_http_info(*namespace*, ***kwargs*)

delete collection of NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_network_policy_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the

specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_replica_set(*namespace*, ***kwargs*)

delete collection of ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_replica_set(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will

be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_replica_set_with_http_info(*namespace*, ***kwargs*)

delete collection of ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_replica_set_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_pod_security_policy(***kwargs*)
delete collection of PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_pod_security_policy(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_pod_security_policy_with_http_info(***kwargs*)
delete collection of PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_pod_security_policy_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a

list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_daemon_set(*name*, *namespace*, *body*, ***kwargs*)

delete a DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_daemon_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_daemon_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_daemon_set_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent

objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_deployment(*name*, *namespace*, *body*, ***kwargs*)
delete a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_deployment(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_deployment_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
delete a Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_deployment_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_ingress(*name*, *namespace*, *body*, ***kwargs*)
delete an Ingress This method makes a synchronous HTTP request by default. To make an asynchronous

HTTP request, please pass async=True >>> thread = api.delete_namespaced_ingress(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_ingress_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete an Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_ingress_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_network_policy(*name*, *namespace*, *body*, ***kwargs*)

delete a NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_network_policy(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the NetworkPolicy (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param

str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_network_policy_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_network_policy_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the NetworkPolicy (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_replica_set(*name*, *namespace*, *body*, ***kwargs*)

delete a ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_replica_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_replica_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_replica_set_with_http_info(name, namespace, body, async=True) >>> result

= thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_pod_security_policy(*name*, *body*, ***kwargs*)

delete a PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_pod_security_policy(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_pod_security_policy_with_http_info(*name*, *body*, ***kwargs*)

delete a PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_pod_security_policy_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are:

'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_daemon_set_for_all_namespaces(***kwargs*)

list or watch objects of kind DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_daemon_set_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call,

regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1DaemonSetList

If the method is called asynchronously, returns the request thread.

list_daemon_set_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_daemon_set_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1DaemonSetList

If the method is called asynchronously, returns the request thread.

list_deployment_for_all_namespaces(***kwargs*)

list or watch objects of kind Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_deployment_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This

field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: ExtensionsV1beta1DeploymentList

If the method is called asynchronously, returns the request thread.

list_deployment_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_deployment_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that

is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: ExtensionsV1beta1DeploymentList

If the method is called asynchronously, returns the request thread.

list_ingress_for_all_namespaces(***kwargs*)

list or watch objects of kind Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_ingress_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1IngressList

If the method is called asynchronously, returns the request thread.

list_ingress_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Ingress This method makes a synchronous HTTP request by

default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_ingress_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1IngressList

If the method is called asynchronously, returns the request thread.

list_namespaced_daemon_set(*namespace*, ***kwargs*)

list or watch objects of kind DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_daemon_set(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return

for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1DaemonSetList

If the method is called asynchronously, returns the request thread.

list_namespaced_daemon_set_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_daemon_set_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then

the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1DaemonSetList

If the method is called asynchronously, returns the request thread.

list_namespaced_deployment(*namespace*, ***kwargs*)

list or watch objects of kind Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_deployment(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: ExtensionsV1beta1DeploymentList

If the method is called asynchronously, returns the request thread.

list_namespaced_deployment_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_deployment_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue

option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: ExtensionsV1beta1DeploymentList

If the method is called asynchronously, returns the request thread.

list_namespaced_ingress(*namespace*, ***kwargs*)

list or watch objects of kind Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_ingress(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and

clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1IngressList

If the method is called asynchronously, returns the request thread.

list_namespaced_ingress_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_ingress_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call,

regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1IngressList

If the method is called asynchronously, returns the request thread.

list_namespaced_network_policy(*namespace*, ***kwargs*)

list or watch objects of kind NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_network_policy(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1NetworkPolicyList

If the method is called asynchronously, returns the request thread.

list_namespaced_network_policy_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_network_policy_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the

specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1NetworkPolicyList

If the method is called asynchronously, returns the request thread.

list_namespaced_replica_set(*namespace*, ***kwargs*)

list or watch objects of kind ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_replica_set(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field

is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1ReplicaSetList

If the method is called asynchronously, returns the request thread.

list_namespaced_replica_set_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_replica_set_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1ReplicaSetList

If the method is called asynchronously, returns the request thread.

list_network_policy_for_all_namespaces(***kwargs*)

list or watch objects of kind NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_network_policy_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1NetworkPolicyList

If the method is called asynchronously, returns the request thread.

list_network_policy_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_network_policy_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true,

partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1NetworkPolicyList

If the method is called asynchronously, returns the request thread.

list_pod_security_policy(***kwargs*)

list or watch objects of kind PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_security_policy(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str

resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: ExtensionsV1beta1PodSecurityPolicyList

If the method is called asynchronously, returns the request thread.

list_pod_security_policy_with_http_info(***kwargs*)

list or watch objects of kind PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_security_policy_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: ExtensionsV1beta1PodSecurityPolicyList

If the method is called asynchronously, returns the request thread.

list_replica_set_for_all_namespaces(***kwargs*)

list or watch objects of kind ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_replica_set_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results

from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1ReplicaSetList

If the method is called asynchronously, returns the request thread.

list_replica_set_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_replica_set_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of

the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1ReplicaSetList

If the method is called asynchronously, returns the request thread.

patch_namespaced_daemon_set(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_daemon_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_daemon_set_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_daemon_set_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_daemon_set_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_daemon_set_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_daemon_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_daemon_set_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_scale(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

patch_namespaced_deployment_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_deployment_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

patch_namespaced_ingress(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_ingress(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

patch_namespaced_ingress_status(*name*, *namespace*, *body*, ***kwargs*)
partially update status of the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_ingress_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

patch_namespaced_ingress_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update status of the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_ingress_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

patch_namespaced_ingress_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_ingress_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

patch_namespaced_network_policy(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_network_policy(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the NetworkPolicy (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1NetworkPolicy

If the method is called asynchronously, returns the request thread.

patch_namespaced_network_policy_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_network_policy_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the NetworkPolicy (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1NetworkPolicy

If the method is called asynchronously, returns the request thread.

patch_namespaced_replica_set(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replica_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_replica_set_scale(*name*, *namespace*, *body*, ***kwargs*)
partially update scale of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replica_set_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_replica_set_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update scale of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replica_set_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_replica_set_status(*name*, *namespace*, *body*, ***kwargs*)
partially update status of the specified ReplicaSet This method makes a synchronous HTTP re-

quest by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replica_set_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_replica_set_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replica_set_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_replica_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replica_set_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

patch_namespaced_replication_controller_dummy_scale(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified ReplicationControllerDummy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replication_controller_dummy_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_namespaced_replication_controller_dummy_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update scale of the specified ReplicationControllerDummy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_replication_controller_dummy_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

patch_pod_security_policy(*name*, *body*, ***kwargs*)

partially update the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_pod_security_policy(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

patch_pod_security_policy_with_http_info(*name*, *body*, ***kwargs*)

partially update the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_pod_security_policy_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

read_namespaced_daemon_set(*name*, *namespace*, ***kwargs*)

read the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_daemon_set(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

read_namespaced_daemon_set_status(*name*, *namespace*, ***kwargs*)

read status of the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_daemon_set_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

read_namespaced_daemon_set_status_with_http_info(*name*, *namespace*, ***kwargs*)

read status of the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_daemon_set_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

read_namespaced_daemon_set_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_daemon_set_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment(*name*, *namespace*, ***kwargs*)
read the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_scale(*name*, *namespace*, ***kwargs*)
read scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_scale(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_scale_with_http_info(*name*, *namespace*, ***kwargs*)
read scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_scale_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_status(*name*, *namespace*, ***kwargs*)
read status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

read_namespaced_deployment_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_deployment_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

read_namespaced_ingress(*name*, *namespace*, ***kwargs*)
read the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_ingress(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

read_namespaced_ingress_status(*name*, *namespace*, ***kwargs*)
read status of the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_ingress_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

read_namespaced_ingress_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_ingress_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

read_namespaced_ingress_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_ingress_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

read_namespaced_network_policy(*name*, *namespace*, ***kwargs*)
read the specified NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_network_policy(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the NetworkPolicy (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1NetworkPolicy

If the method is called asynchronously, returns the request thread.

read_namespaced_network_policy_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_network_policy_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the NetworkPolicy (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1NetworkPolicy

If the method is called asynchronously, returns the request thread.

read_namespaced_replica_set(*name*, *namespace*, ***kwargs*)
read the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replica_set(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

read_namespaced_replica_set_scale(*name*, *namespace*, ***kwargs*)
read scale of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replica_set_scale(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty

printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_replica_set_scale_with_http_info(*name*, *namespace*, ***kwargs*)
read scale of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replica_set_scale_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_replica_set_status(*name*, *namespace*, ***kwargs*)
read status of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replica_set_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

read_namespaced_replica_set_status_with_http_info(*name*, *namespace*, ***kwargs*)
read status of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replica_set_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

read_namespaced_replica_set_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replica_set_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

read_namespaced_replication_controller_dummy_scale(*name*, *namespace*, ***kwargs*)
read scale of the specified ReplicationControllerDummy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replication_controller_dummy_scale(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_namespaced_replication_controller_dummy_scale_with_http_info(*name*, *namespace*, ***kwargs*)

read scale of the specified ReplicationControllerDummy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_replication_controller_dummy_scale_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

read_pod_security_policy(*name*, ***kwargs*)

read the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_pod_security_policy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: ExtensionsV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

read_pod_security_policy_with_http_info(*name*, ***kwargs*)

read the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_pod_security_policy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: ExtensionsV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

replace_namespaced_daemon_set(*name*, *namespace*, *body*, ***kwargs*)

replace the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_daemon_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1DaemonSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_daemon_set_status(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_daemon_set_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1DaemonSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_daemon_set_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_daemon_set_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1DaemonSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_daemon_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified DaemonSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_daemon_set_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the DaemonSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1DaemonSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1DaemonSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment(*name*, *namespace*, *body*, ***kwargs*)

replace the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_scale(*name*, *namespace*, *body*, ***kwargs*)

replace scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace scale of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_status(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

replace_namespaced_deployment_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Deployment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_deployment_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Deployment (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Deployment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Deployment

If the method is called asynchronously, returns the request thread.

replace_namespaced_ingress(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_ingress(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1Ingress body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

replace_namespaced_ingress_status(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_ingress_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1Ingress body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

replace_namespaced_ingress_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_ingress_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1Ingress body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

replace_namespaced_ingress_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified Ingress This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_ingress_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Ingress (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1Ingress body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1Ingress

If the method is called asynchronously, returns the request thread.

replace_namespaced_network_policy(*name*, *namespace*, *body*, ***kwargs*)

replace the specified NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_network_policy(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the NetworkPolicy (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1NetworkPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1NetworkPolicy

If the method is called asynchronously, returns the request thread.

replace_namespaced_network_policy_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified NetworkPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_network_policy_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the NetworkPolicy (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1NetworkPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1NetworkPolicy

If the method is called asynchronously, returns the request thread.

replace_namespaced_replica_set(*name*, *namespace*, *body*, ***kwargs*)

replace the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replica_set(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ReplicaSet body: (required)

:param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_replica_set_scale(*name*, *namespace*, *body*, ***kwargs*)
replace scale of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replica_set_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_replica_set_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace scale of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replica_set_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_replica_set_status(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replica_set_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ReplicaSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_replica_set_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace status of the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replica_set_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ReplicaSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_replica_set_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified ReplicaSet This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replica_set_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ReplicaSet (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1ReplicaSet body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1ReplicaSet

If the method is called asynchronously, returns the request thread.

replace_namespaced_replication_controller_dummy_scale(*name*, *namespace*, *body*, ***kwargs*)

replace scale of the specified ReplicationControllerDummy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replication_controller_dummy_scale(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_namespaced_replication_controller_dummy_scale_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace scale of the specified ReplicationControllerDummy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_replication_controller_dummy_scale_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Scale (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param ExtensionsV1beta1Scale body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1Scale

If the method is called asynchronously, returns the request thread.

replace_pod_security_policy(*name*, *body*, ***kwargs*)

replace the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_pod_security_policy(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param ExtensionsV1beta1PodSecurityPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

replace_pod_security_policy_with_http_info(*name*, *body*, ***kwargs*)

replace the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_pod_security_policy_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param ExtensionsV1beta1PodSecurityPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: ExtensionsV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.logs_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.logs_api.**LogsApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

log_file_handler(*logpath*, ***kwargs*)
This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.log_file_handler(logpath, async=True) >>> result = thread.get()

:param async bool :param str logpath: path to the log (required) :return: None

If the method is called asynchronously, returns the request thread.

log_file_handler_with_http_info(*logpath*, ***kwargs*)
This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.log_file_handler_with_http_info(logpath, async=True) >>> result = thread.get()

:param async bool :param str logpath: path to the log (required) :return: None

If the method is called asynchronously, returns the request thread.

log_file_list_handler(***kwargs*)
This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.log_file_list_handler(async=True) >>> result = thread.get()

:param async bool :return: None

If the method is called asynchronously, returns the request thread.

log_file_list_handler_with_http_info(***kwargs*)
This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.log_file_list_handler_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: None

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.policy_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.policy_api.**PolicyApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.policy_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.policy_v1beta1_api.**PolicyV1beta1Api**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_namespaced_pod_disruption_budget(*namespace*, *body*, ***kwargs*)
create a PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_disruption_budget(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1PodDisruptionBudget body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

create_namespaced_pod_disruption_budget_with_http_info(*namespace*, *body*, ***kwargs*)
create a PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_pod_disruption_budget_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1PodDisruptionBudget body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

create_pod_security_policy(*body*, ***kwargs*)
create a PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_pod_security_policy(body, async=True) >>> result = thread.get()

:param async bool :param PolicyV1beta1PodSecurityPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: PolicyV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

create_pod_security_policy_with_http_info(*body*, ***kwargs*)
create a PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_pod_security_policy_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param PolicyV1beta1PodSecurityPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: PolicyV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_pod_disruption_budget(*namespace*, ***kwargs*)
delete collection of PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_pod_disruption_budget(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_pod_disruption_budget_with_http_info(*namespace*, ***kwargs*)
delete collection of PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_pod_disruption_budget_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_pod_security_policy(***kwargs*)

delete collection of PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_pod_security_policy(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and

clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_pod_security_policy_with_http_info(***kwargs*)

delete collection of PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_pod_security_policy_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_pod_disruption_budget(*name*, *namespace*, *body*, ***kwargs*)

delete a PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_pod_disruption_budget(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_pod_disruption_budget_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_pod_disruption_budget_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_pod_security_policy(*name*, *body*, ***kwargs*)

delete a PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_pod_security_policy(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period

for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_pod_security_policy_with_http_info(*name*, *body*, ***kwargs*)

delete a PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_pod_security_policy_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_namespaced_pod_disruption_budget(*namespace*, ***kwargs*)

list or watch objects of kind PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_pod_disruption_budget(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue

option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1PodDisruptionBudgetList

If the method is called asynchronously, returns the request thread.

list_namespaced_pod_disruption_budget_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_pod_disruption_budget_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer

than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1PodDisruptionBudgetList

If the method is called asynchronously, returns the request thread.

list_pod_disruption_budget_for_all_namespaces(***kwargs*)

list or watch objects of kind PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_disruption_budget_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call,

regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1PodDisruptionBudgetList

If the method is called asynchronously, returns the request thread.

list_pod_disruption_budget_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_disruption_budget_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1PodDisruptionBudgetList

If the method is called asynchronously, returns the request thread.

list_pod_security_policy(***kwargs*)

list or watch objects of kind PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_security_policy(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or

a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: PolicyV1beta1PodSecurityPolicyList

If the method is called asynchronously, returns the request thread.

list_pod_security_policy_with_http_info(***kwargs*)

list or watch objects of kind PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_pod_security_policy_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted

after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: PolicyV1beta1PodSecurityPolicyList

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_disruption_budget(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_disruption_budget(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_disruption_budget_status(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_disruption_budget_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_disruption_budget_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update status of the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_disruption_budget_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

patch_namespaced_pod_disruption_budget_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

partially update the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_pod_disruption_budget_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

patch_pod_security_policy(*name*, *body*, ***kwargs*)

partially update the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_pod_security_policy(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: PolicyV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

patch_pod_security_policy_with_http_info(*name*, *body*, ***kwargs*)

partially update the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_pod_security_policy_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: PolicyV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_disruption_budget(*name*, *namespace*, ***kwargs*)

read the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_disruption_budget(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_disruption_budget_status(*name*, *namespace*, ***kwargs*)

read status of the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_disruption_budget_status(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_disruption_budget_status_with_http_info(*name*, *namespace*, ***kwargs*)

read status of the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_disruption_budget_status_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

read_namespaced_pod_disruption_budget_with_http_info(*name*, *namespace*, ***kwargs*)

read the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_pod_disruption_budget_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

read_pod_security_policy(*name*, ***kwargs*)

read the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_pod_security_policy(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: PolicyV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

read_pod_security_policy_with_http_info(*name*, ***kwargs*)

read the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_pod_security_policy_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: PolicyV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_disruption_budget(*name*, *namespace*, *body*, ***kwargs*)

replace the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_disruption_budget(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1PodDisruptionBudget body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_disruption_budget_status(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified PodDisruptionBudget This method makes a synchronous HTTP re-

quest by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_disruption_budget_status(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1PodDisruptionBudget body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_disruption_budget_status_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace status of the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_disruption_budget_status_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1PodDisruptionBudget body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

replace_namespaced_pod_disruption_budget_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

replace the specified PodDisruptionBudget This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_pod_disruption_budget_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodDisruptionBudget (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1beta1PodDisruptionBudget body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1PodDisruptionBudget

If the method is called asynchronously, returns the request thread.

replace_pod_security_policy(*name*, *body*, ***kwargs*)

replace the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_pod_security_policy(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param PolicyV1beta1PodSecurityPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: PolicyV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

replace_pod_security_policy_with_http_info(*name*, *body*, ***kwargs*)

replace the specified PodSecurityPolicy This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_pod_security_policy_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the PodSecurityPolicy (required) :param PolicyV1beta1PodSecurityPolicy body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: PolicyV1beta1PodSecurityPolicy

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.rbac_authorization_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.rbac_authorization_api.**RbacAuthorizationApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.rbac_authorization_v1alpha1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.rbac_authorization_v1alpha1_api.**RbacAuthorizationV1alpha1Api**(*
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_cluster_role(*body*, ***kwargs*)
create a ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_cluster_role(body, async=True) >>> result = thread.get()

:param async bool :param V1alpha1ClusterRole body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRole

If the method is called asynchronously, returns the request thread.

create_cluster_role_binding(*body*, ***kwargs*)
create a ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_cluster_role_binding(body, async=True) >>> result = thread.get()

:param async bool :param V1alpha1ClusterRoleBinding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRoleBinding

If the method is called asynchronously, returns the request thread.

create_cluster_role_binding_with_http_info(*body*, ***kwargs*)
create a ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_cluster_role_binding_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1alpha1ClusterRoleBinding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRoleBinding

If the method is called asynchronously, returns the request thread.

create_cluster_role_with_http_info(*body*, ***kwargs*)
create a ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_cluster_role_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1alpha1ClusterRole body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRole

If the method is called asynchronously, returns the request thread.

create_namespaced_role(*namespace*, *body*, ***kwargs*)
create a Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_role(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1alpha1Role body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1Role

If the method is called asynchronously, returns the request thread.

create_namespaced_role_binding(*namespace*, *body*, ***kwargs*)
create a RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_role_binding(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1alpha1RoleBinding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1RoleBinding

If the method is called asynchronously, returns the request thread.

create_namespaced_role_binding_with_http_info(*namespace*, *body*, ***kwargs*)
create a RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_role_binding_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1alpha1RoleBinding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1RoleBinding

If the method is called asynchronously, returns the request thread.

create_namespaced_role_with_http_info(*namespace*, *body*, ***kwargs*)
create a Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_namespaced_role_with_http_info(namespace, body, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1alpha1Role body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1Role

If the method is called asynchronously, returns the request thread.

delete_cluster_role(*name*, *body*, ***kwargs*)
delete a ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_cluster_role(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRole (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_cluster_role_binding(*name*, *body*, ***kwargs*)
delete a ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_cluster_role_binding(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRoleBinding (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_cluster_role_binding_with_http_info(*name*, *body*, ***kwargs*)
delete a ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.delete_cluster_role_binding_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRoleBinding (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_cluster_role_with_http_info(*name*, *body*, ***kwargs*)

delete a ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_cluster_role_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRole (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_cluster_role(***kwargs*)

delete collection of ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_cluster_role(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a

list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_cluster_role_binding(***kwargs*)

delete collection of ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_cluster_role_binding(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv.

:param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_cluster_role_binding_with_http_info(***kwargs*)

delete collection of ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_cluster_role_binding_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_cluster_role_with_http_info(***kwargs*)

delete collection of ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_cluster_role_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true.

Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_role(*namespace*, ***kwargs*)

delete collection of Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_role(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of

a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_role_binding(*namespace*, ***kwargs*)

delete collection of RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_role_binding(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_role_binding_with_http_info(*namespace*, ***kwargs*)

delete collection of RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_role_binding_with_http_info(namespace, async=True) >>> result =

thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_namespaced_role_with_http_info(*namespace*, ***kwargs*)

delete collection of Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_namespaced_role_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that

can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_role(*name*, *namespace*, *body*, ***kwargs*)

delete a Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_role(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Role (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_role_binding(*name*, *namespace*, *body*, ***kwargs*)

delete a RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_role_binding(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the RoleBinding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param

str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_role_binding_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_role_binding_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the RoleBinding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_namespaced_role_with_http_info(*name*, *namespace*, *body*, ***kwargs*)

delete a Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_namespaced_role_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Role (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)

get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_cluster_role(***kwargs*)
list or watch objects of kind ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_cluster_role(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1ClusterRoleList

If the method is called asynchronously, returns the request thread.

list_cluster_role_binding(***kwargs*)
list or watch objects of kind ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_cluster_role_binding(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1ClusterRoleBindingList

If the method is called asynchronously, returns the request thread.

list_cluster_role_binding_with_http_info(***kwargs*)

list or watch objects of kind ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_cluster_role_binding_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and

clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1ClusterRoleBindingList

If the method is called asynchronously, returns the request thread.

list_cluster_role_with_http_info(***kwargs*)

list or watch objects of kind ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_cluster_role_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources

and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1ClusterRoleList

If the method is called asynchronously, returns the request thread.

list_namespaced_role(*namespace*, ***kwargs*)

list or watch objects of kind Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_role(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1RoleList

If the method is called asynchronously, returns the request thread.

list_namespaced_role_binding(*namespace*, ***kwargs*)

list or watch objects of kind RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_role_binding(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or

a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1RoleBindingList

If the method is called asynchronously, returns the request thread.

list_namespaced_role_binding_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_role_binding_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will

be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1RoleBindingList

If the method is called asynchronously, returns the request thread.

list_namespaced_role_with_http_info(*namespace*, ***kwargs*)

list or watch objects of kind Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_namespaced_role_with_http_info(namespace, async=True) >>> result = thread.get()

:param async bool :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1RoleList

If the method is called asynchronously, returns the request thread.

list_role_binding_for_all_namespaces(***kwargs*)

list or watch objects of kind RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_role_binding_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1RoleBindingList

If the method is called asynchronously, returns the request thread.

list_role_binding_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_role_binding_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum

number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1RoleBindingList

If the method is called asynchronously, returns the request thread.

list_role_for_all_namespaces(***kwargs*)

list or watch objects of kind Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_role_for_all_namespaces(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then

the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1RoleList

If the method is called asynchronously, returns the request thread.

list_role_for_all_namespaces_with_http_info(***kwargs*)

list or watch objects of kind Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_role_for_all_namespaces_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str pretty: If 'true', then the output is pretty printed. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1alpha1RoleList

If the method is called asynchronously, returns the request thread.

patch_cluster_role(*name*, *body*, ***kwargs*)

partially update the specified ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_cluster_role(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRole (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRole

If the method is called asynchronously, returns the request thread.

patch_cluster_role_binding(*name*, *body*, ***kwargs*)
partially update the specified ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_cluster_role_binding(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRoleBinding (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRoleBinding

If the method is called asynchronously, returns the request thread.

patch_cluster_role_binding_with_http_info(*name*, *body*, ***kwargs*)
partially update the specified ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_cluster_role_binding_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRoleBinding (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRoleBinding

If the method is called asynchronously, returns the request thread.

patch_cluster_role_with_http_info(*name*, *body*, ***kwargs*)
partially update the specified ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_cluster_role_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRole (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRole

If the method is called asynchronously, returns the request thread.

patch_namespaced_role(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_role(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Role (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1Role

If the method is called asynchronously, returns the request thread.

patch_namespaced_role_binding(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_role_binding(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the RoleBinding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1RoleBinding

If the method is called asynchronously, returns the request thread.

patch_namespaced_role_binding_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_role_binding_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the RoleBinding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str

pretty: If 'true', then the output is pretty printed. :return: V1alpha1RoleBinding

If the method is called asynchronously, returns the request thread.

patch_namespaced_role_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
partially update the specified Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_namespaced_role_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Role (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1Role

If the method is called asynchronously, returns the request thread.

read_cluster_role(*name*, ***kwargs*)
read the specified ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_cluster_role(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRole (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRole

If the method is called asynchronously, returns the request thread.

read_cluster_role_binding(*name*, ***kwargs*)
read the specified ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_cluster_role_binding(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRoleBinding (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRoleBinding

If the method is called asynchronously, returns the request thread.

read_cluster_role_binding_with_http_info(*name*, ***kwargs*)
read the specified ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_cluster_role_binding_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRoleBinding (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRoleBinding

If the method is called asynchronously, returns the request thread.

read_cluster_role_with_http_info(*name*, ***kwargs*)
read the specified ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_cluster_role_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRole (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRole

If the method is called asynchronously, returns the request thread.

read_namespaced_role(*name*, *namespace*, ***kwargs*)
read the specified Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_role(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Role (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1Role

If the method is called asynchronously, returns the request thread.

read_namespaced_role_binding(*name*, *namespace*, ***kwargs*)
read the specified RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_role_binding(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the RoleBinding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1RoleBinding

If the method is called asynchronously, returns the request thread.

read_namespaced_role_binding_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_role_binding_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the RoleBinding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1RoleBinding

If the method is called asynchronously, returns the request thread.

read_namespaced_role_with_http_info(*name*, *namespace*, ***kwargs*)
read the specified Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_namespaced_role_with_http_info(name, namespace, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Role (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1Role

If the method is called asynchronously, returns the request thread.

replace_cluster_role(*name*, *body*, ***kwargs*)
replace the specified ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_cluster_role(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRole (required) :param V1alpha1ClusterRole body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRole

If the method is called asynchronously, returns the request thread.

replace_cluster_role_binding(*name*, *body*, ***kwargs*)
replace the specified ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_cluster_role_binding(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRoleBinding (required) :param V1alpha1ClusterRoleBinding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRoleBinding

If the method is called asynchronously, returns the request thread.

replace_cluster_role_binding_with_http_info(*name*, *body*, ***kwargs*)
replace the specified ClusterRoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_cluster_role_binding_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRoleBinding (required) :param V1alpha1ClusterRoleBinding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRoleBinding

If the method is called asynchronously, returns the request thread.

replace_cluster_role_with_http_info(*name*, *body*, ***kwargs*)
replace the specified ClusterRole This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_cluster_role_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the ClusterRole (required) :param V1alpha1ClusterRole body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1ClusterRole

If the method is called asynchronously, returns the request thread.

replace_namespaced_role(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_role(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Role (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1alpha1Role body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1Role

If the method is called asynchronously, returns the request thread.

replace_namespaced_role_binding(*name*, *namespace*, *body*, ***kwargs*)
replace the specified RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_role_binding(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the RoleBinding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1alpha1RoleBinding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1RoleBinding

If the method is called asynchronously, returns the request thread.

replace_namespaced_role_binding_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified RoleBinding This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_role_binding_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the RoleBinding (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1alpha1RoleBinding body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1RoleBinding

If the method is called asynchronously, returns the request thread.

replace_namespaced_role_with_http_info(*name*, *namespace*, *body*, ***kwargs*)
replace the specified Role This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_namespaced_role_with_http_info(name, namespace, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the Role (required) :param str namespace: object name and auth scope, such as for teams and projects (required) :param V1alpha1Role body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1alpha1Role

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.storage_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.storage_api.**StorageApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_api_group(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

get_api_group_with_http_info(***kwargs*)
get information of a group This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_group_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIGroup

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.storage_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.storage_v1beta1_api.**StorageV1beta1Api**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

create_storage_class(*body*, ***kwargs*)
create a StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_storage_class(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1StorageClass body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StorageClass

If the method is called asynchronously, returns the request thread.

create_storage_class_with_http_info(*body*, ***kwargs*)
create a StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_storage_class_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1StorageClass body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StorageClass

If the method is called asynchronously, returns the request thread.

create_volume_attachment(*body*, ***kwargs*)
create a VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_volume_attachment(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1VolumeAttachment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1VolumeAttachment

If the method is called asynchronously, returns the request thread.

create_volume_attachment_with_http_info(*body*, ***kwargs*)
create a VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.create_volume_attachment_with_http_info(body, async=True) >>> result = thread.get()

:param async bool :param V1beta1VolumeAttachment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1VolumeAttachment

If the method is called asynchronously, returns the request thread.

delete_collection_storage_class(***kwargs*)
delete collection of StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_storage_class(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted

after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_storage_class_with_http_info(***kwargs*)

delete collection of StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_storage_class_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_volume_attachment(***kwargs*)

delete collection of VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread =

api.delete_collection_volume_attachment(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_collection_volume_attachment_with_http_info(***kwargs*)

delete collection of VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_collection_volume_attachment_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and

clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_storage_class(*name*, *body*, ***kwargs*)

delete a StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_storage_class(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StorageClass (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_storage_class_with_http_info(*name*, *body*, ***kwargs*)

delete a StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_storage_class_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StorageClass (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background;

'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_volume_attachment(*name*, *body*, ***kwargs*)
delete a VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_volume_attachment(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the VolumeAttachment (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

delete_volume_attachment_with_http_info(*name*, *body*, ***kwargs*)
delete a VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.delete_volume_attachment_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the VolumeAttachment (required) :param V1DeleteOptions body: (required) :param str pretty: If 'true', then the output is pretty printed. :param int grace_period_seconds: The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately. :param bool orphan_dependents: Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both. :param str propagation_policy: Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground. :return: V1Status

If the method is called asynchronously, returns the request thread.

get_api_resources(***kwargs*)
get available resources This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

get_api_resources_with_http_info(***kwargs*)
get available resources This method makes a synchronous HTTP request by de-

fault. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_api_resources_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: V1APIResourceList

If the method is called asynchronously, returns the request thread.

list_storage_class(***kwargs*)

list or watch objects of kind StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_storage_class(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1StorageClassList

If the method is called asynchronously, returns the request thread.

list_storage_class_with_http_info(***kwargs*)

list or watch objects of kind StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_storage_class_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or

a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1StorageClassList

If the method is called asynchronously, returns the request thread.

list_volume_attachment(***kwargs*)

list or watch objects of kind VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_volume_attachment(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted

after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1VolumeAttachmentList

If the method is called asynchronously, returns the request thread.

list_volume_attachment_with_http_info(***kwargs*)

list or watch objects of kind VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.list_volume_attachment_with_http_info(async=True) >>> result = thread.get()

:param async bool :param str pretty: If 'true', then the output is pretty printed. :param str _continue: The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server the server will respond with a 410 ResourceExpired error indicating the client must restart their list without the continue field. This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications. :param str field_selector: A selector to restrict the list of returned objects by their fields. Defaults to everything. :param bool include_uninitialized: If true, partially initialized resources are included in the response. :param str label_selector: A selector to restrict the list of returned objects by their labels. Defaults to everything. :param int limit: limit is a maximum number of responses to return for a list call. If more items exist, the server will set the *continue* field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned. :param str resource_version: When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it's 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv. :param int timeout_seconds: Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity. :param bool watch: Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion. :return: V1beta1VolumeAttachmentList

If the method is called asynchronously, returns the request thread.

patch_storage_class(*name*, *body*, ***kwargs*)

partially update the specified StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_storage_class(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StorageClass (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StorageClass

If the method is called asynchronously, returns the request thread.

patch_storage_class_with_http_info(*name*, *body*, ***kwargs*)
partially update the specified StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_storage_class_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StorageClass (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StorageClass

If the method is called asynchronously, returns the request thread.

patch_volume_attachment(*name*, *body*, ***kwargs*)
partially update the specified VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_volume_attachment(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the VolumeAttachment (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1VolumeAttachment

If the method is called asynchronously, returns the request thread.

patch_volume_attachment_with_http_info(*name*, *body*, ***kwargs*)
partially update the specified VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.patch_volume_attachment_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the VolumeAttachment (required) :param object body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1VolumeAttachment

If the method is called asynchronously, returns the request thread.

read_storage_class(*name*, ***kwargs*)
read the specified StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_storage_class(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StorageClass (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1StorageClass

If the method is called asynchronously, returns the request thread.

read_storage_class_with_http_info(*name*, ***kwargs*)
read the specified StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_storage_class_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StorageClass (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1StorageClass

If the method is called asynchronously, returns the request thread.

read_volume_attachment(*name*, ***kwargs*)
read the specified VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_volume_attachment(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the VolumeAttachment (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1VolumeAttachment

If the method is called asynchronously, returns the request thread.

read_volume_attachment_with_http_info(*name*, ***kwargs*)
read the specified VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.read_volume_attachment_with_http_info(name, async=True) >>> result = thread.get()

:param async bool :param str name: name of the VolumeAttachment (required) :param str pretty: If 'true', then the output is pretty printed. :param bool exact: Should the export be exact. Exact export maintains cluster-specific fields like 'Namespace'. :param bool export: Should this value be exported. Export strips fields that a user can not specify. :return: V1beta1VolumeAttachment

If the method is called asynchronously, returns the request thread.

replace_storage_class(*name*, *body*, ***kwargs*)
replace the specified StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_storage_class(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StorageClass (required) :param V1beta1StorageClass body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StorageClass

If the method is called asynchronously, returns the request thread.

replace_storage_class_with_http_info(*name*, *body*, ***kwargs*)
replace the specified StorageClass This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_storage_class_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the StorageClass (required) :param V1beta1StorageClass body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1StorageClass

If the method is called asynchronously, returns the request thread.

replace_volume_attachment(*name*, *body*, ***kwargs*)
replace the specified VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_volume_attachment(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the VolumeAttachment (required) :param V1beta1VolumeAttachment body: (required) :param str pretty: If 'true', then the output is pretty printed. :return: V1beta1VolumeAttachment

If the method is called asynchronously, returns the request thread.

replace_volume_attachment_with_http_info(*name*, *body*, ***kwargs*)
replace the specified VolumeAttachment This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.replace_volume_attachment_with_http_info(name, body, async=True) >>> result = thread.get()

:param async bool :param str name: name of the VolumeAttachment (required) :param V1beta1VolumeAttachment body: (required) :param str pretty: If 'true', then the output is pretty printed.

:return: V1beta1VolumeAttachment

If the method is called asynchronously, returns the request thread.

kubernetes.client.apis.version_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.apis.version_api.**VersionApi**(*api_client=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually. Ref: https://github.com/swagger-api/swagger-codegen

get_code(**kwargs*)
get the code version This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_code(async=True) >>> result = thread.get()

:param async bool :return: VersionInfo

If the method is called asynchronously, returns the request thread.

get_code_with_http_info(**kwargs*)
get the code version This method makes a synchronous HTTP request by default. To make an asynchronous HTTP request, please pass async=True >>> thread = api.get_code_with_http_info(async=True) >>> result = thread.get()

:param async bool :return: VersionInfo

If the method is called asynchronously, returns the request thread.

Module contents

kubernetes.client.models package

Submodules

kubernetes.client.models.intstr_int_or_string module

kubernetes.client.models.resource_quantity module

kubernetes.client.models.runtime_raw_extension module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.runtime_raw_extension.**RuntimeRawExtension**(*raw=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'raw': 'Raw'}

raw
Gets the raw of this RuntimeRawExtension. Raw is the underlying serialization of this object.

Returns The raw of this RuntimeRawExtension.

Return type str

swagger_types = {'raw': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.unversioned_api_group module

kubernetes.client.models.unversioned_api_group_list module

kubernetes.client.models.unversioned_api_resource module

kubernetes.client.models.unversioned_api_resource_list module

kubernetes.client.models.unversioned_api_versions module

kubernetes.client.models.unversioned_group_version_for_discovery module

kubernetes.client.models.unversioned_label_selector module

kubernetes.client.models.unversioned_label_selector_requirement module

kubernetes.client.models.unversioned_list_meta module

kubernetes.client.models.unversioned_server_address_by_client_cidr module

kubernetes.client.models.unversioned_status module

kubernetes.client.models.unversioned_status_cause module

kubernetes.client.models.unversioned_status_details module

kubernetes.client.models.unversioned_time module

kubernetes.client.models.v1_attached_volume module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_attached_volume.**V1AttachedVolume**(*device_path=None*, *name=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'device_path': 'devicePath', 'name': 'name'}

device_path

Gets the device_path of this V1AttachedVolume. DevicePath represents the device path where the volume should be available

Returns The device_path of this V1AttachedVolume.

Return type str

name
Gets the name of this V1AttachedVolume. Name of the attached volume

Returns The name of this V1AttachedVolume.

Return type str

swagger_types = {'device_path': 'str', 'name': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_aws_elastic_block_store_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_aws_elastic_block_store_volume_source.**V1AWSElasticBlockSt**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'fs_type': 'fsType', 'partition': 'partition', 'read_only': 'readO

fs_type
Gets the fs_type of this V1AWSElasticBlockStoreVolumeSource. Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Returns The fs_type of this V1AWSElasticBlockStoreVolumeSource.

Return type str

partition
Gets the partition of this V1AWSElasticBlockStoreVolumeSource. The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).

Returns The partition of this V1AWSElasticBlockStoreVolumeSource.

Return type int

read_only
Gets the read_only of this V1AWSElasticBlockStoreVolumeSource. Specify "true" to force and set the ReadOnly property in VolumeMounts to "true". If omitted, the default is "false". More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Returns The read_only of this V1AWSElasticBlockStoreVolumeSource.

Return type bool

swagger_types = {'fs_type': 'str', 'partition': 'int', 'read_only': 'bool', 'volume

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

volume_id
Gets the volume_id of this V1AWSElasticBlockStoreVolumeSource. Unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Returns The volume_id of this V1AWSElasticBlockStoreVolumeSource.

Return type str

kubernetes.client.models.v1_azure_disk_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_azure_disk_volume_source.**V1AzureDiskVolumeSource**(*caching_m*
disk_name
disk_uri=N
fs_type=No
kind=None
read_only=

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'caching_mode': 'cachingMode', 'disk_name': 'diskName', 'disk_uri':

caching_mode
Gets the caching_mode of this V1AzureDiskVolumeSource. Host Caching mode: None, Read Only, Read Write.

Returns The caching_mode of this V1AzureDiskVolumeSource.

Return type str

disk_name
Gets the disk_name of this V1AzureDiskVolumeSource. The Name of the data disk in the blob storage

Returns The disk_name of this V1AzureDiskVolumeSource.

Return type str

disk_uri
Gets the disk_uri of this V1AzureDiskVolumeSource. The URI the data disk in the blob storage

Returns The disk_uri of this V1AzureDiskVolumeSource.

Return type str

fs_type
Gets the fs_type of this V1AzureDiskVolumeSource. Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

Returns The fs_type of this V1AzureDiskVolumeSource.

Return type str

kind
Gets the kind of this V1AzureDiskVolumeSource. Expected values Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared

Returns The kind of this V1AzureDiskVolumeSource.

Return type str

read_only
Gets the read_only of this V1AzureDiskVolumeSource. Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Returns The read_only of this V1AzureDiskVolumeSource.

Return type bool

swagger_types = {'caching_mode': 'str', 'disk_name': 'str', 'disk_uri': 'str', 'fs_

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_azure_file_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_azure_file_volume_source.**V1AzureFileVolumeSource**(*read_only=*
se-
cret_name=
share_nam

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'read_only': 'readOnly', 'secret_name': 'secretName', 'share_name':

read_only
Gets the read_only of this V1AzureFileVolumeSource. Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Returns The read_only of this V1AzureFileVolumeSource.

Return type bool

secret_name
Gets the secret_name of this V1AzureFileVolumeSource. the name of secret that contains Azure Storage Account Name and Key

Returns The secret_name of this V1AzureFileVolumeSource.

Return type str

share_name
Gets the share_name of this V1AzureFileVolumeSource. Share Name

Returns The share_name of this V1AzureFileVolumeSource.

Return type str

swagger_types = {'read_only': 'bool', 'secret_name': 'str', 'share_name': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_binding module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_binding.**V1Binding**(*api_version=None*, *kind=None*, *metadata=None*, *target=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1Binding. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Binding.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1Binding. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Binding.

Return type str

metadata
Gets the metadata of this V1Binding. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Binding.

Return type *V1ObjectMeta*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

target
Gets the target of this V1Binding. The target object that you want to bind to the standard object.

Returns The target of this V1Binding.

Return type *V1ObjectReference*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_capabilities module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_capabilities.**V1Capabilities**(*add=None*, *drop=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

add
Gets the add of this V1Capabilities. Added capabilities

Returns The add of this V1Capabilities.

Return type list[str]

attribute_map = {'add': 'add', 'drop': 'drop'}

drop
Gets the drop of this V1Capabilities. Removed capabilities

Returns The drop of this V1Capabilities.

Return type list[str]

swagger_types = {'add': 'list[str]', 'drop': 'list[str]'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_ceph_fs_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_ceph_fs_volume_source.**V1CephFSVolumeSource**(*monitors=None*, *path=None*, *read_only=None*, *secret_file=None*, *secret_ref=None*, *user=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'monitors': 'monitors', 'path': 'path', 'read_only': 'readOnly', '

monitors

Gets the monitors of this V1CephFSVolumeSource. Required: Monitors is a collection of Ceph monitors More info: https://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

Returns The monitors of this V1CephFSVolumeSource.

Return type list[str]

path

Gets the path of this V1CephFSVolumeSource. Optional: Used as the mounted root, rather than the full Ceph tree, default is /

Returns The path of this V1CephFSVolumeSource.

Return type str

read_only

Gets the read_only of this V1CephFSVolumeSource. Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

Returns The read_only of this V1CephFSVolumeSource.

Return type bool

secret_file

Gets the secret_file of this V1CephFSVolumeSource. Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

Returns The secret_file of this V1CephFSVolumeSource.

Return type str

secret_ref

Gets the secret_ref of this V1CephFSVolumeSource. Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

Returns The secret_ref of this V1CephFSVolumeSource.

Return type *V1LocalObjectReference*

swagger_types = {'monitors': 'list[str]', 'path': 'str', 'read_only': 'bool', 'secr

to_dict()

Returns the model properties as a dict

to_str()
Returns the string representation of the model

user
Gets the user of this V1CephFSVolumeSource. Optional: User is the rados user name, default is admin More info: https://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

Returns The user of this V1CephFSVolumeSource.

Return type str

kubernetes.client.models.v1_cinder_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_cinder_volume_source.**V1CinderVolumeSource**(*fs_type=None*, *read_only=None*, *volume_id=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'fs_type': 'fsType', 'read_only': 'readOnly', 'volume_id': 'volume

fs_type
Gets the fs_type of this V1CinderVolumeSource. Filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

Returns The fs_type of this V1CinderVolumeSource.

Return type str

read_only
Gets the read_only of this V1CinderVolumeSource. Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

Returns The read_only of this V1CinderVolumeSource.

Return type bool

swagger_types = {'fs_type': 'str', 'read_only': 'bool', 'volume_id': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

volume_id
Gets the volume_id of this V1CinderVolumeSource. volume id used to identify the volume in cinder More info: https://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

Returns The volume_id of this V1CinderVolumeSource.

Return type str

kubernetes.client.models.v1_component_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_component_condition.**V1ComponentCondition**(*error=None*, *message=None*, *status=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'error': 'error', 'message': 'message', 'status': 'status', 'type'

error
Gets the error of this V1ComponentCondition. Condition error code for a component. For example, a health check error code.

Returns The error of this V1ComponentCondition.

Return type str

message
Gets the message of this V1ComponentCondition. Message about the condition for a component. For example, information about a health check.

Returns The message of this V1ComponentCondition.

Return type str

status
Gets the status of this V1ComponentCondition. Status of the condition for a component. Valid values for "Healthy": "True", "False", or "Unknown".

Returns The status of this V1ComponentCondition.

Return type str

swagger_types = {'error': 'str', 'message': 'str', 'status': 'str', 'type': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1ComponentCondition. Type of condition for a component. Valid value: "Healthy"

Returns The type of this V1ComponentCondition.

Return type str

kubernetes.client.models.v1_component_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_component_status.**V1ComponentStatus**(*api_version=None*, *conditions=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ComponentStatus. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ComponentStatus.

Return type str

attribute_map = {'api_version': 'apiVersion', 'conditions': 'conditions', 'kind': '

conditions
Gets the conditions of this V1ComponentStatus. List of component conditions observed

Returns The conditions of this V1ComponentStatus.

Return type list[*V1ComponentCondition*]

kind
Gets the kind of this V1ComponentStatus. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ComponentStatus.

Return type str

metadata
Gets the metadata of this V1ComponentStatus. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1ComponentStatus.

Return type *V1ObjectMeta*

swagger_types = {'api_version': 'str', 'conditions': 'list[V1ComponentCondition]', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_component_status_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_component_status_list.**V1ComponentStatusList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ComponentStatusList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ComponentStatusList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1ComponentStatusList. List of ComponentStatus objects.

Returns The items of this V1ComponentStatusList.

Return type list[*V1ComponentStatus*]

kind
Gets the kind of this V1ComponentStatusList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ComponentStatusList.

Return type str

metadata
Gets the metadata of this V1ComponentStatusList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1ComponentStatusList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1ComponentStatus]', 'kind':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_config_map module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_config_map.**V1ConfigMap**(*api_version=None*, *binary_data=None*, *data=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1ConfigMap. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ConfigMap.

Return type str

attribute_map = {'api_version': 'apiVersion', 'binary_data': 'binaryData', 'data':

binary_data

Gets the binary_data of this V1ConfigMap. BinaryData contains the binary data. Each key must consist of alphanumeric characters, '-', '_' or '.'. BinaryData can contain byte sequences that are not in the UTF-8 range. The keys stored in BinaryData must not overlap with the ones in the Data field, this is enforced during validation process. Using this field will require 1.10+ apiserver and kubelet.

Returns The binary_data of this V1ConfigMap.

Return type dict(str, str)

data

Gets the data of this V1ConfigMap. Data contains the configuration data. Each key must consist of alphanumeric characters, '-', '_' or '.'. Values with non-UTF-8 byte sequences must use the BinaryData field. The keys stored in Data must not overlap with the keys in the BinaryData field, this is enforced during validation process.

Returns The data of this V1ConfigMap.

Return type dict(str, str)

kind

Gets the kind of this V1ConfigMap. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ConfigMap.

Return type str

metadata

Gets the metadata of this V1ConfigMap. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1ConfigMap.

Return type *V1ObjectMeta*

swagger_types = {'api_version': 'str', 'binary_data': 'dict(str, str)', 'data': 'di

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_config_map_key_selector module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_config_map_key_selector.**V1ConfigMapKeySelector**(*key=None*, *name=None*, *optional=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'key': 'key', 'name': 'name', 'optional': 'optional'}

key
Gets the key of this V1ConfigMapKeySelector. The key to select.

Returns The key of this V1ConfigMapKeySelector.

Return type str

name
Gets the name of this V1ConfigMapKeySelector. Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Returns The name of this V1ConfigMapKeySelector.

Return type str

optional
Gets the optional of this V1ConfigMapKeySelector. Specify whether the ConfigMap or it's key must be defined

Returns The optional of this V1ConfigMapKeySelector.

Return type bool

swagger_types = {'key': 'str', 'name': 'str', 'optional': 'bool'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_config_map_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_config_map_list.**V1ConfigMapList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ConfigMapList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ConfigMapList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1ConfigMapList. Items is the list of ConfigMaps.

Returns The items of this V1ConfigMapList.

Return type list[*V1ConfigMap*]

kind
Gets the kind of this V1ConfigMapList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ConfigMapList.

Return type str

metadata
Gets the metadata of this V1ConfigMapList. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1ConfigMapList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1ConfigMap]', 'kind': 'str',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_config_map_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_config_map_volume_source.**V1ConfigMapVolumeSource**(*default_mo*
items=Non
name=Non
op-
tional=Non

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'default_mode': 'defaultMode', 'items': 'items', 'name': 'name', '

default_mode
Gets the default_mode of this V1ConfigMapVolumeSource. Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Returns The default_mode of this V1ConfigMapVolumeSource.

Return type int

items
Gets the items of this V1ConfigMapVolumeSource. If unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Returns The items of this V1ConfigMapVolumeSource.

Return type list[*V1KeyToPath*]

name
Gets the name of this V1ConfigMapVolumeSource. Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Returns The name of this V1ConfigMapVolumeSource.

Return type str

optional
Gets the optional of this V1ConfigMapVolumeSource. Specify whether the ConfigMap or it's keys must be defined

Returns The optional of this V1ConfigMapVolumeSource.

Return type bool

swagger_types = {'default_mode': 'int', 'items': 'list[V1KeyToPath]', 'name': 'str'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_container module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_container.**V1Container**(*args=None*, *command=None*, *env=None*, *env_from=None*, *image=None*, *image_pull_policy=None*, *lifecycle=None*, *liveness_probe=None*, *name=None*, *ports=None*, *readiness_probe=None*, *resources=None*, *security_context=None*, *stdin=None*, *stdin_once=None*, *termination_message_path=None*, *termination_message_policy=None*, *tty=None*, *volume_devices=None*, *volume_mounts=None*, *working_dir=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

args
Gets the args of this V1Container. Arguments to the entrypoint. The docker image's CMD is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

Returns The args of this V1Container.

Return type list[str]

attribute_map = {'args': 'args', 'command': 'command', 'env': 'env', 'env_from': '

command
Gets the command of this V1Container. Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated.

More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

Returns The command of this V1Container.

Return type list[str]

env

Gets the env of this V1Container. List of environment variables to set in the container. Cannot be updated.

Returns The env of this V1Container.

Return type list[*V1EnvVar*]

env_from

Gets the env_from of this V1Container. List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Returns The env_from of this V1Container.

Return type list[V1EnvFromSource]

image

Gets the image of this V1Container. Docker image name. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Returns The image of this V1Container.

Return type str

image_pull_policy

Gets the image_pull_policy of this V1Container. Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images

Returns The image_pull_policy of this V1Container.

Return type str

lifecycle

Gets the lifecycle of this V1Container. Actions that the management system should take in response to container lifecycle events. Cannot be updated.

Returns The lifecycle of this V1Container.

Return type *V1Lifecycle*

liveness_probe

Gets the liveness_probe of this V1Container. Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Returns The liveness_probe of this V1Container.

Return type *V1Probe*

name

Gets the name of this V1Container. Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.

Returns The name of this V1Container.

Return type str

ports

Gets the ports of this V1Container. List of ports to expose from the container. Exposing a port here gives the system additional information about the network connections a container uses, but is primarily informational. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Cannot be updated.

Returns The ports of this V1Container.

Return type list[*V1ContainerPort*]

readiness_probe

Gets the readiness_probe of this V1Container. Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Returns The readiness_probe of this V1Container.

Return type *V1Probe*

resources

Gets the resources of this V1Container. Compute Resources required by this container. Cannot be updated. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

Returns The resources of this V1Container.

Return type *V1ResourceRequirements*

security_context

Gets the security_context of this V1Container. Security options the pod should run with. More info: https://kubernetes.io/docs/concepts/policy/security-context/ More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/

Returns The security_context of this V1Container.

Return type *V1SecurityContext*

stdin

Gets the stdin of this V1Container. Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.

Returns The stdin of this V1Container.

Return type bool

stdin_once

Gets the stdin_once of this V1Container. Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false

Returns The stdin_once of this V1Container.

Return type bool

swagger_types = {'args': 'list[str]', 'command': 'list[str]', 'env': 'list[V1EnvVar

termination_message_path

Gets the termination_message_path of this V1Container. Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written

is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.

Returns The termination_message_path of this V1Container.

Return type str

termination_message_policy
Gets the termination_message_policy of this V1Container. Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated.

Returns The termination_message_policy of this V1Container.

Return type str

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

tty
Gets the tty of this V1Container. Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.

Returns The tty of this V1Container.

Return type bool

volume_devices
Gets the volume_devices of this V1Container. volumeDevices is the list of block devices to be used by the container. This is an alpha feature and may change in the future.

Returns The volume_devices of this V1Container.

Return type list[V1VolumeDevice]

volume_mounts
Gets the volume_mounts of this V1Container. Pod volumes to mount into the container's filesystem. Cannot be updated.

Returns The volume_mounts of this V1Container.

Return type list[*V1VolumeMount*]

working_dir
Gets the working_dir of this V1Container. Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

Returns The working_dir of this V1Container.

Return type str

kubernetes.client.models.v1_container_image module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_container_image.**V1ContainerImage**(*names=None*, *size_bytes=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'names': 'names', 'size_bytes': 'sizeBytes'}

names

Gets the names of this V1ContainerImage. Names by which this image is known. e.g. ["k8s.gcr.io/hyperkube:v1.0.7", "dockerhub.io/google_containers/hyperkube:v1.0.7"]

Returns The names of this V1ContainerImage.

Return type list[str]

size_bytes

Gets the size_bytes of this V1ContainerImage. The size of the image in bytes.

Returns The size_bytes of this V1ContainerImage.

Return type int

swagger_types = {'names': 'list[str]', 'size_bytes': 'int'}

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1_container_port module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_container_port.**V1ContainerPort**(*container_port=None*, *host_ip=None*, *host_port=None*, *name=None*, *protocol=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'container_port': 'containerPort', 'host_ip': 'hostIP', 'host_port'

container_port

Gets the container_port of this V1ContainerPort. Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.

Returns The container_port of this V1ContainerPort.

Return type int

host_ip
Gets the host_ip of this V1ContainerPort. What host IP to bind the external port to.

Returns The host_ip of this V1ContainerPort.

Return type str

host_port
Gets the host_port of this V1ContainerPort. Number of port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.

Returns The host_port of this V1ContainerPort.

Return type int

name
Gets the name of this V1ContainerPort. If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.

Returns The name of this V1ContainerPort.

Return type str

protocol
Gets the protocol of this V1ContainerPort. Protocol for port. Must be UDP or TCP. Defaults to "TCP".

Returns The protocol of this V1ContainerPort.

Return type str

swagger_types = {'container_port': 'int', 'host_ip': 'str', 'host_port': 'int', 'na

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_container_state module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_container_state.**V1ContainerState**(*running=None*, *terminated=None*, *waiting=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'running': 'running', 'terminated': 'terminated', 'waiting': 'wait

running
Gets the running of this V1ContainerState. Details about a running container

Returns The running of this V1ContainerState.

Return type *V1ContainerStateRunning*

swagger_types = {'running': 'V1ContainerStateRunning', 'terminated': 'V1ContainerSta

terminated
Gets the terminated of this V1ContainerState. Details about a terminated container

Returns The terminated of this V1ContainerState.

Return type *V1ContainerStateTerminated*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

waiting
Gets the waiting of this V1ContainerState. Details about a waiting container

Returns The waiting of this V1ContainerState.

Return type *V1ContainerStateWaiting*

kubernetes.client.models.v1_container_state_running module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_container_state_running.**V1ContainerStateRunning**(*started_at=*
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'started_at': 'startedAt'}

started_at
Gets the started_at of this V1ContainerStateRunning. Time at which the container was last (re-)started

Returns The started_at of this V1ContainerStateRunning.

Return type datetime

swagger_types = {'started_at': 'datetime'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_container_state_terminated module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_container_state_terminated.**V1ContainerStateTerminated**(*container_id=None*, *exit_code=None*, *finished_at=None*, *message=None*, *reason=None*, *signal=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'container_id': 'containerID', 'exit_code': 'exitCode', 'finished_at'

container_id
Gets the container_id of this V1ContainerStateTerminated. Container's ID in the format 'docker://<container_id>'

Returns The container_id of this V1ContainerStateTerminated.

Return type str

exit_code
Gets the exit_code of this V1ContainerStateTerminated. Exit status from the last termination of the container

Returns The exit_code of this V1ContainerStateTerminated.

Return type int

finished_at
Gets the finished_at of this V1ContainerStateTerminated. Time at which the container last terminated

Returns The finished_at of this V1ContainerStateTerminated.

Return type datetime

message
Gets the message of this V1ContainerStateTerminated. Message regarding the last termination of the container

Returns The message of this V1ContainerStateTerminated.

Return type str

reason
Gets the reason of this V1ContainerStateTerminated. (brief) reason from the last termination of the container

Returns The reason of this V1ContainerStateTerminated.

Return type str

signal
Gets the signal of this V1ContainerStateTerminated. Signal from the last termination of the container

Returns The signal of this V1ContainerStateTerminated.

Return type int

started_at
Gets the started_at of this V1ContainerStateTerminated. Time at which previous execution of the container started

Returns The started_at of this V1ContainerStateTerminated.

Return type datetime

swagger_types = {'container_id': 'str', 'exit_code': 'int', 'finished_at': 'datetime

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_container_state_waiting module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_container_state_waiting.**V1ContainerStateWaiting**(*message=None*, *reason=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'message': 'message', 'reason': 'reason'}

message
Gets the message of this V1ContainerStateWaiting. Message regarding why the container is not yet running.

Returns The message of this V1ContainerStateWaiting.

Return type str

reason
Gets the reason of this V1ContainerStateWaiting. (brief) reason the container is not yet running.

Returns The reason of this V1ContainerStateWaiting.

Return type str

swagger_types = {'message': 'str', 'reason': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_container_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_container_status.**V1ContainerStatus**(*container_id=None*, *image=None*, *image_id=None*, *last_state=None*, *name=None*, *ready=None*, *restart_count=None*, *state=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'container_id': 'containerID', 'image': 'image', 'image_id': 'imag

container_id

Gets the container_id of this V1ContainerStatus. Container's ID in the format 'docker://<container_id>'.

Returns The container_id of this V1ContainerStatus.

Return type str

image

Gets the image of this V1ContainerStatus. The image the container is running. More info: https://kubernetes.io/docs/concepts/containers/images

Returns The image of this V1ContainerStatus.

Return type str

image_id

Gets the image_id of this V1ContainerStatus. ImageID of the container's image.

Returns The image_id of this V1ContainerStatus.

Return type str

last_state

Gets the last_state of this V1ContainerStatus. Details about the container's last termination condition.

Returns The last_state of this V1ContainerStatus.

Return type *V1ContainerState*

name

Gets the name of this V1ContainerStatus. This must be a DNS_LABEL. Each container in a pod must have a unique name. Cannot be updated.

Returns The name of this V1ContainerStatus.

Return type str

ready

Gets the ready of this V1ContainerStatus. Specifies whether the container has passed its readiness probe.

Returns The ready of this V1ContainerStatus.

Return type bool

restart_count
Gets the restart_count of this V1ContainerStatus. The number of times the container has been restarted, currently based on the number of dead containers that have not yet been removed. Note that this is calculated from dead containers. But those containers are subject to garbage collection. This value will get capped at 5 by GC.

Returns The restart_count of this V1ContainerStatus.

Return type int

state
Gets the state of this V1ContainerStatus. Details about the container's current condition.

Returns The state of this V1ContainerStatus.

Return type *V1ContainerState*

swagger_types = {'container_id': 'str', 'image': 'str', 'image_id': 'str', 'last_st

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_cross_version_object_reference module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_cross_version_object_reference.**V1CrossVersionObjectRefere**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1CrossVersionObjectReference. API version of the referent

Returns The api_version of this V1CrossVersionObjectReference.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'name': 'name'}

kind
Gets the kind of this V1CrossVersionObjectReference. Kind of the referent; More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds"

Returns The kind of this V1CrossVersionObjectReference.

Return type str

name
Gets the name of this V1CrossVersionObjectReference. Name of the referent; More info: http://kubernetes.io/docs/user-guide/identifiers#names

Returns The name of this V1CrossVersionObjectReference.

Return type str

swagger_types = {'api_version': 'str', 'kind': 'str', 'name': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_daemon_endpoint module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_daemon_endpoint.**V1DaemonEndpoint**(*port=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'port': 'Port'}

port
Gets the port of this V1DaemonEndpoint. Port number of the given endpoint.

Returns The port of this V1DaemonEndpoint.

Return type int

swagger_types = {'port': 'int'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_delete_options module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_delete_options.**V1DeleteOptions**(*api_version=None*, *grace_period_seconds=None*, *kind=None*, *orphan_dependents=None*, *preconditions=None*, *propagation_policy=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1DeleteOptions. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1DeleteOptions.

Return type str

attribute_map = {'api_version': 'apiVersion', 'grace_period_seconds': 'gracePeriodSe

grace_period_seconds

Gets the grace_period_seconds of this V1DeleteOptions. The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.

Returns The grace_period_seconds of this V1DeleteOptions.

Return type int

kind

Gets the kind of this V1DeleteOptions. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1DeleteOptions.

Return type str

orphan_dependents

Gets the orphan_dependents of this V1DeleteOptions. Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.

Returns The orphan_dependents of this V1DeleteOptions.

Return type bool

preconditions

Gets the preconditions of this V1DeleteOptions. Must be fulfilled before a deletion is carried out. If not possible, a 409 Conflict status will be returned.

Returns The preconditions of this V1DeleteOptions.

Return type *V1Preconditions*

propagation_policy
Gets the propagation_policy of this V1DeleteOptions. Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Returns The propagation_policy of this V1DeleteOptions.

Return type str

swagger_types = {'api_version': 'str', 'grace_period_seconds': 'int', 'kind': 'str'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_downward_api_volume_file module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_downward_api_volume_file.**V1DownwardAPIVolumeFile**(*field_ref=N mode=Non path=None re- source_fiel*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'field_ref': 'fieldRef', 'mode': 'mode', 'path': 'path', 'resource_

field_ref
Gets the field_ref of this V1DownwardAPIVolumeFile. Required: Selects a field of the pod: only annotations, labels, name and namespace are supported.

Returns The field_ref of this V1DownwardAPIVolumeFile.

Return type *V1ObjectFieldSelector*

mode
Gets the mode of this V1DownwardAPIVolumeFile. Optional: mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Returns The mode of this V1DownwardAPIVolumeFile.

Return type int

path
Gets the path of this V1DownwardAPIVolumeFile. Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'

Returns The path of this V1DownwardAPIVolumeFile.

Return type str

resource_field_ref
Gets the resource_field_ref of this V1DownwardAPIVolumeFile. Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Returns The resource_field_ref of this V1DownwardAPIVolumeFile.

Return type *V1ResourceFieldSelector*

swagger_types = {'field_ref': 'V1ObjectFieldSelector', 'mode': 'int', 'path': 'str'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_downward_api_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_downward_api_volume_source.**V1DownwardAPIVolumeSource**(*defau*
items

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'default_mode': 'defaultMode', 'items': 'items'}

default_mode
Gets the default_mode of this V1DownwardAPIVolumeSource. Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Returns The default_mode of this V1DownwardAPIVolumeSource.

Return type int

items
Gets the items of this V1DownwardAPIVolumeSource. Items is a list of downward API volume file

Returns The items of this V1DownwardAPIVolumeSource.

Return type list[*V1DownwardAPIVolumeFile*]

swagger_types = {'default_mode': 'int', 'items': 'list[V1DownwardAPIVolumeFile]'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_empty_dir_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_empty_dir_volume_source.**V1EmptyDirVolumeSource**(*medium=None*, *size_limit=No*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'medium': 'medium', 'size_limit': 'sizeLimit'}

medium
Gets the medium of this V1EmptyDirVolumeSource. What type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Returns The medium of this V1EmptyDirVolumeSource.

Return type str

size_limit
Gets the size_limit of this V1EmptyDirVolumeSource. Total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: http://kubernetes.io/docs/user-guide/volumes#emptydir

Returns The size_limit of this V1EmptyDirVolumeSource.

Return type str

swagger_types = {'medium': 'str', 'size_limit': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_endpoint_address module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_endpoint_address.**V1EndpointAddress**(*hostname=None*, *ip=None*, *node_name=None*, *target_ref=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'hostname': 'hostname', 'ip': 'ip', 'node_name': 'nodeName', 'targ

hostname
Gets the hostname of this V1EndpointAddress. The Hostname of this endpoint

Returns The hostname of this V1EndpointAddress.

Return type str

ip
Gets the ip of this V1EndpointAddress. The IP of this endpoint. May not be loopback (127.0.0.0/8), link-local (169.254.0.0/16), or link-local multicast ((224.0.0.0/24). IPv6 is also accepted but not fully supported on all platforms. Also, certain kubernetes components, like kube-proxy, are not IPv6 ready.

Returns The ip of this V1EndpointAddress.

Return type str

node_name
Gets the node_name of this V1EndpointAddress. Optional: Node hosting this endpoint. This can be used to determine endpoints local to a node.

Returns The node_name of this V1EndpointAddress.

Return type str

swagger_types = {'hostname': 'str', 'ip': 'str', 'node_name': 'str', 'target_ref':

target_ref
Gets the target_ref of this V1EndpointAddress. Reference to object providing the endpoint.

Returns The target_ref of this V1EndpointAddress.

Return type *V1ObjectReference*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_endpoint_port module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_endpoint_port.**V1EndpointPort**(*name=None*, *port=None*, *protocol=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'name': 'name', 'port': 'port', 'protocol': 'protocol'}

name
Gets the name of this V1EndpointPort. The name of this port (corresponds to ServicePort.Name). Must be a DNS_LABEL. Optional only if one port is defined.

Returns The name of this V1EndpointPort.

Return type str

port
Gets the port of this V1EndpointPort. The port number of the endpoint.

Returns The port of this V1EndpointPort.

Return type int

protocol
Gets the protocol of this V1EndpointPort. The IP protocol for this port. Must be UDP or TCP. Default is TCP.

Returns The protocol of this V1EndpointPort.

Return type str

swagger_types = {'name': 'str', 'port': 'int', 'protocol': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_endpoint_subset module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_endpoint_subset.**V1EndpointSubset**(*addresses=None*, *not_ready_addresses=None*, *ports=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

addresses
Gets the addresses of this V1EndpointSubset. IP addresses which offer the related ports that are marked as ready. These endpoints should be considered safe for load balancers and clients to utilize.

Returns The addresses of this V1EndpointSubset.

Return type list[*V1EndpointAddress*]

attribute_map = {'addresses': 'addresses', 'not_ready_addresses': 'notReadyAddresses

not_ready_addresses
Gets the not_ready_addresses of this V1EndpointSubset. IP addresses which offer the related ports but are not currently marked as ready because they have not yet finished starting, have recently failed a readiness check, or have recently failed a liveness check.

Returns The not_ready_addresses of this V1EndpointSubset.

Return type list[*V1EndpointAddress*]

ports
Gets the ports of this V1EndpointSubset. Port numbers available on the related IP addresses.

Returns The ports of this V1EndpointSubset.

Return type list[*V1EndpointPort*]

swagger_types = {'addresses': 'list[V1EndpointAddress]', 'not_ready_addresses': 'lis

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_endpoints module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_endpoints.**V1Endpoints**(*api_version=None*, *kind=None*, *metadata=None*, *subsets=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1Endpoints. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Endpoints.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1Endpoints. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Endpoints.

Return type str

metadata
Gets the metadata of this V1Endpoints. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Endpoints.

Return type *V1ObjectMeta*

subsets
Gets the subsets of this V1Endpoints. The set of all endpoints is the union of all subsets. Addresses are placed into subsets according to the IPs they share. A single address with multiple ports, some of which are ready and some of which are not (because they come from different containers) will result in the address

being displayed in different subsets for the different ports. No address will appear in both Addresses and NotReadyAddresses in the same subset. Sets of addresses and ports that comprise a service.

Returns The subsets of this V1Endpoints.

Return type list[*V1EndpointSubset*]

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_endpoints_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_endpoints_list.**V1EndpointsList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1EndpointsList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1EndpointsList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1EndpointsList. List of endpoints.

Returns The items of this V1EndpointsList.

Return type list[*V1Endpoints*]

kind
Gets the kind of this V1EndpointsList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1EndpointsList.

Return type str

metadata
Gets the metadata of this V1EndpointsList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1EndpointsList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1Endpoints]', 'kind': 'str',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_env_var module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_env_var.**V1EnvVar**(*name=None*, *value=None*, *value_from=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'name': 'name', 'value': 'value', 'value_from': 'valueFrom'}

name
Gets the name of this V1EnvVar. Name of the environment variable. Must be a C_IDENTIFIER.

Returns The name of this V1EnvVar.

Return type str

swagger_types = {'name': 'str', 'value': 'str', 'value_from': 'V1EnvVarSource'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

value
Gets the value of this V1EnvVar. Variable references $(VAR_NAME) are expanded using the previous defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".

Returns The value of this V1EnvVar.

Return type str

value_from
Gets the value_from of this V1EnvVar. Source for the environment variable's value. Cannot be used if value is not empty.

Returns The value_from of this V1EnvVar.

Return type *V1EnvVarSource*

kubernetes.client.models.v1_env_var_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_env_var_source.**V1EnvVarSource**(*config_map_key_ref=None*, *field_ref=None*, *resource_field_ref=None*, *secret_key_ref=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'config_map_key_ref': 'configMapKeyRef', 'field_ref': 'fieldRef', '

config_map_key_ref

Gets the config_map_key_ref of this V1EnvVarSource. Selects a key of a ConfigMap.

Returns The config_map_key_ref of this V1EnvVarSource.

Return type *V1ConfigMapKeySelector*

field_ref

Gets the field_ref of this V1EnvVarSource. Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP.

Returns The field_ref of this V1EnvVarSource.

Return type *V1ObjectFieldSelector*

resource_field_ref

Gets the resource_field_ref of this V1EnvVarSource. Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Returns The resource_field_ref of this V1EnvVarSource.

Return type *V1ResourceFieldSelector*

secret_key_ref

Gets the secret_key_ref of this V1EnvVarSource. Selects a key of a secret in the pod's namespace

Returns The secret_key_ref of this V1EnvVarSource.

Return type *V1SecretKeySelector*

swagger_types = {'config_map_key_ref': 'V1ConfigMapKeySelector', 'field_ref': 'V1Obj

to_dict()

Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_event module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_event.**V1Event**(*action=None*, *api_version=None*, *count=None*, *event_time=None*, *first_timestamp=None*, *involved_object=None*, *kind=None*, *last_timestamp=None*, *message=None*, *metadata=None*, *reason=None*, *related=None*, *reporting_component=None*, *reporting_instance=None*, *series=None*, *source=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

action
Gets the action of this V1Event. What action was taken/failed regarding to the Regarding object.

Returns The action of this V1Event.

Return type str

api_version
Gets the api_version of this V1Event. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Event.

Return type str

attribute_map = {'action': 'action', 'api_version': 'apiVersion', 'count': 'count',

count
Gets the count of this V1Event. The number of times this event has occurred.

Returns The count of this V1Event.

Return type int

event_time
Gets the event_time of this V1Event. Time when this Event was first observed.

Returns The event_time of this V1Event.

Return type datetime

first_timestamp
Gets the first_timestamp of this V1Event. The time at which the event was first recorded. (Time of server receipt is in TypeMeta.)

Returns The first_timestamp of this V1Event.

Return type datetime

involved_object
Gets the involved_object of this V1Event. The object that this event is about.

Returns The involved_object of this V1Event.

Return type *V1ObjectReference*

kind
Gets the kind of this V1Event. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Event.

Return type str

last_timestamp
Gets the last_timestamp of this V1Event. The time at which the most recent occurrence of this event was recorded.

Returns The last_timestamp of this V1Event.

Return type datetime

message
Gets the message of this V1Event. A human-readable description of the status of this operation.

Returns The message of this V1Event.

Return type str

metadata
Gets the metadata of this V1Event. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Event.

Return type *V1ObjectMeta*

reason
Gets the reason of this V1Event. This should be a short, machine understandable string that gives the reason for the transition into the object's current status.

Returns The reason of this V1Event.

Return type str

related
Gets the related of this V1Event. Optional secondary object for more complex actions.

Returns The related of this V1Event.

Return type *V1ObjectReference*

reporting_component
Gets the reporting_component of this V1Event. Name of the controller that emitted this Event, e.g. *kubernetes.io/kubelet*.

Returns The reporting_component of this V1Event.

Return type str

reporting_instance
Gets the reporting_instance of this V1Event. ID of the controller instance, e.g. *kubelet-xyzf*.

Returns The reporting_instance of this V1Event.

Return type str

series
Gets the series of this V1Event. Data about the Event series this event represents or nil if it's a singleton Event.

Returns The series of this V1Event.

Return type V1EventSeries

source
Gets the source of this V1Event. The component reporting this event. Should be a short machine understandable string.

Returns The source of this V1Event.

Return type *V1EventSource*

swagger_types = {'action': 'str', 'api_version': 'str', 'count': 'int', 'event_time

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1Event. Type of this event (Normal, Warning), new types could be added in the future

Returns The type of this V1Event.

Return type str

kubernetes.client.models.v1_event_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_event_list.**V1EventList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1EventList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1EventList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1EventList. List of events

Returns The items of this V1EventList.

Return type list[*V1Event*]

kind
Gets the kind of this V1EventList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1EventList.

Return type str

metadata
Gets the metadata of this V1EventList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1EventList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1Event]', 'kind': 'str', 'me

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_event_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_event_source.**V1EventSource**(*component=None*, *host=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'component': 'component', 'host': 'host'}

component
Gets the component of this V1EventSource. Component from which the event is generated.

Returns The component of this V1EventSource.

Return type str

host
Gets the host of this V1EventSource. Node name on which the event is generated.

Returns The host of this V1EventSource.

Return type str

swagger_types = {'component': 'str', 'host': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_exec_action module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_exec_action.**V1ExecAction**(*command=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'command': 'command'}

command
Gets the command of this V1ExecAction. Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Returns The command of this V1ExecAction.

Return type list[str]

swagger_types = {'command': 'list[str]'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_fc_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_fc_volume_source.**V1FCVolumeSource**(*fs_type=None*, *lun=None*, *read_only=None*, *target_ww_ns=None*, *wwids=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'fs_type': 'fsType', 'lun': 'lun', 'read_only': 'readOnly', 'targe

fs_type
Gets the fs_type of this V1FCVolumeSource. Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

Returns The fs_type of this V1FCVolumeSource.

Return type str

lun
Gets the lun of this V1FCVolumeSource. Optional: FC target lun number

Returns The lun of this V1FCVolumeSource.

Return type int

read_only
Gets the read_only of this V1FCVolumeSource. Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Returns The read_only of this V1FCVolumeSource.

Return type bool

swagger_types = {'fs_type': 'str', 'lun': 'int', 'read_only': 'bool', 'target_ww_ns

target_ww_ns
Gets the target_ww_ns of this V1FCVolumeSource. Optional: FC target worldwide names (WWNs)

Returns The target_ww_ns of this V1FCVolumeSource.

Return type list[str]

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

wwids
Gets the wwids of this V1FCVolumeSource. Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

Returns The wwids of this V1FCVolumeSource.

Return type list[str]

kubernetes.client.models.v1_flex_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_flex_volume_source.**V1FlexVolumeSource**(*driver=None*, *fs_type=None*, *options=None*, *read_only=None*, *secret_ref=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'driver': 'driver', 'fs_type': 'fsType', 'options': 'options', 're

driver
Gets the driver of this V1FlexVolumeSource. Driver is the name of the driver to use for this volume.

Returns The driver of this V1FlexVolumeSource.

Return type str

fs_type
Gets the fs_type of this V1FlexVolumeSource. Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.

Returns The fs_type of this V1FlexVolumeSource.

Return type str

options
Gets the options of this V1FlexVolumeSource. Optional: Extra command options if any.

Returns The options of this V1FlexVolumeSource.

Return type dict(str, str)

read_only
Gets the read_only of this V1FlexVolumeSource. Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Returns The read_only of this V1FlexVolumeSource.

Return type bool

secret_ref
Gets the secret_ref of this V1FlexVolumeSource. Optional: SecretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Returns The secret_ref of this V1FlexVolumeSource.

Return type *V1LocalObjectReference*

swagger_types = {'driver': 'str', 'fs_type': 'str', 'options': 'dict(str, str)', 'r

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_flocker_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_flocker_volume_source.**V1FlockerVolumeSource**(*dataset_name=No*
dataset_uuid=Non

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'dataset_name': 'datasetName', 'dataset_uuid': 'datasetUUID'}

dataset_name
Gets the dataset_name of this V1FlockerVolumeSource. Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated

Returns The dataset_name of this V1FlockerVolumeSource.

Return type str

dataset_uuid
Gets the dataset_uuid of this V1FlockerVolumeSource. UUID of the dataset. This is unique identifier of a Flocker dataset

Returns The dataset_uuid of this V1FlockerVolumeSource.

Return type str

swagger_types = {'dataset_name': 'str', 'dataset_uuid': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_gce_persistent_disk_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_gce_persistent_disk_volume_source.**V1GCEPersistentDiskVolu**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'fs_type': 'fsType', 'partition': 'partition', 'pd_name': 'pdName'

fs_type
Gets the fs_type of this V1GCEPersistentDiskVolumeSource. Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Returns The fs_type of this V1GCEPersistentDiskVolumeSource.

Return type str

partition
Gets the partition of this V1GCEPersistentDiskVolumeSource. The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Returns The partition of this V1GCEPersistentDiskVolumeSource.

Return type int

pd_name
Gets the pd_name of this V1GCEPersistentDiskVolumeSource. Unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Returns The pd_name of this V1GCEPersistentDiskVolumeSource.

Return type str

read_only
Gets the read_only of this V1GCEPersistentDiskVolumeSource. ReadOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Returns The read_only of this V1GCEPersistentDiskVolumeSource.

Return type bool

swagger_types = {'fs_type': 'str', 'partition': 'int', 'pd_name': 'str', 'read_only

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_git_repo_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_git_repo_volume_source.**V1GitRepoVolumeSource**(*directory=None*, *repository=None*, *revision=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'directory': 'directory', 'repository': 'repository', 'revision':

directory
Gets the directory of this V1GitRepoVolumeSource. Target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.

Returns The directory of this V1GitRepoVolumeSource.

Return type str

repository
Gets the repository of this V1GitRepoVolumeSource. Repository URL

Returns The repository of this V1GitRepoVolumeSource.

Return type str

revision
Gets the revision of this V1GitRepoVolumeSource. Commit hash for the specified revision.

Returns The revision of this V1GitRepoVolumeSource.

Return type str

swagger_types = {'directory': 'str', 'repository': 'str', 'revision': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_glusterfs_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_glusterfs_volume_source.**V1GlusterfsVolumeSource**(*endpoints=N*, *path=None*, *read_only=N*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'endpoints': 'endpoints', 'path': 'path', 'read_only': 'readOnly'}

endpoints
Gets the endpoints of this V1GlusterfsVolumeSource. EndpointsName is the endpoint name that details Glusterfs topology. More info: https://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod

Returns The endpoints of this V1GlusterfsVolumeSource.

Return type str

path
Gets the path of this V1GlusterfsVolumeSource. Path is the Glusterfs volume path. More info: https://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod

Returns The path of this V1GlusterfsVolumeSource.

Return type str

read_only
Gets the read_only of this V1GlusterfsVolumeSource. ReadOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod

Returns The read_only of this V1GlusterfsVolumeSource.

Return type bool

swagger_types = {'endpoints': 'str', 'path': 'str', 'read_only': 'bool'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_handler module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_handler.**V1Handler**(*_exec=None*, *http_get=None*, *tcp_socket=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'_exec': 'exec', 'http_get': 'httpGet', 'tcp_socket': 'tcpSocket'}

http_get
Gets the http_get of this V1Handler. HTTPGet specifies the http request to perform.

Returns The http_get of this V1Handler.

Return type *V1HTTPGetAction*

swagger_types = {'_exec': 'V1ExecAction', 'http_get': 'V1HTTPGetAction', 'tcp_socket

tcp_socket
Gets the tcp_socket of this V1Handler. TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported

Returns The tcp_socket of this V1Handler.

Return type *V1TCPSocketAction*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_horizontal_pod_autoscaler module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_horizontal_pod_autoscaler.**V1HorizontalPodAutoscaler**(*api_ve kind=N meta- data=N spec=N sta- tus=No*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1HorizontalPodAutoscaler. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1HorizontalPodAutoscaler.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1HorizontalPodAutoscaler. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1HorizontalPodAutoscaler.

Return type str

metadata
Gets the metadata of this V1HorizontalPodAutoscaler. Standard object metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1HorizontalPodAutoscaler.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1HorizontalPodAutoscaler. behaviour of autoscaler. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status.

Returns The spec of this V1HorizontalPodAutoscaler.

Return type *V1HorizontalPodAutoscalerSpec*

status
Gets the status of this V1HorizontalPodAutoscaler. current information about the autoscaler.

Returns The status of this V1HorizontalPodAutoscaler.

Return type *V1HorizontalPodAutoscalerStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_horizontal_pod_autoscaler_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_horizontal_pod_autoscaler_list.**V1HorizontalPodAutoscalerL**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1HorizontalPodAutoscalerList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1HorizontalPodAutoscalerList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1HorizontalPodAutoscalerList. list of horizontal pod autoscaler objects.

Returns The items of this V1HorizontalPodAutoscalerList.

Return type list[*V1HorizontalPodAutoscaler*]

kind
Gets the kind of this V1HorizontalPodAutoscalerList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests

to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1HorizontalPodAutoscalerList.

Return type str

metadata
Gets the metadata of this V1HorizontalPodAutoscalerList. Standard list metadata.

Returns The metadata of this V1HorizontalPodAutoscalerList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1HorizontalPodAutoscaler]', 'l

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_horizontal_pod_autoscaler_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_horizontal_pod_autoscaler_spec.**V1HorizontalPodAutoscalerS**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'max_replicas': 'maxReplicas', 'min_replicas': 'minReplicas', 'scal

max_replicas
Gets the max_replicas of this V1HorizontalPodAutoscalerSpec. upper limit for the number of pods that can be set by the autoscaler; cannot be smaller than MinReplicas.

Returns The max_replicas of this V1HorizontalPodAutoscalerSpec.

Return type int

min_replicas
Gets the min_replicas of this V1HorizontalPodAutoscalerSpec. lower limit for the number of pods that can be set by the autoscaler, default 1.

Returns The min_replicas of this V1HorizontalPodAutoscalerSpec.

Return type int

scale_target_ref
Gets the scale_target_ref of this V1HorizontalPodAutoscalerSpec. reference to scaled resource; horizontal pod autoscaler will learn the current resource consumption and will set the desired number of pods by using its Scale subresource.

Returns The scale_target_ref of this V1HorizontalPodAutoscalerSpec.

Return type *V1CrossVersionObjectReference*

swagger_types = {'max_replicas': 'int', 'min_replicas': 'int', 'scale_target_ref':

target_cpu_utilization_percentage
Gets the target_cpu_utilization_percentage of this V1HorizontalPodAutoscalerSpec. target average CPU utilization (represented as a percentage of requested CPU) over all the pods; if not specified the default autoscaling policy will be used.

Returns The target_cpu_utilization_percentage of this V1HorizontalPodAutoscalerSpec.

Return type int

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_horizontal_pod_autoscaler_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_horizontal_pod_autoscaler_status.**V1HorizontalPodAutoscale**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'current_cpu_utilization_percentage': 'currentCPUUtilizationPercenta

current_cpu_utilization_percentage
Gets the current_cpu_utilization_percentage of this V1HorizontalPodAutoscalerStatus. current average CPU utilization over all pods, represented as a percentage of requested CPU, e.g. 70 means that an average pod is using now 70% of its requested CPU.

Returns The current_cpu_utilization_percentage of this V1HorizontalPodAutoscalerStatus.

Return type int

current_replicas
Gets the current_replicas of this V1HorizontalPodAutoscalerStatus. current number of replicas of pods managed by this autoscaler.

Returns The current_replicas of this V1HorizontalPodAutoscalerStatus.

Return type int

desired_replicas
Gets the desired_replicas of this V1HorizontalPodAutoscalerStatus. desired number of replicas of pods managed by this autoscaler.

Returns The desired_replicas of this V1HorizontalPodAutoscalerStatus.

Return type int

last_scale_time
Gets the last_scale_time of this V1HorizontalPodAutoscalerStatus. last time the HorizontalPodAutoscaler scaled the number of pods; used by the autoscaler to control how often the number of pods is changed.

Returns The last_scale_time of this V1HorizontalPodAutoscalerStatus.

Return type datetime

observed_generation
Gets the observed_generation of this V1HorizontalPodAutoscalerStatus. most recent generation observed by this autoscaler.

Returns The observed_generation of this V1HorizontalPodAutoscalerStatus.

Return type int

swagger_types = {'current_cpu_utilization_percentage': 'int', 'current_replicas': 'i

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_host_path_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_host_path_volume_source.**V1HostPathVolumeSource**(*path=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'path': 'path', 'type': 'type'}

path
Gets the path of this V1HostPathVolumeSource. Path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

Returns The path of this V1HostPathVolumeSource.

Return type str

swagger_types = {'path': 'str', 'type': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1HostPathVolumeSource. Type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

Returns The type of this V1HostPathVolumeSource.

Return type str

kubernetes.client.models.v1_http_get_action module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_http_get_action.**V1HTTPGetAction**(*host=None*, *http_headers=None*, *path=None*, *port=None*, *scheme=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'host': 'host', 'http_headers': 'httpHeaders', 'path': 'path', 'po

host
Gets the host of this V1HTTPGetAction. Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Returns The host of this V1HTTPGetAction.

Return type str

http_headers
Gets the http_headers of this V1HTTPGetAction. Custom headers to set in the request. HTTP allows repeated headers.

Returns The http_headers of this V1HTTPGetAction.

Return type list[*V1HTTPHeader*]

path
Gets the path of this V1HTTPGetAction. Path to access on the HTTP server.

Returns The path of this V1HTTPGetAction.

Return type str

port
Gets the port of this V1HTTPGetAction. Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Returns The port of this V1HTTPGetAction.

Return type object

scheme
Gets the scheme of this V1HTTPGetAction. Scheme to use for connecting to the host. Defaults to HTTP.

Returns The scheme of this V1HTTPGetAction.

Return type str

swagger_types = {'host': 'str', 'http_headers': 'list[V1HTTPHeader]', 'path': 'str'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_http_header module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_http_header.**V1HTTPHeader**(*name=None*, *value=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'name': 'name', 'value': 'value'}

name
Gets the name of this V1HTTPHeader. The header field name

Returns The name of this V1HTTPHeader.

Return type str

swagger_types = {'name': 'str', 'value': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

value
Gets the value of this V1HTTPHeader. The header field value

Returns The value of this V1HTTPHeader.

Return type str

kubernetes.client.models.v1_iscsi_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_iscsi_volume_source.**V1ISCSIVolumeSource**(*chap_auth_discovery=N chap_auth_session=Non fs_type=None, initiator_name=None, iqn=None, iscsi_interface=None, lun=None, portals=None, read_only=None, secret_ref=None, target_portal=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'chap_auth_discovery': 'chapAuthDiscovery', 'chap_auth_session': 'ch

chap_auth_discovery
Gets the chap_auth_discovery of this V1ISCSIVolumeSource. whether support iSCSI Discovery CHAP authentication

Returns The chap_auth_discovery of this V1ISCSIVolumeSource.

Return type bool

chap_auth_session
Gets the chap_auth_session of this V1ISCSIVolumeSource. whether support iSCSI Session CHAP authentication

Returns The chap_auth_session of this V1ISCSIVolumeSource.

Return type bool

fs_type
Gets the fs_type of this V1ISCSIVolumeSource. Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi

Returns The fs_type of this V1ISCSIVolumeSource.

Return type str

initiator_name
Gets the initiator_name of this V1ISCSIVolumeSource. Custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface <target portal>:<volume name> will be created for the connection.

Returns The initiator_name of this V1ISCSIVolumeSource.

Return type str

iqn
Gets the iqn of this V1ISCSIVolumeSource. Target iSCSI Qualified Name.

Returns The iqn of this V1ISCSIVolumeSource.

Return type str

iscsi_interface
Gets the iscsi_interface of this V1ISCSIVolumeSource. iSCSI Interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).

Returns The iscsi_interface of this V1ISCSIVolumeSource.

Return type str

lun
Gets the lun of this V1ISCSIVolumeSource. iSCSI Target Lun number.

Returns The lun of this V1ISCSIVolumeSource.

Return type int

portals
Gets the portals of this V1ISCSIVolumeSource. iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

Returns The portals of this V1ISCSIVolumeSource.

Return type list[str]

read_only
Gets the read_only of this V1ISCSIVolumeSource. ReadOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.

Returns The read_only of this V1ISCSIVolumeSource.

Return type bool

secret_ref
Gets the secret_ref of this V1ISCSIVolumeSource. CHAP Secret for iSCSI target and initiator authentication

Returns The secret_ref of this V1ISCSIVolumeSource.

Return type *V1LocalObjectReference*

swagger_types = {'chap_auth_discovery': 'bool', 'chap_auth_session': 'bool', 'fs_typ

target_portal
Gets the target_portal of this V1ISCSIVolumeSource. iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

Returns The target_portal of this V1ISCSIVolumeSource.

Return type str

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_job module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_job.**V1Job**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1Job. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Job.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind

Gets the kind of this V1Job. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Job.

Return type str

metadata

Gets the metadata of this V1Job. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Job.

Return type *V1ObjectMeta*

spec

Gets the spec of this V1Job. Specification of the desired behavior of a job. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1Job.

Return type *V1JobSpec*

status

Gets the status of this V1Job. Current status of a job. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1Job.

Return type *V1JobStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1_job_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_job_condition.**V1JobCondition**(*last_probe_time=None*, *last_transition_time=None*, *message=None*, *reason=None*, *status=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'last_probe_time': 'lastProbeTime', 'last_transition_time': 'lastTr

last_probe_time
Gets the last_probe_time of this V1JobCondition. Last time the condition was checked.

Returns The last_probe_time of this V1JobCondition.

Return type datetime

last_transition_time
Gets the last_transition_time of this V1JobCondition. Last time the condition transit from one status to another.

Returns The last_transition_time of this V1JobCondition.

Return type datetime

message
Gets the message of this V1JobCondition. Human readable message indicating details about last transition.

Returns The message of this V1JobCondition.

Return type str

reason
Gets the reason of this V1JobCondition. (brief) reason for the condition's last transition.

Returns The reason of this V1JobCondition.

Return type str

status
Gets the status of this V1JobCondition. Status of the condition, one of True, False, Unknown.

Returns The status of this V1JobCondition.

Return type str

swagger_types = {'last_probe_time': 'datetime', 'last_transition_time': 'datetime',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1JobCondition. Type of job condition, Complete or Failed.

Returns The type of this V1JobCondition.

Return type str

kubernetes.client.models.v1_job_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_job_list.**V1JobList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1JobList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1JobList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items

Gets the items of this V1JobList. items is the list of Jobs.

Returns The items of this V1JobList.

Return type list[*V1Job*]

kind

Gets the kind of this V1JobList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1JobList.

Return type str

metadata

Gets the metadata of this V1JobList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1JobList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1Job]', 'kind': 'str', 'meta

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1_job_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_job_spec.**V1JobSpec**(*active_deadline_seconds=None*, *backoff_limit=None*, *completions=None*, *manual_selector=None*, *parallelism=None*, *selector=None*, *template=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

active_deadline_seconds

Gets the active_deadline_seconds of this V1JobSpec. Specifies the duration in seconds relative to the startTime that the job may be active before the system tries to terminate it; value must be positive integer

Returns The active_deadline_seconds of this V1JobSpec.

Return type int

attribute_map = {'active_deadline_seconds': 'activeDeadlineSeconds', 'backoff_limit':

backoff_limit

Gets the backoff_limit of this V1JobSpec. Specifies the number of retries before marking this job failed. Defaults to 6

Returns The backoff_limit of this V1JobSpec.

Return type int

completions

Gets the completions of this V1JobSpec. Specifies the desired number of successfully finished pods the job should be run with. Setting to nil means that the success of any pod signals the success of all pods, and allows parallelism to have any positive value. Setting to 1 means that parallelism is limited to 1 and the success of that pod signals the success of the job. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/

Returns The completions of this V1JobSpec.

Return type int

manual_selector

Gets the manual_selector of this V1JobSpec. manualSelector controls generation of pod labels and pod selectors. Leave *manualSelector* unset unless you are certain what you are doing. When false or unset, the system pick labels unique to this job and appends those labels to the pod template. When true, the user is responsible for picking unique labels and specifying the selector. Failure to pick a unique label may cause this and other jobs to not function correctly. However, You may see *manualSelector=true* in jobs that were created with the old *extensions/v1beta1* API. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/#specifying-your-own-pod-selector

Returns The manual_selector of this V1JobSpec.

Return type bool

parallelism

Gets the parallelism of this V1JobSpec. Specifies the maximum desired number of pods the job should run at any given time. The actual number of pods running in steady state will be less than this number when ((.spec.completions - .status.successful) < .spec.parallelism), i.e. when the work left to

do is less than max parallelism. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/

Returns The parallelism of this V1JobSpec.

Return type int

selector
Gets the selector of this V1JobSpec. A label query over pods that should match the pod count. Normally, the system sets this field for you. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors

Returns The selector of this V1JobSpec.

Return type V1LabelSelector

swagger_types = {'active_deadline_seconds': 'int', 'backoff_limit': 'int', 'completi

template
Gets the template of this V1JobSpec. Describes the pod that will be created when executing a job. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/

Returns The template of this V1JobSpec.

Return type *V1PodTemplateSpec*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_job_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_job_status.**V1JobStatus**(*active=None*, *completion_time=None*, *conditions=None*, *failed=None*, *start_time=None*, *succeeded=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

active
Gets the active of this V1JobStatus. The number of actively running pods.

Returns The active of this V1JobStatus.

Return type int

attribute_map = {'active': 'active', 'completion_time': 'completionTime', 'condition

completion_time
Gets the completion_time of this V1JobStatus. Represents time when the job was completed. It is not

guaranteed to be set in happens-before order across separate operations. It is represented in RFC3339 form and is in UTC.

Returns The completion_time of this V1JobStatus.

Return type datetime

conditions
Gets the conditions of this V1JobStatus. The latest available observations of an object's current state. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/

Returns The conditions of this V1JobStatus.

Return type list[*V1JobCondition*]

failed
Gets the failed of this V1JobStatus. The number of pods which reached phase Failed.

Returns The failed of this V1JobStatus.

Return type int

start_time
Gets the start_time of this V1JobStatus. Represents time when the job was acknowledged by the job controller. It is not guaranteed to be set in happens-before order across separate operations. It is represented in RFC3339 form and is in UTC.

Returns The start_time of this V1JobStatus.

Return type datetime

succeeded
Gets the succeeded of this V1JobStatus. The number of pods which reached phase Succeeded.

Returns The succeeded of this V1JobStatus.

Return type int

swagger_types = {'active': 'int', 'completion_time': 'datetime', 'conditions': 'lis

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_key_to_path module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_key_to_path.**V1KeyToPath**(*key=None*, *mode=None*, *path=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'key': 'key', 'mode': 'mode', 'path': 'path'}

key
Gets the key of this V1KeyToPath. The key to project.

Returns The key of this V1KeyToPath.

Return type str

mode
Gets the mode of this V1KeyToPath. Optional: mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Returns The mode of this V1KeyToPath.

Return type int

path
Gets the path of this V1KeyToPath. The relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

Returns The path of this V1KeyToPath.

Return type str

swagger_types = {'key': 'str', 'mode': 'int', 'path': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_lifecycle module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_lifecycle.**V1Lifecycle**(*post_start=None*, *pre_stop=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'post_start': 'postStart', 'pre_stop': 'preStop'}

post_start
Gets the post_start of this V1Lifecycle. PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Returns The post_start of this V1Lifecycle.

Return type *V1Handler*

pre_stop
Gets the pre_stop of this V1Lifecycle. PreStop is called immediately before a container is terminated. The container is terminated after the handler completes. The reason for termination is passed to the handler. Regardless of the outcome of the handler, the container is eventually terminated. Other management of

the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Returns The pre_stop of this V1Lifecycle.

Return type *V1Handler*

swagger_types = {'post_start': 'V1Handler', 'pre_stop': 'V1Handler'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_limit_range module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_limit_range.**V1LimitRange**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1LimitRange. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1LimitRange.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1LimitRange. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1LimitRange.

Return type str

metadata
Gets the metadata of this V1LimitRange. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1LimitRange.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1LimitRange. Spec defines the limits enforced. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1LimitRange.

Return type *V1LimitRangeSpec*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_limit_range_item module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_limit_range_item.**V1LimitRangeItem**(*default=None*, *default_request=None*, *max=None*, *max_limit_request_ratio=None*, *min=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'default': 'default', 'default_request': 'defaultRequest', 'max':

default
Gets the default of this V1LimitRangeItem. Default resource requirement limit value by resource name if resource limit is omitted.

Returns The default of this V1LimitRangeItem.

Return type dict(str, str)

default_request
Gets the default_request of this V1LimitRangeItem. DefaultRequest is the default resource requirement request value by resource name if resource request is omitted.

Returns The default_request of this V1LimitRangeItem.

Return type dict(str, str)

max
Gets the max of this V1LimitRangeItem. Max usage constraints on this kind by resource name.

Returns The max of this V1LimitRangeItem.

Return type dict(str, str)

max_limit_request_ratio
Gets the max_limit_request_ratio of this V1LimitRangeItem. MaxLimitRequestRatio if specified, the named resource must have a request and limit that are both non-zero where limit divided by request is less than or equal to the enumerated value; this represents the max burst for the named resource.

Returns The max_limit_request_ratio of this V1LimitRangeItem.

Return type dict(str, str)

min
Gets the min of this V1LimitRangeItem. Min usage constraints on this kind by resource name.

Returns The min of this V1LimitRangeItem.

Return type dict(str, str)

swagger_types = {'default': 'dict(str, str)', 'default_request': 'dict(str, str)', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1LimitRangeItem. Type of resource that this limit applies to.

Returns The type of this V1LimitRangeItem.

Return type str

kubernetes.client.models.v1_limit_range_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_limit_range_list.**V1LimitRangeList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1LimitRangeList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1LimitRangeList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1LimitRangeList. Items is a list of LimitRange objects. More info: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/

Returns The items of this V1LimitRangeList.

Return type list[*V1LimitRange*]

kind
Gets the kind of this V1LimitRangeList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1LimitRangeList.

Return type str

metadata
Gets the metadata of this V1LimitRangeList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1LimitRangeList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1LimitRange]', 'kind': 'str'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_limit_range_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_limit_range_spec.**V1LimitRangeSpec**(*limits=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'limits': 'limits'}

limits
Gets the limits of this V1LimitRangeSpec. Limits is the list of LimitRangeItem objects that are enforced.

Returns The limits of this V1LimitRangeSpec.

Return type list[*V1LimitRangeItem*]

swagger_types = {'limits': 'list[V1LimitRangeItem]'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_load_balancer_ingress module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_load_balancer_ingress.**V1LoadBalancerIngress**(*hostname=None*, *ip=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'hostname': 'hostname', 'ip': 'ip'}

hostname

Gets the hostname of this V1LoadBalancerIngress. Hostname is set for load-balancer ingress points that are DNS based (typically AWS load-balancers)

Returns The hostname of this V1LoadBalancerIngress.

Return type str

ip

Gets the ip of this V1LoadBalancerIngress. IP is set for load-balancer ingress points that are IP based (typically GCE or OpenStack load-balancers)

Returns The ip of this V1LoadBalancerIngress.

Return type str

swagger_types = {'hostname': 'str', 'ip': 'str'}

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1_load_balancer_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_load_balancer_status.**V1LoadBalancerStatus**(*ingress=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'ingress': 'ingress'}

ingress

Gets the ingress of this V1LoadBalancerStatus. Ingress is a list containing ingress points for the load-balancer. Traffic intended for the service should be sent to these ingress points.

Returns The ingress of this V1LoadBalancerStatus.

Return type list[*V1LoadBalancerIngress*]

swagger_types = {'ingress': 'list[V1LoadBalancerIngress]'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_local_object_reference module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_local_object_reference.**V1LocalObjectReference**(*name=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'name': 'name'}

name
Gets the name of this V1LocalObjectReference. Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Returns The name of this V1LocalObjectReference.

Return type str

swagger_types = {'name': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_namespace module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_namespace.**V1Namespace**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1Namespace. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Namespace.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1Namespace. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Namespace.

Return type str

metadata
Gets the metadata of this V1Namespace. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Namespace.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1Namespace. Spec defines the behavior of the Namespace. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1Namespace.

Return type *V1NamespaceSpec*

status
Gets the status of this V1Namespace. Status describes the current status of a Namespace. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1Namespace.

Return type *V1NamespaceStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_namespace_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_namespace_list.**V1NamespaceList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1NamespaceList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1NamespaceList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1NamespaceList. Items is the list of Namespace objects in the list. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

Returns The items of this V1NamespaceList.

Return type list[*V1Namespace*]

kind
Gets the kind of this V1NamespaceList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1NamespaceList.

Return type str

metadata
Gets the metadata of this V1NamespaceList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1NamespaceList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1Namespace]', 'kind': 'str',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_namespace_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_namespace_spec.**V1NamespaceSpec**(*finalizers=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'finalizers': 'finalizers'}

finalizers
Gets the finalizers of this V1NamespaceSpec. Finalizers is an opaque list of values that must be empty to permanently remove object from storage. More info: https://kubernetes.io/docs/tasks/administer-cluster/namespaces/

Returns The finalizers of this V1NamespaceSpec.

Return type list[str]

swagger_types = {'finalizers': 'list[str]'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_namespace_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_namespace_status.**V1NamespaceStatus**(*phase=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'phase': 'phase'}

phase
Gets the phase of this V1NamespaceStatus. Phase is the current lifecycle phase of the namespace. More info: https://kubernetes.io/docs/tasks/administer-cluster/namespaces/

Returns The phase of this V1NamespaceStatus.

Return type str

swagger_types = {'phase': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_nfs_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_nfs_volume_source.**V1NFSVolumeSource**(*path=None*, *read_only=None*, *server=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'path': 'path', 'read_only': 'readOnly', 'server': 'server'}

path

Gets the path of this V1NFSVolumeSource. Path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

Returns The path of this V1NFSVolumeSource.

Return type str

read_only

Gets the read_only of this V1NFSVolumeSource. ReadOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

Returns The read_only of this V1NFSVolumeSource.

Return type bool

server

Gets the server of this V1NFSVolumeSource. Server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

Returns The server of this V1NFSVolumeSource.

Return type str

swagger_types = {'path': 'str', 'read_only': 'bool', 'server': 'str'}

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1_node module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_node.**V1Node**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1Node. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Node.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1Node. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Node.

Return type str

metadata
Gets the metadata of this V1Node. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Node.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1Node. Spec defines the behavior of a node. https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1Node.

Return type *V1NodeSpec*

status
Gets the status of this V1Node. Most recently observed status of the node. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1Node.

Return type *V1NodeStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_node_address module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_node_address.**V1NodeAddress**(*address=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

address
Gets the address of this V1NodeAddress. The node address.

Returns The address of this V1NodeAddress.

Return type str

attribute_map = {'address': 'address', 'type': 'type'}

swagger_types = {'address': 'str', 'type': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1NodeAddress. Node address type, one of Hostname, ExternalIP or InternalIP.

Returns The type of this V1NodeAddress.

Return type str

kubernetes.client.models.v1_node_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_node_condition.**V1NodeCondition**(*last_heartbeat_time=None*, *last_transition_time=None*, *message=None*, *reason=None*, *status=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'last_heartbeat_time': 'lastHeartbeatTime', 'last_transition_time':

last_heartbeat_time
Gets the last_heartbeat_time of this V1NodeCondition. Last time we got an update on a given condition.

Returns The last_heartbeat_time of this V1NodeCondition.

Return type datetime

last_transition_time
Gets the last_transition_time of this V1NodeCondition. Last time the condition transit from one status to another.

Returns The last_transition_time of this V1NodeCondition.

Return type datetime

message
Gets the message of this V1NodeCondition. Human readable message indicating details about last transition.

Returns The message of this V1NodeCondition.

Return type str

reason
Gets the reason of this V1NodeCondition. (brief) reason for the condition's last transition.

Returns The reason of this V1NodeCondition.

Return type str

status
Gets the status of this V1NodeCondition. Status of the condition, one of True, False, Unknown.

Returns The status of this V1NodeCondition.

Return type str

swagger_types = {'last_heartbeat_time': 'datetime', 'last_transition_time': 'datetim

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1NodeCondition. Type of node condition.

Returns The type of this V1NodeCondition.

Return type str

kubernetes.client.models.v1_node_daemon_endpoints module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_node_daemon_endpoints.**V1NodeDaemonEndpoints**(*kubelet_endpoint=*
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'kubelet_endpoint': 'kubeletEndpoint'}

kubelet_endpoint
Gets the kubelet_endpoint of this V1NodeDaemonEndpoints. Endpoint on which Kubelet is listening.

Returns The kubelet_endpoint of this V1NodeDaemonEndpoints.

Return type *V1DaemonEndpoint*

swagger_types = {'kubelet_endpoint': 'V1DaemonEndpoint'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_node_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_node_list.**V1NodeList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1NodeList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1NodeList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1NodeList. List of nodes

Returns The items of this V1NodeList.

Return type list[*V1Node*]

kind
Gets the kind of this V1NodeList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1NodeList.

Return type str

metadata
Gets the metadata of this V1NodeList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1NodeList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1Node]', 'kind': 'str', 'meta

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_node_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_node_spec.**V1NodeSpec**(*config_source=None*, *external_id=None*, *pod_cidr=None*, *provider_id=None*, *taints=None*, *unschedulable=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'config_source': 'configSource', 'external_id': 'externalID', 'pod_

config_source
Gets the config_source of this V1NodeSpec. If specified, the source to get node configuration from The DynamicKubeletConfig feature gate must be enabled for the Kubelet to use this field

Returns The config_source of this V1NodeSpec.

Return type V1NodeConfigSource

external_id
Gets the external_id of this V1NodeSpec. External ID of the node assigned by some machine database (e.g. a cloud provider). Deprecated.

Returns The external_id of this V1NodeSpec.

Return type str

pod_cidr
Gets the pod_cidr of this V1NodeSpec. PodCIDR represents the pod IP range assigned to the node.

Returns The pod_cidr of this V1NodeSpec.

Return type str

provider_id
Gets the provider_id of this V1NodeSpec. ID of the node assigned by the cloud provider in the format: <ProviderName>://<ProviderSpecificNodeID>

Returns The provider_id of this V1NodeSpec.

Return type str

swagger_types = {'config_source': 'V1NodeConfigSource', 'external_id': 'str', 'pod_c

taints
Gets the taints of this V1NodeSpec. If specified, the node's taints.

Returns The taints of this V1NodeSpec.

Return type list[V1Taint]

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

unschedulable
Gets the unschedulable of this V1NodeSpec. Unschedulable controls node schedulability of new pods. By default, node is schedulable. More info: https://kubernetes.io/docs/concepts/nodes/node/#manual-node-administration

Returns The unschedulable of this V1NodeSpec.

Return type bool

kubernetes.client.models.v1_node_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_node_status.**V1NodeStatus**(*addresses=None*, *allocatable=None*, *capacity=None*, *conditions=None*, *daemon_endpoints=None*, *images=None*, *node_info=None*, *phase=None*, *volumes_attached=None*, *volumes_in_use=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

addresses
Gets the addresses of this V1NodeStatus. List of addresses reachable to the node. Queried from cloud provider, if available. More info: https://kubernetes.io/docs/concepts/nodes/node/#addresses

Returns The addresses of this V1NodeStatus.

Return type list[*V1NodeAddress*]

allocatable
Gets the allocatable of this V1NodeStatus. Allocatable represents the resources of a node that are available for scheduling. Defaults to Capacity.

Returns The allocatable of this V1NodeStatus.

Return type dict(str, str)

attribute_map = {'addresses': 'addresses', 'allocatable': 'allocatable', 'capacity':

capacity
Gets the capacity of this V1NodeStatus. Capacity represents the total resources of a node. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#capacity

Returns The capacity of this V1NodeStatus.

Return type dict(str, str)

conditions
Gets the conditions of this V1NodeStatus. Conditions is an array of current observed node conditions. More info: https://kubernetes.io/docs/concepts/nodes/node/#condition

Returns The conditions of this V1NodeStatus.

Return type list[*V1NodeCondition*]

daemon_endpoints
Gets the daemon_endpoints of this V1NodeStatus. Endpoints of daemons running on the Node.

Returns The daemon_endpoints of this V1NodeStatus.

Return type *V1NodeDaemonEndpoints*

images
Gets the images of this V1NodeStatus. List of container images on this node

Returns The images of this V1NodeStatus.

Return type list[*V1ContainerImage*]

node_info
Gets the node_info of this V1NodeStatus. Set of ids/uuids to uniquely identify the node. More info: https://kubernetes.io/docs/concepts/nodes/node/#info

Returns The node_info of this V1NodeStatus.

Return type *V1NodeSystemInfo*

phase
Gets the phase of this V1NodeStatus. NodePhase is the recently observed lifecycle phase of the node. More info: https://kubernetes.io/docs/concepts/nodes/node/#phase The field is never populated, and now is deprecated.

Returns The phase of this V1NodeStatus.

Return type str

swagger_types = {'addresses': 'list[V1NodeAddress]', 'allocatable': 'dict(str, str)'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

volumes_attached
Gets the volumes_attached of this V1NodeStatus. List of volumes that are attached to the node.

Returns The volumes_attached of this V1NodeStatus.

Return type list[*V1AttachedVolume*]

volumes_in_use
Gets the volumes_in_use of this V1NodeStatus. List of attachable volumes in use (mounted) by the node.

Returns The volumes_in_use of this V1NodeStatus.

Return type list[str]

kubernetes.client.models.v1_node_system_info module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_node_system_info.**V1NodeSystemInfo**(*architecture=None*, *boot_id=None*, *container_runtime_version=None*, *kernel_version=None*, *kube_proxy_version=None*, *kubelet_version=None*, *machine_id=None*, *operating_system=None*, *os_image=None*, *system_uuid=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

architecture

Gets the architecture of this V1NodeSystemInfo. The Architecture reported by the node

Returns The architecture of this V1NodeSystemInfo.

Return type str

attribute_map = {'architecture': 'architecture', 'boot_id': 'bootID', 'container_runt

boot_id

Gets the boot_id of this V1NodeSystemInfo. Boot ID reported by the node.

Returns The boot_id of this V1NodeSystemInfo.

Return type str

container_runtime_version

Gets the container_runtime_version of this V1NodeSystemInfo. ContainerRuntime Version reported by the node through runtime remote API (e.g. docker://1.5.0).

Returns The container_runtime_version of this V1NodeSystemInfo.

Return type str

kernel_version

Gets the kernel_version of this V1NodeSystemInfo. Kernel Version reported by the node from 'uname -r' (e.g. 3.16.0-0.bpo.4-amd64).

Returns The kernel_version of this V1NodeSystemInfo.

Return type str

kube_proxy_version

Gets the kube_proxy_version of this V1NodeSystemInfo. KubeProxy Version reported by the node.

Returns The kube_proxy_version of this V1NodeSystemInfo.

Return type str

kubelet_version
Gets the kubelet_version of this V1NodeSystemInfo. Kubelet Version reported by the node.

Returns The kubelet_version of this V1NodeSystemInfo.

Return type str

machine_id
Gets the machine_id of this V1NodeSystemInfo. MachineID reported by the node. For unique machine identification in the cluster this field is preferred. Learn more from man(5) machine-id: http://man7.org/linux/man-pages/man5/machine-id.5.html

Returns The machine_id of this V1NodeSystemInfo.

Return type str

operating_system
Gets the operating_system of this V1NodeSystemInfo. The Operating System reported by the node

Returns The operating_system of this V1NodeSystemInfo.

Return type str

os_image
Gets the os_image of this V1NodeSystemInfo. OS Image reported by the node from /etc/os-release (e.g. Debian GNU/Linux 7 (wheezy)).

Returns The os_image of this V1NodeSystemInfo.

Return type str

swagger_types = {'architecture': 'str', 'boot_id': 'str', 'container_runtime_version

system_uuid
Gets the system_uuid of this V1NodeSystemInfo. SystemUUID reported by the node. For unique machine identification MachineID is preferred. This field is specific to Red Hat hosts https://access.redhat.com/documentation/en-US/Red_Hat_Subscription_Management/1/html/RHSM/getting-system-uuid.html

Returns The system_uuid of this V1NodeSystemInfo.

Return type str

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_object_field_selector module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_object_field_selector.**V1ObjectFieldSelector**(*api_version=None*, *field_path=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ObjectFieldSelector. Version of the schema the FieldPath is written in terms of, defaults to "v1".

Returns The api_version of this V1ObjectFieldSelector.

Return type str

attribute_map = {'api_version': 'apiVersion', 'field_path': 'fieldPath'}

field_path
Gets the field_path of this V1ObjectFieldSelector. Path of the field to select in the specified API version.

Returns The field_path of this V1ObjectFieldSelector.

Return type str

swagger_types = {'api_version': 'str', 'field_path': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_object_meta module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_object_meta.**V1ObjectMeta**(*annotations=None*, *cluster_name=None*, *creation_timestamp=None*, *deletion_grace_period_seconds=None*, *deletion_timestamp=None*, *finalizers=None*, *generate_name=None*, *generation=None*, *initializers=None*, *labels=None*, *name=None*, *namespace=None*, *owner_references=None*, *resource_version=None*, *self_link=None*, *uid=None*)

Bases: `object`

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

annotations

Gets the annotations of this V1ObjectMeta. Annotations is an unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. More info: http://kubernetes.io/docs/user-guide/annotations

Returns The annotations of this V1ObjectMeta.

Return type dict(str, str)

attribute_map = {'annotations': 'annotations', 'cluster_name': 'clusterName', 'creat

cluster_name

Gets the cluster_name of this V1ObjectMeta. The name of the cluster which the object belongs to. This is used to distinguish resources with same name and namespace in different clusters. This field is not set anywhere right now and apiserver is going to ignore it if set in create or update request.

Returns The cluster_name of this V1ObjectMeta.

Return type str

creation_timestamp

Gets the creation_timestamp of this V1ObjectMeta. CreationTimestamp is a timestamp representing the server time when this object was created. It is not guaranteed to be set in happens-before order across separate operations. Clients may not set this value. It is represented in RFC3339 form and is in UTC. Populated by the system. Read-only. Null for lists. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The creation_timestamp of this V1ObjectMeta.

Return type datetime

deletion_grace_period_seconds

Gets the deletion_grace_period_seconds of this V1ObjectMeta. Number of seconds allowed for this object to gracefully terminate before it will be removed from the system. Only set when deletionTimestamp is also set. May only be shortened. Read-only.

Returns The deletion_grace_period_seconds of this V1ObjectMeta.

Return type int

deletion_timestamp

Gets the deletion_timestamp of this V1ObjectMeta. DeletionTimestamp is RFC 3339 date and time at which this resource will be deleted. This field is set by the server when a graceful deletion is requested by the user, and is not directly settable by a client. The resource is expected to be deleted (no longer visible from resource lists, and not reachable by name) after the time in this field, once the finalizers list is empty. As long as the finalizers list contains items, deletion is blocked. Once the deletionTimestamp is set, this value may not be unset or be set further into the future, although it may be shortened or the resource may be deleted prior to this time. For example, a user may request that a pod is deleted in 30 seconds. The Kubelet will react by sending a graceful termination signal to the containers in the pod. After that 30 seconds, the Kubelet will send a hard termination signal (SIGKILL) to the container and after cleanup, remove the pod from the API. In the presence of network partitions, this object may still exist after this timestamp, until an administrator or automated process can determine the resource is fully terminated. If not set, graceful deletion of the object has not been requested. Populated by the system when a graceful deletion is requested. Read-only. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The deletion_timestamp of this V1ObjectMeta.

Return type datetime

finalizers

Gets the finalizers of this V1ObjectMeta. Must be empty before the object is deleted from the registry. Each entry is an identifier for the responsible component that will remove the entry from the list. If the deletionTimestamp of the object is non-nil, entries in this list can only be removed.

Returns The finalizers of this V1ObjectMeta.

Return type list[str]

generate_name

Gets the generate_name of this V1ObjectMeta. GenerateName is an optional prefix, used by the server, to generate a unique name ONLY IF the Name field has not been provided. If this field is used, the name returned to the client will be different than the name passed. This value will also be combined with a unique suffix. The provided value has the same validation rules as the Name field, and may be truncated by the length of the suffix required to make the value unique on the server. If this field is specified and the generated name exists, the server will NOT return a 409 - instead, it will either return 201 Created or 500 with Reason ServerTimeout indicating a unique name could not be found in the time allotted, and the client should retry (optionally after the time indicated in the Retry-After header). Applied only if Name is not specified. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#idempotency

Returns The generate_name of this V1ObjectMeta.

Return type str

generation

Gets the generation of this V1ObjectMeta. A sequence number representing a specific generation of the desired state. Populated by the system. Read-only.

Returns The generation of this V1ObjectMeta.

Return type int

initializers

Gets the initializers of this V1ObjectMeta. An initializer is a controller which enforces some system invariant at object creation time. This field is a list of initializers that have not yet acted on this object. If nil or empty, this object has been completely initialized. Otherwise, the object is considered uninitialized and is hidden (in list/watch and get calls) from clients that haven't explicitly asked to observe uninitialized objects. When an object is created, the system will populate this list with the current set of initializers. Only privileged users may set or modify this list. Once it is empty, it may not be modified further by any user.

Returns The initializers of this V1ObjectMeta.

Return type V1Initializers

labels

Gets the labels of this V1ObjectMeta. Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services. More info: http://kubernetes.io/docs/user-guide/labels

Returns The labels of this V1ObjectMeta.

Return type dict(str, str)

name

Gets the name of this V1ObjectMeta. Name must be unique within a namespace. Is required when creating resources, although some resources may allow a client to request the generation of an appropriate name automatically. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names

Returns The name of this V1ObjectMeta.

Return type str

namespace
Gets the namespace of this V1ObjectMeta. Namespace defines the space within each name must be unique. An empty namespace is equivalent to the "default" namespace, but "default" is the canonical representation. Not all objects are required to be scoped to a namespace - the value of this field for those objects will be empty. Must be a DNS_LABEL. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/namespaces

Returns The namespace of this V1ObjectMeta.

Return type str

owner_references
Gets the owner_references of this V1ObjectMeta. List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.

Returns The owner_references of this V1ObjectMeta.

Return type list[*V1OwnerReference*]

resource_version
Gets the resource_version of this V1ObjectMeta. An opaque value that represents the internal version of this object that can be used by clients to determine when objects have changed. May be used for optimistic concurrency, change detection, and the watch operation on a resource or set of resources. Clients must treat these values as opaque and passed unmodified back to the server. They may only be valid for a particular resource or set of resources. Populated by the system. Read-only. Value must be treated as opaque by clients and . More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#concurrency-control-and-consistency

Returns The resource_version of this V1ObjectMeta.

Return type str

self_link
Gets the self_link of this V1ObjectMeta. SelfLink is a URL representing this object. Populated by the system. Read-only.

Returns The self_link of this V1ObjectMeta.

Return type str

swagger_types = {'annotations': 'dict(str, str)', 'cluster_name': 'str', 'creation_t

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

uid
Gets the uid of this V1ObjectMeta. UID is the unique in time and space value for this object. It is typically generated by the server on successful creation of a resource and is not allowed to change on PUT operations. Populated by the system. Read-only. More info: http://kubernetes.io/docs/user-guide/identifiers#uids

Returns The uid of this V1ObjectMeta.

Return type str

kubernetes.client.models.v1_object_reference module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_object_reference.**V1ObjectReference**(*api_version=None*, *field_path=None*, *kind=None*, *name=None*, *namespace=None*, *resource_version=None*, *uid=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1ObjectReference. API version of the referent.

Returns The api_version of this V1ObjectReference.

Return type str

attribute_map = {'api_version': 'apiVersion', 'field_path': 'fieldPath', 'kind': 'k

field_path

Gets the field_path of this V1ObjectReference. If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

Returns The field_path of this V1ObjectReference.

Return type str

kind

Gets the kind of this V1ObjectReference. Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ObjectReference.

Return type str

name

Gets the name of this V1ObjectReference. Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Returns The name of this V1ObjectReference.

Return type str

namespace

Gets the namespace of this V1ObjectReference. Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

Returns The namespace of this V1ObjectReference.

Return type str

resource_version
Gets the resource_version of this V1ObjectReference. Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#concurrency-control-and-consistency

Returns The resource_version of this V1ObjectReference.

Return type str

swagger_types = {'api_version': 'str', 'field_path': 'str', 'kind': 'str', 'name':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

uid
Gets the uid of this V1ObjectReference. UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

Returns The uid of this V1ObjectReference.

Return type str

kubernetes.client.models.v1_owner_reference module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_owner_reference.**V1OwnerReference**(*api_version=None*, *block_owner_deletion=None*, *controller=None*, *kind=None*, *name=None*, *uid=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1OwnerReference. API version of the referent.

Returns The api_version of this V1OwnerReference.

Return type str

attribute_map = {'api_version': 'apiVersion', 'block_owner_deletion': 'blockOwnerDel

block_owner_deletion
Gets the block_owner_deletion of this V1OwnerReference. If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is

removed. Defaults to false. To set this field, a user needs "delete" permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.

Returns The block_owner_deletion of this V1OwnerReference.

Return type bool

controller
Gets the controller of this V1OwnerReference. If true, this reference points to the managing controller.

Returns The controller of this V1OwnerReference.

Return type bool

kind
Gets the kind of this V1OwnerReference. Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1OwnerReference.

Return type str

name
Gets the name of this V1OwnerReference. Name of the referent. More info: http://kubernetes.io/docs/user-guide/identifiers#names

Returns The name of this V1OwnerReference.

Return type str

swagger_types = {'api_version': 'str', 'block_owner_deletion': 'bool', 'controller':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

uid
Gets the uid of this V1OwnerReference. UID of the referent. More info: http://kubernetes.io/docs/user-guide/identifiers#uids

Returns The uid of this V1OwnerReference.

Return type str

kubernetes.client.models.v1_persistent_volume module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume.**V1PersistentVolume**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1PersistentVolume. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1PersistentVolume.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1PersistentVolume. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1PersistentVolume.

Return type str

metadata
Gets the metadata of this V1PersistentVolume. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1PersistentVolume.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1PersistentVolume. Spec defines a specification of a persistent volume owned by the cluster. Provisioned by an administrator. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistent-volumes

Returns The spec of this V1PersistentVolume.

Return type *V1PersistentVolumeSpec*

status
Gets the status of this V1PersistentVolume. Status represents the current information/status for the persistent volume. Populated by the system. Read-only. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistent-volumes

Returns The status of this V1PersistentVolume.

Return type *V1PersistentVolumeStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_persistent_volume_claim module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume_claim.**V1PersistentVolumeClaim**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1PersistentVolumeClaim. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1PersistentVolumeClaim.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind

Gets the kind of this V1PersistentVolumeClaim. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1PersistentVolumeClaim.

Return type str

metadata

Gets the metadata of this V1PersistentVolumeClaim. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1PersistentVolumeClaim.

Return type *V1ObjectMeta*

spec

Gets the spec of this V1PersistentVolumeClaim. Spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Returns The spec of this V1PersistentVolumeClaim.

Return type *V1PersistentVolumeClaimSpec*

status

Gets the status of this V1PersistentVolumeClaim. Status represents the current information/status of a persistent volume claim. Read-only. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Returns The status of this V1PersistentVolumeClaim.

Return type *V1PersistentVolumeClaimStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_persistent_volume_claim_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume_claim_list.**V1PersistentVolumeClaimList**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1PersistentVolumeClaimList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1PersistentVolumeClaimList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1PersistentVolumeClaimList. A list of persistent volume claims. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Returns The items of this V1PersistentVolumeClaimList.

Return type list[*V1PersistentVolumeClaim*]

kind
Gets the kind of this V1PersistentVolumeClaimList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1PersistentVolumeClaimList.

Return type str

metadata
Gets the metadata of this V1PersistentVolumeClaimList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1PersistentVolumeClaimList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1PersistentVolumeClaim]', 'ki

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_persistent_volume_claim_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume_claim_spec.**V1PersistentVolumeClaimSpec**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

access_modes
Gets the access_modes of this V1PersistentVolumeClaimSpec. AccessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1

Returns The access_modes of this V1PersistentVolumeClaimSpec.

Return type list[str]

attribute_map = {'access_modes': 'accessModes', 'resources': 'resources', 'selector'

resources
Gets the resources of this V1PersistentVolumeClaimSpec. Resources represents the minimum resources the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

Returns The resources of this V1PersistentVolumeClaimSpec.

Return type *V1ResourceRequirements*

selector
Gets the selector of this V1PersistentVolumeClaimSpec. A label query over volumes to consider for binding.

Returns The selector of this V1PersistentVolumeClaimSpec.

Return type V1LabelSelector

storage_class_name
Gets the storage_class_name of this V1PersistentVolumeClaimSpec. Name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

Returns The storage_class_name of this V1PersistentVolumeClaimSpec.

Return type str

swagger_types = {'access_modes': 'list[str]', 'resources': 'V1ResourceRequirements',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

volume_mode
Gets the volume_mode of this V1PersistentVolumeClaimSpec. volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec. This is an alpha feature and may change in the future.

Returns The volume_mode of this V1PersistentVolumeClaimSpec.

Return type str

volume_name
Gets the volume_name of this V1PersistentVolumeClaimSpec. VolumeName is the binding reference to the PersistentVolume backing this claim.

Returns The volume_name of this V1PersistentVolumeClaimSpec.

Return type str

kubernetes.client.models.v1_persistent_volume_claim_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume_claim_status.**V1PersistentVolumeClaimSta**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

access_modes
Gets the access_modes of this V1PersistentVolumeClaimStatus. AccessModes contains the actual access modes the volume backing the PVC has. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1

Returns The access_modes of this V1PersistentVolumeClaimStatus.

Return type list[str]

attribute_map = {'access_modes': 'accessModes', 'capacity': 'capacity', 'conditions'

capacity
Gets the capacity of this V1PersistentVolumeClaimStatus. Represents the actual resources of the underlying volume.

Returns The capacity of this V1PersistentVolumeClaimStatus.

Return type dict(str, str)

conditions
Gets the conditions of this V1PersistentVolumeClaimStatus. Current Condition of persistent volume claim. If underlying persistent volume is being resized then the Condition will be set to 'ResizeStarted'.

Returns The conditions of this V1PersistentVolumeClaimStatus.

Return type list[V1PersistentVolumeClaimCondition]

phase
Gets the phase of this V1PersistentVolumeClaimStatus. Phase represents the current phase of PersistentVolumeClaim.

Returns The phase of this V1PersistentVolumeClaimStatus.

Return type str

swagger_types = {'access_modes': 'list[str]', 'capacity': 'dict(str, str)', 'conditi

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_persistent_volume_claim_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume_claim_volume_source.**V1PersistentVolume**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'claim_name': 'claimName', 'read_only': 'readOnly'}

claim_name
Gets the claim_name of this V1PersistentVolumeClaimVolumeSource. ClaimName is the name of a PersistentVolumeClaim in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Returns The claim_name of this V1PersistentVolumeClaimVolumeSource.

Return type str

read_only
Gets the read_only of this V1PersistentVolumeClaimVolumeSource. Will force the ReadOnly setting in VolumeMounts. Default false.

Returns The read_only of this V1PersistentVolumeClaimVolumeSource.

Return type bool

swagger_types = {'claim_name': 'str', 'read_only': 'bool'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_persistent_volume_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume_list.**V1PersistentVolumeList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1PersistentVolumeList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1PersistentVolumeList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1PersistentVolumeList. List of persistent volumes. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes

Returns The items of this V1PersistentVolumeList.

Return type list[*V1PersistentVolume*]

kind
Gets the kind of this V1PersistentVolumeList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1PersistentVolumeList.

Return type str

metadata
Gets the metadata of this V1PersistentVolumeList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1PersistentVolumeList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1PersistentVolume]', 'kind':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_persistent_volume_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume_spec.**V1PersistentVolumeSpec**(*access_modes=
aws_elastic_blo
azure_disk=No
azure_file=Non
ca-
pac-
ity=None,
cephfs=None,
cin-
der=None,
claim_ref=Non
csi=None,
fc=None,
flex_volume=N
flocker=None,
gce_persistent_
glus-
terfs=None,
host_path=Non
iscsi=None,
lo-
cal=None,
mount_options=
nfs=None,
node_affinity=N
per-
sis-
tent_volume_re
pho-
ton_persistent_
port-
worx_volume=
quobyte=None,
rbd=None,
scale_io=None
stor-
age_class_nam
stor-
a-
geos=None,
vol-
ume_mode=No
vsphere_volume*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

access_modes

Gets the access_modes of this V1PersistentVolumeSpec. AccessModes contains all ways the volume can be mounted. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes

Returns The access_modes of this V1PersistentVolumeSpec.

Return type list[str]

attribute_map = {'access_modes': 'accessModes', 'aws_elastic_block_store': 'awsElast

aws_elastic_block_store
Gets the aws_elastic_block_store of this V1PersistentVolumeSpec. AWSElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Returns The aws_elastic_block_store of this V1PersistentVolumeSpec.

Return type *V1AWSElasticBlockStoreVolumeSource*

azure_disk
Gets the azure_disk of this V1PersistentVolumeSpec. AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Returns The azure_disk of this V1PersistentVolumeSpec.

Return type *V1AzureDiskVolumeSource*

azure_file
Gets the azure_file of this V1PersistentVolumeSpec. AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

Returns The azure_file of this V1PersistentVolumeSpec.

Return type V1AzureFilePersistentVolumeSource

capacity
Gets the capacity of this V1PersistentVolumeSpec. A description of the persistent volume's resources and capacity. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#capacity

Returns The capacity of this V1PersistentVolumeSpec.

Return type dict(str, str)

cephfs
Gets the cephfs of this V1PersistentVolumeSpec. CephFS represents a Ceph FS mount on the host that shares a pod's lifetime

Returns The cephfs of this V1PersistentVolumeSpec.

Return type V1CephFSPersistentVolumeSource

cinder
Gets the cinder of this V1PersistentVolumeSpec. Cinder represents a cinder volume attached and mounted on kubelets host machine More info: https://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

Returns The cinder of this V1PersistentVolumeSpec.

Return type *V1CinderVolumeSource*

claim_ref
Gets the claim_ref of this V1PersistentVolumeSpec. ClaimRef is part of a bi-directional binding between PersistentVolume and PersistentVolumeClaim. Expected to be non-nil when bound. claim.VolumeName is the authoritative bind between PV and PVC. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#binding

Returns The claim_ref of this V1PersistentVolumeSpec.

Return type *V1ObjectReference*

csi
Gets the csi of this V1PersistentVolumeSpec. CSI represents storage that handled by an external CSI driver (Beta feature).

Returns The csi of this V1PersistentVolumeSpec.

Return type V1CSIPersistentVolumeSource

fc

Gets the fc of this V1PersistentVolumeSpec. FC represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Returns The fc of this V1PersistentVolumeSpec.

Return type *V1FCVolumeSource*

flex_volume

Gets the flex_volume of this V1PersistentVolumeSpec. FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Returns The flex_volume of this V1PersistentVolumeSpec.

Return type V1FlexPersistentVolumeSource

flocker

Gets the flocker of this V1PersistentVolumeSpec. Flocker represents a Flocker volume attached to a kubelet's host machine and exposed to the pod for its usage. This depends on the Flocker control service being running

Returns The flocker of this V1PersistentVolumeSpec.

Return type *V1FlockerVolumeSource*

gce_persistent_disk

Gets the gce_persistent_disk of this V1PersistentVolumeSpec. GCEPersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Returns The gce_persistent_disk of this V1PersistentVolumeSpec.

Return type *V1GCEPersistentDiskVolumeSource*

glusterfs

Gets the glusterfs of this V1PersistentVolumeSpec. Glusterfs represents a Glusterfs volume that is attached to a host and exposed to the pod. Provisioned by an admin. More info: https://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md

Returns The glusterfs of this V1PersistentVolumeSpec.

Return type *V1GlusterfsVolumeSource*

host_path

Gets the host_path of this V1PersistentVolumeSpec. HostPath represents a directory on the host. Provisioned by a developer or tester. This is useful for single-node development and testing only! On-host storage is not supported in any way and WILL NOT WORK in a multi-node cluster. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

Returns The host_path of this V1PersistentVolumeSpec.

Return type *V1HostPathVolumeSource*

iscsi

Gets the iscsi of this V1PersistentVolumeSpec. ISCSI represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin.

Returns The iscsi of this V1PersistentVolumeSpec.

Return type V1ISCSIPersistentVolumeSource

local
Gets the local of this V1PersistentVolumeSpec. Local represents directly-attached storage with node affinity

Returns The local of this V1PersistentVolumeSpec.

Return type V1LocalVolumeSource

mount_options
Gets the mount_options of this V1PersistentVolumeSpec. A list of mount options, e.g. ["ro", "soft"]. Not validated - mount will simply fail if one is invalid. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes/#mount-options

Returns The mount_options of this V1PersistentVolumeSpec.

Return type list[str]

nfs
Gets the nfs of this V1PersistentVolumeSpec. NFS represents an NFS mount on the host. Provisioned by an admin. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

Returns The nfs of this V1PersistentVolumeSpec.

Return type *V1NFSVolumeSource*

node_affinity
Gets the node_affinity of this V1PersistentVolumeSpec. NodeAffinity defines constraints that limit what nodes this volume can be accessed from. This field influences the scheduling of pods that use this volume.

Returns The node_affinity of this V1PersistentVolumeSpec.

Return type V1VolumeNodeAffinity

persistent_volume_reclaim_policy
Gets the persistent_volume_reclaim_policy of this V1PersistentVolumeSpec. What happens to a persistent volume when released from its claim. Valid options are Retain (default for manually created PersistentVolumes), Delete (default for dynamically provisioned PersistentVolumes), and Recycle (deprecated). Recycle must be supported by the volume plugin underlying this PersistentVolume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#reclaiming

Returns The persistent_volume_reclaim_policy of this V1PersistentVolumeSpec.

Return type str

photon_persistent_disk
Gets the photon_persistent_disk of this V1PersistentVolumeSpec. PhotonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine

Returns The photon_persistent_disk of this V1PersistentVolumeSpec.

Return type *V1PhotonPersistentDiskVolumeSource*

portworx_volume
Gets the portworx_volume of this V1PersistentVolumeSpec. PortworxVolume represents a portworx volume attached and mounted on kubelets host machine

Returns The portworx_volume of this V1PersistentVolumeSpec.

Return type V1PortworxVolumeSource

quobyte
Gets the quobyte of this V1PersistentVolumeSpec. Quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Returns The quobyte of this V1PersistentVolumeSpec.

Return type *V1QuobyteVolumeSource*

rbd
Gets the rbd of this V1PersistentVolumeSpec. RBD represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md

Returns The rbd of this V1PersistentVolumeSpec.

Return type V1RBDPersistentVolumeSource

scale_io
Gets the scale_io of this V1PersistentVolumeSpec. ScaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Returns The scale_io of this V1PersistentVolumeSpec.

Return type V1ScaleIOPersistentVolumeSource

storage_class_name
Gets the storage_class_name of this V1PersistentVolumeSpec. Name of StorageClass to which this persistent volume belongs. Empty value means that this volume does not belong to any StorageClass.

Returns The storage_class_name of this V1PersistentVolumeSpec.

Return type str

storageos
Gets the storageos of this V1PersistentVolumeSpec. StorageOS represents a StorageOS volume that is attached to the kubelet's host machine and mounted into the pod More info: https://releases.k8s.io/HEAD/examples/volumes/storageos/README.md

Returns The storageos of this V1PersistentVolumeSpec.

Return type V1StorageOSPersistentVolumeSource

swagger_types = {'access_modes': 'list[str]', 'aws_elastic_block_store': 'V1AWSElast

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

volume_mode
Gets the volume_mode of this V1PersistentVolumeSpec. volumeMode defines if a volume is intended to be used with a formatted filesystem or to remain in raw block state. Value of Filesystem is implied when not included in spec. This is an alpha feature and may change in the future.

Returns The volume_mode of this V1PersistentVolumeSpec.

Return type str

vsphere_volume
Gets the vsphere_volume of this V1PersistentVolumeSpec. VsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Returns The vsphere_volume of this V1PersistentVolumeSpec.

Return type *V1VsphereVirtualDiskVolumeSource*

kubernetes.client.models.v1_persistent_volume_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_persistent_volume_status.**V1PersistentVolumeStatus**(*message=None*, *phase=None*, *reason=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'message': 'message', 'phase': 'phase', 'reason': 'reason'}

message
Gets the message of this V1PersistentVolumeStatus. A human-readable message indicating details about why the volume is in this state.

Returns The message of this V1PersistentVolumeStatus.

Return type str

phase
Gets the phase of this V1PersistentVolumeStatus. Phase indicates if a volume is available, bound to a claim, or released by a claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#phase

Returns The phase of this V1PersistentVolumeStatus.

Return type str

reason
Gets the reason of this V1PersistentVolumeStatus. Reason is a brief CamelCase string that describes any failure and is meant for machine parsing and tidy display in the CLI.

Returns The reason of this V1PersistentVolumeStatus.

Return type str

swagger_types = {'message': 'str', 'phase': 'str', 'reason': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_photon_persistent_disk_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_photon_persistent_disk_volume_source.**V1PhotonPersistentDi**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'fs_type': 'fsType', 'pd_id': 'pdID'}

fs_type
Gets the fs_type of this V1PhotonPersistentDiskVolumeSource. Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

Returns The fs_type of this V1PhotonPersistentDiskVolumeSource.

Return type str

pd_id
Gets the pd_id of this V1PhotonPersistentDiskVolumeSource. ID that identifies Photon Controller persistent disk

Returns The pd_id of this V1PhotonPersistentDiskVolumeSource.

Return type str

swagger_types = {'fs_type': 'str', 'pd_id': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_pod module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod.**V1Pod**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1Pod. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Pod.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1Pod. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Pod.

Return type str

metadata
Gets the metadata of this V1Pod. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Pod.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1Pod. Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1Pod.

Return type *V1PodSpec*

status
Gets the status of this V1Pod. Most recently observed status of the pod. This data may not be up to date. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1Pod.

Return type *V1PodStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_pod_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod_condition.**V1PodCondition**(*last_probe_time=None*, *last_transition_time=None*, *message=None*, *reason=None*, *status=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'last_probe_time': 'lastProbeTime', 'last_transition_time': 'lastTr

last_probe_time
Gets the last_probe_time of this V1PodCondition. Last time we probed the condition.

Returns The last_probe_time of this V1PodCondition.

Return type datetime

last_transition_time
Gets the last_transition_time of this V1PodCondition. Last time the condition transitioned from one status to another.

Returns The last_transition_time of this V1PodCondition.

Return type datetime

message
Gets the message of this V1PodCondition. Human-readable message indicating details about last transition.

Returns The message of this V1PodCondition.

Return type str

reason
Gets the reason of this V1PodCondition. Unique, one-word, CamelCase reason for the condition's last transition.

Returns The reason of this V1PodCondition.

Return type str

status
Gets the status of this V1PodCondition. Status is the status of the condition. Can be True, False, Unknown. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions

Returns The status of this V1PodCondition.

Return type str

swagger_types = {'last_probe_time': 'datetime', 'last_transition_time': 'datetime',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1PodCondition. Type is the type of the condition. Currently only Ready. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions

Returns The type of this V1PodCondition.

Return type str

kubernetes.client.models.v1_pod_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod_list.**V1PodList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1PodList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1PodList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1PodList. List of pods. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md

Returns The items of this V1PodList.

Return type list[*V1Pod*]

kind
Gets the kind of this V1PodList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1PodList.

Return type str

metadata
Gets the metadata of this V1PodList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1PodList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1Pod]', 'kind': 'str', 'meta

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_pod_security_context module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod_security_context.**V1PodSecurityContext**(*fs_group=None*, *run_as_group=None*, *run_as_non_root=No*, *run_as_user=None*, *se_linux_options=No*, *supplemental_groups=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'fs_group': 'fsGroup', 'run_as_group': 'runAsGroup', 'run_as_non_ro

fs_group

Gets the fs_group of this V1PodSecurityContext. A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw—- If unset, the Kubelet will not modify the ownership and permissions of any volume.

Returns The fs_group of this V1PodSecurityContext.

Return type int

run_as_group

Gets the run_as_group of this V1PodSecurityContext. The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.

Returns The run_as_group of this V1PodSecurityContext.

Return type int

run_as_non_root

Gets the run_as_non_root of this V1PodSecurityContext. Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Returns The run_as_non_root of this V1PodSecurityContext.

Return type bool

run_as_user

Gets the run_as_user of this V1PodSecurityContext. The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.

Returns The run_as_user of this V1PodSecurityContext.

Return type int

se_linux_options

Gets the se_linux_options of this V1PodSecurityContext. The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.

Returns The se_linux_options of this V1PodSecurityContext.

Return type *V1SELinuxOptions*

supplemental_groups
Gets the supplemental_groups of this V1PodSecurityContext. A list of groups applied to the first process run in each container, in addition to the container's primary GID. If unspecified, no groups will be added to any container.

Returns The supplemental_groups of this V1PodSecurityContext.

Return type list[int]

swagger_types = {'fs_group': 'int', 'run_as_group': 'int', 'run_as_non_root': 'bool

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_pod_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod_spec.**V1PodSpec**(*active_deadline_seconds=None*, *affinity=None*, *automount_service_account_token=None*, *containers=None*, *dns_config=None*, *dns_policy=None*, *host_aliases=None*, *host_ipc=None*, *host_network=None*, *host_pid=None*, *hostname=None*, *image_pull_secrets=None*, *init_containers=None*, *node_name=None*, *node_selector=None*, *priority=None*, *priority_class_name=None*, *restart_policy=None*, *scheduler_name=None*, *security_context=None*, *service_account=None*, *service_account_name=None*, *share_process_namespace=None*, *subdomain=None*, *termination_grace_period_seconds=None*, *tolerations=None*, *volumes=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

active_deadline_seconds
Gets the active_deadline_seconds of this V1PodSpec. Optional duration in seconds the pod may be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer.

Returns The active_deadline_seconds of this V1PodSpec.

Return type int

affinity
Gets the affinity of this V1PodSpec. If specified, the pod's scheduling constraints

Returns The affinity of this V1PodSpec.

Return type V1Affinity

attribute_map = {'active_deadline_seconds': 'activeDeadlineSeconds', 'affinity': 'af

automount_service_account_token
Gets the automount_service_account_token of this V1PodSpec. AutomountServiceAccountToken indicates whether a service account token should be automatically mounted.

Returns The automount_service_account_token of this V1PodSpec.

Return type bool

containers
Gets the containers of this V1PodSpec. List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

Returns The containers of this V1PodSpec.

Return type list[*V1Container*]

dns_config
Gets the dns_config of this V1PodSpec. Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy.

Returns The dns_config of this V1PodSpec.

Return type V1PodDNSConfig

dns_policy
Gets the dns_policy of this V1PodSpec. Set DNS policy for the pod. Defaults to "ClusterFirst". Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' or 'None'. DNS parameters given in DNSConfig will be merged with the policy selected with DNSPolicy. To have DNS options set along with hostNetwork, you have to specify DNS policy explicitly to 'ClusterFirstWithHostNet'.

Returns The dns_policy of this V1PodSpec.

Return type str

host_aliases
Gets the host_aliases of this V1PodSpec. HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified. This is only valid for non-hostNetwork pods.

Returns The host_aliases of this V1PodSpec.

Return type list[V1HostAlias]

host_ipc
Gets the host_ipc of this V1PodSpec. Use the host's ipc namespace. Optional: Default to false.

Returns The host_ipc of this V1PodSpec.

Return type bool

host_network
Gets the host_network of this V1PodSpec. Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified. Default to false.

Returns The host_network of this V1PodSpec.

Return type bool

host_pid
Gets the host_pid of this V1PodSpec. Use the host's pid namespace. Optional: Default to false.

Returns The host_pid of this V1PodSpec.

Return type bool

hostname
Gets the hostname of this V1PodSpec. Specifies the hostname of the Pod If not specified, the pod's hostname will be set to a system-defined value.

Returns The hostname of this V1PodSpec.

Return type str

image_pull_secrets
Gets the image_pull_secrets of this V1PodSpec. ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. For example, in the case of docker, only DockerConfig type secrets are honored. More info: https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod

Returns The image_pull_secrets of this V1PodSpec.

Return type list[*V1LocalObjectReference*]

init_containers
Gets the init_containers of this V1PodSpec. List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, or Liveness probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/init-containers/

Returns The init_containers of this V1PodSpec.

Return type list[*V1Container*]

node_name
Gets the node_name of this V1PodSpec. NodeName is a request to schedule this pod onto a specific node. If it is non-empty, the scheduler simply schedules this pod onto that node, assuming that it fits resource requirements.

Returns The node_name of this V1PodSpec.

Return type str

node_selector
Gets the node_selector of this V1PodSpec. NodeSelector is a selector which must be true for the pod to fit

on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/

Returns The node_selector of this V1PodSpec.

Return type dict(str, str)

priority

Gets the priority of this V1PodSpec. The priority value. Various system components use this field to find the priority of the pod. When Priority Admission Controller is enabled, it prevents users from setting this field. The admission controller populates this field from PriorityClassName. The higher the value, the higher the priority.

Returns The priority of this V1PodSpec.

Return type int

priority_class_name

Gets the priority_class_name of this V1PodSpec. If specified, indicates the pod's priority. "system-node-critical" and "system-cluster-critical" are two special keywords which indicate the highest priorities with the former being the highest priority. Any other name must be defined by creating a PriorityClass object with that name. If not specified, the pod priority will be default or zero if there is no default.

Returns The priority_class_name of this V1PodSpec.

Return type str

restart_policy

Gets the restart_policy of this V1PodSpec. Restart policy for all containers within the pod. One of Always, OnFailure, Never. Default to Always. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy

Returns The restart_policy of this V1PodSpec.

Return type str

scheduler_name

Gets the scheduler_name of this V1PodSpec. If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.

Returns The scheduler_name of this V1PodSpec.

Return type str

security_context

Gets the security_context of this V1PodSpec. SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See type description for default values of each field.

Returns The security_context of this V1PodSpec.

Return type *V1PodSecurityContext*

service_account

Gets the service_account of this V1PodSpec. DeprecatedServiceAccount is a depreciated alias for ServiceAccountName. Deprecated: Use serviceAccountName instead.

Returns The service_account of this V1PodSpec.

Return type str

service_account_name

Gets the service_account_name of this V1PodSpec. ServiceAccountName is the name of the Ser-

viceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

Returns The service_account_name of this V1PodSpec.

Return type str

share_process_namespace
Gets the share_process_namespace of this V1PodSpec. Share a single process namespace between all of the containers in a pod. When this is set containers will be able to view and signal processes from other containers in the same pod, and the first process in each container will not be assigned PID 1. HostPID and ShareProcessNamespace cannot both be set. Optional: Default to false. This field is alpha-level and is honored only by servers that enable the PodShareProcessNamespace feature.

Returns The share_process_namespace of this V1PodSpec.

Return type bool

subdomain
Gets the subdomain of this V1PodSpec. If specified, the fully qualified Pod hostname will be "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>". If not specified, the pod will not have a domainname at all.

Returns The subdomain of this V1PodSpec.

Return type str

swagger_types = {'active_deadline_seconds': 'int', 'affinity': 'V1Affinity', 'automo

termination_grace_period_seconds
Gets the termination_grace_period_seconds of this V1PodSpec. Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. Defaults to 30 seconds.

Returns The termination_grace_period_seconds of this V1PodSpec.

Return type int

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

tolerations
Gets the tolerations of this V1PodSpec. If specified, the pod's tolerations.

Returns The tolerations of this V1PodSpec.

Return type list[V1Toleration]

volumes
Gets the volumes of this V1PodSpec. List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes

Returns The volumes of this V1PodSpec.

Return type list[*V1Volume*]

kubernetes.client.models.v1_pod_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod_status.**V1PodStatus**(*conditions=None*, *container_statuses=None*, *host_ip=None*, *init_container_statuses=None*, *message=None*, *nominated_node_name=None*, *phase=None*, *pod_ip=None*, *qos_class=None*, *reason=None*, *start_time=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'conditions': 'conditions', 'container_statuses': 'containerStatuses

conditions

Gets the conditions of this V1PodStatus. Current service state of pod. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions

Returns The conditions of this V1PodStatus.

Return type list[*V1PodCondition*]

container_statuses

Gets the container_statuses of this V1PodStatus. The list has one entry per container in the manifest. Each entry is currently the output of *docker inspect*. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status

Returns The container_statuses of this V1PodStatus.

Return type list[*V1ContainerStatus*]

host_ip

Gets the host_ip of this V1PodStatus. IP address of the host to which the pod is assigned. Empty if not yet scheduled.

Returns The host_ip of this V1PodStatus.

Return type str

init_container_statuses

Gets the init_container_statuses of this V1PodStatus. The list has one entry per init container in the manifest. The most recent successful init container will have ready = true, the most recently started container will have startTime set. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status

Returns The init_container_statuses of this V1PodStatus.

Return type list[*V1ContainerStatus*]

message
Gets the message of this V1PodStatus. A human readable message indicating details about why the pod is in this condition.

Returns The message of this V1PodStatus.

Return type str

nominated_node_name
Gets the nominated_node_name of this V1PodStatus. nominatedNodeName is set only when this pod preempts other pods on the node, but it cannot be scheduled right away as preemption victims receive their graceful termination periods. This field does not guarantee that the pod will be scheduled on this node. Scheduler may decide to place the pod elsewhere if other nodes become available sooner. Scheduler may also decide to give the resources on this node to a higher priority pod that is created after preemption. As a result, this field may be different than PodSpec.nodeName when the pod is scheduled.

Returns The nominated_node_name of this V1PodStatus.

Return type str

phase
Gets the phase of this V1PodStatus. Current condition of the pod. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-phase

Returns The phase of this V1PodStatus.

Return type str

pod_ip
Gets the pod_ip of this V1PodStatus. IP address allocated to the pod. Routable at least within the cluster. Empty if not yet allocated.

Returns The pod_ip of this V1PodStatus.

Return type str

qos_class
Gets the qos_class of this V1PodStatus. The Quality of Service (QOS) classification assigned to the pod based on resource requirements See PodQOSClass type for available QOS classes More info: https://git.k8s.io/community/contributors/design-proposals/node/resource-qos.md

Returns The qos_class of this V1PodStatus.

Return type str

reason
Gets the reason of this V1PodStatus. A brief CamelCase message indicating details about why the pod is in this state. e.g. 'Evicted'

Returns The reason of this V1PodStatus.

Return type str

start_time
Gets the start_time of this V1PodStatus. RFC 3339 date and time at which the object was acknowledged by the Kubelet. This is before the Kubelet pulled the container image(s) for the pod.

Returns The start_time of this V1PodStatus.

Return type datetime

swagger_types = {'conditions': 'list[V1PodCondition]', 'container_statuses': 'list[V

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_pod_template module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod_template.**V1PodTemplate**(*api_version=None*, *kind=None*, *metadata=None*, *template=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1PodTemplate. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1PodTemplate.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1PodTemplate. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1PodTemplate.

Return type str

metadata
Gets the metadata of this V1PodTemplate. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1PodTemplate.

Return type *V1ObjectMeta*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

template
Gets the template of this V1PodTemplate. Template defines the pods that will be created from this pod template. https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The template of this V1PodTemplate.

Return type *V1PodTemplateSpec*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_pod_template_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod_template_list.**V1PodTemplateList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1PodTemplateList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1PodTemplateList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1PodTemplateList. List of pod templates

Returns The items of this V1PodTemplateList.

Return type list[*V1PodTemplate*]

kind
Gets the kind of this V1PodTemplateList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1PodTemplateList.

Return type str

metadata
Gets the metadata of this V1PodTemplateList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1PodTemplateList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1PodTemplate]', 'kind': 'str

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_pod_template_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_pod_template_spec.**V1PodTemplateSpec**(*metadata=None*, *spec=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'metadata': 'metadata', 'spec': 'spec'}

metadata
Gets the metadata of this V1PodTemplateSpec. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1PodTemplateSpec.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1PodTemplateSpec. Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1PodTemplateSpec.

Return type *V1PodSpec*

swagger_types = {'metadata': 'V1ObjectMeta', 'spec': 'V1PodSpec'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_preconditions module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_preconditions.**V1Preconditions**(*uid=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'uid': 'uid'}

swagger_types = {'uid': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

uid
Gets the uid of this V1Preconditions. Specifies the target UID.

Returns The uid of this V1Preconditions.

Return type str

kubernetes.client.models.v1_probe module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_probe.**V1Probe**(*_exec=None*, *failure_threshold=None*, *http_get=None*, *initial_delay_seconds=None*, *period_seconds=None*, *success_threshold=None*, *tcp_socket=None*, *timeout_seconds=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'_exec': 'exec', 'failure_threshold': 'failureThreshold', 'http_get

failure_threshold
Gets the failure_threshold of this V1Probe. Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

Returns The failure_threshold of this V1Probe.

Return type int

http_get
Gets the http_get of this V1Probe. HTTPGet specifies the http request to perform.

Returns The http_get of this V1Probe.

Return type *V1HTTPGetAction*

initial_delay_seconds
Gets the initial_delay_seconds of this V1Probe. Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Returns The initial_delay_seconds of this V1Probe.

Return type int

period_seconds
Gets the period_seconds of this V1Probe. How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.

Returns The period_seconds of this V1Probe.

Return type int

success_threshold
Gets the success_threshold of this V1Probe. Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.

Returns The success_threshold of this V1Probe.

Return type int

swagger_types = {'_exec': 'V1ExecAction', 'failure_threshold': 'int', 'http_get': '

tcp_socket
Gets the tcp_socket of this V1Probe. TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported

Returns The tcp_socket of this V1Probe.

Return type *V1TCPSocketAction*

timeout_seconds
Gets the timeout_seconds of this V1Probe. Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Returns The timeout_seconds of this V1Probe.

Return type int

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_quobyte_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_quobyte_volume_source.**V1QuobyteVolumeSource**(*group=None*, *read_only=None*, *registry=None*, *user=None*, *volume=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'group': 'group', 'read_only': 'readOnly', 'registry': 'registry',

group
Gets the group of this V1QuobyteVolumeSource. Group to map volume access to Default is no group

Returns The group of this V1QuobyteVolumeSource.

Return type str

read_only
Gets the read_only of this V1QuobyteVolumeSource. ReadOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.

Returns The read_only of this V1QuobyteVolumeSource.

Return type bool

registry
Gets the registry of this V1QuobyteVolumeSource. Registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes

Returns The registry of this V1QuobyteVolumeSource.

Return type str

swagger_types = {'group': 'str', 'read_only': 'bool', 'registry': 'str', 'user': '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

user
Gets the user of this V1QuobyteVolumeSource. User to map volume access to Defaults to serivceaccount user

Returns The user of this V1QuobyteVolumeSource.

Return type str

volume
Gets the volume of this V1QuobyteVolumeSource. Volume is a string that references an already created Quobyte volume by name.

Returns The volume of this V1QuobyteVolumeSource.

Return type str

kubernetes.client.models.v1_rbd_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_rbd_volume_source.**V1RBDVolumeSource**(*fs_type=None*, *image=None*, *keyring=None*, *monitors=None*, *pool=None*, *read_only=None*, *secret_ref=None*, *user=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'fs_type': 'fsType', 'image': 'image', 'keyring': 'keyring', 'monit

fs_type
Gets the fs_type of this V1RBDVolumeSource. Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd

Returns The fs_type of this V1RBDVolumeSource.

Return type str

image
Gets the image of this V1RBDVolumeSource. The rados image name. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

Returns The image of this V1RBDVolumeSource.

Return type str

keyring
Gets the keyring of this V1RBDVolumeSource. Keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

Returns The keyring of this V1RBDVolumeSource.

Return type str

monitors
Gets the monitors of this V1RBDVolumeSource. A collection of Ceph monitors. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

Returns The monitors of this V1RBDVolumeSource.

Return type list[str]

pool
Gets the pool of this V1RBDVolumeSource. The rados pool name. Default is rbd. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

Returns The pool of this V1RBDVolumeSource.

Return type str

read_only
Gets the read_only of this V1RBDVolumeSource. ReadOnly here will force the ReadOnly setting

in VolumeMounts. Defaults to false. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

Returns The read_only of this V1RBDVolumeSource.

Return type bool

secret_ref
Gets the secret_ref of this V1RBDVolumeSource. SecretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

Returns The secret_ref of this V1RBDVolumeSource.

Return type *V1LocalObjectReference*

swagger_types = {'fs_type': 'str', 'image': 'str', 'keyring': 'str', 'monitors': '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

user
Gets the user of this V1RBDVolumeSource. The rados user name. Default is admin. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

Returns The user of this V1RBDVolumeSource.

Return type str

kubernetes.client.models.v1_replication_controller module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_replication_controller.**V1ReplicationController**(*api_version=*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ReplicationController. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ReplicationController.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1ReplicationController. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ReplicationController.

Return type str

metadata
Gets the metadata of this V1ReplicationController. If the Labels of a ReplicationController are empty, they are defaulted to be the same as the Pod(s) that the replication controller manages. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1ReplicationController.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1ReplicationController. Spec defines the specification of the desired behavior of the replication controller. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1ReplicationController.

Return type *V1ReplicationControllerSpec*

status
Gets the status of this V1ReplicationController. Status is the most recently observed status of the replication controller. This data may be out of date by some window of time. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1ReplicationController.

Return type *V1ReplicationControllerStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_replication_controller_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_replication_controller_condition.**V1ReplicationControllerC**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'last_transition_time': 'lastTransitionTime', 'message': 'message',

last_transition_time
Gets the last_transition_time of this V1ReplicationControllerCondition. The last time the condition transitioned from one status to another.

Returns The last_transition_time of this V1ReplicationControllerCondition.

Return type datetime

message
Gets the message of this V1ReplicationControllerCondition. A human readable message indicating details about the transition.

Returns The message of this V1ReplicationControllerCondition.

Return type str

reason
Gets the reason of this V1ReplicationControllerCondition. The reason for the condition's last transition.

Returns The reason of this V1ReplicationControllerCondition.

Return type str

status
Gets the status of this V1ReplicationControllerCondition. Status of the condition, one of True, False, Unknown.

Returns The status of this V1ReplicationControllerCondition.

Return type str

swagger_types = {'last_transition_time': 'datetime', 'message': 'str', 'reason': 's

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1ReplicationControllerCondition. Type of replication controller condition.

Returns The type of this V1ReplicationControllerCondition.

Return type str

kubernetes.client.models.v1_replication_controller_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_replication_controller_list.**V1ReplicationControllerList**(*a*
i
k
r
d

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ReplicationControllerList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ReplicationControllerList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1ReplicationControllerList. List of replication controllers. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller

Returns The items of this V1ReplicationControllerList.

Return type list[*V1ReplicationController*]

kind
Gets the kind of this V1ReplicationControllerList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ReplicationControllerList.

Return type str

metadata
Gets the metadata of this V1ReplicationControllerList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1ReplicationControllerList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1ReplicationController]', 'ki

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_replication_controller_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_replication_controller_spec.**V1ReplicationControllerSpec**(

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'min_ready_seconds': 'minReadySeconds', 'replicas': 'replicas', 'se

min_ready_seconds
Gets the min_ready_seconds of this V1ReplicationControllerSpec. Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)

Returns The min_ready_seconds of this V1ReplicationControllerSpec.

Return type int

replicas
Gets the replicas of this V1ReplicationControllerSpec. Replicas is the number of desired replicas. This is a pointer to distinguish between explicit zero and unspecified. Defaults to 1. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#what-is-a-replicationcontroller

Returns The replicas of this V1ReplicationControllerSpec.

Return type int

selector
Gets the selector of this V1ReplicationControllerSpec. Selector is a label query over pods that should match the Replicas count. If Selector is empty, it is defaulted to the labels present on the Pod template. Label keys and values that must match in order to be controlled by this replication controller, if empty defaulted to labels on Pod template. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors

Returns The selector of this V1ReplicationControllerSpec.

Return type dict(str, str)

swagger_types = {'min_ready_seconds': 'int', 'replicas': 'int', 'selector': 'dict(s

template
Gets the template of this V1ReplicationControllerSpec. Template is the object that describes the pod that will be created if insufficient replicas are detected. This takes precedence over a TemplateRef. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#pod-template

Returns The template of this V1ReplicationControllerSpec.

Return type *V1PodTemplateSpec*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_replication_controller_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_replication_controller_status.**V1ReplicationControllerStat**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'available_replicas': 'availableReplicas', 'conditions': 'condition

available_replicas
Gets the available_replicas of this V1ReplicationControllerStatus. The number of available replicas (ready for at least minReadySeconds) for this replication controller.

Returns The available_replicas of this V1ReplicationControllerStatus.

Return type int

conditions
Gets the conditions of this V1ReplicationControllerStatus. Represents the latest available observations of a replication controller's current state.

Returns The conditions of this V1ReplicationControllerStatus.

Return type list[*V1ReplicationControllerCondition*]

fully_labeled_replicas
Gets the fully_labeled_replicas of this V1ReplicationControllerStatus. The number of pods that have labels matching the labels of the pod template of the replication controller.

Returns The fully_labeled_replicas of this V1ReplicationControllerStatus.

Return type int

observed_generation
Gets the observed_generation of this V1ReplicationControllerStatus. ObservedGeneration reflects the generation of the most recently observed replication controller.

Returns The observed_generation of this V1ReplicationControllerStatus.

Return type int

ready_replicas
Gets the ready_replicas of this V1ReplicationControllerStatus. The number of ready replicas for this replication controller.

Returns The ready_replicas of this V1ReplicationControllerStatus.

Return type int

replicas
Gets the replicas of this V1ReplicationControllerStatus. Replicas is the most recently oberved number of replicas. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#what-is-a-replicationcontroller

Returns The replicas of this V1ReplicationControllerStatus.

Return type int

swagger_types = {'available_replicas': 'int', 'conditions': 'list[V1ReplicationContr

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_resource_field_selector module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_resource_field_selector.**V1ResourceFieldSelector**(*container_n*
di-
vi-
sor=None,
re-
source=Non

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'container_name': 'containerName', 'divisor': 'divisor', 'resource'

container_name
Gets the container_name of this V1ResourceFieldSelector. Container name: required for volumes, optional for env vars

Returns The container_name of this V1ResourceFieldSelector.

Return type str

divisor
Gets the divisor of this V1ResourceFieldSelector. Specifies the output format of the exposed resources, defaults to "1"

Returns The divisor of this V1ResourceFieldSelector.

Return type str

resource
Gets the resource of this V1ResourceFieldSelector. Required: resource to select

Returns The resource of this V1ResourceFieldSelector.

Return type str

swagger_types = {'container_name': 'str', 'divisor': 'str', 'resource': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_resource_quota module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_resource_quota.**V1ResourceQuota**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ResourceQuota. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ResourceQuota.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1ResourceQuota. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ResourceQuota.

Return type str

metadata
Gets the metadata of this V1ResourceQuota. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1ResourceQuota.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1ResourceQuota. Spec defines the desired quota. https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1ResourceQuota.

Return type *V1ResourceQuotaSpec*

status
Gets the status of this V1ResourceQuota. Status defines the actual enforced quota and its current usage. https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1ResourceQuota.

Return type *V1ResourceQuotaStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_resource_quota_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_resource_quota_list.**V1ResourceQuotaList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ResourceQuotaList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ResourceQuotaList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1ResourceQuotaList. Items is a list of ResourceQuota objects. More info: https://kubernetes.io/docs/concepts/policy/resource-quotas/

Returns The items of this V1ResourceQuotaList.

Return type list[*V1ResourceQuota*]

kind
Gets the kind of this V1ResourceQuotaList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ResourceQuotaList.

Return type str

metadata
Gets the metadata of this V1ResourceQuotaList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1ResourceQuotaList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1ResourceQuota]', 'kind': 's

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_resource_quota_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_resource_quota_spec.**V1ResourceQuotaSpec**(*hard=None*, *scopes=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'hard': 'hard', 'scopes': 'scopes'}

hard
Gets the hard of this V1ResourceQuotaSpec. Hard is the set of desired hard limits for each named resource. More info: https://kubernetes.io/docs/concepts/policy/resource-quotas/

Returns The hard of this V1ResourceQuotaSpec.

Return type dict(str, str)

scopes
Gets the scopes of this V1ResourceQuotaSpec. A collection of filters that must match each object tracked by a quota. If not specified, the quota matches all objects.

Returns The scopes of this V1ResourceQuotaSpec.

Return type list[str]

swagger_types = {'hard': 'dict(str, str)', 'scopes': 'list[str]'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_resource_quota_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_resource_quota_status.**V1ResourceQuotaStatus**(*hard=None*, *used=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'hard': 'hard', 'used': 'used'}

hard
Gets the hard of this V1ResourceQuotaStatus. Hard is the set of enforced hard limits for each named resource. More info: https://kubernetes.io/docs/concepts/policy/resource-quotas/

Returns The hard of this V1ResourceQuotaStatus.

Return type dict(str, str)

swagger_types = {'hard': 'dict(str, str)', 'used': 'dict(str, str)'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

used
Gets the used of this V1ResourceQuotaStatus. Used is the current observed total usage of the resource in the namespace.

Returns The used of this V1ResourceQuotaStatus.

Return type dict(str, str)

kubernetes.client.models.v1_resource_requirements module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_resource_requirements.**V1ResourceRequirements**(*limits=None*, *requests=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'limits': 'limits', 'requests': 'requests'}

limits
Gets the limits of this V1ResourceRequirements. Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/

Returns The limits of this V1ResourceRequirements.

Return type dict(str, str)

requests
Gets the requests of this V1ResourceRequirements. Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. More info: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/

Returns The requests of this V1ResourceRequirements.

Return type dict(str, str)

swagger_types = {'limits': 'dict(str, str)', 'requests': 'dict(str, str)'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_scale module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_scale.**V1Scale**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1Scale. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Scale.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1Scale. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Scale.

Return type str

metadata
Gets the metadata of this V1Scale. Standard object metadata; More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata.

Returns The metadata of this V1Scale.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1Scale. defines the behavior of the scale. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status.

Returns The spec of this V1Scale.

Return type *V1ScaleSpec*

status
Gets the status of this V1Scale. current status of the scale. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status. Read-only.

Returns The status of this V1Scale.

Return type *V1ScaleStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_scale_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_scale_spec.**V1ScaleSpec**(*replicas=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'replicas': 'replicas'}

replicas
Gets the replicas of this V1ScaleSpec. desired number of instances for the scaled object.

Returns The replicas of this V1ScaleSpec.

Return type int

swagger_types = {'replicas': 'int'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_scale_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_scale_status.**V1ScaleStatus**(*replicas=None*, *selector=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'replicas': 'replicas', 'selector': 'selector'}

replicas

Gets the replicas of this V1ScaleStatus. actual number of observed instances of the scaled object.

Returns The replicas of this V1ScaleStatus.

Return type int

selector

Gets the selector of this V1ScaleStatus. label query over pods that should match the replicas count. This is same as the label selector but in the string format to avoid introspection by clients. The string will be in the same format as the query-param syntax. More info about label selectors: http://kubernetes.io/docs/user-guide/labels#label-selectors

Returns The selector of this V1ScaleStatus.

Return type str

swagger_types = {'replicas': 'int', 'selector': 'str'}

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1_se_linux_options module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_se_linux_options.**V1SELinuxOptions**(*level=None*, *role=None*, *type=None*, *user=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'level': 'level', 'role': 'role', 'type': 'type', 'user': 'user'}

level
Gets the level of this V1SELinuxOptions. Level is SELinux level label that applies to the container.

Returns The level of this V1SELinuxOptions.

Return type str

role
Gets the role of this V1SELinuxOptions. Role is a SELinux role label that applies to the container.

Returns The role of this V1SELinuxOptions.

Return type str

swagger_types = {'level': 'str', 'role': 'str', 'type': 'str', 'user': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1SELinuxOptions. Type is a SELinux type label that applies to the container.

Returns The type of this V1SELinuxOptions.

Return type str

user
Gets the user of this V1SELinuxOptions. User is a SELinux user label that applies to the container.

Returns The user of this V1SELinuxOptions.

Return type str

kubernetes.client.models.v1_secret module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_secret.**V1Secret**(*api_version=None*, *data=None*, *kind=None*, *metadata=None*, *string_data=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1Secret. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Secret.

Return type str

attribute_map = {'api_version': 'apiVersion', 'data': 'data', 'kind': 'kind', 'meta

data
Gets the data of this V1Secret. Data contains the secret data. Each key must consist of alphanumeric characters, '-', '_' or '.'. The serialized form of the secret data is a base64 encoded string, representing the arbitrary (possibly non-string) data value here. Described in https://tools.ietf.org/html/rfc4648#section-4

Returns The data of this V1Secret.

Return type dict(str, str)

kind
Gets the kind of this V1Secret. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Secret.

Return type str

metadata
Gets the metadata of this V1Secret. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Secret.

Return type *V1ObjectMeta*

string_data
Gets the string_data of this V1Secret. stringData allows specifying non-binary secret data in string form. It is provided as a write-only convenience method. All keys and values are merged into the data field on write, overwriting any existing values. It is never output when reading from the API.

Returns The string_data of this V1Secret.

Return type dict(str, str)

swagger_types = {'api_version': 'str', 'data': 'dict(str, str)', 'kind': 'str', 'me

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1Secret. Used to facilitate programmatic handling of secret data.

Returns The type of this V1Secret.

Return type str

kubernetes.client.models.v1_secret_key_selector module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_secret_key_selector.**V1SecretKeySelector**(*key=None*, *name=None*, *optional=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'key': 'key', 'name': 'name', 'optional': 'optional'}

key
Gets the key of this V1SecretKeySelector. The key of the secret to select from. Must be a valid secret key.

Returns The key of this V1SecretKeySelector.

Return type str

name
Gets the name of this V1SecretKeySelector. Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Returns The name of this V1SecretKeySelector.

Return type str

optional
Gets the optional of this V1SecretKeySelector. Specify whether the Secret or it's key must be defined

Returns The optional of this V1SecretKeySelector.

Return type bool

swagger_types = {'key': 'str', 'name': 'str', 'optional': 'bool'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_secret_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_secret_list.**V1SecretList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1SecretList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1SecretList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1SecretList. Items is a list of secret objects. More info: https://kubernetes.io/docs/concepts/configuration/secret

Returns The items of this V1SecretList.

Return type list[*V1Secret*]

kind
Gets the kind of this V1SecretList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1SecretList.

Return type str

metadata
Gets the metadata of this V1SecretList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1SecretList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1Secret]', 'kind': 'str', 'm

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_secret_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_secret_volume_source.**V1SecretVolumeSource**(*default_mode=None*, *items=None*, *optional=None*, *secret_name=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'default_mode': 'defaultMode', 'items': 'items', 'optional': 'optio

default_mode
Gets the default_mode of this V1SecretVolumeSource. Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not

affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Returns The default_mode of this V1SecretVolumeSource.

Return type int

items
Gets the items of this V1SecretVolumeSource. If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Returns The items of this V1SecretVolumeSource.

Return type list[*V1KeyToPath*]

optional
Gets the optional of this V1SecretVolumeSource. Specify whether the Secret or it's keys must be defined

Returns The optional of this V1SecretVolumeSource.

Return type bool

secret_name
Gets the secret_name of this V1SecretVolumeSource. Name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

Returns The secret_name of this V1SecretVolumeSource.

Return type str

swagger_types = {'default_mode': 'int', 'items': 'list[V1KeyToPath]', 'optional': 'l

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_security_context module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_security_context.**V1SecurityContext**(*allow_privilege_escalation=No*
capabilities=None,
privileged=None,
read_only_root_filesystem=Non
run_as_group=None,
run_as_non_root=None,
run_as_user=None,
se_linux_options=None)

Bases: `object`

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

allow_privilege_escalation
Gets the allow_privilege_escalation of this V1SecurityContext. AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN

Returns The allow_privilege_escalation of this V1SecurityContext.

Return type bool

attribute_map = {'allow_privilege_escalation': 'allowPrivilegeEscalation', 'capabilit

capabilities
Gets the capabilities of this V1SecurityContext. The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.

Returns The capabilities of this V1SecurityContext.

Return type *V1Capabilities*

privileged
Gets the privileged of this V1SecurityContext. Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false.

Returns The privileged of this V1SecurityContext.

Return type bool

read_only_root_filesystem
Gets the read_only_root_filesystem of this V1SecurityContext. Whether this container has a read-only root filesystem. Default is false.

Returns The read_only_root_filesystem of this V1SecurityContext.

Return type bool

run_as_group
Gets the run_as_group of this V1SecurityContext. The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Returns The run_as_group of this V1SecurityContext.

Return type int

run_as_non_root
Gets the run_as_non_root of this V1SecurityContext. Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Returns The run_as_non_root of this V1SecurityContext.

Return type bool

run_as_user
Gets the run_as_user of this V1SecurityContext. The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Returns The run_as_user of this V1SecurityContext.

Return type int

se_linux_options
Gets the se_linux_options of this V1SecurityContext. The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Returns The se_linux_options of this V1SecurityContext.

Return type *V1SELinuxOptions*

swagger_types = {'allow_privilege_escalation': 'bool', 'capabilities': 'V1Capabiliti

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_service module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_service.**V1Service**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1Service. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1Service.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1Service. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1Service.

Return type str

metadata
Gets the metadata of this V1Service. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1Service.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1Service. Spec defines the behavior of a service. https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1Service.

Return type *V1ServiceSpec*

status
Gets the status of this V1Service. Most recently observed status of the service. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1Service.

Return type *V1ServiceStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_service_account module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_service_account.**V1ServiceAccount**(*api_version=None*, *automount_service_account_token=No*, *image_pull_secrets=None*, *kind=None*, *metadata=None*, *secrets=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ServiceAccount. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ServiceAccount.

Return type str

attribute_map = {'api_version': 'apiVersion', 'automount_service_account_token': 'au

automount_service_account_token
Gets the automount_service_account_token of this V1ServiceAccount. AutomountServiceAccountToken indicates whether pods running as this service account should have an API token automatically mounted. Can be overridden at the pod level.

Returns The automount_service_account_token of this V1ServiceAccount.

Return type bool

image_pull_secrets
Gets the image_pull_secrets of this V1ServiceAccount. ImagePullSecrets is a list of references to secrets in the same namespace to use for pulling any images in pods that reference this ServiceAccount. ImagePullSecrets are distinct from Secrets because Secrets can be mounted in the pod, but ImagePullSecrets are only accessed by the kubelet. More info: https://kubernetes.io/docs/concepts/containers/images/#specifying-imagepullsecrets-on-a-pod

Returns The image_pull_secrets of this V1ServiceAccount.

Return type list[*V1LocalObjectReference*]

kind
Gets the kind of this V1ServiceAccount. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ServiceAccount.

Return type str

metadata
Gets the metadata of this V1ServiceAccount. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1ServiceAccount.

Return type *V1ObjectMeta*

secrets
Gets the secrets of this V1ServiceAccount. Secrets is the list of secrets allowed to be used by pods running using this ServiceAccount. More info: https://kubernetes.io/docs/concepts/configuration/secret

Returns The secrets of this V1ServiceAccount.

Return type list[*V1ObjectReference*]

swagger_types = {'api_version': 'str', 'automount_service_account_token': 'bool', 'i

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_service_account_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_service_account_list.**V1ServiceAccountList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ServiceAccountList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ServiceAccountList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1ServiceAccountList. List of ServiceAccounts. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

Returns The items of this V1ServiceAccountList.

Return type list[*V1ServiceAccount*]

kind
Gets the kind of this V1ServiceAccountList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ServiceAccountList.

Return type str

metadata
Gets the metadata of this V1ServiceAccountList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1ServiceAccountList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1ServiceAccount]', 'kind': '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_service_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_service_list.**V1ServiceList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1ServiceList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1ServiceList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1ServiceList. List of services

Returns The items of this V1ServiceList.

Return type list[*V1Service*]

kind
Gets the kind of this V1ServiceList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1ServiceList.

Return type str

metadata
Gets the metadata of this V1ServiceList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1ServiceList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1Service]', 'kind': 'str', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_service_port module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_service_port.**V1ServicePort**(*name=None*, *node_port=None*, *port=None*, *protocol=None*, *target_port=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'name': 'name', 'node_port': 'nodePort', 'port': 'port', 'protocol

name

Gets the name of this V1ServicePort. The name of this port within the service. This must be a DNS_LABEL. All ports within a ServiceSpec must have unique names. This maps to the 'Name' field in EndpointPort objects. Optional if only one ServicePort is defined on this service.

Returns The name of this V1ServicePort.

Return type str

node_port

Gets the node_port of this V1ServicePort. The port on each node on which this service is exposed when type=NodePort or LoadBalancer. Usually assigned by the system. If specified, it will be allocated to the service if unused or else creation of the service will fail. Default is to auto-allocate a port if the ServiceType of this Service requires one. More info: https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport

Returns The node_port of this V1ServicePort.

Return type int

port

Gets the port of this V1ServicePort. The port that will be exposed by this service.

Returns The port of this V1ServicePort.

Return type int

protocol

Gets the protocol of this V1ServicePort. The IP protocol for this port. Supports "TCP" and "UDP". Default is TCP.

Returns The protocol of this V1ServicePort.

Return type str

swagger_types = {'name': 'str', 'node_port': 'int', 'port': 'int', 'protocol': 'st

target_port

Gets the target_port of this V1ServicePort. Number or name of the port to access on the pods targeted by the service. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME. If this is a string, it will be looked up as a named port in the target Pod's container ports. If this is not specified, the value of the 'port' field is used (an identity map). This field is ignored for services with clusterIP=None, and should be omitted or set equal to the 'port' field. More info: https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service

Returns The target_port of this V1ServicePort.

Return type object

to_dict()

Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_service_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_service_spec.**V1ServiceSpec**(*cluster_ip=None*, *external_i_ps=None*, *external_name=None*, *external_traffic_policy=None*, *health_check_node_port=None*, *load_balancer_ip=None*, *load_balancer_source_ranges=None*, *ports=None*, *publish_not_ready_addresses=None*, *selector=None*, *session_affinity=None*, *session_affinity_config=None*, *type=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'cluster_ip': 'clusterIP', 'external_i_ps': 'externalIPs', 'externa

cluster_ip
Gets the cluster_ip of this V1ServiceSpec. clusterIP is the IP address of the service and is usually assigned randomly by the master. If an address is specified manually and is not in use by others, it will be allocated to the service; otherwise, creation of the service will fail. This field can not be changed through updates. Valid values are "None", empty string (""), or a valid IP address. "None" can be specified for headless services when proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. Ignored if type is ExternalName. More info: https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies

Returns The cluster_ip of this V1ServiceSpec.

Return type str

external_i_ps
Gets the external_i_ps of this V1ServiceSpec. externalIPs is a list of IP addresses for which nodes in the cluster will also accept traffic for this service. These IPs are not managed by Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load-balancers that are not part of the Kubernetes system.

Returns The external_i_ps of this V1ServiceSpec.

Return type list[str]

external_name
Gets the external_name of this V1ServiceSpec. externalName is the external reference that kubedns or equivalent will return as a CNAME record for this service. No proxying will be involved. Must be a valid RFC-1123 hostname (https://tools.ietf.org/html/rfc1123) and requires Type to be ExternalName.

Returns The external_name of this V1ServiceSpec.

Return type str

external_traffic_policy
Gets the external_traffic_policy of this V1ServiceSpec. externalTrafficPolicy denotes if this Service desires to route external traffic to node-local or cluster-wide endpoints. "Local" preserves the client source IP and avoids a second hop for LoadBalancer and Nodeport type services, but risks potentially imbalanced traffic spreading. "Cluster" obscures the client source IP and may cause a second hop to another node, but should have good overall load-spreading.

Returns The external_traffic_policy of this V1ServiceSpec.

Return type str

health_check_node_port
Gets the health_check_node_port of this V1ServiceSpec. healthCheckNodePort specifies the healthcheck nodePort for the service. If not specified, HealthCheckNodePort is created by the service api backend with the allocated nodePort. Will use user-specified nodePort value if specified by the client. Only effects when Type is set to LoadBalancer and ExternalTrafficPolicy is set to Local.

Returns The health_check_node_port of this V1ServiceSpec.

Return type int

load_balancer_ip
Gets the load_balancer_ip of this V1ServiceSpec. Only applies to Service Type: LoadBalancer LoadBalancer will get created with the IP specified in this field. This feature depends on whether the underlying cloud-provider supports specifying the loadBalancerIP when a load balancer is created. This field will be ignored if the cloud-provider does not support the feature.

Returns The load_balancer_ip of this V1ServiceSpec.

Return type str

load_balancer_source_ranges
Gets the load_balancer_source_ranges of this V1ServiceSpec. If specified and supported by the platform, this will restrict traffic through the cloud-provider load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature." More info: https://kubernetes.io/docs/tasks/access-application-cluster/configure-cloud-provider-firewall/

Returns The load_balancer_source_ranges of this V1ServiceSpec.

Return type list[str]

ports
Gets the ports of this V1ServiceSpec. The list of ports that are exposed by this service. More info: https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies

Returns The ports of this V1ServiceSpec.

Return type list[*V1ServicePort*]

publish_not_ready_addresses
Gets the publish_not_ready_addresses of this V1ServiceSpec. publishNotReadyAddresses, when set to true, indicates that DNS implementations must publish the notReadyAddresses of subsets for the Endpoints associated with the Service. The default value is false. The primary use case for setting this field is to use a StatefulSet's Headless Service to propagate SRV records for its Pods without respect to their readiness

for purpose of peer discovery. This field will replace the service.alpha.kubernetes.io/tolerate-unready-endpoints when that annotation is deprecated and all clients have been converted to use this field.

Returns The publish_not_ready_addresses of this V1ServiceSpec.

Return type bool

selector
Gets the selector of this V1ServiceSpec. Route service traffic to pods with label keys and values matching this selector. If empty or not present, the service is assumed to have an external process managing its endpoints, which Kubernetes will not modify. Only applies to types ClusterIP, NodePort, and LoadBalancer. Ignored if type is ExternalName. More info: https://kubernetes.io/docs/concepts/services-networking/service/

Returns The selector of this V1ServiceSpec.

Return type dict(str, str)

session_affinity
Gets the session_affinity of this V1ServiceSpec. Supports "ClientIP" and "None". Used to maintain session affinity. Enable client IP based session affinity. Must be ClientIP or None. Defaults to None. More info: https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies

Returns The session_affinity of this V1ServiceSpec.

Return type str

session_affinity_config
Gets the session_affinity_config of this V1ServiceSpec. sessionAffinityConfig contains the configurations of session affinity.

Returns The session_affinity_config of this V1ServiceSpec.

Return type V1SessionAffinityConfig

swagger_types = {'cluster_ip': 'str', 'external_i_ps': 'list[str]', 'external_name':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1ServiceSpec. type determines how the Service is exposed. Defaults to ClusterIP. Valid options are ExternalName, ClusterIP, NodePort, and LoadBalancer. "ExternalName" maps to the specified externalName. "ClusterIP" allocates a cluster-internal IP address for load-balancing to endpoints. Endpoints are determined by the selector or if that is not specified, by manual construction of an Endpoints object. If clusterIP is "None", no virtual IP is allocated and the endpoints are published as a set of endpoints rather than a stable IP. "NodePort" builds on ClusterIP and allocates a port on every node which routes to the clusterIP. "LoadBalancer" builds on NodePort and creates an external load-balancer (if supported in the current cloud) which routes to the clusterIP. More info: https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services—service-types

Returns The type of this V1ServiceSpec.

Return type str

kubernetes.client.models.v1_service_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_service_status.**V1ServiceStatus**(*load_balancer=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'load_balancer': 'loadBalancer'}

load_balancer

Gets the load_balancer of this V1ServiceStatus. LoadBalancer contains the current status of the load-balancer, if one is present.

Returns The load_balancer of this V1ServiceStatus.

Return type *V1LoadBalancerStatus*

swagger_types = {'load_balancer': 'V1LoadBalancerStatus'}

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1_tcp_socket_action module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_tcp_socket_action.**V1TCPSocketAction**(*host=None*, *port=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'host': 'host', 'port': 'port'}

host

Gets the host of this V1TCPSocketAction. Optional: Host name to connect to, defaults to the pod IP.

Returns The host of this V1TCPSocketAction.

Return type str

port

Gets the port of this V1TCPSocketAction. Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Returns The port of this V1TCPSocketAction.

Return type object

swagger_types = {'host': 'str', 'port': 'object'}

to_dict()

Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_volume module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_volume.**V1Volume**(*aws_elastic_block_store=None*, *azure_disk=None*, *azure_file=None*, *cephfs=None*, *cinder=None*, *config_map=None*, *downward_api=None*, *empty_dir=None*, *fc=None*, *flex_volume=None*, *flocker=None*, *gce_persistent_disk=None*, *git_repo=None*, *glusterfs=None*, *host_path=None*, *iscsi=None*, *name=None*, *nfs=None*, *persistent_volume_claim=None*, *photon_persistent_disk=None*, *portworx_volume=None*, *projected=None*, *quobyte=None*, *rbd=None*, *scale_io=None*, *secret=None*, *storageos=None*, *vsphere_volume=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'aws_elastic_block_store': 'awsElasticBlockStore', 'azure_disk': 'a

aws_elastic_block_store
Gets the aws_elastic_block_store of this V1Volume. AWSElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Returns The aws_elastic_block_store of this V1Volume.

Return type *V1AWSElasticBlockStoreVolumeSource*

azure_disk
Gets the azure_disk of this V1Volume. AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Returns The azure_disk of this V1Volume.

Return type *V1AzureDiskVolumeSource*

azure_file
Gets the azure_file of this V1Volume. AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

Returns The azure_file of this V1Volume.

Return type *V1AzureFileVolumeSource*

cephfs
Gets the cephfs of this V1Volume. CephFS represents a Ceph FS mount on the host that shares a pod's lifetime

Returns The cephfs of this V1Volume.

Return type *V1CephFSVolumeSource*

cinder
Gets the cinder of this V1Volume. Cinder represents a cinder volume attached and mounted on kubelets host machine More info: https://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

Returns The cinder of this V1Volume.

Return type *V1CinderVolumeSource*

config_map
Gets the config_map of this V1Volume. ConfigMap represents a configMap that should populate this volume

Returns The config_map of this V1Volume.

Return type *V1ConfigMapVolumeSource*

downward_api
Gets the downward_api of this V1Volume. DownwardAPI represents downward API about the pod that should populate this volume

Returns The downward_api of this V1Volume.

Return type *V1DownwardAPIVolumeSource*

empty_dir
Gets the empty_dir of this V1Volume. EmptyDir represents a temporary directory that shares a pod's lifetime. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Returns The empty_dir of this V1Volume.

Return type *V1EmptyDirVolumeSource*

fc
Gets the fc of this V1Volume. FC represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Returns The fc of this V1Volume.

Return type *V1FCVolumeSource*

flex_volume
Gets the flex_volume of this V1Volume. FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Returns The flex_volume of this V1Volume.

Return type *V1FlexVolumeSource*

flocker
Gets the flocker of this V1Volume. Flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running

Returns The flocker of this V1Volume.

Return type *V1FlockerVolumeSource*

gce_persistent_disk
Gets the gce_persistent_disk of this V1Volume. GCEPersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Returns The gce_persistent_disk of this V1Volume.

Return type *V1GCEPersistentDiskVolumeSource*

git_repo
Gets the git_repo of this V1Volume. GitRepo represents a git repository at a particular revision.

Returns The git_repo of this V1Volume.

Return type *V1GitRepoVolumeSource*

glusterfs
Gets the glusterfs of this V1Volume. Glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. More info: https://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md

Returns The glusterfs of this V1Volume.

Return type *V1GlusterfsVolumeSource*

host_path
Gets the host_path of this V1Volume. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

Returns The host_path of this V1Volume.

Return type *V1HostPathVolumeSource*

iscsi
Gets the iscsi of this V1Volume. ISCSI represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://releases.k8s.io/HEAD/examples/volumes/iscsi/README.md

Returns The iscsi of this V1Volume.

Return type *V1ISCSIVolumeSource*

name
Gets the name of this V1Volume. Volume's name. Must be a DNS_LABEL and unique within the pod. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Returns The name of this V1Volume.

Return type str

nfs
Gets the nfs of this V1Volume. NFS represents an NFS mount on the host that shares a pod's lifetime More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

Returns The nfs of this V1Volume.

Return type *V1NFSVolumeSource*

persistent_volume_claim
Gets the persistent_volume_claim of this V1Volume. PersistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Returns The persistent_volume_claim of this V1Volume.

Return type *V1PersistentVolumeClaimVolumeSource*

photon_persistent_disk
Gets the photon_persistent_disk of this V1Volume. PhotonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine

Returns The photon_persistent_disk of this V1Volume.

Return type *V1PhotonPersistentDiskVolumeSource*

portworx_volume
Gets the portworx_volume of this V1Volume. PortworxVolume represents a portworx volume attached and mounted on kubelets host machine

Returns The portworx_volume of this V1Volume.

Return type V1PortworxVolumeSource

projected
Gets the projected of this V1Volume. Items for all in one resources secrets, configmaps, and downward API

Returns The projected of this V1Volume.

Return type V1ProjectedVolumeSource

quobyte
Gets the quobyte of this V1Volume. Quobyte represents a Quobyte mount on the host that shares a pod's lifetime

Returns The quobyte of this V1Volume.

Return type *V1QuobyteVolumeSource*

rbd
Gets the rbd of this V1Volume. RBD represents a Rados Block Device mount on the host that shares a pod's lifetime. More info: https://releases.k8s.io/HEAD/examples/volumes/rbd/README.md

Returns The rbd of this V1Volume.

Return type *V1RBDVolumeSource*

scale_io
Gets the scale_io of this V1Volume. ScaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes.

Returns The scale_io of this V1Volume.

Return type V1ScaleIOVolumeSource

secret
Gets the secret of this V1Volume. Secret represents a secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

Returns The secret of this V1Volume.

Return type *V1SecretVolumeSource*

storageos
Gets the storageos of this V1Volume. StorageOS represents a StorageOS volume attached and mounted on Kubernetes nodes.

Returns The storageos of this V1Volume.

Return type V1StorageOSVolumeSource

swagger_types = {'aws_elastic_block_store': 'V1AWSElasticBlockStoreVolumeSource', 'az

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

vsphere_volume
Gets the vsphere_volume of this V1Volume. VsphereVolume represents a vSphere volume attached and mounted on kubelets host machine

Returns The vsphere_volume of this V1Volume.

Return type *V1VsphereVirtualDiskVolumeSource*

kubernetes.client.models.v1_volume_mount module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_volume_mount.**V1VolumeMount**(*mount_path=None*, *mount_propagation=None*, *name=None*, *read_only=None*, *sub_path=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'mount_path': 'mountPath', 'mount_propagation': 'mountPropagation',

mount_path
Gets the mount_path of this V1VolumeMount. Path within the container at which the volume should be mounted. Must not contain ':'.

Returns The mount_path of this V1VolumeMount.

Return type str

mount_propagation
Gets the mount_propagation of this V1VolumeMount. mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationHostToContainer is used. This field is beta in 1.10.

Returns The mount_propagation of this V1VolumeMount.

Return type str

name
Gets the name of this V1VolumeMount. This must match the Name of a Volume.

Returns The name of this V1VolumeMount.

Return type str

read_only
Gets the read_only of this V1VolumeMount. Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Returns The read_only of this V1VolumeMount.

Return type bool

sub_path
Gets the sub_path of this V1VolumeMount. Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

Returns The sub_path of this V1VolumeMount.

Return type str

swagger_types = {'mount_path': 'str', 'mount_propagation': 'str', 'name': 'str', 'r

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1_vsphere_virtual_disk_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1_vsphere_virtual_disk_volume_source.**V1VsphereVirtualDiskV**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'fs_type': 'fsType', 'storage_policy_id': 'storagePolicyID', 'stora

fs_type
Gets the fs_type of this V1VsphereVirtualDiskVolumeSource. Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

Returns The fs_type of this V1VsphereVirtualDiskVolumeSource.

Return type str

storage_policy_id
Gets the storage_policy_id of this V1VsphereVirtualDiskVolumeSource. Storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.

Returns The storage_policy_id of this V1VsphereVirtualDiskVolumeSource.

Return type str

storage_policy_name
Gets the storage_policy_name of this V1VsphereVirtualDiskVolumeSource. Storage Policy Based Management (SPBM) profile name.

Returns The storage_policy_name of this V1VsphereVirtualDiskVolumeSource.

Return type str

swagger_types = {'fs_type': 'str', 'storage_policy_id': 'str', 'storage_policy_name'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

volume_path
Gets the volume_path of this V1VsphereVirtualDiskVolumeSource. Path that identifies vSphere volume vmdk

Returns The volume_path of this V1VsphereVirtualDiskVolumeSource.

Return type str

kubernetes.client.models.v1alpha1_certificate_signing_request module

kubernetes.client.models.v1alpha1_certificate_signing_request_condition module

kubernetes.client.models.v1alpha1_certificate_signing_request_list module

kubernetes.client.models.v1alpha1_certificate_signing_request_spec module

kubernetes.client.models.v1alpha1_certificate_signing_request_status module

kubernetes.client.models.v1alpha1_cluster_role module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_cluster_role.**V1alpha1ClusterRole**(*aggregation_rule=None*, *api_version=None*, *kind=None*, *metadata=None*, *rules=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

aggregation_rule
Gets the aggregation_rule of this V1alpha1ClusterRole. AggregationRule is an optional field that describes how to build the Rules for this ClusterRole. If AggregationRule is set, then the Rules are controller managed and direct changes to Rules will be stomped by the controller.

Returns The aggregation_rule of this V1alpha1ClusterRole.

Return type V1alpha1AggregationRule

api_version
Gets the api_version of this V1alpha1ClusterRole. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1alpha1ClusterRole.

Return type str

attribute_map = {'aggregation_rule': 'aggregationRule', 'api_version': 'apiVersion',

kind
Gets the kind of this V1alpha1ClusterRole. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1alpha1ClusterRole.

Return type str

metadata
Gets the metadata of this V1alpha1ClusterRole. Standard object's metadata.

Returns The metadata of this V1alpha1ClusterRole.

Return type *V1ObjectMeta*

rules
Gets the rules of this V1alpha1ClusterRole. Rules holds all the PolicyRules for this ClusterRole

Returns The rules of this V1alpha1ClusterRole.

Return type list[*V1alpha1PolicyRule*]

swagger_types = {'aggregation_rule': 'V1alpha1AggregationRule', 'api_version': 'str'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1alpha1_cluster_role_binding module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_cluster_role_binding.**V1alpha1ClusterRoleBinding**(*api*
kin
met
dat
role
sub
ject

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1alpha1ClusterRoleBinding. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1alpha1ClusterRoleBinding.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1alpha1ClusterRoleBinding. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1alpha1ClusterRoleBinding.

Return type str

metadata
Gets the metadata of this V1alpha1ClusterRoleBinding. Standard object's metadata.

Returns The metadata of this V1alpha1ClusterRoleBinding.

Return type *V1ObjectMeta*

role_ref
Gets the role_ref of this V1alpha1ClusterRoleBinding. RoleRef can only reference a ClusterRole in the global namespace. If the RoleRef cannot be resolved, the Authorizer must return an error.

Returns The role_ref of this V1alpha1ClusterRoleBinding.

Return type *V1alpha1RoleRef*

subjects
Gets the subjects of this V1alpha1ClusterRoleBinding. Subjects holds references to the objects the role applies to.

Returns The subjects of this V1alpha1ClusterRoleBinding.

Return type list[*V1alpha1Subject*]

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1alpha1_cluster_role_binding_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_cluster_role_binding_list.**V1alpha1ClusterRoleBindi**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1alpha1ClusterRoleBindingList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1alpha1ClusterRoleBindingList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1alpha1ClusterRoleBindingList. Items is a list of ClusterRoleBindings

Returns The items of this V1alpha1ClusterRoleBindingList.

Return type list[*V1alpha1ClusterRoleBinding*]

kind
Gets the kind of this V1alpha1ClusterRoleBindingList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1alpha1ClusterRoleBindingList.

Return type str

metadata
Gets the metadata of this V1alpha1ClusterRoleBindingList. Standard object's metadata.

Returns The metadata of this V1alpha1ClusterRoleBindingList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1alpha1ClusterRoleBinding]',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1alpha1_cluster_role_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_cluster_role_list.**V1alpha1ClusterRoleList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1alpha1ClusterRoleList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1alpha1ClusterRoleList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1alpha1ClusterRoleList. Items is a list of ClusterRoles

Returns The items of this V1alpha1ClusterRoleList.

Return type list[*V1alpha1ClusterRole*]

kind
Gets the kind of this V1alpha1ClusterRoleList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1alpha1ClusterRoleList.

Return type str

metadata
Gets the metadata of this V1alpha1ClusterRoleList. Standard object's metadata.

Returns The metadata of this V1alpha1ClusterRoleList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1alpha1ClusterRole]', 'kind':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1alpha1_policy_rule module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_policy_rule.**V1alpha1PolicyRule**(*api_groups=None*, *non_resource_ur_ls=None*, *resource_names=None*, *resources=None*, *verbs=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_groups
Gets the api_groups of this V1alpha1PolicyRule. APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed.

Returns The api_groups of this V1alpha1PolicyRule.

Return type list[str]

attribute_map = {'api_groups': 'apiGroups', 'non_resource_ur_ls': 'nonResourceURLs',

non_resource_ur_ls
Gets the non_resource_ur_ls of this V1alpha1PolicyRule. NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path This name is intentionally different than the internal type so that the DefaultConvert works nicely and because the ordering may be different. Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.

Returns The non_resource_ur_ls of this V1alpha1PolicyRule.

Return type list[str]

resource_names
Gets the resource_names of this V1alpha1PolicyRule. ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.

Returns The resource_names of this V1alpha1PolicyRule.

Return type list[str]

resources
Gets the resources of this V1alpha1PolicyRule. Resources is a list of resources this rule applies to. ResourceAll represents all resources.

Returns The resources of this V1alpha1PolicyRule.

Return type list[str]

swagger_types = {'api_groups': 'list[str]', 'non_resource_ur_ls': 'list[str]', 'reso

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

verbs
Gets the verbs of this V1alpha1PolicyRule. Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule. VerbAll represents all kinds.

Returns The verbs of this V1alpha1PolicyRule.

Return type list[str]

kubernetes.client.models.v1alpha1_role module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_role.**V1alpha1Role**(*api_version=None*, *kind=None*, *metadata=None*, *rules=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1alpha1Role. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1alpha1Role.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind

Gets the kind of this V1alpha1Role. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1alpha1Role.

Return type str

metadata

Gets the metadata of this V1alpha1Role. Standard object's metadata.

Returns The metadata of this V1alpha1Role.

Return type *V1ObjectMeta*

rules

Gets the rules of this V1alpha1Role. Rules holds all the PolicyRules for this Role

Returns The rules of this V1alpha1Role.

Return type list[*V1alpha1PolicyRule*]

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1alpha1_role_binding module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_role_binding.**V1alpha1RoleBinding**(*api_version=None*, *kind=None*, *metadata=None*, *role_ref=None*, *subjects=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1alpha1RoleBinding. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1alpha1RoleBinding.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind

Gets the kind of this V1alpha1RoleBinding. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1alpha1RoleBinding.

Return type str

metadata

Gets the metadata of this V1alpha1RoleBinding. Standard object's metadata.

Returns The metadata of this V1alpha1RoleBinding.

Return type *V1ObjectMeta*

role_ref

Gets the role_ref of this V1alpha1RoleBinding. RoleRef can reference a Role in the current namespace or a ClusterRole in the global namespace. If the RoleRef cannot be resolved, the Authorizer must return an error.

Returns The role_ref of this V1alpha1RoleBinding.

Return type *V1alpha1RoleRef*

subjects

Gets the subjects of this V1alpha1RoleBinding. Subjects holds references to the objects the role applies to.

Returns The subjects of this V1alpha1RoleBinding.

Return type list[*V1alpha1Subject*]

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1alpha1_role_binding_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_role_binding_list.**V1alpha1RoleBindingList**(*api_version=None, items=None, kind=None, metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1alpha1RoleBindingList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1alpha1RoleBindingList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1alpha1RoleBindingList. Items is a list of RoleBindings

Returns The items of this V1alpha1RoleBindingList.

Return type list[*V1alpha1RoleBinding*]

kind
Gets the kind of this V1alpha1RoleBindingList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1alpha1RoleBindingList.

Return type str

metadata
Gets the metadata of this V1alpha1RoleBindingList. Standard object's metadata.

Returns The metadata of this V1alpha1RoleBindingList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1alpha1RoleBinding]', 'kind':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1alpha1_role_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_role_list.**V1alpha1RoleList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1alpha1RoleList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1alpha1RoleList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1alpha1RoleList. Items is a list of Roles

Returns The items of this V1alpha1RoleList.

Return type list[*V1alpha1Role*]

kind
Gets the kind of this V1alpha1RoleList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1alpha1RoleList.

Return type str

metadata
Gets the metadata of this V1alpha1RoleList. Standard object's metadata.

Returns The metadata of this V1alpha1RoleList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1alpha1Role]', 'kind': 'str'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1alpha1_role_ref module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_role_ref.**V1alpha1RoleRef**(*api_group=None*, *kind=None*, *name=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_group
Gets the api_group of this V1alpha1RoleRef. APIGroup is the group for the resource being referenced

Returns The api_group of this V1alpha1RoleRef.

Return type str

attribute_map = {'api_group': 'apiGroup', 'kind': 'kind', 'name': 'name'}

kind
Gets the kind of this V1alpha1RoleRef. Kind is the type of resource being referenced

Returns The kind of this V1alpha1RoleRef.

Return type str

name
Gets the name of this V1alpha1RoleRef. Name is the name of resource being referenced

Returns The name of this V1alpha1RoleRef.

Return type str

swagger_types = {'api_group': 'str', 'kind': 'str', 'name': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1alpha1_subject module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1alpha1_subject.**V1alpha1Subject**(*api_version=None*, *kind=None*, *name=None*, *namespace=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1alpha1Subject. APIVersion holds the API group and version of the referenced subject. Defaults to "v1" for ServiceAccount subjects. Defaults to "rbac.authorization.k8s.io/v1alpha1" for User and Group subjects.

Returns The api_version of this V1alpha1Subject.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'name': 'name', 'name

kind

Gets the kind of this V1alpha1Subject. Kind of object being referenced. Values defined by this API group are "User", "Group", and "ServiceAccount". If the Authorizer does not recognized the kind value, the Authorizer should report an error.

Returns The kind of this V1alpha1Subject.

Return type str

name

Gets the name of this V1alpha1Subject. Name of the object being referenced.

Returns The name of this V1alpha1Subject.

Return type str

namespace

Gets the namespace of this V1alpha1Subject. Namespace of the referenced object. If the object kind is non-namespace, such as "User" or "Group", and this value is not empty the Authorizer should report an error.

Returns The namespace of this V1alpha1Subject.

Return type str

swagger_types = {'api_version': 'str', 'kind': 'str', 'name': 'str', 'namespace':

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1beta1_api_version module

kubernetes.client.models.v1beta1_cpu_target_utilization module

kubernetes.client.models.v1beta1_daemon_set module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_daemon_set.**V1beta1DaemonSet**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1beta1DaemonSet. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1DaemonSet.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind

Gets the kind of this V1beta1DaemonSet. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1DaemonSet.

Return type str

metadata

Gets the metadata of this V1beta1DaemonSet. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1DaemonSet.

Return type *V1ObjectMeta*

spec

Gets the spec of this V1beta1DaemonSet. The desired behavior of this daemon set. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1beta1DaemonSet.

Return type *V1beta1DaemonSetSpec*

status

Gets the status of this V1beta1DaemonSet. The current status of this daemon set. This data may be out of date by some window of time. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1beta1DaemonSet.

Return type *V1beta1DaemonSetStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_daemon_set_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_daemon_set_list.**V1beta1DaemonSetList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1DaemonSetList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1DaemonSetList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1beta1DaemonSetList. A list of daemon sets.

Returns The items of this V1beta1DaemonSetList.

Return type list[*V1beta1DaemonSet*]

kind
Gets the kind of this V1beta1DaemonSetList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1DaemonSetList.

Return type str

metadata
Gets the metadata of this V1beta1DaemonSetList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1DaemonSetList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1beta1DaemonSet]', 'kind': '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_daemon_set_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_daemon_set_spec.**V1beta1DaemonSetSpec**(*min_ready_seconds=None*, *revision_history_limit=None*, *selector=None*, *template=None*, *template_generation=None*, *update_strategy=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'min_ready_seconds': 'minReadySeconds', 'revision_history_limit': '

min_ready_seconds
Gets the min_ready_seconds of this V1beta1DaemonSetSpec. The minimum number of seconds for which a newly created DaemonSet pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready).

Returns The min_ready_seconds of this V1beta1DaemonSetSpec.

Return type int

revision_history_limit
Gets the revision_history_limit of this V1beta1DaemonSetSpec. The number of old history to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 10.

Returns The revision_history_limit of this V1beta1DaemonSetSpec.

Return type int

selector
Gets the selector of this V1beta1DaemonSetSpec. A label query over pods that are managed by the daemon set. Must match in order to be controlled. If empty, defaulted to labels on Pod template. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors

Returns The selector of this V1beta1DaemonSetSpec.

Return type V1LabelSelector

swagger_types = {'min_ready_seconds': 'int', 'revision_history_limit': 'int', 'selec

template
Gets the template of this V1beta1DaemonSetSpec. An object that describes the pod that will be created. The DaemonSet will create exactly one copy of this pod on every node that matches the template's node selector (or on every node if no node selector is specified). More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#pod-template

Returns The template of this V1beta1DaemonSetSpec.

Return type *V1PodTemplateSpec*

template_generation
Gets the template_generation of this V1beta1DaemonSetSpec. DEPRECATED. A sequence number representing a specific generation of the template. Populated by the system. It can be set only during the creation.

Returns The template_generation of this V1beta1DaemonSetSpec.

Return type int

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

update_strategy
Gets the update_strategy of this V1beta1DaemonSetSpec. An update strategy to replace existing DaemonSet pods with new pods.

Returns The update_strategy of this V1beta1DaemonSetSpec.

Return type V1beta1DaemonSetUpdateStrategy

kubernetes.client.models.v1beta1_daemon_set_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_daemon_set_status.**V1beta1DaemonSetStatus**(*collision_count*, *conditions=None*, *current_number_scheduled*, *desired_number_scheduled*, *number_available=None*, *number_misscheduled=None*, *number_ready=None*, *number_unavailable=None*, *observed_generation=None*, *updated_number_scheduled=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'collision_count': 'collisionCount', 'conditions': 'conditions', 'c

collision_count

Gets the collision_count of this V1beta1DaemonSetStatus. Count of hash collisions for the DaemonSet. The DaemonSet controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ControllerRevision.

Returns The collision_count of this V1beta1DaemonSetStatus.

Return type int

conditions

Gets the conditions of this V1beta1DaemonSetStatus. Represents the latest available observations of a DaemonSet's current state.

Returns The conditions of this V1beta1DaemonSetStatus.

Return type list[V1beta1DaemonSetCondition]

current_number_scheduled

Gets the current_number_scheduled of this V1beta1DaemonSetStatus. The number of nodes that are running at least 1 daemon pod and are supposed to run the daemon pod. More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/

Returns The current_number_scheduled of this V1beta1DaemonSetStatus.

Return type int

desired_number_scheduled

Gets the desired_number_scheduled of this V1beta1DaemonSetStatus. The total number of nodes that should be running the daemon pod (including nodes correctly running the daemon pod). More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/

Returns The desired_number_scheduled of this V1beta1DaemonSetStatus.

Return type int

number_available
Gets the number_available of this V1beta1DaemonSetStatus. The number of nodes that should be running the daemon pod and have one or more of the daemon pod running and available (ready for at least spec.minReadySeconds)

Returns The number_available of this V1beta1DaemonSetStatus.

Return type int

number_misscheduled
Gets the number_misscheduled of this V1beta1DaemonSetStatus. The number of nodes that are running the daemon pod, but are not supposed to run the daemon pod. More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/

Returns The number_misscheduled of this V1beta1DaemonSetStatus.

Return type int

number_ready
Gets the number_ready of this V1beta1DaemonSetStatus. The number of nodes that should be running the daemon pod and have one or more of the daemon pod running and ready.

Returns The number_ready of this V1beta1DaemonSetStatus.

Return type int

number_unavailable
Gets the number_unavailable of this V1beta1DaemonSetStatus. The number of nodes that should be running the daemon pod and have none of the daemon pod running and available (ready for at least spec.minReadySeconds)

Returns The number_unavailable of this V1beta1DaemonSetStatus.

Return type int

observed_generation
Gets the observed_generation of this V1beta1DaemonSetStatus. The most recent generation observed by the daemon set controller.

Returns The observed_generation of this V1beta1DaemonSetStatus.

Return type int

swagger_types = {'collision_count': 'int', 'conditions': 'list[V1beta1DaemonSetCondi

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

updated_number_scheduled
Gets the updated_number_scheduled of this V1beta1DaemonSetStatus. The total number of nodes that are running updated daemon pod

Returns The updated_number_scheduled of this V1beta1DaemonSetStatus.

Return type int

kubernetes.client.models.v1beta1_deployment module

kubernetes.client.models.v1beta1_deployment_condition module

kubernetes.client.models.v1beta1_deployment_list module

kubernetes.client.models.v1beta1_deployment_rollback module

kubernetes.client.models.v1beta1_deployment_spec module

kubernetes.client.models.v1beta1_deployment_status module

kubernetes.client.models.v1beta1_deployment_strategy module

kubernetes.client.models.v1beta1_eviction module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_eviction.**V1beta1Eviction**(*api_version=None*, *delete_options=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V1beta1Eviction. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1Eviction.

Return type str

attribute_map = {'api_version': 'apiVersion', 'delete_options': 'deleteOptions', 'ki

delete_options

Gets the delete_options of this V1beta1Eviction. DeleteOptions may be provided

Returns The delete_options of this V1beta1Eviction.

Return type *V1DeleteOptions*

kind

Gets the kind of this V1beta1Eviction. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1Eviction.

Return type str

metadata
Gets the metadata of this V1beta1Eviction. ObjectMeta describes the pod that is being evicted.

Returns The metadata of this V1beta1Eviction.

Return type *V1ObjectMeta*

swagger_types = {'api_version': 'str', 'delete_options': 'V1DeleteOptions', 'kind':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_horizontal_pod_autoscaler module

kubernetes.client.models.v1beta1_horizontal_pod_autoscaler_list module

kubernetes.client.models.v1beta1_horizontal_pod_autoscaler_spec module

kubernetes.client.models.v1beta1_horizontal_pod_autoscaler_status module

kubernetes.client.models.v1beta1_http_ingress_path module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_http_ingress_path.**V1beta1HTTPIngressPath**(*backend=None*, *path=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'backend': 'backend', 'path': 'path'}

backend
Gets the backend of this V1beta1HTTPIngressPath. Backend defines the referenced service endpoint to which the traffic will be forwarded to.

Returns The backend of this V1beta1HTTPIngressPath.

Return type *V1beta1IngressBackend*

path
Gets the path of this V1beta1HTTPIngressPath. Path is an extended POSIX regex as defined by IEEE Std 1003.1, (i.e this follows the egrep/unix syntax, not the perl syntax) matched against the path of an incoming request. Currently it can contain characters disallowed from the conventional "path" part of a URL as defined by RFC 3986. Paths must begin with a '/'. If unspecified, the path defaults to a catch all sending traffic to the backend.

Returns The path of this V1beta1HTTPIngressPath.

Return type str

swagger_types = {'backend': 'V1beta1IngressBackend', 'path': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_http_ingress_rule_value module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_http_ingress_rule_value.**V1beta1HTTPIngressRuleValue**
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'paths': 'paths'}

paths
Gets the paths of this V1beta1HTTPIngressRuleValue. A collection of paths that map requests to backends.

Returns The paths of this V1beta1HTTPIngressRuleValue.

Return type list[*V1beta1HTTPIngressPath*]

swagger_types = {'paths': 'list[V1beta1HTTPIngressPath]'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_ingress module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_ingress.**V1beta1Ingress**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1Ingress. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may

reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1Ingress.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind

Gets the kind of this V1beta1Ingress. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1Ingress.

Return type str

metadata

Gets the metadata of this V1beta1Ingress. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1Ingress.

Return type *V1ObjectMeta*

spec

Gets the spec of this V1beta1Ingress. Spec is the desired state of the Ingress. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1beta1Ingress.

Return type *V1beta1IngressSpec*

status

Gets the status of this V1beta1Ingress. Status is the current state of the Ingress. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1beta1Ingress.

Return type *V1beta1IngressStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1beta1_ingress_backend module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_ingress_backend.**V1beta1IngressBackend**(*service_name=None*, *service_port=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'service_name': 'serviceName', 'service_port': 'servicePort'}

service_name
Gets the service_name of this V1beta1IngressBackend. Specifies the name of the referenced service.

Returns The service_name of this V1beta1IngressBackend.

Return type str

service_port
Gets the service_port of this V1beta1IngressBackend. Specifies the port of the referenced service.

Returns The service_port of this V1beta1IngressBackend.

Return type object

swagger_types = {'service_name': 'str', 'service_port': 'object'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_ingress_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_ingress_list.**V1beta1IngressList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1IngressList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1IngressList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1beta1IngressList. Items is the list of Ingress.

Returns The items of this V1beta1IngressList.

Return type list[*V1beta1Ingress*]

kind
Gets the kind of this V1beta1IngressList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1IngressList.

Return type str

metadata
Gets the metadata of this V1beta1IngressList. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1IngressList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1beta1Ingress]', 'kind': 'st

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_ingress_rule module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_ingress_rule.**V1beta1IngressRule**(*host=None*, *http=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'host': 'host', 'http': 'http'}

host
Gets the host of this V1beta1IngressRule. Host is the fully qualified domain name of a network host, as defined by RFC 3986. Note the following deviations from the "host" part of the URI as defined in the RFC: 1. IPs are not allowed. Currently an IngressRuleValue can only apply to the IP in the Spec of the parent Ingress. 2. The *:* delimiter is not respected because ports are not allowed. Currently the port of an Ingress is implicitly :80 for http and :443 for https. Both these may change in the future. Incoming requests are matched against the host before the IngressRuleValue. If the host is unspecified, the Ingress routes all traffic based on the specified IngressRuleValue.

Returns The host of this V1beta1IngressRule.

Return type str

http
Gets the http of this V1beta1IngressRule.

Returns The http of this V1beta1IngressRule.

Return type *V1beta1HTTPIngressRuleValue*

swagger_types = {'host': 'str', 'http': 'V1beta1HTTPIngressRuleValue'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_ingress_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_ingress_spec.**V1beta1IngressSpec**(*backend=None*, *rules=None*, *tls=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'backend': 'backend', 'rules': 'rules', 'tls': 'tls'}

backend
Gets the backend of this V1beta1IngressSpec. A default backend capable of servicing requests that don't match any rule. At least one of 'backend' or 'rules' must be specified. This field is optional to allow the loadbalancer controller or defaulting logic to specify a global default.

Returns The backend of this V1beta1IngressSpec.

Return type *V1beta1IngressBackend*

rules
Gets the rules of this V1beta1IngressSpec. A list of host rules used to configure the Ingress. If unspecified, or no rule matches, all traffic is sent to the default backend.

Returns The rules of this V1beta1IngressSpec.

Return type list[*V1beta1IngressRule*]

swagger_types = {'backend': 'V1beta1IngressBackend', 'rules': 'list[V1beta1IngressRu

tls
Gets the tls of this V1beta1IngressSpec. TLS configuration. Currently the Ingress only supports a single TLS port, 443. If multiple members of this list specify different hosts, they will be multiplexed on the same port according to the hostname specified through the SNI TLS extension, if the ingress controller fulfilling the ingress supports SNI.

Returns The tls of this V1beta1IngressSpec.

Return type list[*V1beta1IngressTLS*]

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_ingress_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_ingress_status.**V1beta1IngressStatus**(*load_balancer=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'load_balancer': 'loadBalancer'}

load_balancer

Gets the load_balancer of this V1beta1IngressStatus. LoadBalancer contains the current status of the load-balancer.

Returns The load_balancer of this V1beta1IngressStatus.

Return type *V1LoadBalancerStatus*

swagger_types = {'load_balancer': 'V1LoadBalancerStatus'}

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v1beta1_ingress_tls module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_ingress_tls.**V1beta1IngressTLS**(*hosts=None*, *secret_name=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'hosts': 'hosts', 'secret_name': 'secretName'}

hosts

Gets the hosts of this V1beta1IngressTLS. Hosts are a list of hosts included in the TLS certificate. The values in this list must match the name/s used in the tlsSecret. Defaults to the wildcard host setting for the loadbalancer controller fulfilling this Ingress, if left unspecified.

Returns The hosts of this V1beta1IngressTLS.

Return type list[str]

secret_name

Gets the secret_name of this V1beta1IngressTLS. SecretName is the name of the secret used to terminate SSL traffic on 443. Field is left optional to allow SSL routing based on SNI hostname alone. If the SNI

host in a listener conflicts with the "Host" header field used by an IngressRule, the SNI host is used for termination and value of the Host header is used for routing.

Returns The secret_name of this V1beta1IngressTLS.

Return type str

swagger_types = {'hosts': 'list[str]', 'secret_name': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_job module

kubernetes.client.models.v1beta1_job_condition module

kubernetes.client.models.v1beta1_job_list module

kubernetes.client.models.v1beta1_job_spec module

kubernetes.client.models.v1beta1_job_status module

kubernetes.client.models.v1beta1_local_subject_access_review module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_local_subject_access_review.**V1beta1LocalSubjectAcces**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1LocalSubjectAccessReview. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1LocalSubjectAccessReview.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1beta1LocalSubjectAccessReview. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1LocalSubjectAccessReview.

Return type str

metadata
Gets the metadata of this V1beta1LocalSubjectAccessReview.

Returns The metadata of this V1beta1LocalSubjectAccessReview.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1beta1LocalSubjectAccessReview. Spec holds information about the request being evaluated. spec.namespace must be equal to the namespace you made the request against. If empty, it is defaulted.

Returns The spec of this V1beta1LocalSubjectAccessReview.

Return type *V1beta1SubjectAccessReviewSpec*

status
Gets the status of this V1beta1LocalSubjectAccessReview. Status is filled in by the server and indicates whether the request is allowed or not

Returns The status of this V1beta1LocalSubjectAccessReview.

Return type *V1beta1SubjectAccessReviewStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_network_policy module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_network_policy.**V1beta1NetworkPolicy**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1NetworkPolicy. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1NetworkPolicy.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1beta1NetworkPolicy. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1NetworkPolicy.

Return type str

metadata
Gets the metadata of this V1beta1NetworkPolicy. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1NetworkPolicy.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1beta1NetworkPolicy. Specification of the desired behavior for this NetworkPolicy.

Returns The spec of this V1beta1NetworkPolicy.

Return type *V1beta1NetworkPolicySpec*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_network_policy_ingress_rule module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_network_policy_ingress_rule.**V1beta1NetworkPolicyIngr**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'_from': 'from', 'ports': 'ports'}

ports
Gets the ports of this V1beta1NetworkPolicyIngressRule. List of ports which should be made accessible on the pods selected for this rule. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.

Returns The ports of this V1beta1NetworkPolicyIngressRule.

Return type list[*V1beta1NetworkPolicyPort*]

swagger_types = {'_from': 'list[V1beta1NetworkPolicyPeer]', 'ports': 'list[V1beta1Ne

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_network_policy_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_network_policy_list.**V1beta1NetworkPolicyList**(*api_versi items=No kind=Nor meta- data=Nor*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1NetworkPolicyList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1NetworkPolicyList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1beta1NetworkPolicyList. Items is a list of schema objects.

Returns The items of this V1beta1NetworkPolicyList.

Return type list[*V1beta1NetworkPolicy*]

kind
Gets the kind of this V1beta1NetworkPolicyList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1NetworkPolicyList.

Return type str

metadata
Gets the metadata of this V1beta1NetworkPolicyList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1NetworkPolicyList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1beta1NetworkPolicy]', 'kind'

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_network_policy_peer module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_network_policy_peer.**V1beta1NetworkPolicyPeer**(*ip_block=*
names-
pace_sele
pod_selec

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'ip_block': 'ipBlock', 'namespace_selector': 'namespaceSelector', '

ip_block
Gets the ip_block of this V1beta1NetworkPolicyPeer. IPBlock defines policy on a particular IPBlock

Returns The ip_block of this V1beta1NetworkPolicyPeer.

Return type V1beta1IPBlock

namespace_selector
Gets the namespace_selector of this V1beta1NetworkPolicyPeer. Selects Namespaces using cluster scoped-labels. This matches all pods in all namespaces selected by this label selector. This field follows standard label selector semantics. If present but empty, this selector selects all namespaces.

Returns The namespace_selector of this V1beta1NetworkPolicyPeer.

Return type V1LabelSelector

pod_selector
Gets the pod_selector of this V1beta1NetworkPolicyPeer. This is a label selector which selects Pods in this namespace. This field follows standard label selector semantics. If present but empty, this selector selects all pods in this namespace.

Returns The pod_selector of this V1beta1NetworkPolicyPeer.

Return type V1LabelSelector

swagger_types = {'ip_block': 'V1beta1IPBlock', 'namespace_selector': 'V1LabelSelecto

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_network_policy_port module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_network_policy_port.**V1beta1NetworkPolicyPort**(*port=None*, *protocol=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'port': 'port', 'protocol': 'protocol'}

port
Gets the port of this V1beta1NetworkPolicyPort. If specified, the port on the given protocol. This can either be a numerical or named port on a pod. If this field is not provided, this matches all port names and numbers. If present, only traffic on the specified protocol AND port will be matched.

Returns The port of this V1beta1NetworkPolicyPort.

Return type object

protocol
Gets the protocol of this V1beta1NetworkPolicyPort. Optional. The protocol (TCP or UDP) which traffic must match. If not specified, this field defaults to TCP.

Returns The protocol of this V1beta1NetworkPolicyPort.

Return type str

swagger_types = {'port': 'object', 'protocol': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_network_policy_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_network_policy_spec.**V1beta1NetworkPolicySpec**(*egress=N ingress=I pod_selec pol- icy_types*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'egress': 'egress', 'ingress': 'ingress', 'pod_selector': 'podSele

egress

Gets the egress of this V1beta1NetworkPolicySpec. List of egress rules to be applied to the selected pods. Outgoing traffic is allowed if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic matches at least one egress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy limits all outgoing traffic (and serves solely to ensure that the pods it selects are isolated by default). This field is beta-level in 1.8

Returns The egress of this V1beta1NetworkPolicySpec.

Return type list[V1beta1NetworkPolicyEgressRule]

ingress

Gets the ingress of this V1beta1NetworkPolicySpec. List of ingress rules to be applied to the selected pods. Traffic is allowed to a pod if there are no NetworkPolicies selecting the pod OR if the traffic source is the pod's local node, OR if the traffic matches at least one ingress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy does not allow any traffic (and serves solely to ensure that the pods it selects are isolated by default).

Returns The ingress of this V1beta1NetworkPolicySpec.

Return type list[*V1beta1NetworkPolicyIngressRule*]

pod_selector

Gets the pod_selector of this V1beta1NetworkPolicySpec. Selects the pods to which this NetworkPolicy object applies. The array of ingress rules is applied to any pods selected by this field. Multiple network policies can select the same set of pods. In this case, the ingress rules for each are combined additively. This field is NOT optional and follows standard label selector semantics. An empty podSelector matches all pods in this namespace.

Returns The pod_selector of this V1beta1NetworkPolicySpec.

Return type V1LabelSelector

policy_types

Gets the policy_types of this V1beta1NetworkPolicySpec. List of rule types that the NetworkPolicy relates to. Valid options are Ingress, Egress, or Ingress,Egress. If this field is not specified, it will default based on the existence of Ingress or Egress rules; policies that contain an Egress section are assumed to affect Egress, and all policies (whether or not they contain an Ingress section) are assumed to affect Ingress. If you want to write an egress-only policy, you must explicitly specify policyTypes ["Egress"]. Likewise, if you want to write a policy that specifies that no egress is allowed, you must specify a policyTypes value that include "Egress" (since such a policy would not include an Egress section and would otherwise default to just ["Ingress"]). This field is beta-level in 1.8

Returns The policy_types of this V1beta1NetworkPolicySpec.

Return type list[str]

swagger_types = {'egress': 'list[V1beta1NetworkPolicyEgressRule]', 'ingress': 'list[

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_non_resource_attributes module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_non_resource_attributes.**V1beta1NonResourceAttributes**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'path': 'path', 'verb': 'verb'}

path
Gets the path of this V1beta1NonResourceAttributes. Path is the URL path of the request

Returns The path of this V1beta1NonResourceAttributes.

Return type str

swagger_types = {'path': 'str', 'verb': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

verb
Gets the verb of this V1beta1NonResourceAttributes. Verb is the standard HTTP verb

Returns The verb of this V1beta1NonResourceAttributes.

Return type str

kubernetes.client.models.v1beta1_pod_disruption_budget module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_pod_disruption_budget.**V1beta1PodDisruptionBudget**(*api_*
kind
met
dat
spe
sta-
tus=

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1PodDisruptionBudget. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1PodDisruptionBudget.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1beta1PodDisruptionBudget. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1PodDisruptionBudget.

Return type str

metadata
Gets the metadata of this V1beta1PodDisruptionBudget.

Returns The metadata of this V1beta1PodDisruptionBudget.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1beta1PodDisruptionBudget. Specification of the desired behavior of the PodDisruptionBudget.

Returns The spec of this V1beta1PodDisruptionBudget.

Return type *V1beta1PodDisruptionBudgetSpec*

status
Gets the status of this V1beta1PodDisruptionBudget. Most recently observed status of the PodDisruptionBudget.

Returns The status of this V1beta1PodDisruptionBudget.

Return type *V1beta1PodDisruptionBudgetStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_pod_disruption_budget_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_pod_disruption_budget_list.**V1beta1PodDisruptionBudge**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1PodDisruptionBudgetList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1PodDisruptionBudgetList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1beta1PodDisruptionBudgetList.

Returns The items of this V1beta1PodDisruptionBudgetList.

Return type list[*V1beta1PodDisruptionBudget*]

kind
Gets the kind of this V1beta1PodDisruptionBudgetList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1PodDisruptionBudgetList.

Return type str

metadata
Gets the metadata of this V1beta1PodDisruptionBudgetList.

Returns The metadata of this V1beta1PodDisruptionBudgetList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1beta1PodDisruptionBudget]',

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_pod_disruption_budget_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_pod_disruption_budget_spec.**V1beta1PodDisruptionBudge**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'max_unavailable': 'maxUnavailable', 'min_available': 'minAvailable

max_unavailable
Gets the max_unavailable of this V1beta1PodDisruptionBudgetSpec. An eviction is allowed if at most "maxUnavailable" pods selected by "selector" are unavailable after the eviction, i.e. even in absence of the evicted pod. For example, one can prevent all voluntary evictions by specifying 0. This is a mutually exclusive setting with "minAvailable".

Returns The max_unavailable of this V1beta1PodDisruptionBudgetSpec.

Return type object

min_available
Gets the min_available of this V1beta1PodDisruptionBudgetSpec. An eviction is allowed if at least "minAvailable" pods selected by "selector" will still be available after the eviction, i.e. even in the absence of the evicted pod. So for example you can prevent all voluntary evictions by specifying "100%".

Returns The min_available of this V1beta1PodDisruptionBudgetSpec.

Return type object

selector
Gets the selector of this V1beta1PodDisruptionBudgetSpec. Label query over pods whose evictions are managed by the disruption budget.

Returns The selector of this V1beta1PodDisruptionBudgetSpec.

Return type V1LabelSelector

swagger_types = {'max_unavailable': 'object', 'min_available': 'object', 'selector':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_pod_disruption_budget_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_pod_disruption_budget_status.**V1beta1PodDisruptionBu**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'current_healthy': 'currentHealthy', 'desired_healthy': 'desiredHea

current_healthy
Gets the current_healthy of this V1beta1PodDisruptionBudgetStatus. current number of healthy pods

Returns The current_healthy of this V1beta1PodDisruptionBudgetStatus.

Return type int

desired_healthy
Gets the desired_healthy of this V1beta1PodDisruptionBudgetStatus. minimum desired number of healthy pods

Returns The desired_healthy of this V1beta1PodDisruptionBudgetStatus.

Return type int

disrupted_pods
Gets the disrupted_pods of this V1beta1PodDisruptionBudgetStatus. DisruptedPods contains information about pods whose eviction was processed by the API server eviction subresource handler but has not yet been observed by the PodDisruptionBudget controller. A pod will be in this map from the time when the API server processed the eviction request to the time when the pod is seen by PDB controller as having been marked for deletion (or after a timeout). The key in the map is the name of the pod and the value is the time when the API server processed the eviction request. If the deletion didn't occur and a pod is still there it will be removed from the list automatically by PodDisruptionBudget controller after some time. If everything goes smooth this map should be empty for the most of the time. Large number of entries in the map may indicate problems with pod deletions.

Returns The disrupted_pods of this V1beta1PodDisruptionBudgetStatus.

Return type dict(str, datetime)

disruptions_allowed
Gets the disruptions_allowed of this V1beta1PodDisruptionBudgetStatus. Number of pod disruptions that are currently allowed.

Returns The disruptions_allowed of this V1beta1PodDisruptionBudgetStatus.

Return type int

expected_pods
Gets the expected_pods of this V1beta1PodDisruptionBudgetStatus. total number of pods counted by this disruption budget

Returns The expected_pods of this V1beta1PodDisruptionBudgetStatus.

Return type int

observed_generation
Gets the observed_generation of this V1beta1PodDisruptionBudgetStatus. Most recent generation observed when updating this PDB status. PodDisruptionsAllowed and other status informatio is valid only if observedGeneration equals to PDB's object generation.

Returns The observed_generation of this V1beta1PodDisruptionBudgetStatus.

Return type int

swagger_types = {'current_healthy': 'int', 'desired_healthy': 'int', 'disrupted_pods

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_replica_set module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_replica_set.**V1beta1ReplicaSet**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1ReplicaSet. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1ReplicaSet.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1beta1ReplicaSet. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1ReplicaSet.

Return type str

metadata
Gets the metadata of this V1beta1ReplicaSet. If the Labels of a ReplicaSet are empty, they are defaulted

to be the same as the Pod(s) that the ReplicaSet manages. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1ReplicaSet.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1beta1ReplicaSet. Spec defines the specification of the desired behavior of the ReplicaSet. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V1beta1ReplicaSet.

Return type *V1beta1ReplicaSetSpec*

status
Gets the status of this V1beta1ReplicaSet. Status is the most recently observed status of the ReplicaSet. This data may be out of date by some window of time. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V1beta1ReplicaSet.

Return type *V1beta1ReplicaSetStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_replica_set_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_replica_set_condition.**V1beta1ReplicaSetCondition**(*last mes sag rea son sta- tus= type*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'last_transition_time': 'lastTransitionTime', 'message': 'message',

last_transition_time
Gets the last_transition_time of this V1beta1ReplicaSetCondition. The last time the condition transitioned from one status to another.

Returns The last_transition_time of this V1beta1ReplicaSetCondition.

Return type datetime

message
Gets the message of this V1beta1ReplicaSetCondition. A human readable message indicating details about the transition.

Returns The message of this V1beta1ReplicaSetCondition.

Return type str

reason
Gets the reason of this V1beta1ReplicaSetCondition. The reason for the condition's last transition.

Returns The reason of this V1beta1ReplicaSetCondition.

Return type str

status
Gets the status of this V1beta1ReplicaSetCondition. Status of the condition, one of True, False, Unknown.

Returns The status of this V1beta1ReplicaSetCondition.

Return type str

swagger_types = {'last_transition_time': 'datetime', 'message': 'str', 'reason': 'st

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

type
Gets the type of this V1beta1ReplicaSetCondition. Type of replica set condition.

Returns The type of this V1beta1ReplicaSetCondition.

Return type str

kubernetes.client.models.v1beta1_replica_set_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_replica_set_list.**V1beta1ReplicaSetList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1ReplicaSetList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1ReplicaSetList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1beta1ReplicaSetList. List of ReplicaSets. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller

Returns The items of this V1beta1ReplicaSetList.

Return type list[*V1beta1ReplicaSet*]

kind
Gets the kind of this V1beta1ReplicaSetList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1ReplicaSetList.

Return type str

metadata
Gets the metadata of this V1beta1ReplicaSetList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The metadata of this V1beta1ReplicaSetList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1beta1ReplicaSet]', 'kind':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_replica_set_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_replica_set_spec.**V1beta1ReplicaSetSpec**(*min_ready_second*, *replicas=None*, *selector=None*, *template=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'min_ready_seconds': 'minReadySeconds', 'replicas': 'replicas', 'se

min_ready_seconds
Gets the min_ready_seconds of this V1beta1ReplicaSetSpec. Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)

Returns The min_ready_seconds of this V1beta1ReplicaSetSpec.

Return type int

replicas
Gets the replicas of this V1beta1ReplicaSetSpec. Replicas is the number of desired replicas. This is a pointer to distinguish between explicit zero and unspecified. Defaults to 1. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller/#what-is-a-replicationcontroller

Returns The replicas of this V1beta1ReplicaSetSpec.

Return type int

selector
Gets the selector of this V1beta1ReplicaSetSpec. Selector is a label query over pods that should match the replica count. If the selector is empty, it is defaulted to the labels present on the pod template. Label keys and values that must match in order to be controlled by this replica set. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors

Returns The selector of this V1beta1ReplicaSetSpec.

Return type V1LabelSelector

swagger_types = {'min_ready_seconds': 'int', 'replicas': 'int', 'selector': 'V1Labe

template
Gets the template of this V1beta1ReplicaSetSpec. Template is the object that describes the pod that will be created if insufficient replicas are detected. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#pod-template

Returns The template of this V1beta1ReplicaSetSpec.

Return type *V1PodTemplateSpec*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_replica_set_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_replica_set_status.**V1beta1ReplicaSetStatus**(*available_re*
con-
di-
tions=None,
fully_labeled
ob-
served_gene
ready_replic
repli-
cas=None)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'available_replicas': 'availableReplicas', 'conditions': 'condition

available_replicas
Gets the available_replicas of this V1beta1ReplicaSetStatus. The number of available replicas (ready for at least minReadySeconds) for this replica set.

Returns The available_replicas of this V1beta1ReplicaSetStatus.

Return type int

conditions
Gets the conditions of this V1beta1ReplicaSetStatus. Represents the latest available observations of a replica set's current state.

Returns The conditions of this V1beta1ReplicaSetStatus.

Return type list[*V1beta1ReplicaSetCondition*]

fully_labeled_replicas
Gets the fully_labeled_replicas of this V1beta1ReplicaSetStatus. The number of pods that have labels matching the labels of the pod template of the replicaset.

Returns The fully_labeled_replicas of this V1beta1ReplicaSetStatus.

Return type int

observed_generation
Gets the observed_generation of this V1beta1ReplicaSetStatus. ObservedGeneration reflects the generation of the most recently observed ReplicaSet.

Returns The observed_generation of this V1beta1ReplicaSetStatus.

Return type int

ready_replicas
Gets the ready_replicas of this V1beta1ReplicaSetStatus. The number of ready replicas for this replica set.

Returns The ready_replicas of this V1beta1ReplicaSetStatus.

Return type int

replicas
Gets the replicas of this V1beta1ReplicaSetStatus. Replicas is the most recently oberved number of replicas. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller/#what-is-a-replicationcontroller

Returns The replicas of this V1beta1ReplicaSetStatus.

Return type int

swagger_types = {'available_replicas': 'int', 'conditions': 'list[V1beta1ReplicaSetC

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_resource_attributes module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_resource_attributes.**V1beta1ResourceAttributes**(*group=N name=N namespace=N resource=N subresource=N verb=N version=N*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'group': 'group', 'name': 'name', 'namespace': 'namespace', 'resou

group
Gets the group of this V1beta1ResourceAttributes. Group is the API Group of the Resource. "*" means all.

Returns The group of this V1beta1ResourceAttributes.

Return type str

name
Gets the name of this V1beta1ResourceAttributes. Name is the name of the resource being requested for a "get" or deleted for a "delete". "" (empty) means all.

Returns The name of this V1beta1ResourceAttributes.

Return type str

namespace
Gets the namespace of this V1beta1ResourceAttributes. Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces "" (empty) is defaulted for LocalSubjectAccessReviews "" (empty) is empty for cluster-scoped resources "" (empty) means "all" for namespace scoped resources from a SubjectAccessReview or SelfSubjectAccessReview

Returns The namespace of this V1beta1ResourceAttributes.

Return type str

resource
Gets the resource of this V1beta1ResourceAttributes. Resource is one of the existing resource types. "*" means all.

Returns The resource of this V1beta1ResourceAttributes.

Return type str

subresource
Gets the subresource of this V1beta1ResourceAttributes. Subresource is one of the existing resource types. "" means none.

Returns The subresource of this V1beta1ResourceAttributes.

Return type str

swagger_types = {'group': 'str', 'name': 'str', 'namespace': 'str', 'resource': 's

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

verb
Gets the verb of this V1beta1ResourceAttributes. Verb is a kubernetes resource API verb, like: get, list, watch, create, update, delete, proxy. "*" means all.

Returns The verb of this V1beta1ResourceAttributes.

Return type str

version
Gets the version of this V1beta1ResourceAttributes. Version is the API Version of the Resource. "*" means all.

Returns The version of this V1beta1ResourceAttributes.

Return type str

kubernetes.client.models.v1beta1_rollback_config module

kubernetes.client.models.v1beta1_rolling_update_deployment module

kubernetes.client.models.v1beta1_scale module

kubernetes.client.models.v1beta1_scale_spec module

kubernetes.client.models.v1beta1_scale_status module

kubernetes.client.models.v1beta1_self_subject_access_review module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_self_subject_access_review.**V1beta1SelfSubjectAccessR**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1SelfSubjectAccessReview. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1SelfSubjectAccessReview.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1beta1SelfSubjectAccessReview. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1SelfSubjectAccessReview.

Return type str

metadata
Gets the metadata of this V1beta1SelfSubjectAccessReview.

Returns The metadata of this V1beta1SelfSubjectAccessReview.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1beta1SelfSubjectAccessReview. Spec holds information about the request being evaluated. user and groups must be empty

Returns The spec of this V1beta1SelfSubjectAccessReview.

Return type *V1beta1SelfSubjectAccessReviewSpec*

status
Gets the status of this V1beta1SelfSubjectAccessReview. Status is filled in by the server and indicates whether the request is allowed or not

Returns The status of this V1beta1SelfSubjectAccessReview.

Return type *V1beta1SubjectAccessReviewStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_self_subject_access_review_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_self_subject_access_review_spec.**V1beta1SelfSubjectAc**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'non_resource_attributes': 'nonResourceAttributes', 'resource_attrib

non_resource_attributes
Gets the non_resource_attributes of this V1beta1SelfSubjectAccessReviewSpec. NonResourceAttributes describes information for a non-resource access request

Returns The non_resource_attributes of this V1beta1SelfSubjectAccessReviewSpec.

Return type *V1beta1NonResourceAttributes*

resource_attributes
Gets the resource_attributes of this V1beta1SelfSubjectAccessReviewSpec. ResourceAuthorizationAttributes describes information for a resource access request

Returns The resource_attributes of this V1beta1SelfSubjectAccessReviewSpec.

Return type *V1beta1ResourceAttributes*

swagger_types = {'non_resource_attributes': 'V1beta1NonResourceAttributes', 'resource_

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_stateful_set module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_stateful_set.**V1beta1StatefulSet**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1StatefulSet. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1StatefulSet.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1beta1StatefulSet. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1StatefulSet.

Return type str

metadata
Gets the metadata of this V1beta1StatefulSet.

Returns The metadata of this V1beta1StatefulSet.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1beta1StatefulSet. Spec defines the desired identities of pods in this set.

Returns The spec of this V1beta1StatefulSet.

Return type *V1beta1StatefulSetSpec*

status
Gets the status of this V1beta1StatefulSet. Status is the current status of Pods in this StatefulSet. This data may be out of date by some window of time.

Returns The status of this V1beta1StatefulSet.

Return type *V1beta1StatefulSetStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_stateful_set_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_stateful_set_list.**V1beta1StatefulSetList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1StatefulSetList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1StatefulSetList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1beta1StatefulSetList.

Returns The items of this V1beta1StatefulSetList.

Return type list[*V1beta1StatefulSet*]

kind
Gets the kind of this V1beta1StatefulSetList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1StatefulSetList.

Return type str

metadata
Gets the metadata of this V1beta1StatefulSetList.

Returns The metadata of this V1beta1StatefulSetList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1beta1StatefulSet]', 'kind':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_stateful_set_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_stateful_set_spec.**V1beta1StatefulSetSpec**(*pod_manageme repli- cas=None, re- vi- sion_history_li se- lec- tor=None, ser- vice_name=No tem- plate=None, up- date_strategy= vol- ume_claim_tem*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'pod_management_policy': 'podManagementPolicy', 'replicas': 'replic

pod_management_policy

Gets the pod_management_policy of this V1beta1StatefulSetSpec. podManagementPolicy controls how pods are created during initial scale up, when replacing pods on nodes, or when scaling down. The default policy is *OrderedReady*, where pods are created in increasing order (pod-0, then pod-1, etc) and the controller will wait until each pod is ready before continuing. When scaling down, the pods are removed in the opposite order. The alternative policy is *Parallel* which will create pods in parallel to match the desired scale without waiting, and on scale down will delete all pods at once.

Returns The pod_management_policy of this V1beta1StatefulSetSpec.

Return type str

replicas

Gets the replicas of this V1beta1StatefulSetSpec. replicas is the desired number of replicas of the given Template. These are replicas in the sense that they are instantiations of the same Template, but individual replicas also have a consistent identity. If unspecified, defaults to 1.

Returns The replicas of this V1beta1StatefulSetSpec.

Return type int

revision_history_limit

Gets the revision_history_limit of this V1beta1StatefulSetSpec. revisionHistoryLimit is the maximum number of revisions that will be maintained in the StatefulSet's revision history. The revision history consists of all revisions not represented by a currently applied StatefulSetSpec version. The default value is 10.

Returns The revision_history_limit of this V1beta1StatefulSetSpec.

Return type int

selector

Gets the selector of this V1beta1StatefulSetSpec. selector is a label query over pods that should match the replica count. If empty, defaulted to labels on the pod template. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors

Returns The selector of this V1beta1StatefulSetSpec.

Return type V1LabelSelector

service_name
Gets the service_name of this V1beta1StatefulSetSpec. serviceName is the name of the service that governs this StatefulSet. This service must exist before the StatefulSet, and is responsible for the network identity of the set. Pods get DNS/hostnames that follow the pattern: pod-specific-string.serviceName.default.svc.cluster.local where "pod-specific-string" is managed by the StatefulSet controller.

Returns The service_name of this V1beta1StatefulSetSpec.

Return type str

swagger_types = {'pod_management_policy': 'str', 'replicas': 'int', 'revision_histor

template
Gets the template of this V1beta1StatefulSetSpec. template is the object that describes the pod that will be created if insufficient replicas are detected. Each pod stamped out by the StatefulSet will fulfill this Template, but have a unique identity from the rest of the StatefulSet.

Returns The template of this V1beta1StatefulSetSpec.

Return type *V1PodTemplateSpec*

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

update_strategy
Gets the update_strategy of this V1beta1StatefulSetSpec. updateStrategy indicates the StatefulSetUpdateStrategy that will be employed to update Pods in the StatefulSet when a revision is made to Template.

Returns The update_strategy of this V1beta1StatefulSetSpec.

Return type V1beta1StatefulSetUpdateStrategy

volume_claim_templates
Gets the volume_claim_templates of this V1beta1StatefulSetSpec. volumeClaimTemplates is a list of claims that pods are allowed to reference. The StatefulSet controller is responsible for mapping network identities to claims in a way that maintains the identity of a pod. Every claim in this list must have at least one matching (by name) volumeMount in one container in the template. A claim in this list takes precedence over any volumes in the template, with the same name.

Returns The volume_claim_templates of this V1beta1StatefulSetSpec.

Return type list[*V1PersistentVolumeClaim*]

kubernetes.client.models.v1beta1_stateful_set_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_stateful_set_status.**V1beta1StatefulSetStatus**(*collision_count=None*, *conditions=None*, *current_replicas=None*, *current_revision=None*, *observed_generation=None*, *ready_replicas=None*, *replicas=None*, *update_revision=None*, *updated_replicas=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'collision_count': 'collisionCount', 'conditions': 'conditions', 'c

collision_count
Gets the collision_count of this V1beta1StatefulSetStatus. collisionCount is the count of hash collisions for the StatefulSet. The StatefulSet controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ControllerRevision.

Returns The collision_count of this V1beta1StatefulSetStatus.

Return type int

conditions
Gets the conditions of this V1beta1StatefulSetStatus. Represents the latest available observations of a statefulset's current state.

Returns The conditions of this V1beta1StatefulSetStatus.

Return type list[V1beta1StatefulSetCondition]

current_replicas
Gets the current_replicas of this V1beta1StatefulSetStatus. currentReplicas is the number of Pods created by the StatefulSet controller from the StatefulSet version indicated by currentRevision.

Returns The current_replicas of this V1beta1StatefulSetStatus.

Return type int

current_revision
Gets the current_revision of this V1beta1StatefulSetStatus. currentRevision, if not empty, indicates the version of the StatefulSet used to generate Pods in the sequence [0,currentReplicas).

Returns The current_revision of this V1beta1StatefulSetStatus.

Return type str

observed_generation
Gets the observed_generation of this V1beta1StatefulSetStatus. observedGeneration is the most recent generation observed for this StatefulSet. It corresponds to the StatefulSet's generation, which is updated on mutation by the API Server.

Returns The observed_generation of this V1beta1StatefulSetStatus.

Return type int

ready_replicas
Gets the ready_replicas of this V1beta1StatefulSetStatus. readyReplicas is the number of Pods created by the StatefulSet controller that have a Ready Condition.

Returns The ready_replicas of this V1beta1StatefulSetStatus.

Return type int

replicas
Gets the replicas of this V1beta1StatefulSetStatus. replicas is the number of Pods created by the StatefulSet controller.

Returns The replicas of this V1beta1StatefulSetStatus.

Return type int

swagger_types = {'collision_count': 'int', 'conditions': 'list[V1beta1StatefulSetCon

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

update_revision
Gets the update_revision of this V1beta1StatefulSetStatus. updateRevision, if not empty, indicates the version of the StatefulSet used to generate Pods in the sequence [replicas-updatedReplicas,replicas)

Returns The update_revision of this V1beta1StatefulSetStatus.

Return type str

updated_replicas
Gets the updated_replicas of this V1beta1StatefulSetStatus. updatedReplicas is the number of Pods created by the StatefulSet controller from the StatefulSet version indicated by updateRevision.

Returns The updated_replicas of this V1beta1StatefulSetStatus.

Return type int

kubernetes.client.models.v1beta1_storage_class module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_storage_class.**V1beta1StorageClass**(*allow_volume_expansion=None*, *api_version=None*, *kind=None*, *metadata=None*, *mount_options=None*, *parameters=None*, *provisioner=None*, *reclaim_policy=None*, *volume_binding_mode=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

allow_volume_expansion
Gets the allow_volume_expansion of this V1beta1StorageClass. AllowVolumeExpansion shows whether the storage class allow volume expand

Returns The allow_volume_expansion of this V1beta1StorageClass.

Return type bool

api_version
Gets the api_version of this V1beta1StorageClass. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1StorageClass.

Return type str

attribute_map = {'allow_volume_expansion': 'allowVolumeExpansion', 'api_version': 'ap

kind
Gets the kind of this V1beta1StorageClass. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1StorageClass.

Return type str

metadata
Gets the metadata of this V1beta1StorageClass. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1StorageClass.

Return type *V1ObjectMeta*

mount_options
Gets the mount_options of this V1beta1StorageClass. Dynamically provisioned PersistentVolumes of this

storage class are created with these mountOptions, e.g. [“ro”, “soft”]. Not validated - mount of the PVs will simply fail if one is invalid.

Returns The mount_options of this V1beta1StorageClass.

Return type list[str]

parameters
Gets the parameters of this V1beta1StorageClass. Parameters holds the parameters for the provisioner that should create volumes of this storage class.

Returns The parameters of this V1beta1StorageClass.

Return type dict(str, str)

provisioner
Gets the provisioner of this V1beta1StorageClass. Provisioner indicates the type of the provisioner.

Returns The provisioner of this V1beta1StorageClass.

Return type str

reclaim_policy
Gets the reclaim_policy of this V1beta1StorageClass. Dynamically provisioned PersistentVolumes of this storage class are created with this reclaimPolicy. Defaults to Delete.

Returns The reclaim_policy of this V1beta1StorageClass.

Return type str

swagger_types = {'allow_volume_expansion': 'bool', 'api_version': 'str', 'kind': 'st

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

volume_binding_mode
Gets the volume_binding_mode of this V1beta1StorageClass. VolumeBindingMode indicates how PersistentVolumeClaims should be provisioned and bound. When unset, VolumeBindingImmediate is used. This field is alpha-level and is only honored by servers that enable the VolumeScheduling feature.

Returns The volume_binding_mode of this V1beta1StorageClass.

Return type str

kubernetes.client.models.v1beta1_storage_class_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_storage_class_list.**V1beta1StorageClassList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1StorageClassList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1StorageClassList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V1beta1StorageClassList. Items is the list of StorageClasses

Returns The items of this V1beta1StorageClassList.

Return type list[*V1beta1StorageClass*]

kind
Gets the kind of this V1beta1StorageClassList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1StorageClassList.

Return type str

metadata
Gets the metadata of this V1beta1StorageClassList. Standard list metadata More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V1beta1StorageClassList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V1beta1StorageClass]', 'kind':

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_subject_access_review module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_subject_access_review.**V1beta1SubjectAccessReview**(*api_*
kind
met
dat
spe
sta-
tus=

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1SubjectAccessReview. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1SubjectAccessReview.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1beta1SubjectAccessReview. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1SubjectAccessReview.

Return type str

metadata
Gets the metadata of this V1beta1SubjectAccessReview.

Returns The metadata of this V1beta1SubjectAccessReview.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1beta1SubjectAccessReview. Spec holds information about the request being evaluated

Returns The spec of this V1beta1SubjectAccessReview.

Return type *V1beta1SubjectAccessReviewSpec*

status
Gets the status of this V1beta1SubjectAccessReview. Status is filled in by the server and indicates whether the request is allowed or not

Returns The status of this V1beta1SubjectAccessReview.

Return type *V1beta1SubjectAccessReviewStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_subject_access_review_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_subject_access_review_spec.**V1beta1SubjectAccessRevie**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'extra': 'extra', 'group': 'group', 'non_resource_attributes': 'no

extra
Gets the extra of this V1beta1SubjectAccessReviewSpec. Extra corresponds to the user.Info.GetExtra() method from the authenticator. Since that is input to the authorizer it needs a reflection here.

Returns The extra of this V1beta1SubjectAccessReviewSpec.

Return type dict(str, list[str])

group
Gets the group of this V1beta1SubjectAccessReviewSpec. Groups is the groups you're testing for.

Returns The group of this V1beta1SubjectAccessReviewSpec.

Return type list[str]

non_resource_attributes
Gets the non_resource_attributes of this V1beta1SubjectAccessReviewSpec. NonResourceAttributes describes information for a non-resource access request

Returns The non_resource_attributes of this V1beta1SubjectAccessReviewSpec.

Return type *V1beta1NonResourceAttributes*

resource_attributes
Gets the resource_attributes of this V1beta1SubjectAccessReviewSpec. ResourceAuthorizationAttributes describes information for a resource access request

Returns The resource_attributes of this V1beta1SubjectAccessReviewSpec.

Return type *V1beta1ResourceAttributes*

swagger_types = {'extra': 'dict(str, list[str])', 'group': 'list[str]', 'non_resourc

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

uid
Gets the uid of this V1beta1SubjectAccessReviewSpec. UID information about the requesting user.

Returns The uid of this V1beta1SubjectAccessReviewSpec.

Return type str

user
Gets the user of this V1beta1SubjectAccessReviewSpec. User is the user you're testing for. If you specify "User" but not "Group", then is it interpreted as "What if User were not a member of any groups

Returns The user of this V1beta1SubjectAccessReviewSpec.

Return type str

kubernetes.client.models.v1beta1_subject_access_review_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_subject_access_review_status.**V1beta1SubjectAccessRev**

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

allowed
Gets the allowed of this V1beta1SubjectAccessReviewStatus. Allowed is required. True if the action would be allowed, false otherwise.

Returns The allowed of this V1beta1SubjectAccessReviewStatus.

Return type bool

attribute_map = {'allowed': 'allowed', 'denied': 'denied', 'evaluation_error': 'eva

denied
Gets the denied of this V1beta1SubjectAccessReviewStatus. Denied is optional. True if the action would be denied, otherwise false. If both allowed is false and denied is false, then the authorizer has no opinion on whether to authorize the action. Denied may not be true if Allowed is true.

Returns The denied of this V1beta1SubjectAccessReviewStatus.

Return type bool

evaluation_error
Gets the evaluation_error of this V1beta1SubjectAccessReviewStatus. EvaluationError is an indication that some error occurred during the authorization check. It is entirely possible to get an error and be able to continue determine authorization status in spite of it. For instance, RBAC can be missing a role, but enough roles are still present and bound to reason about the request.

Returns The evaluation_error of this V1beta1SubjectAccessReviewStatus.

Return type str

reason
Gets the reason of this V1beta1SubjectAccessReviewStatus. Reason is optional. It indicates why a request was allowed or denied.

Returns The reason of this V1beta1SubjectAccessReviewStatus.

Return type str

swagger_types = {'allowed': 'bool', 'denied': 'bool', 'evaluation_error': 'str', 'r

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_subresource_reference module

kubernetes.client.models.v1beta1_third_party_resource module

kubernetes.client.models.v1beta1_third_party_resource_list module

kubernetes.client.models.v1beta1_token_review module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_token_review.**V1beta1TokenReview**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V1beta1TokenReview. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V1beta1TokenReview.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind
Gets the kind of this V1beta1TokenReview. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V1beta1TokenReview.

Return type str

metadata
Gets the metadata of this V1beta1TokenReview.

Returns The metadata of this V1beta1TokenReview.

Return type *V1ObjectMeta*

spec
Gets the spec of this V1beta1TokenReview. Spec holds information about the request being evaluated

Returns The spec of this V1beta1TokenReview.

Return type *V1beta1TokenReviewSpec*

status
Gets the status of this V1beta1TokenReview. Status is filled in by the server and indicates whether the request can be authenticated.

Returns The status of this V1beta1TokenReview.

Return type *V1beta1TokenReviewStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta', '

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v1beta1_token_review_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_token_review_spec.**V1beta1TokenReviewSpec**(*token=None*)
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'token': 'token'}

swagger_types = {'token': 'str'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

token
Gets the token of this V1beta1TokenReviewSpec. Token is the opaque bearer token.

Returns The token of this V1beta1TokenReviewSpec.

Return type str

kubernetes.client.models.v1beta1_token_review_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_token_review_status.**V1beta1TokenReviewStatus**(*authenticated=None*, *error=None*, *user=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'authenticated': 'authenticated', 'error': 'error', 'user': 'user'}

authenticated
Gets the authenticated of this V1beta1TokenReviewStatus. Authenticated indicates that the token was associated with a known user.

Returns The authenticated of this V1beta1TokenReviewStatus.

Return type bool

error
Gets the error of this V1beta1TokenReviewStatus. Error indicates that the token couldn't be checked

Returns The error of this V1beta1TokenReviewStatus.

Return type str

swagger_types = {'authenticated': 'bool', 'error': 'str', 'user': 'V1beta1UserInfo'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

user
Gets the user of this V1beta1TokenReviewStatus. User is the UserInfo associated with the provided token.

Returns The user of this V1beta1TokenReviewStatus.

Return type *V1beta1UserInfo*

kubernetes.client.models.v1beta1_user_info module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v1beta1_user_info.**V1beta1UserInfo**(*extra=None*, *groups=None*, *uid=None*, *username=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'extra': 'extra', 'groups': 'groups', 'uid': 'uid', 'username': '

extra
Gets the extra of this V1beta1UserInfo. Any additional information provided by the authenticator.

Returns The extra of this V1beta1UserInfo.

Return type dict(str, list[str])

groups
Gets the groups of this V1beta1UserInfo. The names of groups this user is a part of.

Returns The groups of this V1beta1UserInfo.

Return type list[str]

swagger_types = {'extra': 'dict(str, list[str])', 'groups': 'list[str]', 'uid': 'st

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

uid
Gets the uid of this V1beta1UserInfo. A unique value that identifies this user across time. If this user is deleted and another user by the same name is added, they will have different UIDs.

Returns The uid of this V1beta1UserInfo.

Return type str

username
Gets the username of this V1beta1UserInfo. The name that uniquely identifies this user among all active users.

Returns The username of this V1beta1UserInfo.

Return type str

kubernetes.client.models.v2alpha1_cron_job module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v2alpha1_cron_job.**V2alpha1CronJob**(*api_version=None*, *kind=None*, *metadata=None*, *spec=None*, *status=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version

Gets the api_version of this V2alpha1CronJob. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V2alpha1CronJob.

Return type str

attribute_map = {'api_version': 'apiVersion', 'kind': 'kind', 'metadata': 'metadata

kind

Gets the kind of this V2alpha1CronJob. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V2alpha1CronJob.

Return type str

metadata

Gets the metadata of this V2alpha1CronJob. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V2alpha1CronJob.

Return type *V1ObjectMeta*

spec

Gets the spec of this V2alpha1CronJob. Specification of the desired behavior of a cron job, including the schedule. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V2alpha1CronJob.

Return type *V2alpha1CronJobSpec*

status

Gets the status of this V2alpha1CronJob. Current status of a cron job. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The status of this V2alpha1CronJob.

Return type *V2alpha1CronJobStatus*

swagger_types = {'api_version': 'str', 'kind': 'str', 'metadata': 'V1ObjectMeta',

to_dict()

Returns the model properties as a dict

to_str()

Returns the string representation of the model

kubernetes.client.models.v2alpha1_cron_job_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v2alpha1_cron_job_list.**V2alpha1CronJobList**(*api_version=None*, *items=None*, *kind=None*, *metadata=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

api_version
Gets the api_version of this V2alpha1CronJobList. APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources

Returns The api_version of this V2alpha1CronJobList.

Return type str

attribute_map = {'api_version': 'apiVersion', 'items': 'items', 'kind': 'kind', 'me

items
Gets the items of this V2alpha1CronJobList. items is the list of CronJobs.

Returns The items of this V2alpha1CronJobList.

Return type list[*V2alpha1CronJob*]

kind
Gets the kind of this V2alpha1CronJobList. Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds

Returns The kind of this V2alpha1CronJobList.

Return type str

metadata
Gets the metadata of this V2alpha1CronJobList. Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V2alpha1CronJobList.

Return type V1ListMeta

swagger_types = {'api_version': 'str', 'items': 'list[V2alpha1CronJob]', 'kind': 's

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v2alpha1_cron_job_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v2alpha1_cron_job_spec.**V2alpha1CronJobSpec**(*concurrency_policy=No failed_jobs_history_limi job_template=None, sched- ule=None, start- ing_deadline_seconds=I suc- cess- ful_jobs_history_limit=N sus- pend=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'concurrency_policy': 'concurrencyPolicy', 'failed_jobs_history_limi

concurrency_policy
Gets the concurrency_policy of this V2alpha1CronJobSpec. Specifies how to treat concurrent executions of a Job. Valid values are: - "Allow" (default): allows CronJobs to run concurrently; - "Forbid": forbids concurrent runs, skipping next run if previous run hasn't finished yet; - "Replace": cancels currently running job and replaces it with a new one

Returns The concurrency_policy of this V2alpha1CronJobSpec.

Return type str

failed_jobs_history_limit
Gets the failed_jobs_history_limit of this V2alpha1CronJobSpec. The number of failed finished jobs to retain. This is a pointer to distinguish between explicit zero and not specified.

Returns The failed_jobs_history_limit of this V2alpha1CronJobSpec.

Return type int

job_template
Gets the job_template of this V2alpha1CronJobSpec. Specifies the job that will be created when executing a CronJob.

Returns The job_template of this V2alpha1CronJobSpec.

Return type *V2alpha1JobTemplateSpec*

schedule
Gets the schedule of this V2alpha1CronJobSpec. The schedule in Cron format, see https://en.wikipedia.org/wiki/Cron.

Returns The schedule of this V2alpha1CronJobSpec.

Return type str

starting_deadline_seconds
Gets the starting_deadline_seconds of this V2alpha1CronJobSpec. Optional deadline in seconds for starting the job if it misses scheduled time for any reason. Missed jobs executions will be counted as failed ones.

Returns The starting_deadline_seconds of this V2alpha1CronJobSpec.

Return type int

successful_jobs_history_limit
Gets the successful_jobs_history_limit of this V2alpha1CronJobSpec. The number of successful finished jobs to retain. This is a pointer to distinguish between explicit zero and not specified.

Returns The successful_jobs_history_limit of this V2alpha1CronJobSpec.

Return type int

suspend
Gets the suspend of this V2alpha1CronJobSpec. This flag tells the controller to suspend subsequent executions, it does not apply to already started executions. Defaults to false.

Returns The suspend of this V2alpha1CronJobSpec.

Return type bool

swagger_types = {'concurrency_policy': 'str', 'failed_jobs_history_limit': 'int', 'j

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v2alpha1_cron_job_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v2alpha1_cron_job_status.**V2alpha1CronJobStatus**(*active=None*, *last_schedule_time*

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

active
Gets the active of this V2alpha1CronJobStatus. A list of pointers to currently running jobs.

Returns The active of this V2alpha1CronJobStatus.

Return type list[*V1ObjectReference*]

attribute_map = {'active': 'active', 'last_schedule_time': 'lastScheduleTime'}

last_schedule_time
Gets the last_schedule_time of this V2alpha1CronJobStatus. Information when was the last time the job was successfully scheduled.

Returns The last_schedule_time of this V2alpha1CronJobStatus.

Return type datetime

swagger_types = {'active': 'list[V1ObjectReference]', 'last_schedule_time': 'datetim

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.v2alpha1_job module

kubernetes.client.models.v2alpha1_job_condition module

kubernetes.client.models.v2alpha1_job_list module

kubernetes.client.models.v2alpha1_job_spec module

kubernetes.client.models.v2alpha1_job_status module

kubernetes.client.models.v2alpha1_job_template_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.v2alpha1_job_template_spec.**V2alpha1JobTemplateSpec**(*metadata=N*
spec=None)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'metadata': 'metadata', 'spec': 'spec'}

metadata
Gets the metadata of this V2alpha1JobTemplateSpec. Standard object's metadata of the jobs created from this template. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

Returns The metadata of this V2alpha1JobTemplateSpec.

Return type *V1ObjectMeta*

spec
Gets the spec of this V2alpha1JobTemplateSpec. Specification of the desired behavior of the job. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

Returns The spec of this V2alpha1JobTemplateSpec.

Return type *V1JobSpec*

swagger_types = {'metadata': 'V1ObjectMeta', 'spec': 'V1JobSpec'}

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.version_info module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.models.version_info.**VersionInfo**(*build_date=None*, *compiler=None*, *git_commit=None*, *git_tree_state=None*, *git_version=None*, *go_version=None*, *major=None*, *minor=None*, *platform=None*)

Bases: object

NOTE: This class is auto generated by the swagger code generator program. Do not edit the class manually.

attribute_map = {'build_date': 'buildDate', 'compiler': 'compiler', 'git_commit': '

build_date
Gets the build_date of this VersionInfo.

Returns The build_date of this VersionInfo.

Return type str

compiler
Gets the compiler of this VersionInfo.

Returns The compiler of this VersionInfo.

Return type str

git_commit
Gets the git_commit of this VersionInfo.

Returns The git_commit of this VersionInfo.

Return type str

git_tree_state
Gets the git_tree_state of this VersionInfo.

Returns The git_tree_state of this VersionInfo.

Return type str

git_version
Gets the git_version of this VersionInfo.

Returns The git_version of this VersionInfo.

Return type str

go_version
Gets the go_version of this VersionInfo.

Returns The go_version of this VersionInfo.

Return type str

major
Gets the major of this VersionInfo.

Returns The major of this VersionInfo.

Return type str

minor
Gets the minor of this VersionInfo.

Returns The minor of this VersionInfo.

Return type str

platform
Gets the platform of this VersionInfo.

Returns The platform of this VersionInfo.

Return type str

swagger_types = {'build_date': 'str', 'compiler': 'str', 'git_commit': 'str', 'git_

to_dict()
Returns the model properties as a dict

to_str()
Returns the string representation of the model

kubernetes.client.models.versioned_event module

Module contents

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

Submodules

kubernetes.client.api_client module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.api_client.**ApiClient**(*configuration=None*, *header_name=None*, *header_value=None*, *cookie=None*)

Bases: object

Generic API client for Swagger client library builds.

Swagger generic API client. This client handles the client- server communication, and is invariant across implementations. Specifics of the methods and models for each application are generated from the Swagger templates.

NOTE: This class is auto generated by the swagger code generator program. Ref: https://github.com/swagger-api/swagger-codegen Do not edit the class manually.

Parameters

- **host** – The base path for the server to call.
- **header_name** – a header to pass when making calls to the API.
- **header_value** – a header value to pass when making calls to the API.

NATIVE_TYPES_MAPPING = {'bool': <type 'bool'>, 'date': <type 'datetime.date'>, 'date

PRIMITIVE_TYPES = (<type 'float'>, <type 'bool'>, <type 'str'>, <type 'unicode'>, <typ

call_api(*resource_path*, *method*, *path_params=None*, *query_params=None*, *header_params=None*, *body=None*, *post_params=None*, *files=None*, *response_type=None*, *auth_settings=None*, *async=None*, *_return_http_data_only=None*, *collection_formats=None*, *_preload_content=True*, *_request_timeout=None*)

Makes the HTTP request (synchronous) and return the deserialized data. To make an async request, set the async parameter.

Parameters

- **resource_path** – Path to method endpoint.
- **method** – Method to call.
- **path_params** – Path parameters in the url.
- **query_params** – Query parameters in the url.
- **header_params** – Header parameters to be placed in the request header.
- **body** – Request body.
- **dict** (*files*) – Request post form parameters, for *application/x-www-form-urlencoded*, *multipart/form-data*.
- **list** (*auth_settings*) – Auth Settings names for the request.
- **response** – Response data type.
- **dict** – key -> filename, value -> filepath, for *multipart/form-data*.
- **bool** (*async*) – execute request asynchronously
- **_return_http_data_only** – response data without head status code and headers
- **collection_formats** – dict of collection formats for path, query, header, and post parameters.
- **_preload_content** – if False, the urllib3.HTTPResponse object will be returned without reading/decoding response data. Default is True.
- **_request_timeout** – timeout setting for this request. If one number provided, it will be total request timeout. It can also be a pair (tuple) of (connection, read) timeouts.

Returns If async parameter is True, the request will be called asynchronously. The method will return the request thread. If parameter async is False or missing, then the method will return the response directly.

deserialize(*response*, *response_type*)

Deserializes response into an object.

Parameters

- **response** – RESTResponse object to be deserialized.

- **response_type** – class literal for deserialized object, or string of class name.

Returns deserialized object.

parameters_to_tuples(*params*, *collection_formats*)
Get parameters as list of tuples, formatting collections.

Parameters

- **params** – Parameters as dict or list of two-tuples
- **collection_formats** (*dict*) – Parameter collection formats

Returns Parameters as list of tuples, collections formatted

prepare_post_parameters(*post_params=None*, *files=None*)
Builds form parameters.

Parameters

- **post_params** – Normal form parameters.
- **files** – File parameters.

Returns Form parameters with files.

request(*method*, *url*, *query_params=None*, *headers=None*, *post_params=None*, *body=None*, *_preload_content=True*, *_request_timeout=None*)
Makes the HTTP request using RESTClient.

sanitize_for_serialization(*obj*)
Builds a JSON POST object.

If obj is None, return None. If obj is str, int, long, float, bool, return directly. If obj is datetime.datetime, datetime.date

convert to string in iso8601 format.

If obj is list, sanitize each element in the list. If obj is dict, return the dict. If obj is swagger model, return the properties dict.

Parameters **obj** – The data to serialize.

Returns The serialized form of data.

select_header_accept(*accepts*)
Returns *Accept* based on an array of accepts provided.

Parameters **accepts** – List of headers.

Returns Accept (e.g. application/json).

select_header_content_type(*content_types*)
Returns *Content-Type* based on an array of content_types provided.

Parameters **content_types** – List of content-types.

Returns Content-Type (e.g. application/json).

set_default_header(*header_name*, *header_value*)

update_params_for_auth(*headers*, *querys*, *auth_settings*)
Updates header and query params based on authentication setting.

Parameters

- **headers** – Header parameters dict to be updated.
- **querys** – Query parameters tuple list to be updated.

- **auth_settings** – Authentication setting identifiers list.

user_agent
Gets user agent.

kubernetes.client.configuration module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.client.configuration.**Configuration**
Bases: object

NOTE: This class is auto generated by the swagger code generator program. Ref: https://github.com/swagger-api/swagger-codegen Do not edit the class manually.

auth_settings()
Gets Auth Settings dict for api client.

Returns The Auth Settings information dict.

debug
Gets the debug status.

get_api_key_with_prefix(*identifier*)
Gets API key (with prefix if set).

Parameters **identifier** – The identifier of apiKey.

Returns The token for api key authentication.

get_basic_auth_token()
Gets HTTP basic authentication header (string).

Returns The token for basic HTTP authentication.

logger_file
Gets the logger_file.

logger_format
Gets the logger_format.

to_debug_report()
Gets the essential information for debugging.

Returns The report for debugging.

class kubernetes.client.configuration.**TypeWithDefault**(*name*, *bases*, *dct*)
Bases: type

set_default(*default*)

kubernetes.client.rest module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

exception kubernetes.client.rest.**ApiException**(*status=None*, *reason=None*, *http_resp=None*)

Bases: exceptions.Exception

class kubernetes.client.rest.**RESTClientObject**(*configuration*, *pools_size=4*, *maxsize=None*)

Bases: object

DELETE(*url*, *headers=None*, *query_params=None*, *body=None*, *_preload_content=True*, *_request_timeout=None*)

GET(*url*, *headers=None*, *query_params=None*, *_preload_content=True*, *_request_timeout=None*)

HEAD(*url*, *headers=None*, *query_params=None*, *_preload_content=True*, *_request_timeout=None*)

OPTIONS(*url*, *headers=None*, *query_params=None*, *post_params=None*, *body=None*, *_preload_content=True*, *_request_timeout=None*)

PATCH(*url*, *headers=None*, *query_params=None*, *post_params=None*, *body=None*, *_preload_content=True*, *_request_timeout=None*)

POST(*url*, *headers=None*, *query_params=None*, *post_params=None*, *body=None*, *_preload_content=True*, *_request_timeout=None*)

PUT(*url*, *headers=None*, *query_params=None*, *post_params=None*, *body=None*, *_preload_content=True*, *_request_timeout=None*)

request(*method*, *url*, *query_params=None*, *headers=None*, *body=None*, *post_params=None*, *_preload_content=True*, *_request_timeout=None*)

Parameters

- **method** – http request method
- **url** – http request url
- **query_params** – query parameters in the url
- **headers** – http request headers
- **body** – request json body, for *application/json*
- **post_params** – request post parameters, *application/x-www-form-urlencoded* and *multipart/form-data*
- **_preload_content** – if False, the urllib3.HTTPResponse object will be returned without reading/decoding response data. Default is True.
- **_request_timeout** – timeout setting for this request. If one number provided, it will be total request timeout. It can also be a pair (tuple) of (connection, read) timeouts.

class kubernetes.client.rest.**RESTResponse**(*resp*)

Bases: io.IOBase

getheader(*name*, *default=None*)

Returns a given response header.

getheaders()

Returns a dictionary of the response headers.

Module contents

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

kubernetes.config package

Submodules

kubernetes.config.config_exception module

exception kubernetes.config.config_exception.**ConfigException**
Bases: exceptions.Exception

kubernetes.config.incluster_config module

class kubernetes.config.incluster_config.**InClusterConfigLoader**(*token_filename*, *cert_filename*, *environ={'LANG': 'C.UTF-8', 'READTHEDOCS_PROJECT': 'jashandeep-sohik8s-python', 'READTHEDOCS': 'True', 'APPDIR': '/app', 'DEBIAN_FRONTEND': 'noninteractive', 'OLDPWD': '/home/docs', 'HOSTNAME': 'build-7177534-project-214688-jashandeep-sohik8s-python', 'PWD': '/home/docs/checkouts/readthedocs.org/user_builds/sohik8s-python/checkouts/fix-sphinx-readme/doc/source', 'BIN_PATH': '/home/docs/checkouts/readthedocs.org/user_builds/sohik8s-python/envs/fix-sphinx-readme/bin', 'READTHEDOCS_VERSION': 'fix-sphinx-readme', 'PATH': '/home/docs/checkouts/readthedocs.org/user_builds/sohik8s-python/envs/fix-sphinx-readme/bin:/usr/local/sbin:/usr/local/bin', 'HOME': '/home/docs'}*)

Bases: object

load_and_set()

kubernetes.config.incluster_config.**load_incluster_config**()

Use the service account kubernetes gives to pods to connect to kubernetes cluster. It's intended for clients that

expect to be running inside a pod running on kubernetes. It will raise an exception if called from a process not running in a kubernetes environment.

kubernetes.config.incluster_config_test module

class kubernetes.config.incluster_config_test.**InClusterConfigTest**(*methodName='runTest'*)
Bases: unittest.case.TestCase

get_test_loader(*token_filename=None*, *cert_filename=None*, *environ={'KUBERNETES_SERVICE_HOST': '127.0.0.1', 'KUBERNETES_SERVICE_PORT': '80'}*)

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_empty_cert_file()

test_empty_host()

test_empty_port()

test_empty_token_file()

test_join_host_port()

test_load_config()

test_no_cert_file()

test_no_host()

test_no_port()

test_no_token_file()

kubernetes.config.kube_config module

class kubernetes.config.kube_config.**ConfigNode**(*name*, *value*)
Bases: object

Remembers each config key's path and construct a relevant exception message in case of missing keys. The assumption is all access keys are present in a well-formed kube-config.

get_with_name(*name*, *safe=False*)

safe_get(*key*)

class kubernetes.config.kube_config.**FileOrData**(*obj*, *file_key_name*, *data_key_name=None*, *file_base_path=''*, *base64_file_content=True*)
Bases: object

Utility class to read content of obj[%data_key_name] or file's content of obj[%file_key_name] and represent it as file or data. Note that the data is preferred. The obj[%file_key_name] will be used iff obj['%data_key_name'] is not set or empty. Assumption is file content is raw data and data field is base64 string. The assumption can be changed with base64_file_content flag. If set to False, the content of the file will assumed to be base64 and read as is. The default True value will result in base64 encode of the file content after read.

as_data()
If obj[%data_key_name] exists, Return obj[%data_key_name] otherwise base64 encoded string of obj[%file_key_name] file content.

as_file()
If obj[%data_key_name] exists, return name of a file with base64 decoded obj[%data_key_name] content otherwise obj[%file_key_name].

class kubernetes.config.kube_config.**KubeConfigLoader**(*config_dict*, *active_context=None*, *get_google_credentials=None*, *config_base_path=''*, *config_persister=None*)
Bases: object

current_context

list_contexts()

load_and_set(*client_configuration*)

set_active_context(*context_name=None*)

kubernetes.config.kube_config.**list_kube_config_contexts**(*config_file=None*)

kubernetes.config.kube_config.**load_kube_config**(*config_file=None*, *context=None*, *client_configuration=None*, *persist_config=True*)
Loads authentication and cluster information from kube-config file and stores them in kubernetes.client.configuration.

Parameters

- **config_file** – Name of the kube-config file.
- **context** – set the active context. If is set to None, current_context from config file will be used.
- **client_configuration** – The kubernetes.client.Configuration to set configs to.
- **persist_config** – If True, config file will be updated when changed (e.g GCP token refresh).

kubernetes.config.kube_config.**new_client_from_config**(*config_file=None*, *context=None*, *persist_config=True*)
Loads configuration the same as load_kube_config but returns an ApiClient to be used with any API object. This will allow the caller to concurrently talk with multiple clusters.

kubernetes.config.kube_config_test module

class kubernetes.config.kube_config_test.**BaseTestCase**(*methodName='runTest'*)
Bases: unittest.case.TestCase

expect_exception(*func*, *message_part*, **args*, ***kwargs*)

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

class kubernetes.config.kube_config_test.**FakeConfig**(*token=None*, ***kwargs*)

FILE_KEYS = ['ssl_ca_cert', 'key_file', 'cert_file']

class kubernetes.config.kube_config_test.**TestConfigNode**(*methodName='runTest'*)
Bases: *kubernetes.config.kube_config_test.BaseTestCase*

setUp()
Hook method for setting up the test fixture before exercising it.

test_get_with_name()

test_get_with_name_on_duplicate_name()

test_get_with_name_on_invalid_object()

test_get_with_name_on_name_does_not_exists()

test_get_with_name_on_non_list_object()

test_key_does_not_exists()

test_normal_map_array_operations()

test_obj = {'key1': 'test', 'key2': ['a', 'b', 'c'], 'key3': {'inner_key': 'inner_

class kubernetes.config.kube_config_test.**TestFileOrData**(*methodName='runTest'*)
Bases: *kubernetes.config.kube_config_test.BaseTestCase*

static get_file_content(*filename*)

test_create_temp_file_with_content()

test_data_given_data()

test_data_given_file()

test_data_given_file_and_data()

test_data_given_file_no_base64()

test_file_given_data()

test_file_given_data_no_base64()

test_file_given_file()

test_file_given_file_and_data()

test_file_given_non_existing_file()

test_file_with_custom_dirname()

class kubernetes.config.kube_config_test.**TestKubeConfigLoader**(*methodName='runTest'*)
Bases: *kubernetes.config.kube_config_test.BaseTestCase*

TEST_KUBE_CONFIG = {'clusters': [{'cluster': {'server': 'test-host'}, 'name': 'defa

test_current_context()

test_gcp_no_refresh()

test_list_contexts()

test_list_kube_config_contexts()

test_load_gcp_token_no_refresh()

test_load_gcp_token_with_refresh()

test_load_kube_config()

test_load_user_pass_token()

test_load_user_token()

test_new_client_from_config()

test_no_user_context()

test_no_users_section()

test_non_existing_user()

test_oidc_no_refresh()

test_oidc_with_refresh(***keywargs*)

test_set_active_context()

test_simple_token()

test_ssl()

test_ssl_no_cert_files()

test_ssl_no_verification()

test_ssl_with_relative_ssl_files()

test_user_pass()

Module contents

kubernetes.test package

Submodules

kubernetes.test.test_apis_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_apis_api.**TestApisApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

ApisApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_versions()
Test case for get_api_versions

kubernetes.test.test_apps_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_apps_api.**TestAppsApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

AppsApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_apps_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_apps_v1beta1_api.**TestAppsV1beta1Api**(*methodName='runTest'*)
Bases: unittest.case.TestCase

AppsV1beta1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_namespaced_controller_revision()
Test case for create_namespaced_controller_revision

test_create_namespaced_deployment()
Test case for create_namespaced_deployment

test_create_namespaced_deployment_rollback()
Test case for create_namespaced_deployment_rollback

test_create_namespaced_stateful_set()
Test case for create_namespaced_stateful_set

test_delete_collection_namespaced_controller_revision()
Test case for delete_collection_namespaced_controller_revision

test_delete_collection_namespaced_deployment()
Test case for delete_collection_namespaced_deployment

test_delete_collection_namespaced_stateful_set()
Test case for delete_collection_namespaced_stateful_set

test_delete_namespaced_controller_revision()
Test case for delete_namespaced_controller_revision

test_delete_namespaced_deployment()
Test case for delete_namespaced_deployment

test_delete_namespaced_stateful_set()
Test case for delete_namespaced_stateful_set

test_get_api_resources()
Test case for get_api_resources

test_list_controller_revision_for_all_namespaces()
Test case for list_controller_revision_for_all_namespaces

test_list_deployment_for_all_namespaces()
Test case for list_deployment_for_all_namespaces

test_list_namespaced_controller_revision()
Test case for list_namespaced_controller_revision

test_list_namespaced_deployment()
Test case for list_namespaced_deployment

test_list_namespaced_stateful_set()
Test case for list_namespaced_stateful_set

test_list_stateful_set_for_all_namespaces()
Test case for list_stateful_set_for_all_namespaces

test_patch_namespaced_controller_revision()
Test case for patch_namespaced_controller_revision

test_patch_namespaced_deployment()
Test case for patch_namespaced_deployment

test_patch_namespaced_deployment_scale()
Test case for patch_namespaced_deployment_scale

test_patch_namespaced_deployment_status()
Test case for patch_namespaced_deployment_status

test_patch_namespaced_stateful_set()
Test case for patch_namespaced_stateful_set

test_patch_namespaced_stateful_set_scale()
Test case for patch_namespaced_stateful_set_scale

test_patch_namespaced_stateful_set_status()
Test case for patch_namespaced_stateful_set_status

test_read_namespaced_controller_revision()
Test case for read_namespaced_controller_revision

test_read_namespaced_deployment()
Test case for read_namespaced_deployment

test_read_namespaced_deployment_scale()
Test case for read_namespaced_deployment_scale

test_read_namespaced_deployment_status()
Test case for read_namespaced_deployment_status

test_read_namespaced_stateful_set()
Test case for read_namespaced_stateful_set

test_read_namespaced_stateful_set_scale()
Test case for read_namespaced_stateful_set_scale

test_read_namespaced_stateful_set_status()
Test case for read_namespaced_stateful_set_status

test_replace_namespaced_controller_revision()
Test case for replace_namespaced_controller_revision

test_replace_namespaced_deployment()
Test case for replace_namespaced_deployment

test_replace_namespaced_deployment_scale()
Test case for replace_namespaced_deployment_scale

test_replace_namespaced_deployment_status()
Test case for replace_namespaced_deployment_status

test_replace_namespaced_stateful_set()
Test case for replace_namespaced_stateful_set

test_replace_namespaced_stateful_set_scale()
Test case for replace_namespaced_stateful_set_scale

test_replace_namespaced_stateful_set_status()
Test case for replace_namespaced_stateful_set_status

kubernetes.test.test_authentication_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_authentication_api.**TestAuthenticationApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

AuthenticationApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_authentication_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_authentication_v1beta1_api.**TestAuthenticationV1beta1Api**(*methodNa*
Bases: unittest.case.TestCase

AuthenticationV1beta1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_token_review()
Test case for create_token_review

test_get_api_resources()
Test case for get_api_resources

kubernetes.test.test_authorization_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_authorization_api.**TestAuthorizationApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

AuthorizationApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_authorization_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_authorization_v1beta1_api.**TestAuthorizationV1beta1Api**(*methodName=*
Bases: unittest.case.TestCase

AuthorizationV1beta1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_namespaced_local_subject_access_review()
Test case for create_namespaced_local_subject_access_review

test_create_self_subject_access_review()
Test case for create_self_subject_access_review

test_create_self_subject_rules_review()
Test case for create_self_subject_rules_review

test_create_subject_access_review()
Test case for create_subject_access_review

test_get_api_resources()
Test case for get_api_resources

kubernetes.test.test_autoscaling_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_autoscaling_api.**TestAutoscalingApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

AutoscalingApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_autoscaling_v1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_autoscaling_v1_api.**TestAutoscalingV1Api**(*methodName='runTest'*)
Bases: unittest.case.TestCase

AutoscalingV1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_namespaced_horizontal_pod_autoscaler()
Test case for create_namespaced_horizontal_pod_autoscaler

test_delete_collection_namespaced_horizontal_pod_autoscaler()
Test case for delete_collection_namespaced_horizontal_pod_autoscaler

test_delete_namespaced_horizontal_pod_autoscaler()
Test case for delete_namespaced_horizontal_pod_autoscaler

test_get_api_resources()
Test case for get_api_resources

test_list_horizontal_pod_autoscaler_for_all_namespaces()
Test case for list_horizontal_pod_autoscaler_for_all_namespaces

test_list_namespaced_horizontal_pod_autoscaler()
Test case for list_namespaced_horizontal_pod_autoscaler

test_patch_namespaced_horizontal_pod_autoscaler()
Test case for patch_namespaced_horizontal_pod_autoscaler

test_patch_namespaced_horizontal_pod_autoscaler_status()
Test case for patch_namespaced_horizontal_pod_autoscaler_status

test_read_namespaced_horizontal_pod_autoscaler()
Test case for read_namespaced_horizontal_pod_autoscaler

test_read_namespaced_horizontal_pod_autoscaler_status()
Test case for read_namespaced_horizontal_pod_autoscaler_status

test_replace_namespaced_horizontal_pod_autoscaler()
Test case for replace_namespaced_horizontal_pod_autoscaler

test_replace_namespaced_horizontal_pod_autoscaler_status()
Test case for replace_namespaced_horizontal_pod_autoscaler_status

kubernetes.test.test_batch_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_batch_api.**TestBatchApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

BatchApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_batch_v1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_batch_v1_api.**TestBatchV1Api**(*methodName='runTest'*)
Bases: unittest.case.TestCase

BatchV1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_namespaced_job()
Test case for create_namespaced_job

test_delete_collection_namespaced_job()
Test case for delete_collection_namespaced_job

test_delete_namespaced_job()
Test case for delete_namespaced_job

test_get_api_resources()
Test case for get_api_resources

test_list_job_for_all_namespaces()
Test case for list_job_for_all_namespaces

test_list_namespaced_job()
Test case for list_namespaced_job

test_patch_namespaced_job()
Test case for patch_namespaced_job

test_patch_namespaced_job_status()
Test case for patch_namespaced_job_status

test_read_namespaced_job()
Test case for read_namespaced_job

test_read_namespaced_job_status()
Test case for read_namespaced_job_status

test_replace_namespaced_job()
Test case for replace_namespaced_job

test_replace_namespaced_job_status()
Test case for replace_namespaced_job_status

kubernetes.test.test_batch_v2alpha1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_batch_v2alpha1_api.**TestBatchV2alpha1Api**(*methodName='runTest'*)
Bases: unittest.case.TestCase

BatchV2alpha1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_namespaced_cron_job()
Test case for create_namespaced_cron_job

test_delete_collection_namespaced_cron_job()
Test case for delete_collection_namespaced_cron_job

test_delete_namespaced_cron_job()
Test case for delete_namespaced_cron_job

test_get_api_resources()
Test case for get_api_resources

test_list_cron_job_for_all_namespaces()
Test case for list_cron_job_for_all_namespaces

test_list_namespaced_cron_job()
Test case for list_namespaced_cron_job

test_patch_namespaced_cron_job()
Test case for patch_namespaced_cron_job

test_patch_namespaced_cron_job_status()
Test case for patch_namespaced_cron_job_status

test_read_namespaced_cron_job()
Test case for read_namespaced_cron_job

test_read_namespaced_cron_job_status()
Test case for read_namespaced_cron_job_status

test_replace_namespaced_cron_job()
Test case for replace_namespaced_cron_job

test_replace_namespaced_cron_job_status()
Test case for replace_namespaced_cron_job_status

kubernetes.test.test_certificates_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_certificates_api.**TestCertificatesApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

CertificatesApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_certificates_v1alpha1_api module

kubernetes.test.test_core_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_core_api.**TestCoreApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

CoreApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_versions()
Test case for get_api_versions

kubernetes.test.test_core_v1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_core_v1_api.**TestCoreV1Api**(*methodName='runTest'*)
Bases: unittest.case.TestCase

CoreV1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_connect_delete_namespaced_pod_proxy()
Test case for connect_delete_namespaced_pod_proxy

test_connect_delete_namespaced_pod_proxy_with_path()
Test case for connect_delete_namespaced_pod_proxy_with_path

test_connect_delete_namespaced_service_proxy()
Test case for connect_delete_namespaced_service_proxy

test_connect_delete_namespaced_service_proxy_with_path()
Test case for connect_delete_namespaced_service_proxy_with_path

test_connect_delete_node_proxy()
Test case for connect_delete_node_proxy

test_connect_delete_node_proxy_with_path()
Test case for connect_delete_node_proxy_with_path

test_connect_get_namespaced_pod_attach()
Test case for connect_get_namespaced_pod_attach

test_connect_get_namespaced_pod_exec()
Test case for connect_get_namespaced_pod_exec

test_connect_get_namespaced_pod_portforward()
Test case for connect_get_namespaced_pod_portforward

test_connect_get_namespaced_pod_proxy()
Test case for connect_get_namespaced_pod_proxy

test_connect_get_namespaced_pod_proxy_with_path()
Test case for connect_get_namespaced_pod_proxy_with_path

test_connect_get_namespaced_service_proxy()
Test case for connect_get_namespaced_service_proxy

test_connect_get_namespaced_service_proxy_with_path()
Test case for connect_get_namespaced_service_proxy_with_path

test_connect_get_node_proxy()
Test case for connect_get_node_proxy

test_connect_get_node_proxy_with_path()
Test case for connect_get_node_proxy_with_path

test_connect_head_namespaced_pod_proxy()
Test case for connect_head_namespaced_pod_proxy

test_connect_head_namespaced_pod_proxy_with_path()
Test case for connect_head_namespaced_pod_proxy_with_path

test_connect_head_namespaced_service_proxy()
Test case for connect_head_namespaced_service_proxy

test_connect_head_namespaced_service_proxy_with_path()
Test case for connect_head_namespaced_service_proxy_with_path

test_connect_head_node_proxy()
Test case for connect_head_node_proxy

test_connect_head_node_proxy_with_path()
Test case for connect_head_node_proxy_with_path

test_connect_options_namespaced_pod_proxy()
Test case for connect_options_namespaced_pod_proxy

test_connect_options_namespaced_pod_proxy_with_path()
Test case for connect_options_namespaced_pod_proxy_with_path

test_connect_options_namespaced_service_proxy()
Test case for connect_options_namespaced_service_proxy

test_connect_options_namespaced_service_proxy_with_path()
Test case for connect_options_namespaced_service_proxy_with_path

test_connect_options_node_proxy()
Test case for connect_options_node_proxy

test_connect_options_node_proxy_with_path()
Test case for connect_options_node_proxy_with_path

test_connect_patch_namespaced_pod_proxy()
Test case for connect_patch_namespaced_pod_proxy

test_connect_patch_namespaced_pod_proxy_with_path()
Test case for connect_patch_namespaced_pod_proxy_with_path

test_connect_patch_namespaced_service_proxy()
Test case for connect_patch_namespaced_service_proxy

test_connect_patch_namespaced_service_proxy_with_path()
Test case for connect_patch_namespaced_service_proxy_with_path

test_connect_patch_node_proxy()
Test case for connect_patch_node_proxy

test_connect_patch_node_proxy_with_path()
Test case for connect_patch_node_proxy_with_path

test_connect_post_namespaced_pod_attach()
Test case for connect_post_namespaced_pod_attach

test_connect_post_namespaced_pod_exec()
Test case for connect_post_namespaced_pod_exec

test_connect_post_namespaced_pod_portforward()
Test case for connect_post_namespaced_pod_portforward

test_connect_post_namespaced_pod_proxy()
Test case for connect_post_namespaced_pod_proxy

test_connect_post_namespaced_pod_proxy_with_path()
Test case for connect_post_namespaced_pod_proxy_with_path

test_connect_post_namespaced_service_proxy()
Test case for connect_post_namespaced_service_proxy

test_connect_post_namespaced_service_proxy_with_path()
Test case for connect_post_namespaced_service_proxy_with_path

test_connect_post_node_proxy()
Test case for connect_post_node_proxy

test_connect_post_node_proxy_with_path()
Test case for connect_post_node_proxy_with_path

test_connect_put_namespaced_pod_proxy()
Test case for connect_put_namespaced_pod_proxy

test_connect_put_namespaced_pod_proxy_with_path()
Test case for connect_put_namespaced_pod_proxy_with_path

test_connect_put_namespaced_service_proxy()
Test case for connect_put_namespaced_service_proxy

test_connect_put_namespaced_service_proxy_with_path()
Test case for connect_put_namespaced_service_proxy_with_path

test_connect_put_node_proxy()
Test case for connect_put_node_proxy

test_connect_put_node_proxy_with_path()
Test case for connect_put_node_proxy_with_path

test_create_namespace()
Test case for create_namespace

test_create_namespaced_binding()
Test case for create_namespaced_binding

test_create_namespaced_config_map()
Test case for create_namespaced_config_map

test_create_namespaced_endpoints()
Test case for create_namespaced_endpoints

test_create_namespaced_event()
Test case for create_namespaced_event

test_create_namespaced_limit_range()
Test case for create_namespaced_limit_range

test_create_namespaced_persistent_volume_claim()
Test case for create_namespaced_persistent_volume_claim

test_create_namespaced_pod()
Test case for create_namespaced_pod

test_create_namespaced_pod_binding()
Test case for create_namespaced_pod_binding

test_create_namespaced_pod_eviction()
Test case for create_namespaced_pod_eviction

test_create_namespaced_pod_template()
Test case for create_namespaced_pod_template

test_create_namespaced_replication_controller()
Test case for create_namespaced_replication_controller

test_create_namespaced_resource_quota()
Test case for create_namespaced_resource_quota

test_create_namespaced_secret()
Test case for create_namespaced_secret

test_create_namespaced_service()
Test case for create_namespaced_service

test_create_namespaced_service_account()
Test case for create_namespaced_service_account

test_create_node()
Test case for create_node

test_create_persistent_volume()
Test case for create_persistent_volume

test_delete_collection_namespaced_config_map()
Test case for delete_collection_namespaced_config_map

test_delete_collection_namespaced_endpoints()
Test case for delete_collection_namespaced_endpoints

test_delete_collection_namespaced_event()
Test case for delete_collection_namespaced_event

test_delete_collection_namespaced_limit_range()
Test case for delete_collection_namespaced_limit_range

test_delete_collection_namespaced_persistent_volume_claim()
Test case for delete_collection_namespaced_persistent_volume_claim

test_delete_collection_namespaced_pod()
Test case for delete_collection_namespaced_pod

test_delete_collection_namespaced_pod_template()
Test case for delete_collection_namespaced_pod_template

test_delete_collection_namespaced_replication_controller()
Test case for delete_collection_namespaced_replication_controller

test_delete_collection_namespaced_resource_quota()
Test case for delete_collection_namespaced_resource_quota

test_delete_collection_namespaced_secret()
Test case for delete_collection_namespaced_secret

test_delete_collection_namespaced_service_account()
Test case for delete_collection_namespaced_service_account

test_delete_collection_node()
Test case for delete_collection_node

test_delete_collection_persistent_volume()
Test case for delete_collection_persistent_volume

test_delete_namespace()
Test case for delete_namespace

test_delete_namespaced_config_map()
Test case for delete_namespaced_config_map

test_delete_namespaced_endpoints()
Test case for delete_namespaced_endpoints

test_delete_namespaced_event()
Test case for delete_namespaced_event

test_delete_namespaced_limit_range()
Test case for delete_namespaced_limit_range

test_delete_namespaced_persistent_volume_claim()
Test case for delete_namespaced_persistent_volume_claim

test_delete_namespaced_pod()
Test case for delete_namespaced_pod

test_delete_namespaced_pod_template()
Test case for delete_namespaced_pod_template

test_delete_namespaced_replication_controller()
Test case for delete_namespaced_replication_controller

test_delete_namespaced_resource_quota()
Test case for delete_namespaced_resource_quota

test_delete_namespaced_secret()
Test case for delete_namespaced_secret

test_delete_namespaced_service()
Test case for delete_namespaced_service

test_delete_namespaced_service_account()
Test case for delete_namespaced_service_account

test_delete_node()
Test case for delete_node

test_delete_persistent_volume()
Test case for delete_persistent_volume

test_get_api_resources()
Test case for get_api_resources

test_list_component_status()
Test case for list_component_status

test_list_config_map_for_all_namespaces()
Test case for list_config_map_for_all_namespaces

test_list_endpoints_for_all_namespaces()
Test case for list_endpoints_for_all_namespaces

test_list_event_for_all_namespaces()
Test case for list_event_for_all_namespaces

test_list_limit_range_for_all_namespaces()
Test case for list_limit_range_for_all_namespaces

test_list_namespace()
Test case for list_namespace

test_list_namespaced_config_map()
Test case for list_namespaced_config_map

test_list_namespaced_endpoints()
Test case for list_namespaced_endpoints

test_list_namespaced_event()
Test case for list_namespaced_event

test_list_namespaced_limit_range()
Test case for list_namespaced_limit_range

test_list_namespaced_persistent_volume_claim()
Test case for list_namespaced_persistent_volume_claim

test_list_namespaced_pod()
Test case for list_namespaced_pod

test_list_namespaced_pod_template()
Test case for list_namespaced_pod_template

test_list_namespaced_replication_controller()
Test case for list_namespaced_replication_controller

test_list_namespaced_resource_quota()
Test case for list_namespaced_resource_quota

test_list_namespaced_secret()
Test case for list_namespaced_secret

test_list_namespaced_service()
Test case for list_namespaced_service

test_list_namespaced_service_account()
Test case for list_namespaced_service_account

test_list_node()
Test case for list_node

test_list_persistent_volume()
Test case for list_persistent_volume

test_list_persistent_volume_claim_for_all_namespaces()
Test case for list_persistent_volume_claim_for_all_namespaces

test_list_pod_for_all_namespaces()
Test case for list_pod_for_all_namespaces

test_list_pod_template_for_all_namespaces()
Test case for list_pod_template_for_all_namespaces

test_list_replication_controller_for_all_namespaces()
Test case for list_replication_controller_for_all_namespaces

test_list_resource_quota_for_all_namespaces()
Test case for list_resource_quota_for_all_namespaces

test_list_secret_for_all_namespaces()
Test case for list_secret_for_all_namespaces

test_list_service_account_for_all_namespaces()
Test case for list_service_account_for_all_namespaces

test_list_service_for_all_namespaces()
Test case for list_service_for_all_namespaces

test_patch_namespace()
Test case for patch_namespace

test_patch_namespace_status()
Test case for patch_namespace_status

test_patch_namespaced_config_map()
Test case for patch_namespaced_config_map

test_patch_namespaced_endpoints()
Test case for patch_namespaced_endpoints

test_patch_namespaced_event()
Test case for patch_namespaced_event

test_patch_namespaced_limit_range()
Test case for patch_namespaced_limit_range

test_patch_namespaced_persistent_volume_claim()
Test case for patch_namespaced_persistent_volume_claim

test_patch_namespaced_persistent_volume_claim_status()
Test case for patch_namespaced_persistent_volume_claim_status

test_patch_namespaced_pod()
Test case for patch_namespaced_pod

test_patch_namespaced_pod_status()
Test case for patch_namespaced_pod_status

test_patch_namespaced_pod_template()
Test case for patch_namespaced_pod_template

test_patch_namespaced_replication_controller()
Test case for patch_namespaced_replication_controller

test_patch_namespaced_replication_controller_scale()
Test case for patch_namespaced_replication_controller_scale

test_patch_namespaced_replication_controller_status()
Test case for patch_namespaced_replication_controller_status

test_patch_namespaced_resource_quota()
Test case for patch_namespaced_resource_quota

test_patch_namespaced_resource_quota_status()
Test case for patch_namespaced_resource_quota_status

test_patch_namespaced_secret()
Test case for patch_namespaced_secret

test_patch_namespaced_service()
Test case for patch_namespaced_service

test_patch_namespaced_service_account()
Test case for patch_namespaced_service_account

test_patch_namespaced_service_status()
Test case for patch_namespaced_service_status

test_patch_node()
Test case for patch_node

test_patch_node_status()
Test case for patch_node_status

test_patch_persistent_volume()
Test case for patch_persistent_volume

test_patch_persistent_volume_status()
Test case for patch_persistent_volume_status

test_read_component_status()
Test case for read_component_status

test_read_namespace()
Test case for read_namespace

test_read_namespace_status()
Test case for read_namespace_status

test_read_namespaced_config_map()
Test case for read_namespaced_config_map

test_read_namespaced_endpoints()
Test case for read_namespaced_endpoints

test_read_namespaced_event()
Test case for read_namespaced_event

test_read_namespaced_limit_range()
Test case for read_namespaced_limit_range

test_read_namespaced_persistent_volume_claim()
Test case for read_namespaced_persistent_volume_claim

test_read_namespaced_persistent_volume_claim_status()
Test case for read_namespaced_persistent_volume_claim_status

test_read_namespaced_pod()
Test case for read_namespaced_pod

test_read_namespaced_pod_log()
Test case for read_namespaced_pod_log

test_read_namespaced_pod_status()
Test case for read_namespaced_pod_status

test_read_namespaced_pod_template()
Test case for read_namespaced_pod_template

test_read_namespaced_replication_controller()
Test case for read_namespaced_replication_controller

test_read_namespaced_replication_controller_scale()
Test case for read_namespaced_replication_controller_scale

test_read_namespaced_replication_controller_status()
Test case for read_namespaced_replication_controller_status

test_read_namespaced_resource_quota()
Test case for read_namespaced_resource_quota

test_read_namespaced_resource_quota_status()
Test case for read_namespaced_resource_quota_status

test_read_namespaced_secret()
Test case for read_namespaced_secret

test_read_namespaced_service()
Test case for read_namespaced_service

test_read_namespaced_service_account()
Test case for read_namespaced_service_account

test_read_namespaced_service_status()
Test case for read_namespaced_service_status

test_read_node()
Test case for read_node

test_read_node_status()
Test case for read_node_status

test_read_persistent_volume()
Test case for read_persistent_volume

test_read_persistent_volume_status()
Test case for read_persistent_volume_status

test_replace_namespace()
Test case for replace_namespace

test_replace_namespace_finalize()
Test case for replace_namespace_finalize

test_replace_namespace_status()
Test case for replace_namespace_status

test_replace_namespaced_config_map()
Test case for replace_namespaced_config_map

test_replace_namespaced_endpoints()
Test case for replace_namespaced_endpoints

test_replace_namespaced_event()
Test case for replace_namespaced_event

test_replace_namespaced_limit_range()
Test case for replace_namespaced_limit_range

test_replace_namespaced_persistent_volume_claim()
Test case for replace_namespaced_persistent_volume_claim

test_replace_namespaced_persistent_volume_claim_status()
Test case for replace_namespaced_persistent_volume_claim_status

test_replace_namespaced_pod()
Test case for replace_namespaced_pod

test_replace_namespaced_pod_status()
Test case for replace_namespaced_pod_status

test_replace_namespaced_pod_template()
Test case for replace_namespaced_pod_template

test_replace_namespaced_replication_controller()
Test case for replace_namespaced_replication_controller

test_replace_namespaced_replication_controller_scale()
Test case for replace_namespaced_replication_controller_scale

test_replace_namespaced_replication_controller_status()
Test case for replace_namespaced_replication_controller_status

test_replace_namespaced_resource_quota()
Test case for replace_namespaced_resource_quota

test_replace_namespaced_resource_quota_status()
Test case for replace_namespaced_resource_quota_status

test_replace_namespaced_secret()
Test case for replace_namespaced_secret

test_replace_namespaced_service()
Test case for replace_namespaced_service

test_replace_namespaced_service_account()
Test case for replace_namespaced_service_account

test_replace_namespaced_service_status()
Test case for replace_namespaced_service_status

test_replace_node()
Test case for replace_node

test_replace_node_status()
Test case for replace_node_status

test_replace_persistent_volume()
Test case for replace_persistent_volume

test_replace_persistent_volume_status()
Test case for replace_persistent_volume_status

kubernetes.test.test_extensions_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_extensions_api.**TestExtensionsApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

ExtensionsApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_extensions_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_extensions_v1beta1_api.**TestExtensionsV1beta1Api**(*methodName='runTest*
Bases: unittest.case.TestCase

ExtensionsV1beta1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_namespaced_daemon_set()
Test case for create_namespaced_daemon_set

test_create_namespaced_deployment()
Test case for create_namespaced_deployment

test_create_namespaced_deployment_rollback()
Test case for create_namespaced_deployment_rollback

test_create_namespaced_ingress()
Test case for create_namespaced_ingress

test_create_namespaced_network_policy()
Test case for create_namespaced_network_policy

test_create_namespaced_replica_set()
Test case for create_namespaced_replica_set

test_create_pod_security_policy()
Test case for create_pod_security_policy

test_delete_collection_namespaced_daemon_set()
Test case for delete_collection_namespaced_daemon_set

test_delete_collection_namespaced_deployment()
Test case for delete_collection_namespaced_deployment

test_delete_collection_namespaced_ingress()
Test case for delete_collection_namespaced_ingress

test_delete_collection_namespaced_network_policy()
Test case for delete_collection_namespaced_network_policy

test_delete_collection_namespaced_replica_set()
Test case for delete_collection_namespaced_replica_set

test_delete_collection_pod_security_policy()
Test case for delete_collection_pod_security_policy

test_delete_namespaced_daemon_set()
Test case for delete_namespaced_daemon_set

test_delete_namespaced_deployment()
Test case for delete_namespaced_deployment

test_delete_namespaced_ingress()
Test case for delete_namespaced_ingress

test_delete_namespaced_network_policy()
Test case for delete_namespaced_network_policy

test_delete_namespaced_replica_set()
Test case for delete_namespaced_replica_set

test_delete_pod_security_policy()
Test case for delete_pod_security_policy

test_get_api_resources()
Test case for get_api_resources

test_list_daemon_set_for_all_namespaces()
Test case for list_daemon_set_for_all_namespaces

test_list_deployment_for_all_namespaces()
Test case for list_deployment_for_all_namespaces

test_list_ingress_for_all_namespaces()
Test case for list_ingress_for_all_namespaces

test_list_namespaced_daemon_set()
Test case for list_namespaced_daemon_set

test_list_namespaced_deployment()
Test case for list_namespaced_deployment

test_list_namespaced_ingress()
Test case for list_namespaced_ingress

test_list_namespaced_network_policy()
Test case for list_namespaced_network_policy

test_list_namespaced_replica_set()
Test case for list_namespaced_replica_set

test_list_network_policy_for_all_namespaces()
Test case for list_network_policy_for_all_namespaces

test_list_pod_security_policy()
Test case for list_pod_security_policy

test_list_replica_set_for_all_namespaces()
Test case for list_replica_set_for_all_namespaces

test_patch_namespaced_daemon_set()
Test case for patch_namespaced_daemon_set

test_patch_namespaced_daemon_set_status()
Test case for patch_namespaced_daemon_set_status

test_patch_namespaced_deployment()
Test case for patch_namespaced_deployment

test_patch_namespaced_deployment_scale()
Test case for patch_namespaced_deployment_scale

test_patch_namespaced_deployment_status()
Test case for patch_namespaced_deployment_status

test_patch_namespaced_ingress()
Test case for patch_namespaced_ingress

test_patch_namespaced_ingress_status()
Test case for patch_namespaced_ingress_status

test_patch_namespaced_network_policy()
Test case for patch_namespaced_network_policy

test_patch_namespaced_replica_set()
Test case for patch_namespaced_replica_set

test_patch_namespaced_replica_set_scale()
Test case for patch_namespaced_replica_set_scale

test_patch_namespaced_replica_set_status()
Test case for patch_namespaced_replica_set_status

test_patch_namespaced_replication_controller_dummy_scale()
Test case for patch_namespaced_replication_controller_dummy_scale

test_patch_pod_security_policy()
Test case for patch_pod_security_policy

test_read_namespaced_daemon_set()
Test case for read_namespaced_daemon_set

test_read_namespaced_daemon_set_status()
Test case for read_namespaced_daemon_set_status

test_read_namespaced_deployment()
Test case for read_namespaced_deployment

test_read_namespaced_deployment_scale()
Test case for read_namespaced_deployment_scale

test_read_namespaced_deployment_status()
Test case for read_namespaced_deployment_status

test_read_namespaced_ingress()
Test case for read_namespaced_ingress

test_read_namespaced_ingress_status()
Test case for read_namespaced_ingress_status

test_read_namespaced_network_policy()
Test case for read_namespaced_network_policy

test_read_namespaced_replica_set()
Test case for read_namespaced_replica_set

test_read_namespaced_replica_set_scale()
Test case for read_namespaced_replica_set_scale

test_read_namespaced_replica_set_status()
Test case for read_namespaced_replica_set_status

test_read_namespaced_replication_controller_dummy_scale()
Test case for read_namespaced_replication_controller_dummy_scale

test_read_pod_security_policy()
Test case for read_pod_security_policy

test_replace_namespaced_daemon_set()
Test case for replace_namespaced_daemon_set

test_replace_namespaced_daemon_set_status()
Test case for replace_namespaced_daemon_set_status

test_replace_namespaced_deployment()
Test case for replace_namespaced_deployment

test_replace_namespaced_deployment_scale()
Test case for replace_namespaced_deployment_scale

test_replace_namespaced_deployment_status()
Test case for replace_namespaced_deployment_status

test_replace_namespaced_ingress()
Test case for replace_namespaced_ingress

test_replace_namespaced_ingress_status()
Test case for replace_namespaced_ingress_status

test_replace_namespaced_network_policy()
Test case for replace_namespaced_network_policy

test_replace_namespaced_replica_set()
Test case for replace_namespaced_replica_set

test_replace_namespaced_replica_set_scale()
Test case for replace_namespaced_replica_set_scale

test_replace_namespaced_replica_set_status()
Test case for replace_namespaced_replica_set_status

test_replace_namespaced_replication_controller_dummy_scale()
Test case for replace_namespaced_replication_controller_dummy_scale

test_replace_pod_security_policy()
Test case for replace_pod_security_policy

kubernetes.test.test_intstr_int_or_string module

kubernetes.test.test_logs_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_logs_api.**TestLogsApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

LogsApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_log_file_handler()
Test case for log_file_handler

test_log_file_list_handler()
Test case for log_file_list_handler

kubernetes.test.test_policy_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_policy_api.**TestPolicyApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

PolicyApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_policy_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_policy_v1beta1_api.**TestPolicyV1beta1Api**(*methodName='runTest'*)
Bases: unittest.case.TestCase

PolicyV1beta1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_namespaced_pod_disruption_budget()
Test case for create_namespaced_pod_disruption_budget

test_create_pod_security_policy()
Test case for create_pod_security_policy

test_delete_collection_namespaced_pod_disruption_budget()
Test case for delete_collection_namespaced_pod_disruption_budget

test_delete_collection_pod_security_policy()
Test case for delete_collection_pod_security_policy

test_delete_namespaced_pod_disruption_budget()
Test case for delete_namespaced_pod_disruption_budget

test_delete_pod_security_policy()
Test case for delete_pod_security_policy

test_get_api_resources()
Test case for get_api_resources

test_list_namespaced_pod_disruption_budget()
Test case for list_namespaced_pod_disruption_budget

test_list_pod_disruption_budget_for_all_namespaces()
Test case for list_pod_disruption_budget_for_all_namespaces

test_list_pod_security_policy()
Test case for list_pod_security_policy

test_patch_namespaced_pod_disruption_budget()
Test case for patch_namespaced_pod_disruption_budget

test_patch_namespaced_pod_disruption_budget_status()
Test case for patch_namespaced_pod_disruption_budget_status

test_patch_pod_security_policy()
Test case for patch_pod_security_policy

test_read_namespaced_pod_disruption_budget()
Test case for read_namespaced_pod_disruption_budget

test_read_namespaced_pod_disruption_budget_status()
Test case for read_namespaced_pod_disruption_budget_status

test_read_pod_security_policy()
Test case for read_pod_security_policy

test_replace_namespaced_pod_disruption_budget()
Test case for replace_namespaced_pod_disruption_budget

test_replace_namespaced_pod_disruption_budget_status()
Test case for replace_namespaced_pod_disruption_budget_status

test_replace_pod_security_policy()
Test case for replace_pod_security_policy

kubernetes.test.test_rbac_authorization_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_rbac_authorization_api.**TestRbacAuthorizationApi**(*methodName='runTest*
Bases: unittest.case.TestCase

RbacAuthorizationApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_rbac_authorization_v1alpha1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_rbac_authorization_v1alpha1_api.**TestRbacAuthorizationV1alpha1Api**
Bases: unittest.case.TestCase

RbacAuthorizationV1alpha1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_cluster_role()
Test case for create_cluster_role

test_create_cluster_role_binding()
Test case for create_cluster_role_binding

test_create_namespaced_role()
Test case for create_namespaced_role

test_create_namespaced_role_binding()
Test case for create_namespaced_role_binding

test_delete_cluster_role()
Test case for delete_cluster_role

test_delete_cluster_role_binding()
Test case for delete_cluster_role_binding

test_delete_collection_cluster_role()
Test case for delete_collection_cluster_role

test_delete_collection_cluster_role_binding()
Test case for delete_collection_cluster_role_binding

test_delete_collection_namespaced_role()
Test case for delete_collection_namespaced_role

test_delete_collection_namespaced_role_binding()
Test case for delete_collection_namespaced_role_binding

test_delete_namespaced_role()
Test case for delete_namespaced_role

test_delete_namespaced_role_binding()
Test case for delete_namespaced_role_binding

test_get_api_resources()
Test case for get_api_resources

test_list_cluster_role()
Test case for list_cluster_role

test_list_cluster_role_binding()
Test case for list_cluster_role_binding

test_list_namespaced_role()
Test case for list_namespaced_role

test_list_namespaced_role_binding()
Test case for list_namespaced_role_binding

test_list_role_binding_for_all_namespaces()
Test case for list_role_binding_for_all_namespaces

test_list_role_for_all_namespaces()
Test case for list_role_for_all_namespaces

test_patch_cluster_role()
Test case for patch_cluster_role

test_patch_cluster_role_binding()
Test case for patch_cluster_role_binding

test_patch_namespaced_role()
Test case for patch_namespaced_role

test_patch_namespaced_role_binding()
Test case for patch_namespaced_role_binding

test_read_cluster_role()
Test case for read_cluster_role

test_read_cluster_role_binding()
Test case for read_cluster_role_binding

test_read_namespaced_role()
Test case for read_namespaced_role

test_read_namespaced_role_binding()
Test case for read_namespaced_role_binding

test_replace_cluster_role()
Test case for replace_cluster_role

test_replace_cluster_role_binding()
Test case for replace_cluster_role_binding

test_replace_namespaced_role()
Test case for replace_namespaced_role

test_replace_namespaced_role_binding()
Test case for replace_namespaced_role_binding

kubernetes.test.test_resource_quantity module

kubernetes.test.test_runtime_raw_extension module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_runtime_raw_extension.**TestRuntimeRawExtension**(*methodName='runTest'*)
Bases: unittest.case.TestCase

RuntimeRawExtension unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testRuntimeRawExtension()
Test RuntimeRawExtension

kubernetes.test.test_storage_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_storage_api.**TestStorageApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

StorageApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_api_group()
Test case for get_api_group

kubernetes.test.test_storage_v1beta1_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_storage_v1beta1_api.**TestStorageV1beta1Api**(*methodName='runTest'*)
Bases: unittest.case.TestCase

StorageV1beta1Api unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_create_storage_class()
Test case for create_storage_class

test_create_volume_attachment()
Test case for create_volume_attachment

test_delete_collection_storage_class()
Test case for delete_collection_storage_class

test_delete_collection_volume_attachment()
Test case for delete_collection_volume_attachment

test_delete_storage_class()
Test case for delete_storage_class

test_delete_volume_attachment()
Test case for delete_volume_attachment

test_get_api_resources()
Test case for get_api_resources

test_list_storage_class()
Test case for list_storage_class

test_list_volume_attachment()
Test case for list_volume_attachment

test_patch_storage_class()
Test case for patch_storage_class

test_patch_volume_attachment()
Test case for patch_volume_attachment

test_read_storage_class()
Test case for read_storage_class

test_read_volume_attachment()
Test case for read_volume_attachment

test_replace_storage_class()
Test case for replace_storage_class

test_replace_volume_attachment()
Test case for replace_volume_attachment

kubernetes.test.test_unversioned_api_group module

kubernetes.test.test_unversioned_api_group_list module

kubernetes.test.test_unversioned_api_resource module

kubernetes.test.test_unversioned_api_resource_list module

kubernetes.test.test_unversioned_api_versions module

kubernetes.test.test_unversioned_group_version_for_discovery module

kubernetes.test.test_unversioned_label_selector module

kubernetes.test.test_unversioned_label_selector_requirement module

kubernetes.test.test_unversioned_list_meta module

kubernetes.test.test_unversioned_server_address_by_client_cidr module

kubernetes.test.test_unversioned_status module

kubernetes.test.test_unversioned_status_cause module

kubernetes.test.test_unversioned_status_details module

kubernetes.test.test_unversioned_time module

kubernetes.test.test_v1_attached_volume module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_attached_volume.**TestV1AttachedVolume**(*methodName='runTest'*)

Bases: unittest.case.TestCase

V1AttachedVolume unit test stubs

setUp()

Hook method for setting up the test fixture before exercising it.

tearDown()

Hook method for deconstructing the test fixture after testing it.

testV1AttachedVolume()

Test V1AttachedVolume

kubernetes.test.test_v1_aws_elastic_block_store_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_aws_elastic_block_store_volume_source.**TestV1AWSElasticBlockSt**
Bases: unittest.case.TestCase

V1AWSElasticBlockStoreVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1AWSElasticBlockStoreVolumeSource()
Test V1AWSElasticBlockStoreVolumeSource

kubernetes.test.test_v1_azure_disk_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_azure_disk_volume_source.**TestV1AzureDiskVolumeSource**(*methodNa*
Bases: unittest.case.TestCase

V1AzureDiskVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1AzureDiskVolumeSource()
Test V1AzureDiskVolumeSource

kubernetes.test.test_v1_azure_file_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_azure_file_volume_source.**TestV1AzureFileVolumeSource**(*methodNa*
Bases: unittest.case.TestCase

V1AzureFileVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1AzureFileVolumeSource()
Test V1AzureFileVolumeSource

kubernetes.test.test_v1_binding module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_binding.**TestV1Binding**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Binding unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Binding()
Test V1Binding

kubernetes.test.test_v1_capabilities module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_capabilities.**TestV1Capabilities**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Capabilities unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Capabilities()
Test V1Capabilities

kubernetes.test.test_v1_ceph_fs_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_ceph_fs_volume_source.**TestV1CephFSVolumeSource**(*methodName='runT*
Bases: unittest.case.TestCase

V1CephFSVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1CephFSVolumeSource()
Test V1CephFSVolumeSource

kubernetes.test.test_v1_cinder_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_cinder_volume_source.**TestV1CinderVolumeSource**(*methodName='runTe*
Bases: unittest.case.TestCase

V1CinderVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1CinderVolumeSource()
Test V1CinderVolumeSource

kubernetes.test.test_v1_component_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_component_condition.**TestV1ComponentCondition**(*methodName='runTest*
Bases: unittest.case.TestCase

V1ComponentCondition unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ComponentCondition()
Test V1ComponentCondition

kubernetes.test.test_v1_component_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_component_status.**TestV1ComponentStatus**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ComponentStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ComponentStatus()
Test V1ComponentStatus

kubernetes.test.test_v1_component_status_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_component_status_list.**TestV1ComponentStatusList**(*methodName='ru*
Bases: unittest.case.TestCase

V1ComponentStatusList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ComponentStatusList()
Test V1ComponentStatusList

kubernetes.test.test_v1_config_map module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_config_map.**TestV1ConfigMap**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ConfigMap unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ConfigMap()
Test V1ConfigMap

kubernetes.test.test_v1_config_map_key_selector module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_config_map_key_selector.**TestV1ConfigMapKeySelector**(*methodName=*
Bases: unittest.case.TestCase

V1ConfigMapKeySelector unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ConfigMapKeySelector()
Test V1ConfigMapKeySelector

kubernetes.test.test_v1_config_map_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_config_map_list.**TestV1ConfigMapList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ConfigMapList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ConfigMapList()
Test V1ConfigMapList

kubernetes.test.test_v1_config_map_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_config_map_volume_source.**TestV1ConfigMapVolumeSource**(*methodNa*
Bases: unittest.case.TestCase

V1ConfigMapVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ConfigMapVolumeSource()
Test V1ConfigMapVolumeSource

kubernetes.test.test_v1_container module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_container.**TestV1Container**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Container unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Container()
Test V1Container

kubernetes.test.test_v1_container_image module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_container_image.**TestV1ContainerImage**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ContainerImage unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ContainerImage()
Test V1ContainerImage

kubernetes.test.test_v1_container_port module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_container_port.**TestV1ContainerPort**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ContainerPort unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ContainerPort()
Test V1ContainerPort

kubernetes.test.test_v1_container_state module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_container_state.**TestV1ContainerState**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ContainerState unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ContainerState()
Test V1ContainerState

kubernetes.test.test_v1_container_state_running module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_container_state_running.**TestV1ContainerStateRunning**(*methodNam*
Bases: unittest.case.TestCase

V1ContainerStateRunning unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ContainerStateRunning()
Test V1ContainerStateRunning

kubernetes.test.test_v1_container_state_terminated module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_container_state_terminated.**TestV1ContainerStateTerminated**(*met*
Bases: unittest.case.TestCase

V1ContainerStateTerminated unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ContainerStateTerminated()
Test V1ContainerStateTerminated

kubernetes.test.test_v1_container_state_waiting module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_container_state_waiting.**TestV1ContainerStateWaiting**(*methodName*
Bases: unittest.case.TestCase

V1ContainerStateWaiting unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ContainerStateWaiting()
Test V1ContainerStateWaiting

kubernetes.test.test_v1_container_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_container_status.**TestV1ContainerStatus**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ContainerStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ContainerStatus()
Test V1ContainerStatus

kubernetes.test.test_v1_cross_version_object_reference module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_cross_version_object_reference.**TestV1CrossVersionObjectRefere**
Bases: unittest.case.TestCase

V1CrossVersionObjectReference unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1CrossVersionObjectReference()
Test V1CrossVersionObjectReference

kubernetes.test.test_v1_daemon_endpoint module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_daemon_endpoint.**TestV1DaemonEndpoint**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1DaemonEndpoint unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1DaemonEndpoint()
Test V1DaemonEndpoint

kubernetes.test.test_v1_delete_options module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_delete_options.**TestV1DeleteOptions**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1DeleteOptions unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1DeleteOptions()
Test V1DeleteOptions

kubernetes.test.test_v1_downward_api_volume_file module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_downward_api_volume_file.**TestV1DownwardAPIVolumeFile**(*methodNa*
Bases: unittest.case.TestCase

V1DownwardAPIVolumeFile unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1DownwardAPIVolumeFile()
Test V1DownwardAPIVolumeFile

kubernetes.test.test_v1_downward_api_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_downward_api_volume_source.**TestV1DownwardAPIVolumeSource**(*meth*
Bases: unittest.case.TestCase

V1DownwardAPIVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1DownwardAPIVolumeSource()
Test V1DownwardAPIVolumeSource

kubernetes.test.test_v1_empty_dir_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_empty_dir_volume_source.**TestV1EmptyDirVolumeSource**(*methodName=*
Bases: unittest.case.TestCase

V1EmptyDirVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EmptyDirVolumeSource()
Test V1EmptyDirVolumeSource

kubernetes.test.test_v1_endpoint_address module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_endpoint_address.**TestV1EndpointAddress**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1EndpointAddress unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EndpointAddress()
Test V1EndpointAddress

kubernetes.test.test_v1_endpoint_port module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_endpoint_port.**TestV1EndpointPort**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1EndpointPort unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EndpointPort()
Test V1EndpointPort

kubernetes.test.test_v1_endpoint_subset module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_endpoint_subset.**TestV1EndpointSubset**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1EndpointSubset unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EndpointSubset()
Test V1EndpointSubset

kubernetes.test.test_v1_endpoints module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_endpoints.**TestV1Endpoints**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Endpoints unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Endpoints()
Test V1Endpoints

kubernetes.test.test_v1_endpoints_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_endpoints_list.**TestV1EndpointsList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1EndpointsList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EndpointsList()
Test V1EndpointsList

kubernetes.test.test_v1_env_var module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_env_var.**TestV1EnvVar**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1EnvVar unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EnvVar()
Test V1EnvVar

kubernetes.test.test_v1_env_var_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_env_var_source.**TestV1EnvVarSource**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1EnvVarSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EnvVarSource()
Test V1EnvVarSource

kubernetes.test.test_v1_event module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_event.**TestV1Event**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Event unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Event()
Test V1Event

kubernetes.test.test_v1_event_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_event_list.**TestV1EventList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1EventList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EventList()
Test V1EventList

kubernetes.test.test_v1_event_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_event_source.**TestV1EventSource**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1EventSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1EventSource()
Test V1EventSource

kubernetes.test.test_v1_exec_action module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_exec_action.**TestV1ExecAction**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ExecAction unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ExecAction()
Test V1ExecAction

kubernetes.test.test_v1_fc_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_fc_volume_source.**TestV1FCVolumeSource**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1FCVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1FCVolumeSource()
Test V1FCVolumeSource

kubernetes.test.test_v1_flex_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_flex_volume_source.**TestV1FlexVolumeSource**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1FlexVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1FlexVolumeSource()
Test V1FlexVolumeSource

kubernetes.test.test_v1_flocker_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_flocker_volume_source.**TestV1FlockerVolumeSource**(*methodName='ru*
Bases: unittest.case.TestCase

V1FlockerVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1FlockerVolumeSource()
Test V1FlockerVolumeSource

kubernetes.test.test_v1_gce_persistent_disk_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_gce_persistent_disk_volume_source.**TestV1GCEPersistentDiskVolu**
Bases: unittest.case.TestCase

V1GCEPersistentDiskVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1GCEPersistentDiskVolumeSource()
Test V1GCEPersistentDiskVolumeSource

kubernetes.test.test_v1_git_repo_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_git_repo_volume_source.**TestV1GitRepoVolumeSource**(*methodName='r*
Bases: unittest.case.TestCase

V1GitRepoVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1GitRepoVolumeSource()
Test V1GitRepoVolumeSource

kubernetes.test.test_v1_glusterfs_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_glusterfs_volume_source.**TestV1GlusterfsVolumeSource**(*methodNam*
Bases: unittest.case.TestCase

V1GlusterfsVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1GlusterfsVolumeSource()
Test V1GlusterfsVolumeSource

kubernetes.test.test_v1_handler module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_handler.**TestV1Handler**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Handler unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Handler()
Test V1Handler

kubernetes.test.test_v1_horizontal_pod_autoscaler module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_horizontal_pod_autoscaler.**TestV1HorizontalPodAutoscaler**(*metho*
Bases: unittest.case.TestCase

V1HorizontalPodAutoscaler unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1HorizontalPodAutoscaler()
Test V1HorizontalPodAutoscaler

kubernetes.test.test_v1_horizontal_pod_autoscaler_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_horizontal_pod_autoscaler_list.**TestV1HorizontalPodAutoscalerL**
Bases: unittest.case.TestCase

V1HorizontalPodAutoscalerList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1HorizontalPodAutoscalerList()
Test V1HorizontalPodAutoscalerList

kubernetes.test.test_v1_horizontal_pod_autoscaler_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_horizontal_pod_autoscaler_spec.**TestV1HorizontalPodAutoscaler**
Bases: unittest.case.TestCase

V1HorizontalPodAutoscalerSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1HorizontalPodAutoscalerSpec()
Test V1HorizontalPodAutoscalerSpec

kubernetes.test.test_v1_horizontal_pod_autoscaler_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_horizontal_pod_autoscaler_status.**TestV1HorizontalPodAutoscale**
Bases: unittest.case.TestCase

V1HorizontalPodAutoscalerStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1HorizontalPodAutoscalerStatus()
Test V1HorizontalPodAutoscalerStatus

kubernetes.test.test_v1_host_path_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_host_path_volume_source.**TestV1HostPathVolumeSource**(*methodName=*
Bases: unittest.case.TestCase

V1HostPathVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1HostPathVolumeSource()
Test V1HostPathVolumeSource

kubernetes.test.test_v1_http_get_action module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_http_get_action.**TestV1HTTPGetAction**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1HTTPGetAction unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1HTTPGetAction()
Test V1HTTPGetAction

kubernetes.test.test_v1_http_header module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_http_header.**TestV1HTTPHeader**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1HTTPHeader unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1HTTPHeader()
Test V1HTTPHeader

kubernetes.test.test_v1_iscsi_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_iscsi_volume_source.**TestV1ISCSIVolumeSource**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ISCSIVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ISCSIVolumeSource()
Test V1ISCSIVolumeSource

kubernetes.test.test_v1_job module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_job.**TestV1Job**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Job unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Job()
Test V1Job

kubernetes.test.test_v1_job_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_job_condition.**TestV1JobCondition**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1JobCondition unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1JobCondition()
Test V1JobCondition

kubernetes.test.test_v1_job_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_job_list.**TestV1JobList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1JobList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1JobList()
Test V1JobList

kubernetes.test.test_v1_job_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_job_spec.**TestV1JobSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1JobSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1JobSpec()
Test V1JobSpec

kubernetes.test.test_v1_job_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_job_status.**TestV1JobStatus**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1JobStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1JobStatus()
Test V1JobStatus

kubernetes.test.test_v1_key_to_path module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_key_to_path.**TestV1KeyToPath**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1KeyToPath unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1KeyToPath()
Test V1KeyToPath

kubernetes.test.test_v1_lifecycle module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_lifecycle.**TestV1Lifecycle**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Lifecycle unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Lifecycle()
Test V1Lifecycle

kubernetes.test.test_v1_limit_range module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_limit_range.**TestV1LimitRange**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1LimitRange unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1LimitRange()
Test V1LimitRange

kubernetes.test.test_v1_limit_range_item module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_limit_range_item.**TestV1LimitRangeItem**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1LimitRangeItem unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1LimitRangeItem()
Test V1LimitRangeItem

kubernetes.test.test_v1_limit_range_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_limit_range_list.**TestV1LimitRangeList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1LimitRangeList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1LimitRangeList()
Test V1LimitRangeList

kubernetes.test.test_v1_limit_range_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_limit_range_spec.**TestV1LimitRangeSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1LimitRangeSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1LimitRangeSpec()
Test V1LimitRangeSpec

kubernetes.test.test_v1_load_balancer_ingress module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_load_balancer_ingress.**TestV1LoadBalancerIngress**(*methodName='ru*
Bases: unittest.case.TestCase

V1LoadBalancerIngress unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1LoadBalancerIngress()
Test V1LoadBalancerIngress

kubernetes.test.test_v1_load_balancer_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_load_balancer_status.**TestV1LoadBalancerStatus**(*methodName='runTe*
Bases: unittest.case.TestCase

V1LoadBalancerStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1LoadBalancerStatus()
Test V1LoadBalancerStatus

kubernetes.test.test_v1_local_object_reference module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_local_object_reference.**TestV1LocalObjectReference**(*methodName=*
Bases: unittest.case.TestCase

V1LocalObjectReference unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1LocalObjectReference()
Test V1LocalObjectReference

kubernetes.test.test_v1_namespace module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_namespace.**TestV1Namespace**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Namespace unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Namespace()
Test V1Namespace

kubernetes.test.test_v1_namespace_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_namespace_list.**TestV1NamespaceList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NamespaceList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NamespaceList()
Test V1NamespaceList

kubernetes.test.test_v1_namespace_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_namespace_spec.**TestV1NamespaceSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NamespaceSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NamespaceSpec()
Test V1NamespaceSpec

kubernetes.test.test_v1_namespace_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_namespace_status.**TestV1NamespaceStatus**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NamespaceStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NamespaceStatus()
Test V1NamespaceStatus

kubernetes.test.test_v1_nfs_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_nfs_volume_source.**TestV1NFSVolumeSource**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NFSVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NFSVolumeSource()
Test V1NFSVolumeSource

kubernetes.test.test_v1_node module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_node.**TestV1Node**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Node unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Node()
Test V1Node

kubernetes.test.test_v1_node_address module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_node_address.**TestV1NodeAddress**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NodeAddress unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NodeAddress()
Test V1NodeAddress

kubernetes.test.test_v1_node_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_node_condition.**TestV1NodeCondition**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NodeCondition unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NodeCondition()
Test V1NodeCondition

kubernetes.test.test_v1_node_daemon_endpoints module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_node_daemon_endpoints.**TestV1NodeDaemonEndpoints**(*methodName='ru*
Bases: unittest.case.TestCase

V1NodeDaemonEndpoints unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NodeDaemonEndpoints()
Test V1NodeDaemonEndpoints

kubernetes.test.test_v1_node_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_node_list.**TestV1NodeList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NodeList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NodeList()
Test V1NodeList

kubernetes.test.test_v1_node_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_node_spec.**TestV1NodeSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NodeSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NodeSpec()
Test V1NodeSpec

kubernetes.test.test_v1_node_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_node_status.**TestV1NodeStatus**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NodeStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NodeStatus()
Test V1NodeStatus

kubernetes.test.test_v1_node_system_info module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_node_system_info.**TestV1NodeSystemInfo**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1NodeSystemInfo unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1NodeSystemInfo()
Test V1NodeSystemInfo

kubernetes.test.test_v1_object_field_selector module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_object_field_selector.**TestV1ObjectFieldSelector**(*methodName='ru*
Bases: unittest.case.TestCase

V1ObjectFieldSelector unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ObjectFieldSelector()
Test V1ObjectFieldSelector

kubernetes.test.test_v1_object_meta module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_object_meta.**TestV1ObjectMeta**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ObjectMeta unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ObjectMeta()
Test V1ObjectMeta

kubernetes.test.test_v1_object_reference module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_object_reference.**TestV1ObjectReference**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ObjectReference unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ObjectReference()
Test V1ObjectReference

kubernetes.test.test_v1_owner_reference module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_owner_reference.**TestV1OwnerReference**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1OwnerReference unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1OwnerReference()
Test V1OwnerReference

kubernetes.test.test_v1_persistent_volume module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume.**TestV1PersistentVolume**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1PersistentVolume unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolume()
Test V1PersistentVolume

kubernetes.test.test_v1_persistent_volume_claim module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume_claim.**TestV1PersistentVolumeClaim**(*methodName*
Bases: unittest.case.TestCase

V1PersistentVolumeClaim unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolumeClaim()
Test V1PersistentVolumeClaim

kubernetes.test.test_v1_persistent_volume_claim_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume_claim_list.**TestV1PersistentVolumeClaimList**
Bases: unittest.case.TestCase

V1PersistentVolumeClaimList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolumeClaimList()
Test V1PersistentVolumeClaimList

kubernetes.test.test_v1_persistent_volume_claim_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume_claim_spec.**TestV1PersistentVolumeClaimSpec**
Bases: unittest.case.TestCase

V1PersistentVolumeClaimSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolumeClaimSpec()
Test V1PersistentVolumeClaimSpec

kubernetes.test.test_v1_persistent_volume_claim_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume_claim_status.**TestV1PersistentVolumeClaimSta**
Bases: unittest.case.TestCase

V1PersistentVolumeClaimStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolumeClaimStatus()
Test V1PersistentVolumeClaimStatus

kubernetes.test.test_v1_persistent_volume_claim_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume_claim_volume_source.**TestV1PersistentVolumeC**
Bases: unittest.case.TestCase

V1PersistentVolumeClaimVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolumeClaimVolumeSource()
Test V1PersistentVolumeClaimVolumeSource

kubernetes.test.test_v1_persistent_volume_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume_list.**TestV1PersistentVolumeList**(*methodName='*
Bases: unittest.case.TestCase

V1PersistentVolumeList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolumeList()
Test V1PersistentVolumeList

kubernetes.test.test_v1_persistent_volume_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume_spec.**TestV1PersistentVolumeSpec**(*methodName='*
Bases: unittest.case.TestCase

V1PersistentVolumeSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolumeSpec()
Test V1PersistentVolumeSpec

kubernetes.test.test_v1_persistent_volume_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_persistent_volume_status.**TestV1PersistentVolumeStatus**(*methodN*
Bases: unittest.case.TestCase

V1PersistentVolumeStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PersistentVolumeStatus()
Test V1PersistentVolumeStatus

kubernetes.test.test_v1_photon_persistent_disk_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_photon_persistent_disk_volume_source.**TestV1PhotonPersistentD**
Bases: unittest.case.TestCase

V1PhotonPersistentDiskVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PhotonPersistentDiskVolumeSource()
Test V1PhotonPersistentDiskVolumeSource

kubernetes.test.test_v1_pod module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod.**TestV1Pod**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Pod unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Pod()
Test V1Pod

kubernetes.test.test_v1_pod_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod_condition.**TestV1PodCondition**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1PodCondition unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PodCondition()
Test V1PodCondition

kubernetes.test.test_v1_pod_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod_list.**TestV1PodList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1PodList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PodList()
Test V1PodList

kubernetes.test.test_v1_pod_security_context module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod_security_context.**TestV1PodSecurityContext**(*methodName='runTe*
Bases: unittest.case.TestCase

V1PodSecurityContext unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PodSecurityContext()
Test V1PodSecurityContext

kubernetes.test.test_v1_pod_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod_spec.**TestV1PodSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1PodSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PodSpec()
Test V1PodSpec

kubernetes.test.test_v1_pod_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod_status.**TestV1PodStatus**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1PodStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PodStatus()
Test V1PodStatus

kubernetes.test.test_v1_pod_template module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod_template.**TestV1PodTemplate**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1PodTemplate unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PodTemplate()
Test V1PodTemplate

kubernetes.test.test_v1_pod_template_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod_template_list.**TestV1PodTemplateList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1PodTemplateList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PodTemplateList()
Test V1PodTemplateList

kubernetes.test.test_v1_pod_template_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_pod_template_spec.**TestV1PodTemplateSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1PodTemplateSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1PodTemplateSpec()
Test V1PodTemplateSpec

kubernetes.test.test_v1_preconditions module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_preconditions.**TestV1Preconditions**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Preconditions unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Preconditions()
Test V1Preconditions

kubernetes.test.test_v1_probe module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_probe.**TestV1Probe**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Probe unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Probe()
Test V1Probe

kubernetes.test.test_v1_quobyte_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_quobyte_volume_source.**TestV1QuobyteVolumeSource**(*methodName='ru*
Bases: unittest.case.TestCase

V1QuobyteVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1QuobyteVolumeSource()
Test V1QuobyteVolumeSource

kubernetes.test.test_v1_rbd_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_rbd_volume_source.**TestV1RBDVolumeSource**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1RBDVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1RBDVolumeSource()
Test V1RBDVolumeSource

kubernetes.test.test_v1_replication_controller module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_replication_controller.**TestV1ReplicationController**(*methodName=*
Bases: unittest.case.TestCase

V1ReplicationController unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ReplicationController()
Test V1ReplicationController

kubernetes.test.test_v1_replication_controller_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_replication_controller_condition.**TestV1ReplicationControllerC**
Bases: unittest.case.TestCase

V1ReplicationControllerCondition unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ReplicationControllerCondition()
Test V1ReplicationControllerCondition

kubernetes.test.test_v1_replication_controller_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_replication_controller_list.**TestV1ReplicationControllerList**(*
Bases: unittest.case.TestCase

V1ReplicationControllerList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ReplicationControllerList()
Test V1ReplicationControllerList

kubernetes.test.test_v1_replication_controller_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_replication_controller_spec.**TestV1ReplicationControllerSpec**(*
Bases: unittest.case.TestCase

V1ReplicationControllerSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ReplicationControllerSpec()
Test V1ReplicationControllerSpec

kubernetes.test.test_v1_replication_controller_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_replication_controller_status.**TestV1ReplicationControllerStat
Bases: unittest.case.TestCase

V1ReplicationControllerStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ReplicationControllerStatus()
Test V1ReplicationControllerStatus

kubernetes.test.test_v1_resource_field_selector module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_resource_field_selector.**TestV1ResourceFieldSelector**(*methodNam*
Bases: unittest.case.TestCase

V1ResourceFieldSelector unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ResourceFieldSelector()
Test V1ResourceFieldSelector

kubernetes.test.test_v1_resource_quota module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_resource_quota.**TestV1ResourceQuota**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ResourceQuota unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ResourceQuota()
Test V1ResourceQuota

kubernetes.test.test_v1_resource_quota_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_resource_quota_list.**TestV1ResourceQuotaList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ResourceQuotaList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ResourceQuotaList()
Test V1ResourceQuotaList

kubernetes.test.test_v1_resource_quota_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_resource_quota_spec.**TestV1ResourceQuotaSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ResourceQuotaSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ResourceQuotaSpec()
Test V1ResourceQuotaSpec

kubernetes.test.test_v1_resource_quota_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_resource_quota_status.**TestV1ResourceQuotaStatus**(*methodName='run*
Bases: unittest.case.TestCase

V1ResourceQuotaStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ResourceQuotaStatus()
Test V1ResourceQuotaStatus

kubernetes.test.test_v1_resource_requirements module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_resource_requirements.**TestV1ResourceRequirements**(*methodName='r*
Bases: unittest.case.TestCase

V1ResourceRequirements unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ResourceRequirements()
Test V1ResourceRequirements

kubernetes.test.test_v1_scale module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_scale.**TestV1Scale**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Scale unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Scale()
Test V1Scale

kubernetes.test.test_v1_scale_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_scale_spec.**TestV1ScaleSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ScaleSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ScaleSpec()
Test V1ScaleSpec

kubernetes.test.test_v1_scale_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_scale_status.**TestV1ScaleStatus**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ScaleStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ScaleStatus()
Test V1ScaleStatus

kubernetes.test.test_v1_se_linux_options module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_se_linux_options.**TestV1SELinuxOptions**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1SELinuxOptions unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1SELinuxOptions()
Test V1SELinuxOptions

kubernetes.test.test_v1_secret module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_secret.**TestV1Secret**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Secret unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Secret()
Test V1Secret

kubernetes.test.test_v1_secret_key_selector module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_secret_key_selector.**TestV1SecretKeySelector**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1SecretKeySelector unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1SecretKeySelector()
Test V1SecretKeySelector

kubernetes.test.test_v1_secret_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_secret_list.**TestV1SecretList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1SecretList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1SecretList()
Test V1SecretList

kubernetes.test.test_v1_secret_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_secret_volume_source.**TestV1SecretVolumeSource**(*methodName='runTe*
Bases: unittest.case.TestCase

V1SecretVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1SecretVolumeSource()
Test V1SecretVolumeSource

kubernetes.test.test_v1_security_context module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_security_context.**TestV1SecurityContext**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1SecurityContext unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1SecurityContext()
Test V1SecurityContext

kubernetes.test.test_v1_service module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_service.**TestV1Service**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Service unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Service()
Test V1Service

kubernetes.test.test_v1_service_account module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_service_account.**TestV1ServiceAccount**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ServiceAccount unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ServiceAccount()
Test V1ServiceAccount

kubernetes.test.test_v1_service_account_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_service_account_list.**TestV1ServiceAccountList**(*methodName='runTe*
Bases: unittest.case.TestCase

V1ServiceAccountList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ServiceAccountList()
Test V1ServiceAccountList

kubernetes.test.test_v1_service_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_service_list.**TestV1ServiceList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ServiceList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ServiceList()
Test V1ServiceList

kubernetes.test.test_v1_service_port module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_service_port.**TestV1ServicePort**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ServicePort unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ServicePort()
Test V1ServicePort

kubernetes.test.test_v1_service_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_service_spec.**TestV1ServiceSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ServiceSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ServiceSpec()
Test V1ServiceSpec

kubernetes.test.test_v1_service_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_service_status.**TestV1ServiceStatus**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1ServiceStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1ServiceStatus()
Test V1ServiceStatus

kubernetes.test.test_v1_tcp_socket_action module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_tcp_socket_action.**TestV1TCPSocketAction**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1TCPSocketAction unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1TCPSocketAction()
Test V1TCPSocketAction

kubernetes.test.test_v1_volume module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_volume.**TestV1Volume**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1Volume unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1Volume()
Test V1Volume

kubernetes.test.test_v1_volume_mount module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_volume_mount.**TestV1VolumeMount**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1VolumeMount unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1VolumeMount()
Test V1VolumeMount

kubernetes.test.test_v1_vsphere_virtual_disk_volume_source module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1_vsphere_virtual_disk_volume_source.**TestV1VsphereVirtualDiskV**
Bases: unittest.case.TestCase

V1VsphereVirtualDiskVolumeSource unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1VsphereVirtualDiskVolumeSource()
Test V1VsphereVirtualDiskVolumeSource

kubernetes.test.test_v1alpha1_certificate_signing_request module

kubernetes.test.test_v1alpha1_certificate_signing_request_condition module

kubernetes.test.test_v1alpha1_certificate_signing_request_list module

kubernetes.test.test_v1alpha1_certificate_signing_request_spec module

kubernetes.test.test_v1alpha1_certificate_signing_request_status module

kubernetes.test.test_v1alpha1_cluster_role module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_cluster_role.**TestV1alpha1ClusterRole**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1alpha1ClusterRole unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1ClusterRole()
Test V1alpha1ClusterRole

kubernetes.test.test_v1alpha1_cluster_role_binding module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_cluster_role_binding.**TestV1alpha1ClusterRoleBinding**(*met*
Bases: unittest.case.TestCase

V1alpha1ClusterRoleBinding unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1ClusterRoleBinding()
Test V1alpha1ClusterRoleBinding

kubernetes.test.test_v1alpha1_cluster_role_binding_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_cluster_role_binding_list.**TestV1alpha1ClusterRoleBindi**
Bases: unittest.case.TestCase

V1alpha1ClusterRoleBindingList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1ClusterRoleBindingList()
Test V1alpha1ClusterRoleBindingList

kubernetes.test.test_v1alpha1_cluster_role_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_cluster_role_list.**TestV1alpha1ClusterRoleList**(*methodName*
Bases: unittest.case.TestCase

V1alpha1ClusterRoleList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1ClusterRoleList()
Test V1alpha1ClusterRoleList

kubernetes.test.test_v1alpha1_policy_rule module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_policy_rule.**TestV1alpha1PolicyRule**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1alpha1PolicyRule unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1PolicyRule()
Test V1alpha1PolicyRule

kubernetes.test.test_v1alpha1_role module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_role.**TestV1alpha1Role**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1alpha1Role unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1Role()
Test V1alpha1Role

kubernetes.test.test_v1alpha1_role_binding module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_role_binding.**TestV1alpha1RoleBinding**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1alpha1RoleBinding unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1RoleBinding()
Test V1alpha1RoleBinding

kubernetes.test.test_v1alpha1_role_binding_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_role_binding_list.**TestV1alpha1RoleBindingList**(*methodName*
Bases: unittest.case.TestCase

V1alpha1RoleBindingList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1RoleBindingList()
Test V1alpha1RoleBindingList

kubernetes.test.test_v1alpha1_role_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_role_list.**TestV1alpha1RoleList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1alpha1RoleList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1RoleList()
Test V1alpha1RoleList

kubernetes.test.test_v1alpha1_role_ref module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_role_ref.**TestV1alpha1RoleRef**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1alpha1RoleRef unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1RoleRef()
Test V1alpha1RoleRef

kubernetes.test.test_v1alpha1_subject module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1alpha1_subject.**TestV1alpha1Subject**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1alpha1Subject unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1alpha1Subject()
Test V1alpha1Subject

kubernetes.test.test_v1beta1_api_version module

kubernetes.test.test_v1beta1_cpu_target_utilization module

kubernetes.test.test_v1beta1_daemon_set module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_daemon_set.**TestV1beta1DaemonSet**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1DaemonSet unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1DaemonSet()
Test V1beta1DaemonSet

kubernetes.test.test_v1beta1_daemon_set_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_daemon_set_list.**TestV1beta1DaemonSetList**(*methodName='runTe*
Bases: unittest.case.TestCase

V1beta1DaemonSetList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1DaemonSetList()
Test V1beta1DaemonSetList

kubernetes.test.test_v1beta1_daemon_set_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_daemon_set_spec.**TestV1beta1DaemonSetSpec**(*methodName='runTe*
Bases: unittest.case.TestCase

V1beta1DaemonSetSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1DaemonSetSpec()
Test V1beta1DaemonSetSpec

kubernetes.test.test_v1beta1_daemon_set_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_daemon_set_status.**TestV1beta1DaemonSetStatus**(*methodName='*
Bases: unittest.case.TestCase

V1beta1DaemonSetStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1DaemonSetStatus()
Test V1beta1DaemonSetStatus

kubernetes.test.test_v1beta1_deployment module

kubernetes.test.test_v1beta1_deployment_condition module

kubernetes.test.test_v1beta1_deployment_list module

kubernetes.test.test_v1beta1_deployment_rollback module

kubernetes.test.test_v1beta1_deployment_spec module

kubernetes.test.test_v1beta1_deployment_status module

kubernetes.test.test_v1beta1_deployment_strategy module

kubernetes.test.test_v1beta1_eviction module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_eviction.**TestV1beta1Eviction**(*methodName='runTest'*)

Bases: unittest.case.TestCase

V1beta1Eviction unit test stubs

setUp()

Hook method for setting up the test fixture before exercising it.

tearDown()

Hook method for deconstructing the test fixture after testing it.

testV1beta1Eviction()

Test V1beta1Eviction

kubernetes.test.test_v1beta1_horizontal_pod_autoscaler module

kubernetes.test.test_v1beta1_horizontal_pod_autoscaler_list module

kubernetes.test.test_v1beta1_horizontal_pod_autoscaler_spec module

kubernetes.test.test_v1beta1_horizontal_pod_autoscaler_status module

kubernetes.test.test_v1beta1_http_ingress_path module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_http_ingress_path.**TestV1beta1HTTPIngressPath**(*methodName='*
Bases: unittest.case.TestCase

V1beta1HTTPIngressPath unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1HTTPIngressPath()
Test V1beta1HTTPIngressPath

kubernetes.test.test_v1beta1_http_ingress_rule_value module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_http_ingress_rule_value.**TestV1beta1HTTPIngressRuleValue**
Bases: unittest.case.TestCase

V1beta1HTTPIngressRuleValue unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1HTTPIngressRuleValue()
Test V1beta1HTTPIngressRuleValue

kubernetes.test.test_v1beta1_ingress module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_ingress.**TestV1beta1Ingress**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1Ingress unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1Ingress()
Test V1beta1Ingress

kubernetes.test.test_v1beta1_ingress_backend module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_ingress_backend.**TestV1beta1IngressBackend**(*methodName='run1*
Bases: unittest.case.TestCase

V1beta1IngressBackend unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1IngressBackend()
Test V1beta1IngressBackend

kubernetes.test.test_v1beta1_ingress_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_ingress_list.**TestV1beta1IngressList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1IngressList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1IngressList()
Test V1beta1IngressList

kubernetes.test.test_v1beta1_ingress_rule module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_ingress_rule.**TestV1beta1IngressRule**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1IngressRule unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1IngressRule()
Test V1beta1IngressRule

kubernetes.test.test_v1beta1_ingress_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_ingress_spec.**TestV1beta1IngressSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1IngressSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1IngressSpec()
Test V1beta1IngressSpec

kubernetes.test.test_v1beta1_ingress_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_ingress_status.**TestV1beta1IngressStatus**(*methodName='runTest*
Bases: unittest.case.TestCase

V1beta1IngressStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1IngressStatus()
Test V1beta1IngressStatus

kubernetes.test.test_v1beta1_ingress_tls module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_ingress_tls.**TestV1beta1IngressTLS**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1IngressTLS unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1IngressTLS()
Test V1beta1IngressTLS

kubernetes.test.test_v1beta1_job module

kubernetes.test.test_v1beta1_job_condition module

kubernetes.test.test_v1beta1_job_list module

kubernetes.test.test_v1beta1_job_spec module

kubernetes.test.test_v1beta1_job_status module

kubernetes.test.test_v1beta1_local_subject_access_review module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_local_subject_access_review.**TestV1beta1LocalSubjectAcces**
Bases: unittest.case.TestCase

V1beta1LocalSubjectAccessReview unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1LocalSubjectAccessReview()
Test V1beta1LocalSubjectAccessReview

kubernetes.test.test_v1beta1_network_policy module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_network_policy.**TestV1beta1NetworkPolicy**(*methodName='runTest*
Bases: unittest.case.TestCase

V1beta1NetworkPolicy unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1NetworkPolicy()
Test V1beta1NetworkPolicy

kubernetes.test.test_v1beta1_network_policy_ingress_rule module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_network_policy_ingress_rule.**TestV1beta1NetworkPolicyIngr**
Bases: unittest.case.TestCase

V1beta1NetworkPolicyIngressRule unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1NetworkPolicyIngressRule()
Test V1beta1NetworkPolicyIngressRule

kubernetes.test.test_v1beta1_network_policy_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_network_policy_list.**TestV1beta1NetworkPolicyList**(*methodN*
Bases: unittest.case.TestCase

V1beta1NetworkPolicyList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1NetworkPolicyList()
Test V1beta1NetworkPolicyList

kubernetes.test.test_v1beta1_network_policy_peer module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_network_policy_peer.**TestV1beta1NetworkPolicyPeer**(*methodN*
Bases: unittest.case.TestCase

V1beta1NetworkPolicyPeer unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1NetworkPolicyPeer()
Test V1beta1NetworkPolicyPeer

kubernetes.test.test_v1beta1_network_policy_port module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_network_policy_port.**TestV1beta1NetworkPolicyPort**(*methodN*
Bases: unittest.case.TestCase

V1beta1NetworkPolicyPort unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1NetworkPolicyPort()
Test V1beta1NetworkPolicyPort

kubernetes.test.test_v1beta1_network_policy_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_network_policy_spec.**TestV1beta1NetworkPolicySpec**(*methodN*
Bases: unittest.case.TestCase

V1beta1NetworkPolicySpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1NetworkPolicySpec()
Test V1beta1NetworkPolicySpec

kubernetes.test.test_v1beta1_non_resource_attributes module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_non_resource_attributes.**TestV1beta1NonResourceAttributes**
Bases: unittest.case.TestCase

V1beta1NonResourceAttributes unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1NonResourceAttributes()
Test V1beta1NonResourceAttributes

kubernetes.test.test_v1beta1_pod_disruption_budget module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_pod_disruption_budget.**TestV1beta1PodDisruptionBudget**(*met*
Bases: unittest.case.TestCase

V1beta1PodDisruptionBudget unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1PodDisruptionBudget()
Test V1beta1PodDisruptionBudget

kubernetes.test.test_v1beta1_pod_disruption_budget_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_pod_disruption_budget_list.**TestV1beta1PodDisruptionBudge**
Bases: unittest.case.TestCase

V1beta1PodDisruptionBudgetList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1PodDisruptionBudgetList()
Test V1beta1PodDisruptionBudgetList

kubernetes.test.test_v1beta1_pod_disruption_budget_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_pod_disruption_budget_spec.**TestV1beta1PodDisruptionBudge**
Bases: unittest.case.TestCase

V1beta1PodDisruptionBudgetSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1PodDisruptionBudgetSpec()
Test V1beta1PodDisruptionBudgetSpec

kubernetes.test.test_v1beta1_pod_disruption_budget_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_pod_disruption_budget_status.**TestV1beta1PodDisruptionBu**
Bases: unittest.case.TestCase

V1beta1PodDisruptionBudgetStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1PodDisruptionBudgetStatus()
Test V1beta1PodDisruptionBudgetStatus

kubernetes.test.test_v1beta1_replica_set module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_replica_set.**TestV1beta1ReplicaSet**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1ReplicaSet unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1ReplicaSet()
Test V1beta1ReplicaSet

kubernetes.test.test_v1beta1_replica_set_condition module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_replica_set_condition.**TestV1beta1ReplicaSetCondition**(*met*
Bases: unittest.case.TestCase

V1beta1ReplicaSetCondition unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1ReplicaSetCondition()
Test V1beta1ReplicaSetCondition

kubernetes.test.test_v1beta1_replica_set_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_replica_set_list.**TestV1beta1ReplicaSetList**(*methodName='ru*
Bases: unittest.case.TestCase

V1beta1ReplicaSetList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1ReplicaSetList()
Test V1beta1ReplicaSetList

kubernetes.test.test_v1beta1_replica_set_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_replica_set_spec.**TestV1beta1ReplicaSetSpec**(*methodName='ru*
Bases: unittest.case.TestCase

V1beta1ReplicaSetSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1ReplicaSetSpec()
Test V1beta1ReplicaSetSpec

kubernetes.test.test_v1beta1_replica_set_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_replica_set_status.**TestV1beta1ReplicaSetStatus**(*methodName*
Bases: unittest.case.TestCase

V1beta1ReplicaSetStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1ReplicaSetStatus()
Test V1beta1ReplicaSetStatus

kubernetes.test.test_v1beta1_resource_attributes module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_resource_attributes.**TestV1beta1ResourceAttributes**(*methodl*
Bases: unittest.case.TestCase

V1beta1ResourceAttributes unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1ResourceAttributes()
Test V1beta1ResourceAttributes

kubernetes.test.test_v1beta1_rollback_config module

kubernetes.test.test_v1beta1_rolling_update_deployment module

kubernetes.test.test_v1beta1_scale module

kubernetes.test.test_v1beta1_scale_spec module

kubernetes.test.test_v1beta1_scale_status module

kubernetes.test.test_v1beta1_self_subject_access_review module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_self_subject_access_review.**TestV1beta1SelfSubjectAccessR**
Bases: unittest.case.TestCase

V1beta1SelfSubjectAccessReview unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1SelfSubjectAccessReview()
Test V1beta1SelfSubjectAccessReview

kubernetes.test.test_v1beta1_self_subject_access_review_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_self_subject_access_review_spec.**TestV1beta1SelfSubjectAc**
Bases: unittest.case.TestCase

V1beta1SelfSubjectAccessReviewSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1SelfSubjectAccessReviewSpec()
Test V1beta1SelfSubjectAccessReviewSpec

kubernetes.test.test_v1beta1_stateful_set module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_stateful_set.**TestV1beta1StatefulSet**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1StatefulSet unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1StatefulSet()
Test V1beta1StatefulSet

kubernetes.test.test_v1beta1_stateful_set_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_stateful_set_list.**TestV1beta1StatefulSetList**(*methodName='*
Bases: unittest.case.TestCase

V1beta1StatefulSetList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1StatefulSetList()
Test V1beta1StatefulSetList

kubernetes.test.test_v1beta1_stateful_set_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_stateful_set_spec.**TestV1beta1StatefulSetSpec**(*methodName='*
Bases: unittest.case.TestCase

V1beta1StatefulSetSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1StatefulSetSpec()
Test V1beta1StatefulSetSpec

kubernetes.test.test_v1beta1_stateful_set_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_stateful_set_status.**TestV1beta1StatefulSetStatus**(*methodNa*
Bases: unittest.case.TestCase

V1beta1StatefulSetStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1StatefulSetStatus()
Test V1beta1StatefulSetStatus

kubernetes.test.test_v1beta1_storage_class module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_storage_class.**TestV1beta1StorageClass**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1StorageClass unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1StorageClass()
Test V1beta1StorageClass

kubernetes.test.test_v1beta1_storage_class_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_storage_class_list.**TestV1beta1StorageClassList**(*methodName*
Bases: unittest.case.TestCase

V1beta1StorageClassList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1StorageClassList()
Test V1beta1StorageClassList

kubernetes.test.test_v1beta1_subject_access_review module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_subject_access_review.**TestV1beta1SubjectAccessReview**(*met*
Bases: unittest.case.TestCase

V1beta1SubjectAccessReview unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1SubjectAccessReview()
Test V1beta1SubjectAccessReview

kubernetes.test.test_v1beta1_subject_access_review_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_subject_access_review_spec.**TestV1beta1SubjectAccessRevie**
Bases: unittest.case.TestCase

V1beta1SubjectAccessReviewSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1SubjectAccessReviewSpec()
Test V1beta1SubjectAccessReviewSpec

kubernetes.test.test_v1beta1_subject_access_review_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_subject_access_review_status.**TestV1beta1SubjectAccessRev**
Bases: unittest.case.TestCase

V1beta1SubjectAccessReviewStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1SubjectAccessReviewStatus()
Test V1beta1SubjectAccessReviewStatus

kubernetes.test.test_v1beta1_subresource_reference module

kubernetes.test.test_v1beta1_third_party_resource module

kubernetes.test.test_v1beta1_third_party_resource_list module

kubernetes.test.test_v1beta1_token_review module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_token_review.**TestV1beta1TokenReview**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1TokenReview unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1TokenReview()
Test V1beta1TokenReview

kubernetes.test.test_v1beta1_token_review_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_token_review_spec.**TestV1beta1TokenReviewSpec**(*methodName='*
Bases: unittest.case.TestCase

V1beta1TokenReviewSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1TokenReviewSpec()
Test V1beta1TokenReviewSpec

kubernetes.test.test_v1beta1_token_review_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_token_review_status.**TestV1beta1TokenReviewStatus**(*methodN*
Bases: unittest.case.TestCase

V1beta1TokenReviewStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1TokenReviewStatus()
Test V1beta1TokenReviewStatus

kubernetes.test.test_v1beta1_user_info module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v1beta1_user_info.**TestV1beta1UserInfo**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V1beta1UserInfo unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV1beta1UserInfo()
Test V1beta1UserInfo

kubernetes.test.test_v2alpha1_cron_job module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v2alpha1_cron_job.**TestV2alpha1CronJob**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V2alpha1CronJob unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV2alpha1CronJob()
Test V2alpha1CronJob

kubernetes.test.test_v2alpha1_cron_job_list module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v2alpha1_cron_job_list.**TestV2alpha1CronJobList**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V2alpha1CronJobList unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV2alpha1CronJobList()
Test V2alpha1CronJobList

kubernetes.test.test_v2alpha1_cron_job_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v2alpha1_cron_job_spec.**TestV2alpha1CronJobSpec**(*methodName='runTest'*)
Bases: unittest.case.TestCase

V2alpha1CronJobSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV2alpha1CronJobSpec()
Test V2alpha1CronJobSpec

kubernetes.test.test_v2alpha1_cron_job_status module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v2alpha1_cron_job_status.**TestV2alpha1CronJobStatus**(*methodName='ru*
Bases: unittest.case.TestCase

V2alpha1CronJobStatus unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV2alpha1CronJobStatus()
Test V2alpha1CronJobStatus

kubernetes.test.test_v2alpha1_job module

kubernetes.test.test_v2alpha1_job_condition module

kubernetes.test.test_v2alpha1_job_list module

kubernetes.test.test_v2alpha1_job_spec module

kubernetes.test.test_v2alpha1_job_status module

kubernetes.test.test_v2alpha1_job_template_spec module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_v2alpha1_job_template_spec.**TestV2alpha1JobTemplateSpec**(*methodNam*
Bases: unittest.case.TestCase

V2alpha1JobTemplateSpec unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testV2alpha1JobTemplateSpec()
Test V2alpha1JobTemplateSpec

kubernetes.test.test_version_api module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_version_api.**TestVersionApi**(*methodName='runTest'*)
Bases: unittest.case.TestCase

VersionApi unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

test_get_code()
Test case for get_code

kubernetes.test.test_version_info module

Kubernetes

No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)

OpenAPI spec version: v1.10.0

Generated by: https://github.com/swagger-api/swagger-codegen.git

class kubernetes.test.test_version_info.**TestVersionInfo**(*methodName='runTest'*)
Bases: unittest.case.TestCase

VersionInfo unit test stubs

setUp()
Hook method for setting up the test fixture before exercising it.

tearDown()
Hook method for deconstructing the test fixture after testing it.

testVersionInfo()
Test VersionInfo

kubernetes.test.test_versioned_event module

Module contents

kubernetes.watch package

Submodules

kubernetes.watch.watch module

class kubernetes.watch.watch.**SimpleNamespace**(***kwargs*)

class kubernetes.watch.watch.**Watch**(*return_type=None*)
Bases: object

get_return_type(*func*)

stop()

stream(*func*, **args*, ***kwargs*)
Watch an API resource and stream the result back via a generator.

Parameters **func** – The API function pointer. Any parameter to the function can be passed after this parameter.

Returns

Event object with these keys: 'type': The type of event such as "ADDED", "DELETED", etc. 'raw_object': a dict representing the watched object. 'object': A model representation of raw_object. The name of

model will be determined based on the func's doc string. If it cannot be determined, 'object' value will be the same as 'raw_object'.

Example: v1 = kubernetes.client.CoreV1Api() watch = kubernetes.watch.Watch() for e in watch.stream(v1.list_namespace, resource_version=1127):

type = e['type'] object = e['object'] # object is one of type return_type raw_object = e['raw_object'] # raw_object is a dict ... if should_stop:

watch.stop()

unmarshal_event(*data*, *return_type*)

kubernetes.watch.watch.**iter_resp_lines**(*resp*)

kubernetes.watch.watch_test module

class kubernetes.watch.watch_test.**WatchTests**(*methodName='runTest'*)

Bases: unittest.case.TestCase

test_unmarshal_with_float_object()

test_unmarshal_with_no_return_type()

test_watch_stream_loop()

test_watch_stream_twice()

test_watch_with_decode()

test_watch_with_exception()

Module contents

4.1.2 Module contents

CHAPTER 5

Contributing

Contributing guidelines

How to become a contributor and submit your own code

Contributor License Agreements

We'd love to accept your patches! Before we can take them, we have to jump a couple of legal hurdles.

Please fill out either the individual or corporate Contributor License Agreement (CLA).

- If you are an individual writing original source code and you're sure you own the intellectual property, then you'll need to sign an [individual CLA](https://identity.linuxfoundation.org/node/285/node/285/individual-signup).
- If you work for a company that wants to allow you to contribute your work, then you'll need to sign a [corporate CLA](https://identity.linuxfoundation.org/node/285/organization-signup).

Follow either of the two links above to access the appropriate CLA and instructions for how to sign and return it. Once we receive it, we'll be able to accept your pull requests.

Contributing A Patch

1. Submit an issue describing your proposed change to the repo in question. 1. The [repo owners](OWNERS) will respond to your issue promptly. 1. If your proposed change is accepted, and you haven't already done so, sign a Contributor License Agreement (see details above). 1. Fork the desired repo, develop and test your code changes. 1. Submit a pull request.

Adding dependencies

If your patch depends on new packages, add those packages to requirements.txt and setup.py.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

k

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

V

W