cachet-client

Release 2.0.0

Oct 31, 2019

Contents

cachet-client is a python 3.5+ client library for the open source status page system Cachet .

CHAPTER 1

Guide

1.1 Install

A package is available on PyPI:

```
pip install cachet-client
```

Bulding from source:

```
git clone https://github.com/ZettaIO/cachet-client.git (or use ssh)
python setup.py bdist_wheel
# .whl will be located in dist/ directory and can be installed later with pip
```

1.1.1 Development Setup

Development install:

```
git clone https://github.com/ZettaIO/cachet-client.git (or use ssh)
cd cachet-client
python -m virtualenv .venv
. .venv/bin/activate
pip install -e .
```

Building docs:

```
pip install -r docs/requirements.txt
python setup.py build_sphinx
```

Running unit tests:

```
pip install -r tests/requirements.txt
tox
```

(continues on next page)

(continued from previous page)

```
# Optionally
tox -e py36  # tests only
tox -e pep8  # for pep8 run only

# Running tests wity pytest also works, but this works poorly in combination
# with enviroment variables for the live test script (tox separates enviroments)
pytest tests/
```

1.1.2 Testing with real Cachet service

Do not run this script againt a system in production. This is only for a test service. Cachet can easily be set up locally with docker: https://github.com/CachetHQ/Docker

You need to set the following environment variables:

```
CACHET_ENDPOINT
CACHET_API_TOKEN
```

Running tests:

```
python extras/live_run.py
...
==================================================
Numer of tests    : 10
Succesful         : 10
Failure           : 0
Percentage passed : 100.0%
==================================================
```

1.2 Basic Usage

1.2.1 Creating a client

```
import cachetclient

client = cachetclient.Client(
    endpoint='https://status.test/api/v1',
    api_token='secrettoken',
)
```

1.2.2 Add a new subscriber with email verification

```
sub = client.subscribers.create(email='user@example.test', verify=False)
```

1.2.3 List subscribers paginated

```
# Pagination under the hood scaling better with large numbers of subscribers
for sub in client.subscribers.list(page=1, per_page=100):
    print(sub.id, sub.email, sub.verify_code)
```

1.2.4 Creating a component issue

```
from cachetclient.v1 import enums

# Issue signaling to a component there is a major outage
client.incidents.create(
    name="Something blew up!",
    message="We are looking into it",
    status=enums.INCIDENT_INVESTIGATING,
    component_id=1,
    component_status=enums.COMPONENT_STATUS_MAJOR_OUTAGE,
)
```

1.2.5 Creating component group with components

```
from cachetclient.v1 import enums

group = client.component_groups.create(name="Global Services")
component = client.components.create(
    name="Public webside",
    status=enums.COMPONENT_STATUS_OPERATIONAL,
    description="This is a test",
    tags="test, web, something",
    group_id=group.id,
)
```

1.2.6 Recreating resource from json or dict

Every manager has a method for recreating a resource instance from a json string or dictionary. This can be useful if data from cachet is cached or stored somewhere like memcache or a database.

```
subscriber = client.subscribers.instance_from_json(json_str)
subscriber = client.subscribers.instance_from_dict(data_dict)
```

CHAPTER 2

Reference

2.1 Client

`cachetclient.`**`Client`**(*endpoint: str = None*, *api_token: str = None*, *version: str = None*, *verify_tls: bool = True*) → cachetclient.v1.client.Client

Creates a cachet client. Use this fuction to create clients to ensure compatibility in the future.

Parameters

- **endpoint** (*str*) – The api endpoint. for example 'https://status.examples.test/api/v1'. The endpoint can also be specified using the `CACHET_ENDPOINT` env variable.
- **api_token** (*str*) – The api token. Can also be specified using `CACHET_API_TOKEN` env variable.
- **version** (*str*) – The api version. If not specified the version will be derived from the endpoint url. The value "1" will create a v1 cachet client.
- **verify_tls** (*bool*) – Enable/disable tls verify. When using self signed certificates this has to be `False`.

2.2 enums

Constants / enums for various resources in cachet like component and incident status value.

2.2.1 Component Status

`cachetclient.v1.enums.`**`COMPONENT_STATUS_OPERATIONAL`**

[1] Operational. The component is working.

`cachetclient.v1.enums.`**`COMPONENT_STATUS_PERFORMANCE_ISSUES`**

[2] Performance Issues. The component is experiencing some slowness.

`cachetclient.v1.enums.`**`COMPONENT_STATUS_PARTIAL_OUTAGE`**
[3] Partial Outage. The component may not be working for everybody. This could be a geographical issue for example.

`cachetclient.v1.enums.`**`COMPONENT_STATUS_MAJOR_OUTAGE`**
[4] Major Outage. The component is not working for anybody.

`cachetclient.v1.enums.`**`COMPONENT_STATUS_LIST`**
List of all component statuses

Can be used for:

```
>> status in enums.COMPONENT_STATUS_LIST
True
```

2.2.2 Component Group Collapsed

`cachetclient.v1.enums.`**`COMPONENT_GROUP_COLLAPSED_FALSE`**
[0] No

`cachetclient.v1.enums.`**`COMPONENT_GROUP_COLLAPSED_TRUE`**
[1] Yes

`cachetclient.v1.enums.`**`COMPONENT_GROUP_COLLAPSED_NOT_OPERATIONAL`**
[2] Component is not Operational

2.2.3 Incident Status

`cachetclient.v1.enums.`**`INCIDENT_SCHEDULED`**
[0] Scheduled. This status is reserved for a scheduled status.

`cachetclient.v1.enums.`**`INCIDENT_INVESTIGATING`**
[1] Investigating. You have reports of a problem and you're currently looking into them.

`cachetclient.v1.enums.`**`INCIDENT_IDENTIFIED`**
[2] Identified. You've found the issue and you're working on a fix.

`cachetclient.v1.enums.`**`INCIDENT_WATCHING`**
[3] Watching. You've since deployed a fix and you're currently watching the situation.

`cachetclient.v1.enums.`**`INCIDENT_FIXED`**
[4] Fixed. The fix has worked, you're happy to close the incident.

`cachetclient.v1.enums.`**`incident_status_human`**(*status: int*)
Get human status from incident status id

Example:

```
>> incident_status_human(enums.INCIDENT_FIXED)
Fixed
```

Parameters **`status`** (*`int`*) – Incident status id

Returns Human status

Return type str

2.2.4 Schedule Status

`cachetclient.v1.enums.SCHEDULE_STATUS_UPCOMING`
[0] Upcoming

`cachetclient.v1.enums.SCHEDULE_STATUS_IN_PROGRESS`
[1] In progress

`cachetclient.v1.enums.SCHEDULE_STATUS_COMPLETE`
[2] Completed

2.3 Ping

2.3.1 Methods

`PingManager.get()` → bool
Check if the cachet api is responding.

Example:

```
>> client.ping.get()
True
```

Returns `True` if a successful response. Otherwise `False`.

Return type bool

`PingManager.__call__()` → bool
Shortcut for the *`get`* method.

Example:

```
>> client.ping()
True
```

2.4 Version

2.4.1 Resource

`Version.value`
Version string from Cachet service

Type str

`Version.on_latest`
Are we on latest version? Requires beacon enabled on server.

Type bool

`Version.latest`
Obtains info dict about latest version. Requires beacon enabled on server.

Dict format is:

```
{
    "tag_name": "v2.3.10",
    "prelease": false,
    "draft": false
}
```

Type dict

2.4.2 Manager

`VersionManager.get()` → cachetclient.v1.version.Version

Get version info from the server

Example:

```
>> version = client.version.get()
>> version.value
v2.3.10
```

Returns `Version` instance

`VersionManager.__call__()` → cachetclient.v1.version.Version

Shortcut to *get*

Example:

```
>> version = client.version()
>> version.value
v2.3.10
```

2.5 Subscribers

2.5.1 Resource

Methods

`Subscriber.update()`

Posts the values in the resource to the server.

Example:

```
# Change an attribute and save the resource
>> resource.value = something
>> updated_resource = resource.update()
```

Returns The updated resource from the server

`Subscriber.get(name)` → Union[int, str, float, bool]

Safely obtain any attribute name for the resource

Parameters **name** (*str*) – Key name in json response

Returns Value from the raw json response. If the key doesn't exist `None` is returned.

`Subscriber.delete()` → None
Deletes the resource from the server

Attributes

`Subscriber.attrs`
The raw json response from the server

Type dict

`Subscriber.id`
Resource ID

Type int

`Subscriber.email`
email address

Type str

`Subscriber.verify_code`
Auto generated unique verify code

Type str

`Subscriber.is_global`
Is the user subscribed to all components?

Type bool

`Subscriber.created_at`
When the subscription was created

Type datetime

`Subscriber.updated_at`
Last time the subscription was updated

Type datetime

`Subscriber.verified_at`
When the subscription was verified. `None` if not verified

Type datetime

2.5.2 Manager

`SubscriberManager.create(*, email: str, components: List[int] = None, verify: bool = True)` → cachetclient.v1.subscribers.Subscriber
Create a subscriber. If a subscriber already exists the existing one will be returned. Note that this endoint cannot be used to edit the user.

Keyword Arguments

- **email** (*str*) – Email address to subscribe
- **components** (*List[int]*) – The components to subscribe to. If omitted all components are subscribed.
- **verify** (*bool*) – Verification status. If `False` a verification email is sent to the user

Returns `Subscriber` instance

`SubscriberManager.list`(*page: int = 1, per_page: int = 20*) → Generator[cachetclient.v1.subscribers.Subscriber, None, None]

List all subscribers

Keyword Arguments

- **page** (*int*) – The page to start listing
- **per_page** – Number of entries per page

Returns Generator of Subscriber instances

`SubscriberManager.delete`(*subscriber_id: int*) → None

Delete a specific subscriber id

Parameters **subscriber_id** (*int*) – Subscriber id to delete

Raises `requests.exceptions.HttpError` – if subscriber do not exist

`SubscriberManager.count`() → int

Count the total number of subscribers

Returns Number of subscribers

Return type int

`SubscriberManager.instance_from_dict`(*data: dict*) → cachetclient.base.Resource

Creates a resource instance from a dictionary.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from dictionary data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*dict*) – dictionary containing the instance data

Returns The resource class instance

Return type Resource

`SubscriberManager.instance_from_json`(*data: str*) → cachetclient.base.Resource

Creates a resource instance from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

`SubscriberManager.instance_list_from_json`(*data: str*) → List[cachetclient.base.Resource]

Creates a resource instance list from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instances from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

Raises `ValueError` – if json data do not deserialize into a list

2.6 Components

2.6.1 Resource

Methods

`Component.add_tag(name: str) → None`
Add a new tag.

Parameters **name** (*str*) – Name of the tag

`Component.del_tag(name: str) → None`
Delete a tag.

Parameters **name** (*str*) – Name of tag to remove

Raises `KeyError` – if tag does not exist

`Component.has_tag(name: str) → bool`
Check if a tag exists.

Parameters **name** (*str*) – Tag name

Returns If the tag exists

Return type bool

`Component.update()`
Posts the values in the resource to the server.

Example:

```
# Change an attribute and save the resource
>> resource.value = something
>> updated_resource = resource.update()
```

Returns The updated resource from the server

`Component.delete() → None`
Deletes the resource from the server

Attributes

`Component.id`
The unique ID of the component

Type int

`Component.name`
Get or set name of the component

Type str

`Component.description`
Get or set component description

Type str

`Component.link`
Get or set http link to the component

Type str

`Component.status`
Get or set dtatus id of the component (see `enums`)

Type int

`Component.status_name`
Human readable status representation

Type str

`Component.order`
Get or set order of the component in a group

Type int

`Component.group_id`
Get or set the component group id

Type int

`Component.enabled`
Get or set enabled state

Type bool

`Component.tags`
Get or set tags for the component

Also see *add_tag*, *del_tag* and *has_tag* methods.

Type set

`Component.created_at`
When the component was created

Type datetime

`Component.updated_at`
Last time the component was updated

Type datetime

2.6.2 Manager

`ComponentManager.create`(*, *name: str*, *status: int*, *description: str = None*, *link: str = None*, *order: int = None*, *group_id: int = None*, *enabled: bool = True*, *tags: Set[str] = None*)

Create a component.

Keyword Arguments

- **name** (*str*) – Name of the component
- **status** (*int*) – Status if of the component (see enums module)
- **description** (*str*) – Description of the component (required)
- **link** (*str*) – Link to the component
- **order** (*int*) – Order of the component in its group
- **group_id** (*int*) – The group it belongs to
- **enabled** (*bool*) – Enabled status

- **tags** (list) – String tags

Returns Component instance

ComponentManager.**update**(*component_id: int, *, status: int, name: str = None, description: str = None, link: str = None, order: int = None, group_id: int = None, enabled: bool = None, tags: Set[str] = None, **kwargs*) → cachetclient.v1.components.Component

Update a component by id.

Parameters **component_id** (*int*) – The component to update

Keyword Arguments

- **status** (*int*) – Status of the component (see enums)
- **name** (*str*) – New name
- **description** (*str*) – New description
- **link** (*str*) – Link to component
- **order** (*int*) – Order in component group
- **group_id** (*int*) – Component group id
- **enabled** (*bool*) – Enable status of component
- **tags** (list) – List of strings

Returns Updated Component from server

ComponentManager.**list**(*page: int = 1, per_page: int = 20*) → Generator[cachetclient.v1.components.Component, None, None]

List all components

Keyword Arguments

- **page** (*int*) – The page to start listing
- **per_page** (*int*) – Number of entries per page

Returns Generator of Component instances

ComponentManager.**get**(*component_id: int*) → cachetclient.v1.components.Component

Get a component by id

Parameters **component_id** (*int*) – Id of the component

Returns Component instance

Raises HttpError – if not found

ComponentManager.**count**() → int

Count the number of components

Returns Total number of components

Return type int

ComponentManager.**delete**(*component_id: int*) → None

Delete a component

Parameters **component_id** (*int*) – Id of the component

Raises HTTPError – if component do not exist

`ComponentManager.instance_from_dict(data: dict)` → cachetclient.base.Resource

Creates a resource instance from a dictionary.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from dictionary data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*dict*) – dictionary containing the instance data

Returns The resource class instance

Return type Resource

`ComponentManager.instance_from_json(data: str)` → cachetclient.base.Resource

Creates a resource instance from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

`ComponentManager.instance_list_from_json(data: str)` → List[cachetclient.base.Resource]

Creates a resource instance list from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instances from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

Raises `ValueError` – if json data do not deserialize into a list

2.7 Component Groups

2.7.1 Resource

Methods

`ComponentGroup.update()`

Posts the values in the resource to the server.

Example:

```
# Change an attribute and save the resource
>> resource.value = something
>> updated_resource = resource.update()
```

Returns The updated resource from the server

`ComponentGroup.delete()` → None

Deletes the resource from the server

Attributes

`ComponentGroup.id`
Id of the component group

Type int

`ComponentGroup.name`
Set or get name of component group

Type str

`ComponentGroup.enabled_components`
Enabled components in this group

Type List[Component]

`ComponentGroup.order`
Get or set order value for group

Type int

`ComponentGroup.collapsed`
Get or set collapsed status. See `enums` module for values.

Type int

`ComponentGroup.lowest_human_status`
Lowest component status, human readable

Type str

`ComponentGroup.is_collapsed`
Does the current collapsed value indicate the group is collapsed? Note that the collapsed value may also indicate the group is not operational.

Type bool

`ComponentGroup.is_open`
Does the current collapsed value indicate the group is open? Note that the collapsed value may also indicate the group is not operational.

Type bool

`ComponentGroup.is_operational`
Does the current collapsed value indicate the group not operational?

Type bool

`ComponentGroup.created_at`
When the group was created

Type datetime

`ComponentGroup.updated_at`
Last time updated

Type datetime

2.7.2 Manager

`ComponentGroupManager.create`(*, *name: str*, *order: int = 0*, *collapsed: int = 0*, *visible: bool = False*) → cachetclient.v1.component_groups.ComponentGroup
Create a component group

Keyword Arguments

- **name** (*str*) – Name of the group
- **order** (*int*) – group order
- **collapsed** (*int*) – Collapse value (see enums)
- **visible** (*bool*) – Publicly visible group

Returns `ComponentGroup` instance

`ComponentGroupManager.update`(*group_id: int, *, name: str, order: int = None, collapsed: int = None, visible: bool = None, **kwargs*) → cachetclient.v1.component_groups.ComponentGroup

Update component group

Parameters **group_id** (*int*) – The group id to update

Keyword Arguments

- **name** (*str*) – New name for group
- **order** (*int*) – Order value of the group
- **collapsed** (*int*) – Collapsed value. See enums module.
- **visible** (*bool*) – Publicly visible group

`ComponentGroupManager.count`() → int

Count the number of component groups

Returns Number of component groups

Return type int

`ComponentGroupManager.list`(*page: int = 1, per_page: int = 20*) → Generator[cachetclient.v1.component_groups.ComponentGroup, None, None]

List all component groups

Keyword Arguments

- **page** (*int*) – The page to start listing
- **per_page** – Number of entries per page

Returns Generator of `ComponentGroup` instances

`ComponentGroupManager.get`(*group_id*) → cachetclient.v1.component_groups.ComponentGroup

Get a component group by id

Parameters **group_id** (*int*) – Id of the component group

Returns `ComponentGroup` instance

Raises *requests.exceptions.HttpError* – if not found

`ComponentGroupManager.delete`(*group_id: int*) → None

Delete a component group

Parameters **group_id** (*int*) – Id of the component

Raises *requests.exceptions.HttpError* – if not found

`ComponentGroupManager.instance_from_dict`(*data: dict*) → cachetclient.base.Resource

Creates a resource instance from a dictionary.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from dictionary data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*dict*) – dictionary containing the instance data

Returns The resource class instance

Return type Resource

`ComponentGroupManager.`**`instance_from_json`**(*data: str*) → cachetclient.base.Resource

Creates a resource instance from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

`ComponentGroupManager.`**`instance_list_from_json`**(*data: str*) → List[cachetclient.base.Resource]

Creates a resource instance list from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instances from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

Raises `ValueError` – if json data do not deserialize into a list

2.8 Incidents

2.8.1 Resource

Methods

`Incident.`**`updates`**() → Generator[cachetclient.v1.incidents.Incident, None, None]

Generator['Incident', None, None]: Incident updates for this issue

`Incident.`**`update`**()

Posts the values in the resource to the server.

Example:

```
# Change an attribute and save the resource
>> resource.value = something
>> updated_resource = resource.update()
```

Returns The updated resource from the server

`Incident.`**`delete`**() → None

Deletes the resource from the server

Attributes

`Incident.id`
unique id of the incident

Type int

`Incident.component_id`
Get or set component id for this incident

Type int

`Incident.name`
Get or set name/title of the incident

Type str

`Incident.message`
Get or set message

Type str

`Incident.notify`
Get or set notification flag

Type bool

`Incident.status`
Get or set status. See `enums`

Type int

`Incident.human_status`
Human representation of the status

Type str

`Incident.visible`
Get or set visibility of the incident

Type bool

`Incident.scheduled_at`
Scheduled time. This is used for scheduled events like maintenance in Cachet 2.3 were incident status is `INCIDENT_SCHEDULED`. 2.4 has its own schedule resource and endpoints.

Type datetime

`Incident.created_at`
When the issue was created

Type datetime

`Incident.updated_at`
Last time the issue was updated

Type datetime

`Incident.deleted_at`
When the issue was deleted

Type datetime

2.8.2 Manager

`IncidentManager.create`(*, *name: str*, *message: str*, *status: int*, *visible: bool = True*, *component_id: int = None*, *component_status: int = None*, *notify: bool = True*, *created_at: datetime.datetime = None*, *template: str = None*, *template_vars: List[str] = None*) → cachetclient.v1.incidents.Incident

Create and general issue or issue for a component. component_id and component_status must be supplied when making a component issue.

Keyword Arguments

- **name** (*str*) – Name/title of the issue
- **message** (*str*) – Mesage body for the issue
- **status** (*int*) – Status of the incident (see enums)
- **visible** (*bool*) – Publicly visible incident
- **component_id** (*int*) – The component to update
- **component_status** (*int*) – The status to apply on component
- **notify** (*bool*) – If users should be notified
- **created_at** – when the indicent was created
- **template** (*str*) – Slug of template to use
- **template_vars** (*list*) – Variables to the template

Returns Incident instance

`IncidentManager.update`(*incident_id: int*, *name: str = None*, *message: str = None*, *status: int = None*, *visible: bool = None*, *component_id: int = None*, *component_status: int = None*, *notify: bool = True*, *created_at: datetime.datetime = None*, *template: str = None*, *template_vars: List[str] = None*, ***kwargs*) → cachetclient.v1.incidents.Incident

Update an incident.

Parameters **incident_id** (*int*) – The incident to update

Keyword Arguments

- **name** (*str*) – Name/title of the issue
- **message** (*str*) – Mesage body for the issue
- **status** (*int*) – Status of the incident (see enums)
- **visible** (*bool*) – Publicly visible incident
- **component_id** (*int*) – The component to update
- **component_status** (*int*) – The status to apply on component
- **notify** (*bool*) – If users should be notified
- **created_at** – when the indicent was created
- **template** (*str*) – Slug of template to use
- **template_vars** (*list*) – Variables to the template

Returns Updated incident Instance

`IncidentManager.list`(*page: int = 1, per_page: int = 1*) → Generator[cachetclient.v1.incidents.Incident, None, None]

List all incidents paginated

Keyword Arguments

- **page** (*int*) – Page to start on
- **per_page** (*int*) – entries per page

Returns Generator of :py:data:‘Incident‘s

`IncidentManager.get`(*incident_id: int*) → cachetclient.v1.incidents.Incident

Get a single incident

Parameters **incident_id** (*int*) – The incident id to get

Returns `Incident` instance

Raises `requests.exception.HttpError` – if incident do not exist

`IncidentManager.count`() → int

Count the number of incidents

Returns Total number of incidents

Return type int

`IncidentManager.delete`(*incident_id: int*) → None

Delete an incident

Parameters **incident_id** (*int*) – The incident id

`IncidentManager.instance_from_dict`(*data: dict*) → cachetclient.base.Resource

Creates a resource instance from a dictionary.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from dictionary data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*dict*) – dictionary containing the instance data

Returns The resource class instance

Return type Resource

`IncidentManager.instance_from_json`(*data: str*) → cachetclient.base.Resource

Creates a resource instance from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

`IncidentManager.instance_list_from_json`(*data: str*) → List[cachetclient.base.Resource]

Creates a resource instance list from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instances from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

Raises `ValueError` – if json data do not deserialize into a list

2.9 IncidentUpdates

2.9.1 Resource

Methods

`IndicentUpdate.update()` → cachetclient.v1.incident_updates.IndicentUpdate
Update/save changes

Returns Updated IncidentUpdate instance

`IndicentUpdate.delete()` → None
Deletes the incident update

Attributes

`IndicentUpdate.id`
Resource id

Type int

`IndicentUpdate.incident_id`
The incident id this update belongs to

Type int

`IndicentUpdate.status`
Get or set incident status. See `enums`.

Type int

`IndicentUpdate.message`
Get or set message

Type str

`IndicentUpdate.user_id`
The user id creating the update

Type int

`IndicentUpdate.created_at`
when the resource was created

Type datetime

`IndicentUpdate.updated_at`
When the resource as last updated

Type datetime

`IndicentUpdate.human_status`
Human readable status

Type str

2.9.2 Manager

`IncidentUpdatesManager.create`(*, *incident_id: int*, *status: int*, *message: str*) → cachetclient.v1.incident_updates.IndicentUpdate

Create an incident update

Keyword Arguments

- **incident_id** (*int*) – The incident to update
- **status** (*int*) – New status id
- **message** (*str*) – Update message

Returns `IndicentUpdate` instance

`IncidentUpdatesManager.update`(*, *id: int*, *incident_id: int*, *status: int = None*, *message: str = None*, ***kwargs*) → cachetclient.v1.incident_updates.IndicentUpdate

Update an incident update

Parameters

- **incident_id** (*int*) – The incident
- **id** (*int*) – The incident update id to update

Keyword Arguments

- **status** (*int*) – New status id
- **message** (*str*) – New message

Returns The updated `IncidentUpdate` instance

`IncidentUpdatesManager.count`(*incident_id*) → int

Count the number of indicent update for an incident

Parameters **incident_id** (*int*) – The incident

Returns Number of incident updates for the incident

Return type int

`IncidentUpdatesManager.list`(*incident_id: int*, *page: int = 1*, *per_page: int = 20*) → Generator[cachetclient.v1.incident_updates.IndicentUpdate, None, None]

List updates for an issue

Parameters **incident_id** – The incident to list updates

Keyword Arguments

- **page** (*int*) – The first page to request
- **per_page** (*int*) – Entries per page

Returns Generator of :py:data:‘IncidentUpdate‘s

`IncidentUpdatesManager.get`(*incident_id: int*, *update_id: int*) → cachetclient.v1.incident_updates.IndicentUpdate

Get an incident update

Parameters

- **incident_id** (*int*) – The incident
- **update_id** (*int*) – The indicent update id to obtain

Returns `IncidentUpdate` instance

`IncidentUpdatesManager.delete`(*incident_id: int*, *update_id: int*) → None

Delete an incident update

`IncidentUpdatesManager.instance_from_dict`(*data: dict*) → cachetclient.base.Resource

Creates a resource instance from a dictionary.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from dictionary data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*dict*) – dictionary containing the instance data

Returns The resource class instance

Return type Resource

`IncidentUpdatesManager.instance_from_json`(*data: str*) → cachetclient.base.Resource

Creates a resource instance from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instance from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

`IncidentUpdatesManager.instance_list_from_json`(*data: str*) → List[cachetclient.base.Resource]

Creates a resource instance list from a json string.

This doesn't hit any endpoints in cachet, but rather enables us to create a resource class instances from json data. This can be useful when caching data from cachet in memcache or databases.

Parameters **data** (*str*) – json string containing the instance data

Returns The resource class instance

Return type Resource

Raises `ValueError` – if json data do not deserialize into a list

2.10 Metrics

Needs testing and implementation

2.11 Metric Points

Needs testing and implementation.

2.12 Schedules

Needs testing and implementation. For now stick to issues using status `INCIDENT_SCHEDULED`.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

Index

Symbols

A

C

D

E

G

H

I

L

M

N

O

S

T

U

V