
saltant-py Documentation

Matt Wiens

Apr 19, 2019

Contents

1	Installation	3
2	Client	5
3	Managers	9
4	Models	27
5	Indices and tables	35

saltant-py is a Python SDK for [saltant](#). It lets you perform any action possible on the saltant API (some much more conveniently) within the comforts (hopefully) of Python. As a refresher, you might want to look at [saltant's API reference](#) to see what actions are possible.

The source code for saltant-py is available at github.com/saltant-org/saltant-py and is licensed under the [MIT License](#).

CHAPTER 1

Installation

There are two main ways to install saltant-py: installing with pip and installing with source.

1.1 With pip

saltant is available as a PyPI package at pypi.org/project/saltant-py. To install this package with pip, simply run

```
$ pip install saltant-py
```

1.2 From source

To install from source, first clone the saltant-py repository with

```
$ git clone https://github.com/saltant-org/saltant-py.git
```

Move into saltant-py directory

```
$ cd saltant-py
```

and then install with

```
$ python setup.py install
```

where python must be in versions 2.7 or 3.5+.

CHAPTER 2

Client

There are two ways to instantiate a saltant client:

1. Having it read settings from your environment
2. Giving it settings directly

For (1), you need to have environment variables `SALTANT_API_URL` and `SALTANT_AUTH_TOKEN` defined; once you do, you can instantiate a client like so:

```
from saltant.client import from_env

my_client = from_env()
```

For (2), you supply the API URL and auth token directly:

```
from saltant.client import Client

my_client = Client(
    base_api_url='https://shahlabjobs.ca/api/',
    auth_token='p0gch4mp101fy451do9uod1s1x9i4a')
```

2.1 Client reference

```
class saltant.client.Client (base_api_url, auth_token, default_timeout=90,
                             test_if_authenticated=True)
    API client for communicating with a saltant server.
```

Example

```
>>> from saltant.client import Client
>>> client = Client(
...     base_api_url='https://shahlabjobs.ca/api/',
...     auth_token='p0gch4mp101fy451do9uod1slx9i4a')
```

base_api_url

str – The URL of the saltant API.

session

`requests.Session` – A session object to make requests with.

container_task_instances

`saltant.models.container_task_instance.ContainerTaskInstanceManager` – A manager for performing actions related to container task instances.

container_task_types

`saltant.models.container_task_type.ContainerTaskTypeManager` – A manager for performing actions related to container task types.

executable_task_instances

`saltant.models.executable_task_instance.ExecutableTaskInstanceManager` – A manager for performing actions related to executable task instances.

executable_task_types

`saltant.models.executable_task_type.ExecutableTaskTypeManager` – A manager for performing actions related to executable task types.

task_queues

`saltant.models.task_queues.TaskQueueManager` – A manager for performing actions related to task queues.

task_whitelists

`saltant.models.task_queues.TaskWhitelistManager` – A manager for performing actions related to task whitelists.

users

`saltant.models.user.UserManager` – A manager for performing actions related to users.

classmethod from_env (*default_timeout=90*)

Return a client configured from environment variables.

Essentially copying this: <https://github.com/docker/docker-py/blob/master/docker/client.py#L43>.

The environment variables looked for are the following:

SALTANT_API_URL

The URL of the saltant API. For example, <https://shahlabjobs.ca/api/>.

SALTANT_AUTH_TOKEN

The registered saltant user's authentication token.

Example

```
>>> from saltant.client import from_env
>>> client = from_env()
```

Parameters `default_timeout` (*int*, *optional*) – The maximum number of seconds to wait for a request to complete. Defaults to 90 seconds.

Returns A saltant API client object.

Return type *Client*

Raises `saltant.exceptions.BadEnvironmentError` – The user has an incorrectly configured environment.

test_authentication()

Test that the client is authorized.

This currently assumes that read-only operations require authentication, which is the intended authentication protocol for saltant servers.

Raises `saltant.exceptions.AuthenticationError` – The authentication provided was invalid.

After you've instantiated a `saltant.client.Client`, you can interface with saltant models through managers, which are attributes of the `Client` class; that is,

- `Client.container_task_instances`
- `Client.container_task_types`
- `Client.executable_task_instances`
- `Client.executable_task_types`
- `Client.task_queues`
- `Client.task_whitelists`
- `Client.users`

With these managers you can perform the usual API requests, along with some other convenience functions: for example, waiting for a task instance to finish.

3.1 Manager references

```
class saltant.models.container_task_instance.ContainerTaskInstanceManager(_client)  
    Bases: saltant.models.base_task_instance.BaseTaskInstanceManager
```

Manager for container task instances.

`_client`

saltant.client.Client – An authenticated saltant client.

`list_url`

str – The URL to list task instances.

`detail_url`

str – The URL format to get specific task instances.

clone_url

str – The URL format to clone a task instance.

terminate_url

str – The URL format to terminate a task instance.

model

saltant.models.container_task_instance.ContainerTaskInstance – The model of the task instance being used.

clone (*uuid*)

Clone the task instance with given UUID.

Parameters **uuid** (*str*) – The UUID of the task instance to clone.

Returns A task instance model instance representing the task instance created due to the clone.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

clone_many (*uuids*)

Clone the task instances with given UUIDs.

Parameters **uuids** (*list*) – A list of strings containing the UUIDs of the task instances to clone.

Returns A list of *saltant.models.base_task_instance.BaseTaskInstance* subclass instances representing the task instances created due to the clone.

Return type *list*

create (*task_type_id*, *task_queue_id*, *arguments=None*, *name=""*)

Create a task instance.

Parameters

- **task_type_id** (*int*) – The ID of the task type to base the task instance on.
- **task_queue_id** (*int*) – The ID of the task queue to run the job on.
- **arguments** (*dict*, *optional*) – The arguments to give the task type.
- **name** (*str*, *optional*) – A non-unique name to give the task instance.

Returns A task instance model instance representing the task instance just created.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

get (*uuid*)

Get the task instance with given UUID.

Parameters **uuid** (*str*) – The UUID of the task instance to get.

Returns A task instance model instance representing the task instance requested.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

list (*filters=None*)

List model instances.

Currently this gets *everything* and iterates through all possible pages in the API. This may be unsuitable for production environments with huge databases, so finer grained page support should likely be added at some point.

Parameters **filters** (*dict*, *optional*) – API query filters to apply to the request. For example:

```
{'name__startswith': 'azure',
 'user__in': [1, 2, 3, 4],}
```

See saltant's API reference at <https://saltant-org.github.io/saltant/> for each model's available filters.

Returns A list of `saltant.models.resource.Model` subclass instances (for example, container task type model instances).

Return type list

model

alias of `ContainerTaskInstance`

response_data_to_model_instance (*response_data*)

Convert response data to a task instance model.

Parameters **response_data** (*dict*) – The data from the request's response.

Returns A task instance model instance representing the task instance from the response data.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

response_data_to_model_instances_list (*response_data*)

Convert list response data to a list of models.

Parameters **response_data** (*dict*) – The data from the request's response.

Returns A list of `saltant.models.resource.Model` subclass instances.

Return type list

terminate (*uuid*)

Terminate the task instance with given UUID.

Parameters **uuid** (*str*) – The UUID of the task instance to terminate.

Returns A task instance model instance representing the task instance that was told to terminate.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

terminate_many (*uuids*)

Terminate the task instances with given UUIDs.

Parameters **uuids** (*list*) – A list of strings containing the UUIDs of the task instances to terminate.

Returns A list of `saltant.models.base_task_instance.BaseTaskInstance` instances representing the task instances told to terminate.

Return type list

static validate_request_success (*response_text*, *request_url*, *status_code*, *expected_status_code*)

Validates that a request was successful.

Parameters

- **response_text** (*str*) – The response body of the request.
- **request_url** (*str*) – The URL the request was made at.
- **status_code** (*int*) – The status code of the response.
- **expected_status_code** (*int*) – The expected status code of the response.

Raises `saltant.exceptions.BadHttpRequestError` – The HTTP request failed.

wait_until_finished (*uuid*, *refresh_period*=5)

Wait until a task instance with the given UUID is finished.

Parameters

- **uuid** (*str*) – The UUID of the task instance to wait for.
- **refresh_period** (*float*, *optional*) – How many seconds to wait in between checking the task’s status. Defaults to 5 seconds.

Returns A task instance model instance representing the task instance which we waited for.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

class `saltant.models.container_task_type.ContainerTaskTypeManager` (*_client*)

Bases: `saltant.models.base_task_type.BaseTaskTypeManager`

Manager for container task types.

_client

`saltant.client.Client` – An authenticated saltant client.

list_url

str – The URL to list task types.

detail_url

str – The URL format to get specific task types.

model

`saltant.models.container_task_type.ContainerTaskType` – The model of the task instance being used.

create (*name*, *command_to_run*, *container_image*, *container_type*, *description*=”, *logs_path*=”, *results_path*=”, *environment_variables*=None, *required_arguments*=None, *required_arguments_default_values*=None, *extra_data_to_post*=None)

Create a container task type.

Parameters

- **name** (*str*) – The name of the task.
- **command_to_run** (*str*) – The command to run to execute the task.
- **container_image** (*str*) – The container name and tag. For example, ubuntu:14.04 for Docker; and docker://ubuntu:14:04 or shub://vsoch/hello-world for Singularity.
- **container_type** (*str*) – The type of the container.
- **description** (*str*, *optional*) – The description of the task type.
- **logs_path** (*str*, *optional*) – The path of the logs directory inside the container.
- **results_path** (*str*, *optional*) – The path of the results directory inside the container.
- **environment_variables** (*list*, *optional*) – The environment variables required on the host to execute the task.
- **required_arguments** (*list*, *optional*) – The argument names for the task type.
- **required_arguments_default_values** (*dict*, *optional*) – Default values for the task’s required arguments.
- **extra_data_to_post** (*dict*, *optional*) – Extra key-value pairs to add to the request data. This is useful for subclasses which require extra parameters.

Returns A container task type model instance representing the task type just created.

Return type `saltant.models.container_task_type.ContainerTaskType`

get (*id=None, name=None*)

Get a task type.

Either the id xor the name of the task type must be specified.

Parameters

- **id** (*int, optional*) – The id of the task type to get.
- **name** (*str, optional*) – The name of the task type to get.

Returns A task type model instance representing the task type requested.

Return type `saltant.models.base_task_type.BaseTaskType`

Raises `ValueError` – Neither id nor name were set *or* both id and name were set.

list (*filters=None*)

List model instances.

Currently this gets *everything* and iterates through all possible pages in the API. This may be unsuitable for production environments with huge databases, so finer grained page support should likely be added at some point.

Parameters **filters** (*dict, optional*) – API query filters to apply to the request. For example:

```
{'name__startswith': 'azure',
 'user__in': [1, 2, 3, 4],}
```

See saltant's API reference at <https://saltant-org.github.io/saltant/> for each model's available filters.

Returns A list of `saltant.models.resource.Model` subclass instances (for example, container task type model instances).

Return type `list`

model

alias of `ContainerTaskType`

put (*id, name, description, command_to_run, environment_variables, required_arguments, required_arguments_default_values, logs_path, results_path, container_image, container_type, extra_data_to_put=None*)

Updates a task type on the saltant server.

Parameters

- **id** (*int*) – The ID of the task type.
- **name** (*str*) – The name of the task type.
- **description** (*str*) – The description of the task type.
- **command_to_run** (*str*) – The command to run to execute the task.
- **environment_variables** (*list*) – The environment variables required on the host to execute the task.
- **required_arguments** (*list*) – The argument names for the task type.

- **required_arguments_default_values** (*dict*) – Default values for the tasks required arguments.
- **extra_data_to_put** (*dict*, *optional*) – Extra key-value pairs to add to the request data. This is useful for subclasses which require extra parameters.
- **logs_path** (*str*) – The path of the logs directory inside the container.
- **results_path** (*str*) – The path of the results directory inside the container.
- **container_image** (*str*) – The container name and tag. For example, ubuntu:14.04 for Docker; and docker://ubuntu:14:04 or shub://vsoch/hello-world for Singularity.
- **container_type** (*str*) – The type of the container.

response_data_to_model_instance (*response_data*)

Convert response data to a task type model.

Parameters **response_data** (*dict*) – The data from the request’s response.

Returns A model instance representing the task type from the response data.

Return type `saltant.models.base_task_type.BaseTaskType`

response_data_to_model_instances_list (*response_data*)

Convert list response data to a list of models.

Parameters **response_data** (*dict*) – The data from the request’s response.

Returns A list of `saltant.models.resource.Model` subclass instances.

Return type `list`

static validate_request_success (*response_text*, *request_url*, *status_code*, *expected_status_code*)

Validates that a request was successful.

Parameters

- **response_text** (*str*) – The response body of the request.
- **request_url** (*str*) – The URL the request was made at.
- **status_code** (*int*) – The status code of the response.
- **expected_status_code** (*int*) – The expected status code of the response.

Raises `saltant.exceptions.BadHttpRequestError` – The HTTP request failed.

class `saltant.models.executable_task_instance.ExecutableTaskInstanceManager` (*_client*)

Bases: `saltant.models.base_task_instance.BaseTaskInstanceManager`

Manager for executable task instances.

_client

`saltant.client.Client` – An authenticated saltant client.

list_url

str – The URL to list task instances.

detail_url

str – The URL format to get specific task instances.

clone_url

str – The URL format to clone a task instance.

terminate_url

str – The URL format to terminate a task instance.

model

`saltant.models.executable_task_instance.ExecutableTaskInstance` – The model of the task instance being used.

clone (*uuid*)

Clone the task instance with given UUID.

Parameters *uuid* (*str*) – The UUID of the task instance to clone.

Returns A task instance model instance representing the task instance created due to the clone.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

clone_many (*uuids*)

Clone the task instances with given UUIDs.

Parameters *uuids* (*list*) – A list of strings containing the UUIDs of the task instances to clone.

Returns A list of `saltant.models.base_task_instance.BaseTaskInstance` subclass instances representing the task instances created due to the clone.

Return type *list*

create (*task_type_id*, *task_queue_id*, *arguments=None*, *name=""*)

Create a task instance.

Parameters

- **task_type_id** (*int*) – The ID of the task type to base the task instance on.
- **task_queue_id** (*int*) – The ID of the task queue to run the job on.
- **arguments** (*dict*, *optional*) – The arguments to give the task type.
- **name** (*str*, *optional*) – A non-unique name to give the task instance.

Returns A task instance model instance representing the task instance just created.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

get (*uuid*)

Get the task instance with given UUID.

Parameters *uuid* (*str*) – The UUID of the task instance to get.

Returns A task instance model instance representing the task instance requested.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

list (*filters=None*)

List model instances.

Currently this gets *everything* and iterates through all possible pages in the API. This may be unsuitable for production environments with huge databases, so finer grained page support should likely be added at some point.

Parameters *filters* (*dict*, *optional*) – API query filters to apply to the request. For example:

```
{'name__startswith': 'azure',
 'user__in': [1, 2, 3, 4],}
```

See saltant's API reference at <https://saltant-org.github.io/saltant/> for each model's available filters.

Returns A list of `saltant.models.resource.Model` subclass instances (for example, container task type model instances).

Return type list

model

alias of `ExecutableTaskInstance`

response_data_to_model_instance (*response_data*)

Convert response data to a task instance model.

Parameters **response_data** (*dict*) – The data from the request’s response.

Returns A task instance model instance representing the task instance from the response data.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

response_data_to_model_instances_list (*response_data*)

Convert list response data to a list of models.

Parameters **response_data** (*dict*) – The data from the request’s response.

Returns A list of `saltant.models.resource.Model` subclass instances.

Return type list

terminate (*uuid*)

Terminate the task instance with given UUID.

Parameters **uuid** (*str*) – The UUID of the task instance to terminate.

Returns A task instance model instance representing the task instance that was told to terminate.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

terminate_many (*uuids*)

Terminate the task instances with given UUIDs.

Parameters **uuids** (*list*) – A list of strings containing the UUIDs of the task instances to terminate.

Returns A list of `saltant.models.base_task_instance.BaseTaskInstance` instances representing the task instances told to terminate.

Return type list

static validate_request_success (*response_text*, *request_url*, *status_code*, *expected_status_code*)

Validates that a request was successful.

Parameters

- **response_text** (*str*) – The response body of the request.
- **request_url** (*str*) – The URL the request was made at.
- **status_code** (*int*) – The status code of the response.
- **expected_status_code** (*int*) – The expected status code of the response.

Raises `saltant.exceptions.BadHttpRequestError` – The HTTP request failed.

wait_until_finished (*uuid*, *refresh_period=5*)

Wait until a task instance with the given UUID is finished.

Parameters

- **uuid** (*str*) – The UUID of the task instance to wait for.

- **refresh_period** (*float, optional*) – How many seconds to wait in between checking the task’s status. Defaults to 5 seconds.

Returns A task instance model instance representing the task instance which we waited for.

Return type `saltant.models.base_task_instance.BaseTaskInstance`

class `saltant.models.executable_task_type.ExecutableTaskTypeManager` (*_client*)

Bases: `saltant.models.base_task_type.BaseTaskTypeManager`

Manager for executable task types.

_client

`saltant.client.Client` – An authenticated saltant client.

list_url

str – The URL to list task types.

detail_url

str – The URL format to get specific task types.

model

`saltant.models.executable_task_type.ExecutableTaskType` – The model of the task instance being used.

create (*name, command_to_run, description=”, environment_variables=None, required_arguments=None, required_arguments_default_values=None, json_file_option=None, extra_data_to_post=None*)

Create a container task type.

Parameters

- **name** (*str*) – The name of the task.
- **command_to_run** (*str*) – The command to run to execute the task.
- **description** (*str, optional*) – The description of the task type.
- **environment_variables** (*list, optional*) – The environment variables required on the host to execute the task.
- **required_arguments** (*list, optional*) – The argument names for the task type.
- **required_arguments_default_values** (*dict, optional*) – Default values for the task’s required arguments.
- **json_file_option** (*str, optional*) – The name of a command line option, e.g., `-json-file`, which accepts a JSON-encoded file for the command to run.
- **extra_data_to_post** (*dict, optional*) – Extra key-value pairs to add to the request data. This is useful for subclasses which require extra parameters.

Returns An executable task type model instance representing the task type just created.

Return type `saltant.models.container_task_type.ExecutableTaskType`

get (*id=None, name=None*)

Get a task type.

Either the id xor the name of the task type must be specified.

Parameters

- **id** (*int, optional*) – The id of the task type to get.
- **name** (*str, optional*) – The name of the task type to get.

Returns A task type model instance representing the task type requested.

Return type `saltant.models.base_task_type.BaseTaskType`

Raises `ValueError` – Neither id nor name were set *or* both id and name were set.

list (*filters=None*)

List model instances.

Currently this gets *everything* and iterates through all possible pages in the API. This may be unsuitable for production environments with huge databases, so finer grained page support should likely be added at some point.

Parameters **filters** (*dict, optional*) – API query filters to apply to the request. For example:

```
{'name__startswith': 'azure',  
 'user__in': [1, 2, 3, 4],}
```

See saltant's API reference at <https://saltant-org.github.io/saltant/> for each model's available filters.

Returns A list of `saltant.models.resource.Model` subclass instances (for example, container task type model instances).

Return type `list`

model

alias of `ExecutableTaskType`

put (*id, name, description, command_to_run, environment_variables, required_arguments, required_arguments_default_values, json_file_option, extra_data_to_put=None*)
Updates a task type on the saltant server.

Parameters

- **id** (*int*) – The ID of the task type.
- **name** (*str*) – The name of the task type.
- **description** (*str*) – The description of the task type.
- **command_to_run** (*str*) – The command to run to execute the task.
- **environment_variables** (*list*) – The environment variables required on the host to execute the task.
- **required_arguments** (*list*) – The argument names for the task type.
- **required_arguments_default_values** (*dict*) – Default values for the tasks required arguments.
- **json_file_option** (*str*) – The name of a command line option, e.g., `-json-file`, which accepts a JSON-encoded file for the command to run.
- **extra_data_to_put** (*dict, optional*) – Extra key-value pairs to add to the request data. This is useful for subclasses which require extra parameters.

response_data_to_model_instance (*response_data*)

Convert response data to a task type model.

Parameters **response_data** (*dict*) – The data from the request's response.

Returns A model instance representing the task type from the response data.

Return type `saltant.models.base_task_type.BaseTaskType`

response_data_to_model_instances_list (*response_data*)

Convert list response data to a list of models.

Parameters **response_data** (*dict*) – The data from the request’s response.

Returns A list of `saltant.models.resource.Model` subclass instances.

Return type list

static validate_request_success (*response_text*, *request_url*, *status_code*, *expected_status_code*)

Validates that a request was successful.

Parameters

- **response_text** (*str*) – The response body of the request.
- **request_url** (*str*) – The URL the request was made at.
- **status_code** (*int*) – The status code of the response.
- **expected_status_code** (*int*) – The expected status code of the response.

Raises `saltant.exceptions.BadHttpRequestError` – The HTTP request failed.

class `saltant.models.task_queue.TaskQueueManager` (*_client*)

Bases: `saltant.models.resource.ModelManager`

Manager for task queues.

_client

`saltant.client.Client` – An authenticated saltant client.

list_url

str – The URL to list task queues.

detail_url

str – The URL format to get specific task queues.

model

`saltant.models.task_queue.TaskQueue` – The model of the task queue being used.

create (*name*, *description*=”, *private*=False, *runs_executable_tasks*=True, *runs_docker_container_tasks*=True, *runs_singularity_container_tasks*=True, *active*=True, *whitelists*=None)

Create a task queue.

Parameters

- **name** (*str*) – The name of the task queue.
- **description** (*str*, *optional*) – A description of the task queue.
- **private** (*bool*, *optional*) – A boolean specifying whether the queue is exclusive to its creator. Defaults to False.
- **runs_executable_tasks** (*bool*, *optional*) – A Boolean specifying whether the queue runs executable tasks. Defaults to True.
- **runs_docker_container_tasks** (*bool*, *optional*) – A Boolean specifying whether the queue runs container tasks that run in Docker containers. Defaults to True.
- **runs_singularity_container_tasks** (*bool*, *optional*) – A Boolean specifying whether the queue runs container tasks that run in Singularity containers. Defaults to True.

- **active** (*bool, optional*) – A boolean specifying whether the queue is active. Default to True.
- **whitelists** (*list, optional*) – A list of task whitelist IDs. Defaults to None (which gets translated to []).

Returns A task queue model instance representing the task queue just created.

Return type `saltant.models.task_queue.TaskQueue`

get (*id=None, name=None*)
Get a task queue.

Either the id xor the name of the task type must be specified.

Parameters

- **id** (*int, optional*) – The id of the task type to get.
- **name** (*str, optional*) – The name of the task type to get.

Returns A task queue model instance representing the task queue requested.

Return type `saltant.models.task_queue.TaskQueue`

Raises `ValueError` – Neither id nor name were set *or* both id and name were set.

list (*filters=None*)
List model instances.

Currently this gets *everything* and iterates through all possible pages in the API. This may be unsuitable for production environments with huge databases, so finer grained page support should likely be added at some point.

Parameters **filters** (*dict, optional*) – API query filters to apply to the request. For example:

```
{'name__startswith': 'azure',  
 'user__in': [1, 2, 3, 4],}
```

See saltant's API reference at <https://saltant-org.github.io/saltant/> for each model's available filters.

Returns A list of `saltant.models.resource.Model` subclass instances (for example, container task type model instances).

Return type `list`

model
alias of `TaskQueue`

patch (*id, name=None, description=None, private=None, runs_executable_tasks=None, runs_docker_container_tasks=None, runs_singularity_container_tasks=None, active=None, whitelists=None*)
Partially updates a task queue on the saltant server.

Parameters

- **id** (*int*) – The ID of the task queue.
- **name** (*str, optional*) – The name of the task queue.
- **description** (*str, optional*) – The description of the task queue.
- **private** (*bool, optional*) – A Boolean signalling whether the queue can only be used by its associated user.

- **runs_executable_tasks** (*bool, optional*) – A Boolean specifying whether the queue runs executable tasks.
- **runs_docker_container_tasks** (*bool, optional*) – A Boolean specifying whether the queue runs container tasks that run in Docker containers.
- **runs_singularity_container_tasks** (*bool, optional*) – A Boolean specifying whether the queue runs container tasks that run in Singularity containers.
- **active** (*bool, optional*) – A Boolean signalling whether the queue is active.
- **whitelists** (*list, optional*) – A list of task whitelist IDs.

Returns A task queue model instance representing the task queue just updated.

Return type `saltant.models.task_queue.TaskQueue`

put (*id, name, description, private, runs_executable_tasks, runs_docker_container_tasks, runs_singularity_container_tasks, active, whitelists*)
Updates a task queue on the saltant server.

Parameters

- **id** (*int*) – The ID of the task queue.
- **name** (*str*) – The name of the task queue.
- **description** (*str*) – The description of the task queue.
- **private** (*bool*) – A Boolean signalling whether the queue can only be used by its associated user.
- **runs_executable_tasks** (*bool*) – A Boolean specifying whether the queue runs executable tasks.
- **runs_docker_container_tasks** (*bool*) – A Boolean specifying whether the queue runs container tasks that run in Docker containers.
- **runs_singularity_container_tasks** (*bool*) – A Boolean specifying whether the queue runs container tasks that run in Singularity containers.
- **active** (*bool*) – A Boolean signalling whether the queue is active.
- **whitelists** (*list*) – A list of task whitelist IDs.

Returns A task queue model instance representing the task queue just updated.

Return type `saltant.models.task_queue.TaskQueue`

response_data_to_model_instance (*response_data*)

Convert get response data to a model.

Parameters **response_data** (*dict*) – The data from the request's response.

Returns A `saltant.models.resource.Model` subclass instance representing the resource given in the request's response data.

Return type `saltant.models.resource.Model`

response_data_to_model_instances_list (*response_data*)

Convert list response data to a list of models.

Parameters **response_data** (*dict*) – The data from the request's response.

Returns A list of `saltant.models.resource.Model` subclass instances.

Return type `list`

static validate_request_success (*response_text*, *request_url*, *status_code*, *expected_status_code*)

Validates that a request was successful.

Parameters

- **response_text** (*str*) – The response body of the request.
- **request_url** (*str*) – The URL the request was made at.
- **status_code** (*int*) – The status code of the response.
- **expected_status_code** (*int*) – The expected status code of the response.

Raises `saltant.exceptions.BadHttpRequestError` – The HTTP request failed.

class `saltant.models.task_whitelist.TaskWhitelistManager` (*_client*)

Bases: `saltant.models.resource.ModelManager`

Manager for task whitelists.

_client

`saltant.client.Client` – An authenticated saltant client.

list_url

str – The URL to list task whitelists.

detail_url

str – The URL format to get specific task whitelists.

model

`saltant.models.task_whitelist.TaskWhitelist` – The model of the task whitelist being used.

create (*name*, *description=""*, *whitelisted_container_task_types=None*, *whitelisted_executable_task_types=None*)

Create a task whitelist.

Parameters

- **name** (*str*) – The name of the task whitelist.
- **description** (*str*, *optional*) – A description of the task whitelist.
- **whitelisted_container_task_types** (*list*, *optional*) – A list of whitelisted container task type IDs.
- **whitelisted_executable_task_types** (*list*, *optional*) – A list of whitelisted executable task type IDs.

Returns A task whitelist model instance representing the task whitelist just created.

Return type `saltant.models.task_whitelist.TaskWhitelist`

get (*id=None*, *name=None*)

Get a task whitelist.

Either the id xor the name of the task type must be specified.

Parameters

- **id** (*int*, *optional*) – The id of the task type to get.
- **name** (*str*, *optional*) – The name of the task type to get.

Returns A task whitelist model instance representing the task whitelist requested.

Return type `saltant.models.task_whitelist.TaskWhitelist`

Raises `ValueError` – Neither `id` nor `name` were set *or* both `id` and `name` were set.

list (*filters=None*)

List model instances.

Currently this gets *everything* and iterates through all possible pages in the API. This may be unsuitable for production environments with huge databases, so finer grained page support should likely be added at some point.

Parameters *filters* (*dict*, *optional*) – API query filters to apply to the request. For example:

```
{'name__startswith': 'azure',
 'user__in': [1, 2, 3, 4],}
```

See saltant's API reference at <https://saltant-org.github.io/saltant/> for each model's available filters.

Returns A list of `saltant.models.resource.Model` subclass instances (for example, container task type model instances).

Return type `list`

model

alias of `TaskWhitelist`

patch (*id*, *name=None*, *description=None*, *whitelisted_container_task_types=None*, *whitelisted_executable_task_types=None*)

Partially updates a task whitelist on the saltant server.

Parameters

- **id** (*int*) – The ID of the task whitelist.
- **name** (*str*, *optional*) – The name of the task whitelist.
- **description** (*str*, *optional*) – A description of the task whitelist.
- **whitelisted_container_task_types** (*list*, *optional*) – A list of whitelisted container task type IDs.
- **whitelisted_executable_task_types** (*list*, *optional*) – A list of whitelisted executable task type IDs.

Returns A task whitelist model instance representing the task whitelist just updated.

Return type `saltant.models.task_whitelist.TaskWhitelist`

put (*id*, *name*, *description*, *whitelisted_container_task_types*, *whitelisted_executable_task_types*)

Updates a task whitelist on the saltant server.

Parameters

- **id** (*int*) – The ID of the task whitelist.
- **name** (*str*) – The name of the task whitelist.
- **description** (*str*) – The description of the task whitelist.
- **whitelisted_container_task_types** (*list*) – A list of whitelisted container task type IDs.
- **whitelisted_executable_task_types** (*list*) – A list of whitelisted executable task type IDs.

Returns A task whitelist model instance representing the task whitelist just updated.

Return type `saltant.models.task_whitelist.TaskWhitelist`

response_data_to_model_instance (*response_data*)

Convert get response data to a model.

Parameters **response_data** (*dict*) – The data from the request’s response.

Returns A `saltant.models.resource.Model` subclass instance representing the resource given in the request’s response data.

Return type `saltant.models.resource.Model`

response_data_to_model_instances_list (*response_data*)

Convert list response data to a list of models.

Parameters **response_data** (*dict*) – The data from the request’s response.

Returns A list of `saltant.models.resource.Model` subclass instances.

Return type `list`

static validate_request_success (*response_text*, *request_url*, *status_code*, *expected_status_code*)

Validates that a request was successful.

Parameters

- **response_text** (*str*) – The response body of the request.
- **request_url** (*str*) – The URL the request was made at.
- **status_code** (*int*) – The status code of the response.
- **expected_status_code** (*int*) – The expected status code of the response.

Raises `saltant.exceptions.BadHttpRequestError` – The HTTP request failed.

class `saltant.models.user.UserManager` (*_client*)

Bases: `saltant.models.resource.ModelManager`

Manager for task queues.

_client

`saltant.client.Client` – An authenticated saltant client.

list_url

str – The URL to list task queues.

detail_url

str – The URL format to get specific task queues.

model

`saltant.models.user.User` – The model of the task queue being used.

get (*username*)

Get a user.

Parameters **username** (*str*) – The username of the user to get.

Returns A user model instance representing the user requested.

Return type `saltant.models.user.User`

list (*filters=None*)

List model instances.

Currently this gets *everything* and iterates through all possible pages in the API. This may be unsuitable for production environments with huge databases, so finer grained page support should likely be added at some point.

Parameters `filters` (*dict*, *optional*) – API query filters to apply to the request. For example:

```
{'name__startswith': 'azure',
 'user__in': [1, 2, 3, 4],}
```

See saltant’s API reference at <https://saltant-org.github.io/saltant/> for each model’s available filters.

Returns A list of `saltant.models.resource.Model` subclass instances (for example, container task type model instances).

Return type list

model

alias of *User*

response_data_to_model_instance (*response_data*)

Convert get response data to a model.

Parameters `response_data` (*dict*) – The data from the request’s response.

Returns A `saltant.models.resource.Model` subclass instance representing the resource given in the request’s response data.

Return type `saltant.models.resource.Model`

response_data_to_model_instances_list (*response_data*)

Convert list response data to a list of models.

Parameters `response_data` (*dict*) – The data from the request’s response.

Returns A list of `saltant.models.resource.Model` subclass instances.

Return type list

static validate_request_success (*response_text*, *request_url*, *status_code*, *expected_status_code*)

Validates that a request was successful.

Parameters

- **response_text** (*str*) – The response body of the request.
- **request_url** (*str*) – The URL the request was made at.
- **status_code** (*int*) – The status code of the response.
- **expected_status_code** (*int*) – The expected status code of the response.

Raises `saltant.exceptions.BadHttpRequestError` – The HTTP request failed.

saltant model instances are represented with model classes in saltant-py. These model classes have saltant model instance attributes as class attributes (e.g., UUIDs for task instances).

For some models, notably, the task instance models, there are convenience methods for, in the case of task instances, cloning, terminating, and waiting until completion.

4.1 Model references

```
class saltant.models.container_task_instance.ContainerTaskInstance(uuid,  
                                                                    name,  
                                                                    state,  
                                                                    user,  
                                                                    task_queue,  
                                                                    task_type,  
                                                                    date-  
                                                                    time_created,  
                                                                    date-  
                                                                    time_finished,  
                                                                    argu-  
                                                                    ments,  
                                                                    manager)
```

Bases: saltant.models.base_task_instance.BaseTaskInstance

Model for container task instances.

name
str – The name of the task instance.

uuid
str – The UUID of the task instance.

state
str – The state of the task instance.

user
str – The username of the user who started the task.

task_queue
int – The ID of the task queue the instance is running on.

task_type
int – The ID of the task type for the instance.

datetime_created
datetime.datetime – The datetime when the task instance was created.

datetime_finished
datetime.datetime – The datetime when the task instance finished.

arguments
dict – The arguments the task instance was run with.

manager
saltant.models.container_task_instance.ContainerTaskInstanceManager – The task instance manager which spawned this task instance ... instance.

clone()
Clone this task instance.

Returns A task instance model instance representing the task instance created due to the clone.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

sync()
Sync this model with latest data on the saltant server.

Note that in addition to returning the updated object, it also updates the existing object.

Returns This task instance ... instance after syncing.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

terminate()
Terminate this task instance.

Returns This task instance model after it was told to terminate.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

wait_until_finished(refresh_period=5)
Wait until a task instance with the given UUID is finished.

Parameters **refresh_period** (*int, optional*) – How many seconds to wait before checking the task's status. Defaults to 5 seconds.

Returns This task instance model after it finished.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

```
class saltant.models.container_task_type.ContainerTaskType(id, name, de-
scription, user,
datetime_created,
command_to_run,
environ-
ment_variables, re-
quired_arguments, re-
quired_arguments_default_values,
logs_path, re-
sults_path, con-
tainer_image, con-
tainer_type, man-
ager)
```

Bases: saltant.models.base_task_type.BaseTaskType

Model for container task types.

id

int – The ID of the task type.

name

str – The name of the task type.

description

str – The description of the task type.

user

str – The user associated with the task type.

datetime_created

datetime.datetime – The datetime when the task type was created.

command_to_run

str – The command to run inside the container to execute the task.

environment_variables

list – The environment variables required on the host to execute the task.

required_arguments

list – The argument names for the task type.

required_arguments_default_values

dict – Default values for the tasks required arguments.

logs_path

str – The path of the logs directory inside the container.

results_path

str – The path of the results directory inside the container.

container_image

str – The container name and tag. For example, ubuntu:14.04 for Docker; and docker://ubuntu:14:04 or shub://vsoch/hello-world for Singularity.

container_type

str – The type of the container.

manager

saltant.models.container_task_type.ContainerTaskTypeManager – The task type manager which spawned this task type.

put()

Updates this task type on the saltant server.

Returns A task type model instance representing the task type just updated.

Return type `saltant.models.container_task_type.ContainerTaskType`

sync()

Sync this model with latest data on the saltant server.

Note that in addition to returning the updated object, it also updates the existing object.

Returns This task type instance after syncing.

Return type `saltant.models.base_task_type.BaseTaskType`

```
class saltant.models.executable_task_instance.ExecutableTaskInstance(uuid,  
                                                                    name,  
                                                                    state,  
                                                                    user,  
                                                                    task_queue,  
                                                                    task_type,  
                                                                    date-  
                                                                    time_created,  
                                                                    date-  
                                                                    time_finished,  
                                                                    argu-  
                                                                    ments,  
                                                                    man-  
                                                                    ager)
```

Bases: `saltant.models.base_task_instance.BaseTaskInstance`

Model for executable task instances.

name

str – The name of the task instance.

uuid

str – The UUID of the task instance.

state

str – The state of the task instance.

user

str – The username of the user who started the task.

task_queue

int – The ID of the task queue the instance is running on.

task_type

int – The ID of the task type for the instance.

datetime_created

`datetime.datetime` – The datetime when the task instance was created.

datetime_finished

`datetime.datetime` – The datetime when the task instance finished.

arguments

dict – The arguments the task instance was run with.

manager

saltant.models.executable_task_instance.ExecutableTaskInstanceManager –
The task instance manager which spawned this task instance.

clone()

Clone this task instance.

Returns A task instance model instance representing the task instance created due to the clone.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

sync()

Sync this model with latest data on the saltant server.

Note that in addition to returning the updated object, it also updates the existing object.

Returns This task instance ... instance after syncing.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

terminate()

Terminate this task instance.

Returns This task instance model after it was told to terminate.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

wait_until_finished(refresh_period=5)

Wait until a task instance with the given UUID is finished.

Parameters **refresh_period** (*int, optional*) – How many seconds to wait before checking the task's status. Defaults to 5 seconds.

Returns This task instance model after it finished.

Return type *saltant.models.base_task_instance.BaseTaskInstance*

```
class saltant.models.executable_task_type.ExecutableTaskType(id, name, de-  
scription, user,  
datetime_created,  
command_to_run,  
environ-  
ment_variables, re-  
quired_arguments,  
re-  
quired_arguments_default_values,  
json_file_option,  
manager)
```

Bases: *saltant.models.base_task_type.BaseTaskType*

Model for executable task types.

id

int – The ID of the task type.

name

str – The name of the task type.

description

str – The description of the task type.

user

str – The user associated with the task type.

datetime_created

datetime.datetime – The datetime when the task type was created.

command_to_run

str – The command to run to execute the task.

environment_variables

list – The environment variables required on the host to execute the task.

required_arguments

list – The argument names for the task type.

required_arguments_default_values

dict – Default values for the task's required arguments.

json_file_option

str – The name of a command line option, e.g., `-json-file`, which accepts a JSON-encoded file for the command to run.

manager

saltant.models.executable_task_type.ExecutableTaskTypeManager – The task type manager which spawned this task type instance.

put()

Updates this task type on the saltant server.

Returns An executable task type model instance representing the task type just updated.

Return type *saltant.models.container_task_type.ExecutableTaskType*

sync()

Sync this model with latest data on the saltant server.

Note that in addition to returning the updated object, it also updates the existing object.

Returns This task type instance after syncing.

Return type *saltant.models.base_task_type.BaseTaskType*

```
class saltant.models.task_queue.TaskQueue(id, user, name, description,  
                                         private, runs_executable_tasks,  
                                         runs_docker_container_tasks,  
                                         runs_singularity_container_tasks, active,  
                                         whitelists, manager)
```

Bases: *saltant.models.resource.Model*

Base model for a task queue.

id

int – The ID of the task queue.

user

str – The user who created the task queue.

name

str – The name of the task queue.

description

str – The description of the task queue.

private

bool – A Boolean signalling whether the queue can only be used by its associated user.

runs_executable_tasks

bool – A Boolean specifying whether the queue runs executable tasks.

runs_docker_container_tasks

bool – A Boolean specifying whether the queue runs container tasks that run in Docker containers.

runs_singularity_container_tasks

bool – A Boolean specifying whether the queue runs container tasks that run in Singularity containers.

active

bool – A Boolean signalling whether the queue is active.

whitelists

list – A list of task whitelist IDs.

manager

saltant.models.task_queue.TaskQueueManager – The task queue manager which spawned this task queue.

patch()

Updates this task queue on the saltant server.

This is an alias for the model's put method. (Both are identical operations on the model level.)

Returns A task queue model instance representing the task queue just updated.

Return type *saltant.models.task_queue.TaskQueue*

put()

Updates this task queue on the saltant server.

Returns A task queue model instance representing the task queue just updated.

Return type *saltant.models.task_queue.TaskQueue*

sync()

Sync this model with latest data on the saltant server.

Note that in addition to returning the updated object, it also updates the existing object.

Returns This task queue instance after syncing.

Return type *saltant.models.task_queue.TaskQueue*

```
class saltant.models.task_whitelist.TaskWhitelist(id, user, name, description,
                                                whitelisted_container_task_types,
                                                whitelisted_executable_task_types,
                                                manager)
```

Bases: *saltant.models.resource.Model*

Base model for a task whitelist.

id

int – The ID of the task whitelist.

user

str – The user who created the task whitelist.

name

str – The name of the task whitelist.

description

str – The description of the task whitelist.

whitelisted_container_task_types

list – A list of whitelisted container task type IDs.

whitelisted_executable_task_types

list – A list of whitelisted executable task type IDs.

manager

saltant.models.task_whitelist.TaskWhitelistManager – The task whitelist manager which spawned this task whitelist.

patch()

Updates this task whitelist on the saltant server.

This is an alias for the model's put method. (Both are identical operations on the model level.)

Returns A task whitelist model instance representing the task whitelist just updated.

Return type *saltant.models.task_whitelist.TaskWhitelist*

put()

Updates this task whitelist on the saltant server.

Returns A task whitelist model instance representing the task whitelist just updated.

Return type *saltant.models.task_whitelist.TaskWhitelist*

sync()

Sync this model with latest data on the saltant server.

Note that in addition to returning the updated object, it also updates the existing object.

Returns This task whitelist instance after syncing.

Return type *saltant.models.task_whitelist.TaskWhitelist*

class *saltant.models.user.User*(*username, email, manager*)

Bases: *saltant.models.resource.Model*

Base model for a user.

username

str – The user's username.

email

str – The user's email.

manager

saltant.models.user.UserManager – The manager which spawned this user instance.

sync()

Sync this model with latest data on the saltant server.

Note that in addition to returning the updated object, it also updates the existing object.

Returns This user instance after syncing.

Return type *saltant.models.user.User*

CHAPTER 5

Indices and tables

- `genindex`
- `search`

Symbols

_client (saltant.models.container_task_instance.ContainerTaskInstanceManager attribute), 9
 _client (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 12
 _client (saltant.models.executable_task_instance.ExecutableTaskInstanceManager attribute), 14
 _client (saltant.models.executable_task_type.ExecutableTaskTypeManager attribute), 17
 _client (saltant.models.task_queue.TaskQueueManager attribute), 19
 _client (saltant.models.task_whitelist.TaskWhitelistManager attribute), 22
 _client (saltant.models.user.UserManager attribute), 24
A
 active (saltant.models.task_queue.TaskQueue attribute), 33
 arguments (saltant.models.container_task_instance.ContainerTaskInstanceManager attribute), 28
 arguments (saltant.models.executable_task_instance.ExecutableTaskInstanceManager attribute), 30
B
 base_api_url (saltant.client.Client attribute), 6
C
 Client (class in saltant.client), 5
 clone() (saltant.models.container_task_instance.ContainerTaskInstanceManager method), 28
 clone() (saltant.models.container_task_instance.ContainerTaskInstanceManager method), 10
 clone() (saltant.models.executable_task_instance.ExecutableTaskInstanceManager method), 31
 clone() (saltant.models.executable_task_instance.ExecutableTaskInstanceManager method), 15
 clone_many() (saltant.models.container_task_instance.ContainerTaskInstanceManager method), 10
 clone_many() (saltant.models.executable_task_instance.ExecutableTaskInstanceManager method), 15
 clone_url (saltant.models.container_task_instance.ContainerTaskInstanceManager attribute), 9
 clone_url (saltant.models.executable_task_instance.ExecutableTaskInstanceManager attribute), 14
 command_to_run (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 29
 command_to_run (saltant.models.executable_task_type.ExecutableTaskTypeManager attribute), 32
 container_image (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 29
 container_task_instances (saltant.client.Client attribute), 6
 container_task_types (saltant.client.Client attribute), 6
 container_type (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 29
 ContainerTaskInstance (class in saltant.models.container_task_instance), 27
 ContainerTaskInstanceManager (class in saltant.models.container_task_instance), 9
 ContainerTaskType (class in saltant.models.container_task_type), 28
 ContainerTaskTypeManager (class in saltant.models.container_task_type), 12
 create() (saltant.models.container_task_instance.ContainerTaskInstanceManager method), 10
 create() (saltant.models.container_task_type.ContainerTaskTypeManager method), 12
 create() (saltant.models.executable_task_instance.ExecutableTaskInstanceManager method), 15
 create() (saltant.models.executable_task_type.ExecutableTaskTypeManager method), 17
 create() (saltant.models.task_queue.TaskQueueManager method), 19
 create() (saltant.models.task_whitelist.TaskWhitelistManager method), 22

list_url (saltant.models.executable_task_instance.ExecutableTaskInstanceManager attribute), 14

list_url (saltant.models.executable_task_type.ExecutableTaskTypeManager attribute), 17

list_url (saltant.models.task_queue.TaskQueueManager attribute), 19

list_url (saltant.models.task_whitelist.TaskWhitelistManager attribute), 22

list_url (saltant.models.user.UserManager attribute), 24

logs_path (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 29

M

manager (saltant.models.container_task_instance.ContainerTaskInstanceManager attribute), 28

manager (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 29

manager (saltant.models.executable_task_instance.ExecutableTaskInstanceManager attribute), 30

manager (saltant.models.executable_task_type.ExecutableTaskTypeManager attribute), 32

manager (saltant.models.task_queue.TaskQueueManager attribute), 33

manager (saltant.models.task_whitelist.TaskWhitelistManager attribute), 34

manager (saltant.models.user.UserManager attribute), 34

model (saltant.models.container_task_instance.ContainerTaskInstanceManager attribute), 10, 11

model (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 12, 13

model (saltant.models.executable_task_instance.ExecutableTaskInstanceManager attribute), 14, 16

model (saltant.models.executable_task_type.ExecutableTaskTypeManager attribute), 17, 18

model (saltant.models.task_queue.TaskQueueManager attribute), 19, 20

model (saltant.models.task_whitelist.TaskWhitelistManager attribute), 22, 23

model (saltant.models.user.UserManager attribute), 24, 25

N

name (saltant.models.container_task_instance.ContainerTaskInstanceManager attribute), 27

name (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 29

name (saltant.models.executable_task_instance.ExecutableTaskInstanceManager attribute), 30

name (saltant.models.executable_task_type.ExecutableTaskTypeManager attribute), 31

name (saltant.models.task_queue.TaskQueueManager attribute), 32

name (saltant.models.task_whitelist.TaskWhitelistManager attribute), 33

P

patch() (saltant.models.task_queue.TaskQueueManager method), 20

patch() (saltant.models.task_whitelist.TaskWhitelistManager method), 34

patch() (saltant.models.task_whitelist.TaskWhitelistManager method), 23

private (saltant.models.task_queue.TaskQueueManager attribute), 32

put() (saltant.models.container_task_type.ContainerTaskTypeManager method), 29

put() (saltant.models.container_task_type.ContainerTaskTypeManager method), 13

put() (saltant.models.executable_task_type.ExecutableTaskTypeManager method), 32

put() (saltant.models.executable_task_type.ExecutableTaskTypeManager method), 18

put() (saltant.models.task_queue.TaskQueueManager method), 33

put() (saltant.models.task_queue.TaskQueueManager method), 21

put() (saltant.models.task_whitelist.TaskWhitelistManager method), 34

put() (saltant.models.task_whitelist.TaskWhitelistManager method), 23

R

required_arguments (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 29

required_arguments (saltant.models.executable_task_type.ExecutableTaskTypeManager attribute), 32

required_arguments_default_values (saltant.models.container_task_type.ContainerTaskTypeManager attribute), 29

required_arguments_default_values (saltant.models.executable_task_type.ExecutableTaskTypeManager attribute), 32

response_data_to_model_instance() (saltant.models.container_task_instance.ContainerTaskInstanceManager method), 11

response_data_to_model_instance() (saltant.models.container_task_type.ContainerTaskTypeManager method), 14

response_data_to_model_instance() (saltant.models.executable_task_instance.ExecutableTaskInstanceManager method), 16

response_data_to_model_instance() (saltant.models.executable_task_type.ExecutableTaskTypeManager method), 18

response_data_to_model_instance() (saltant.models.task_queue.TaskQueueManager method), 21

response_data_to_model_instance()
 (saltant.models.task_whitelist.TaskWhitelistManager
 method), 24

response_data_to_model_instance()
 (saltant.models.user.UserManager method), 25

response_data_to_model_instances_list()
 (saltant.models.container_task_instance.ContainerTaskInstanceManager
 method), 11

response_data_to_model_instances_list()
 (saltant.models.container_task_type.ContainerTaskTypeManager
 method), 14

response_data_to_model_instances_list()
 (saltant.models.executable_task_instance.ExecutableTaskInstanceManager
 method), 16

response_data_to_model_instances_list()
 (saltant.models.executable_task_type.ExecutableTaskTypeManager
 method), 18

response_data_to_model_instances_list()
 (saltant.models.task_queue.TaskQueueManager
 method), 21

response_data_to_model_instances_list()
 (saltant.models.task_whitelist.TaskWhitelistManager
 method), 24

response_data_to_model_instances_list()
 (saltant.models.user.UserManager method), 25

results_path (saltant.models.container_task_type.ContainerTaskType
 attribute), 29

runs_docker_container_tasks
 (saltant.models.task_queue.TaskQueue at-
 tribute), 33

runs_executable_tasks (saltant.models.task_queue.TaskQueue
 attribute), 32

runs_singularity_container_tasks
 (saltant.models.task_queue.TaskQueue at-
 tribute), 33

S

session (saltant.client.Client attribute), 6

state (saltant.models.container_task_instance.ContainerTaskInstance
 attribute), 27

state (saltant.models.executable_task_instance.ExecutableTaskInstance
 attribute), 30

sync() (saltant.models.container_task_instance.ContainerTaskInstance
 method), 28

sync() (saltant.models.container_task_type.ContainerTaskType
 method), 30

sync() (saltant.models.executable_task_instance.ExecutableTaskInstance
 method), 31

sync() (saltant.models.executable_task_type.ExecutableTaskType
 method), 32

sync() (saltant.models.task_queue.TaskQueue method),
 33

sync() (saltant.models.task_whitelist.TaskWhitelist
 method), 34

sync() (saltant.models.user.User method), 34

T

task_queue (saltant.models.container_task_instance.ContainerTaskInstance
 attribute), 28

task_queue (saltant.models.executable_task_instance.ExecutableTaskInstance
 attribute), 30

task_queues (saltant.client.Client attribute), 6

task_type (saltant.models.container_task_instance.ContainerTaskInstance
 attribute), 28

task_type (saltant.models.executable_task_instance.ExecutableTaskInstance
 attribute), 30

task_whitelists (saltant.client.Client attribute), 6

TaskQueue (class in saltant.models.task_queue), 32

TaskQueueManager (class in saltant.models.task_queue),
 32

TaskWhitelist (class in saltant.models.task_whitelist), 33

TaskWhitelistManager (class in
 saltant.models.task_whitelist), 22

terminate() (saltant.models.container_task_instance.ContainerTaskInstance
 method), 28

terminate() (saltant.models.container_task_instance.ContainerTaskInstance
 method), 11

terminate() (saltant.models.executable_task_instance.ExecutableTaskInstance
 method), 31

terminate() (saltant.models.executable_task_instance.ExecutableTaskInstance
 method), 16

terminate_many() (saltant.models.container_task_instance.ContainerTaskInstance
 method), 11

terminate_many() (saltant.models.executable_task_instance.ExecutableTaskInstance
 method), 16

terminate_url (saltant.models.container_task_instance.ContainerTaskInstance
 attribute), 10

terminate_url (saltant.models.executable_task_instance.ExecutableTaskInstance
 attribute), 14

test_authentication() (saltant.client.Client method), 7

U

User (class in saltant.models.user), 34

user (saltant.models.container_task_instance.ContainerTaskInstance
 attribute), 27

user (saltant.models.container_task_type.ContainerTaskType
 attribute), 29

user (saltant.models.executable_task_instance.ExecutableTaskInstance
 attribute), 30

user (saltant.models.executable_task_type.ExecutableTaskType
 attribute), 31

user (saltant.models.task_queue.TaskQueue attribute), 32

user (saltant.models.task_whitelist.TaskWhitelist at-
 tribute), 33

UserManager (class in saltant.models.user), 24

username (saltant.models.user.User attribute), 34

users (saltant.client.Client attribute), 6

uuid (saltant.models.container_task_instance.ContainerTaskInstance
attribute), [27](#)

uuid (saltant.models.executable_task_instance.ExecutableTaskInstance
attribute), [30](#)

V

validate_request_success()
(saltant.models.container_task_instance.ContainerTaskInstanceManager
static method), [11](#)

validate_request_success()
(saltant.models.container_task_type.ContainerTaskTypeManager
static method), [14](#)

validate_request_success()
(saltant.models.executable_task_instance.ExecutableTaskInstanceManager
static method), [16](#)

validate_request_success()
(saltant.models.executable_task_type.ExecutableTaskTypeManager
static method), [19](#)

validate_request_success()
(saltant.models.task_queue.TaskQueueManager
static method), [21](#)

validate_request_success()
(saltant.models.task_whitelist.TaskWhitelistManager
static method), [24](#)

validate_request_success()
(saltant.models.user.UserManager static
method), [25](#)

W

wait_until_finished() (saltant.models.container_task_instance.ContainerTaskInstance
method), [28](#)

wait_until_finished() (saltant.models.container_task_instance.ContainerTaskInstanceManager
method), [12](#)

wait_until_finished() (saltant.models.executable_task_instance.ExecutableTaskInstance
method), [31](#)

wait_until_finished() (saltant.models.executable_task_instance.ExecutableTaskInstanceManager
method), [16](#)

whitelisted_container_task_types
(saltant.models.task_whitelist.TaskWhitelist
attribute), [33](#)

whitelisted_executable_task_types
(saltant.models.task_whitelist.TaskWhitelist
attribute), [33](#)

whitelists (saltant.models.task_queue.TaskQueue at-
tribute), [33](#)