
cachet-client

Release 3.1.1

Einar Forselv

Jan 07, 2021

CONTENTS

1	Guide	3
1.1	Install	3
1.2	Basic Usage	4
2	Reference	7
2.1	Client	7
2.2	enums	7
2.3	Ping	9
2.4	Version	9
2.5	Subscribers	11
2.6	Components	14
2.7	Component Groups	19
2.8	Incidents	23
2.9	IncidentUpdates	27
2.10	Metrics	30
2.11	Metric Points	33
2.12	Schedules	36
3	Indices and tables	41
	Python Module Index	43
	Index	45

cachet-client is a python 3.5+ client library for the open source status page system [Cachet](#) .

1.1 Install

A package is available on PyPI:

```
pip install cachet-client
```

Building 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

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

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

1.1.2 Testing with real Cachet service

Do not run this script against 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(
```

(continues on next page)

(continued from previous page)

```
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 website",
    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)
```


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.__init__ (http_client: cachetclient.httpclient.HttpClient)`
 Manager initializer.

Parameters `http_client` – The httpclient

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

Example:

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

`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`

2.3.2 Attributes

`PingManager.path = 'ping'`

`PingManager.resource_class = <class 'cachetclient.base.Resource'>`

2.4 Version

2.4.1 Resource

Methods

`Version.__init__ (manager, data)`
 Resource initializer.

Parameters

- **manager** – The manager this resource belongs to
- **data** – The raw json data

`Version.get (name) → Any`
 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.

Attributes

Version.**attrs**

The raw json response from the server

Type dict

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

Methods

VersionManager.**__init__** (*http_client: cachetclient.httpclient.HttpClient*)

Manager initializer.

Parameters **http_client** – The httpclient

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
```

`VersionManager.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

`VersionManager.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

`VersionManager.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

Attributes

`VersionManager.path = 'version'`

`VersionManager.resource_class = <class 'cachetclient.v1.version.Version'>`

2.5 Subscribers

2.5.1 Resource

Methods

`Subscriber.__init__(manager, data)`

Resource initializer.

Parameters

- **manager** – The manager this resource belongs to
- **data** – The raw json data

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

Return type Resource

Subscriber.get(name) → Any

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.

Raises **HTTPException** if the resource don't exist. –

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

Methods

`SubscriberManager.__init__` (*http_client: cachetclient.httpclient.HttpClient*)

Manager initializer.

Parameters `http_client` – The httpclient

`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 endpoint 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

Attributes

`SubscriberManager.path = 'subscribers'`

`SubscriberManager.resource_class = <class 'cachetclient.v1.subscribers.Subscriber'>`

2.6 Components

2.6.1 Resource

Methods

`Component.__init__(manager, data)`

Resource initializer.

Parameters

- **manager** – The manager this resource belongs to
- **data** – The raw json data

`Component.add_tag(name: str) → None`

Add a new tag.

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

`Component.add_tags(names: Iterable[str]) → None`

Add multiple tags.

Note: We use the provided name also as the slug. When the resource is returned from the server the next time it will be slugified.

Parameters `names` (*Iterable[str]*) – Iterable with names such as a list or set

`Component.set_tags` (*names: Iterable[str]*)

Replace the current tags.

Note: We use the provided names also as the slugs. When the resource is returned from the server the next time it will be slugified.

`Component.del_tag` (*name: str = None, slug: str = None*) → None

Delete a tag.

We can delete a tag by using the slug or actual name. Names and slugs are case insensitive.

Parameters

- **name** (*str*) – name to remove
- **slug** (*str*) – slug to remove

Raises **KeyError** – if tag does not exist

`Component.has_tag` (*name: str = None, slug: str = None*) → bool

Check if a tag exists by name or slug.

Tags and slugs are case insensitive.

Parameters

- **name** (*str*) – Name of the tag
- **slug** (*str*) – Slug for the tag

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

Return type Resource

`Component.get` (*name*) → Any

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.

`Component.delete` () → None

Deletes the resource from the server.

Raises **HTTPException** if the resource don't exist. –

Attributes

Component.attrs

The raw json response from the server

Type dict

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 status 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

name“ tag dictionary.

Example:

```
>> component.tags
{'another-test-tag': 'Another Test Tag', 'test-tag': 'Test Tag'}
```

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

Type dict

Type Get the raw “slug

`Component.tag_names`

Get the tag names as a list

Type List[str]

`Component.tag_slugs`

Get the tag slugs as a list

Type List[str]

`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

Methods

`ComponentManager.__init__` (*http_client: cachetclient.httpclient.HttpClient*)

Manager initializer.

Parameters `http_client` – The httpclient

`ComponentManager.create` (*, *name: str, status: int, description: str = None, link: str = None, order: int = None, group_id: int = None, enabled: bool = True, tags: Iterable[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** (*Iterable[str]*) – A list, set or other iterable containing 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: Iterable[str] = None, **kwargs*) → `cachet-client.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** (*Iterable[str]*) – Iterable of tag 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

Attributes

`ComponentManager.path = 'components'`

`ComponentManager.resource_class = <class 'cachetclient.v1.components.Component'>`

2.7 Component Groups

2.7.1 Resource

Methods

`ComponentGroup.__init__(manager, data)`

Resource initializer.

Parameters

- **manager** – The manager this resource belongs to
- **data** – The raw json data

`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

Return type Resource

`ComponentGroup.get(name) → Any`

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.

`ComponentGroup.delete()` → None

Deletes the resource from the server.

Raises `HTTPException` if the resource don't exist. –

Attributes

`ComponentGroup.attrs`

The raw json response from the server

Type dict

`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

`ComponentGroup.visible`

Get or set visibility of the group

Type bool

2.7.2 Manager

Methods

`ComponentGroupManager.__init__` (*http_client: cachetclient.httpclient.HttpClient, components_manager: cachet-client.v1.components.ComponentManager*)

Manager initializer.

Parameters `http_client` – The httpclient

`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*) → `cachet-client.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

Attributes

`ComponentGroupManager.resource_class = <class 'cachetclient.v1.component_groups.ComponentG`
`ComponentGroupManager.path = 'components/groups'`

2.8 Incidents

2.8.1 Resource

Methods

`Incident.__init__(manager, data)`
 Resource initializer.

Parameters

- **manager** – The manager this resource belongs to
- **data** – The raw json data

`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.get(name)` → Any
 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.

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

Raises `HTTPException` if the resource don't exist. –

Attributes

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

Type dict

`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.stickied
Get or set sticky value of the incident (cachet 2.4)
Type bool

Incident.scheduled_at
Scheduled time. This is used for scheduled events like maintenance in Cachet 2.3 where 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.occurred_at
When the issue was occurred
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

Methods

`IncidentManager.__init__`(*http_client: cachetclient.httpclient.HttpClient, incident_update_manager: cachetclient.v1.incident_updates.IncidentUpdatesManager*)

Manager initializer.

Parameters `http_client` – The httpclient

`IncidentManager.create`(**name: str, message: str, status: int, visible: bool = True, stickied: bool = False, component_id: int = None, component_status: int = None, notify: bool = True, created_at: datetime.datetime = None, occurred_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*) – Message body for the issue
- **status** (*int*) – Status of the incident (see enums)
- **visible** (*bool*) – Publicly visible incident
- **stickied** (*bool*) – Stickied incident
- **component_id** (*int*) – The component to update
- **component_status** (*int*) – The status to apply on component
- **notify** (*bool*) – If users should be notified
- **occurred_at** – when the incident occurred (cachet 2.4)
- **created_at** – when the incident was created (cachet 2.3)
- **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, stickied: bool = False, component_id: int = None, component_status: int = None, notify: bool = True, occurred_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*) – Message body for the issue
- **status** (*int*) – Status of the incident (see enums)
- **visible** (*bool*) – Publicly visible incident
- **stickied** (*bool*) – Stickied incident

- **component_id** (*int*) – The component to update
- **component_status** (*int*) – The status to apply on component
- **notify** (*bool*) – If users should be notified
- **occurred_at** (*datetime*) – when the incident was occurred
- **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

Attributes

`IncidentManager.path = 'incidents'`

`IncidentManager.resource_class = <class 'cachetclient.v1.incidents.Incident'>`

2.9 IncidentUpdates

2.9.1 Resource

Methods

`IncidentUpdate.__init__(manager, data)`

Resource initializer.

Parameters

- **manager** – The manager this resource belongs to
- **data** – The raw json data

`IncidentUpdate.update()` → `cachetclient.v1.incident_updates.IncidentUpdate`

Update/save changes

Returns Updated IncidentUpdate instance

`IncidentUpdate.get(name)` → Any

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.

`IncidentUpdate.delete()` → None

Deletes the incident update

Attributes

`IncidentUpdate.attrs`

The raw json response from the server

Type dict

`IncidentUpdate.id`

Resource id

Type int

`IncidentUpdate.incident_id`

The incident id this update belongs to

Type int

`IncidentUpdate.status`

Get or set incident status. See enums.

Type int

`IncidentUpdate.message`

Get or set message

Type str

`IncidentUpdate.user_id`

The user id creating the update

Type int

`IncidentUpdate.created_at`

when the resource was created

Type datetime

`IncidentUpdate.updated_at`

When the resource as last updated

Type datetime

`IncidentUpdate.human_status`

Human readable status

Type str

`IncidentUpdate.permalink`

Permanent url to the incident update

Type str

2.9.2 Manager

Methods

`IncidentUpdatesManager.__init__` (*http_client: cachetclient.httpclient.HttpClient*)

Manager initializer.

Parameters `http_client` – The httpclient

`IncidentUpdatesManager.create` (*, *incident_id: int, status: int, message: str*) → `cachet-client.v1.incident_updates.IncidentUpdate`

Create an incident update

Keyword Arguments

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

Returns IncidentUpdate instance

```
IncidentUpdatesManager.update(*, id: int, incident_id: int, status: int = None,
                             message: str = None, **kwargs) → cachet-
                             client.v1.incident_updates.IncidentUpdate
```

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 incident 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) → Genera-
                             tor[cachetclient.v1.incident_updates.IncidentUpdate, 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) → cachet-
                             client.v1.incident_updates.IncidentUpdate
```

Get an incident update

Parameters

- **incident_id**(*int*) – The incident
- **update_id**(*int*) – The incident 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

Attributes

`IncidentUpdatesManager.path = 'incidents/{}/updates'`

`IncidentUpdatesManager.resource_class = <class 'cachetclient.v1.incident_updates.IncidentUpdatesManager.Resource'>`

2.10 Metrics

2.10.1 Resource

Methods

`Metric.__init__(manager, data)`

Resource initializer.

Parameters

- **manager** – The manager this resource belongs to
- **data** – The raw json data

`Metric.get(name) → Any`

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.

`Metric.delete() → None`

Deletes the resource from the server.

Raises `HTTPException` if the resource don't exist. –

Metric.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

Return type Resource

Attributes

Metric.attrs

The raw json response from the server

Type dict

Metric.id

Metric.name

Metric.description

Metric.default_value

Metric.display_chart

Metric.places

Metric.points = <function Metric.points>

Metric.threshold

Metric.visible

Metric.order

Metric.suffix

Metric.calc_type

Metric.default_view

Metric.created_at

When the issue was created

Type datetime

Metric.updated_at

Last time the issue was updated

Type datetime

2.10.2 Manager

Methods

`MetricsManager.__init__` (*http_client: cachetclient.httpclient.HttpClient, metric_update_manager: cachetclient.v1.metric_points.MetricPointsManager*)

Manager initializer.

Parameters `http_client` – The httpclient

`MetricsManager.create` (*, *name: str, description: str, suffix: str, default_value: int = 0, display_chart: int = 0*) → `cachetclient.v1.metrics.Metric`

Create a metric.

Keyword Arguments

- **name** (*str*) – Name/title of the metric
- **description** (*str*) – Description of what the metric is measuring
- **suffix** (*str*) – Measurements in
- **default_value** (*int*) – The default value to use when a point is added
- **display_chart** (*int*) – Whether to display the chart on the status page

Returns `Metric` instance

`MetricsManager.get` (*metric_id: int*) → `cachetclient.v1.metrics.Metric`

Get a single metric

Parameters `metric_id` (*int*) – The metric id to get

Returns `Metric` instance

Raises `requests.exception.HTTPError` – if metric do not exist

`MetricsManager.list` (*page: int = 1, per_page: int = 1*) → `Generator[cachetclient.v1.metrics.Metric, None, None]`

List all metrics paginated

Keyword Arguments

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

Returns Generator of :py:data:Metric's

`MetricsManager.count` () → `int`

Count the number of metrics

Returns Total number of metrics

Return type `int`

`MetricsManager.delete` (*metric_id: int*) → `None`

Delete an metric

Parameters `metric_id` (*int*) – The metric id

`MetricsManager.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

`MetricsManager.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

`MetricsManager.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

Attributes

`MetricsManager.path = 'metrics'`

`MetricsManager.resource_class = <class 'cachetclient.v1.metrics.Metric'>`

2.11 Metric Points

2.11.1 Resource

Methods

`MetricPoint.__init__(manager, data)`

Resource initializer.

Parameters

- **manager** – The manager this resource belongs to
- **data** – The raw json data

`MetricPoint.get(name) → Any`

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.

`MetricPoint.delete() → None`

Deletes the resource from the server.

Raises `HTTPException` if the resource don't exist. –

MetricPoint.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

Return type Resource

Attributes

MetricPoint.attrs

The raw json response from the server

Type dict

MetricPoint.metric_id

Get or set metic id for this metric point

Type int

MetricPoint.value

Value to plot on the metric graph

Type float

MetricPoint.created_at

When the metric point was created

Type datetime

MetricPoint.id

unique id of the metric point

Type int

MetricPoint.updated_at

Last time the issue was updated

Type datetime

MetricPoint.calculated_value

The calculated value on metric graph

Type float

MetricPoint.counter

Show the actual calculated value

Type int

2.11.2 Manager

Methods

MetricPointsManager.__init__(*http_client: cachetclient.httpclient.HttpClient*)

Manager initializer.

Parameters `http_client` – The httpclient

`MetricPointsManager.create` (*, `metric_id`: `int`, `value`: `float`) → `cachet-client.v1.metric_points.MetricPoint`

Create an metric point

Keyword Arguments

- `metric_id` (`int`) – The metric to tag with the point
- `value` (`float`) – Metric point value for graph

Returns `MetricPoint` instance

`MetricPointsManager.list` (`metric_id`: `int`, `page`: `int` = 1, `per_page`: `int` = 20) → `Generator[cachetclient.v1.metric_points.MetricPoint, None, None]`

List updates for a metric

Parameters `metric_id` – The metric id to list updates

Keyword Arguments

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

Returns Generator of `MetricPoint`

`MetricPointsManager.count` (`metric_id`) → `int`

Count the number of metric points for a metric

Parameters `metric_id` (`int`) – The metric

Returns Number of metric points for the metric

Return type `int`

`MetricPointsManager.delete` (`metric_id`: `int`, `point_id`: `int`) → `None`

Delete a metric point

`MetricPointsManager.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`

`MetricPointsManager.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`

`MetricPointsManager.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

Attributes

`MetricPointsManager.path = 'metrics/{}/points'`

`MetricPointsManager.resource_class = <class 'cachetclient.v1.metric_points.MetricPoint'>`

2.12 Schedules

2.12.1 Resource

Methods

`Schedule.get(name) → Any`

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.

`Schedule.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

Return type Resource

`Schedule.delete() → None`

Deletes the resource from the server.

Raises **HTTPException** if the resource don't exist. –

Attributes

`Schedule.id`

Resource ID

Type int

`Schedule.name`

Name of the scheduled event

Type str

Schedule.message
Message string

Type str

Schedule.status
Status of the scheduled event

Type int

Schedule.scheduled_at
When the event is schedule for

Type datetime

Schedule.completed_at
When the event is completed

Type datetime

Schedule.attrs
The raw json response from the server

Type dict

2.12.2 Manager

Methods

ScheduleManager.__init__ (*http_client: cachetclient.httpclient.HttpClient*)
Manager initializer.

Parameters **http_client** – The httpclient

ScheduleManager.create (*, *name: str, status: int, message: str = None, scheduled_at: datetime.datetime = None, completed_at: datetime.datetime = None, notify: bool = True*)

Create a schedule.

Keyword Arguments

- **name** (*str*) – Name of the scheduled event
- **status** (*int*) – Schedule status. See enums
- **message** (*str*) – Message string
- **scheduled_at** (*datetime*) – When the event starts
- **completed_at** (*datetime*) – When the event ends
- **notify** (*bool*) – Notify subscribers

Returns Schedule instance

ScheduleManager.update (*schedule_id: int, *, name: str, status: int, message: str = None, scheduled_at: datetime.datetime = None, **kwargs*) → cachet-client.v1.schedules.Schedule

Update a Schedule by id.

Parameters **schedule_id** (*int*) – The schedule to update

Keyword Arguments

- **status** (*int*) – Status of the schedule (see enums)
- **name** (*str*) – New name
- **description** (*str*) – New description

Returns Updated Schedule from server

`ScheduleManager.list` (*page: int = 1, per_page: int = 20*) → Generator[cachetclient.v1.schedules.Schedule, None, None]

List all schedules

Keyword Arguments

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

Returns Generator of Schedules instances

`ScheduleManager.get` (*schedule_id: int*) → cachetclient.v1.schedules.Schedule
Get a schedule by id

Parameters **schedule_id** (*int*) – Id of the schedule

Returns Schedule instance

Raises **HttpError** – if not found

`ScheduleManager.delete` (*schedule_id: int*) → None
Delete a schedule

Parameters **schedule_id** (*int*) – Id of the schedule

Raises **HTTPError** – if schedule do not exist

`ScheduleManager.count` () → int
Count the total number of scheduled events

Returns Number of subscribers

Return type int

`ScheduleManager.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

`ScheduleManager.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

`ScheduleManager.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

Attributes

`ScheduleManager.path` = 'schedules'

`ScheduleManager.resource_class` = <class 'cachetclient.v1.schedules.Schedule'>

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

C

- `cachetclient`, 7
- `cachetclient.v1.component_groups`, 19
- `cachetclient.v1.components`, 14
- `cachetclient.v1.enums`, 7
- `cachetclient.v1.incident_updates`, 27
- `cachetclient.v1.incidents`, 23
- `cachetclient.v1.metric_points`, 33
- `cachetclient.v1.metrics`, 30
- `cachetclient.v1.ping`, 8
- `cachetclient.v1.schedules`, 36
- `cachetclient.v1.subscribers`, 11
- `cachetclient.v1.version`, 9

Symbols

<code>__call__()</code>	(<i>cachetclient.v1.ping.PingManager</i> method), 9
<code>__call__()</code>	(<i>cachetclient.v1.version.VersionManager</i> method), 10
<code>__init__()</code>	(<i>cachetclient.v1.component_groups.ComponentGroup</i> method), 19
<code>__init__()</code>	(<i>cachetclient.v1.component_groups.ComponentGroupManager</i> method), 21
<code>__init__()</code>	(<i>cachetclient.v1.components.Component</i> method), 14
<code>__init__()</code>	(<i>cachetclient.v1.components.ComponentManager</i> method), 17
<code>__init__()</code>	(<i>cachetclient.v1.incident_updates.IncidentUpdate</i> method), 27
<code>__init__()</code>	(<i>cachetclient.v1.incident_updates.IncidentUpdatesManager</i> method), 28
<code>__init__()</code>	(<i>cachetclient.v1.incidents.Incident</i> method), 23
<code>__init__()</code>	(<i>cachetclient.v1.incidents.IncidentManager</i> method), 25
<code>__init__()</code>	(<i>cachetclient.v1.metric_points.MetricPoint</i> method), 33
<code>__init__()</code>	(<i>cachetclient.v1.metric_points.MetricPointsManager</i> method), 34
<code>__init__()</code>	(<i>cachetclient.v1.metrics.Metric</i> method), 30
<code>__init__()</code>	(<i>cachetclient.v1.metrics.MetricsManager</i> method), 32
<code>__init__()</code>	(<i>cachetclient.v1.ping.PingManager</i> method), 9
<code>__init__()</code>	(<i>cachetclient.v1.schedules.ScheduleManager</i> method), 37
<code>__init__()</code>	(<i>cachetclient.v1.subscribers.Subscriber</i> method), 11
<code>__init__()</code>	(<i>cachetclient.v1.subscribers.SubscriberManager</i> method), 13
<code>__init__()</code>	(<i>cachetclient.v1.version.Version</i> method), 9
<code>__init__()</code>	(<i>cachetclient.v1.version.VersionManager</i> method), 10
<code>add_tag()</code>	(<i>cachetclient.v1.components.Component</i> method), 14
<code>add_tags()</code>	(<i>cachetclient.v1.components.Component</i> method), 14
<code>attrs</code>	(<i>cachetclient.v1.component_groups.ComponentGroup</i> attribute), 20
<code>attrs</code>	(<i>cachetclient.v1.components.Component</i> attribute), 16
<code>attrs</code>	(<i>cachetclient.v1.incident_updates.IncidentUpdate</i> attribute), 27
<code>attrs</code>	(<i>cachetclient.v1.incidents.Incident</i> attribute), 23
<code>attrs</code>	(<i>cachetclient.v1.metric_points.MetricPoint</i> attribute), 34
<code>attrs</code>	(<i>cachetclient.v1.metrics.Metric</i> attribute), 31
<code>attrs</code>	(<i>cachetclient.v1.schedules.Schedule</i> attribute), 37
<code>attrs</code>	(<i>cachetclient.v1.subscribers.Subscriber</i> attribute), 12
<code>attrs</code>	(<i>cachetclient.v1.version.Version</i> attribute), 10

C

<code>cachetclient</code>	(module), 7
<code>cachetclient.v1.component_groups</code>	(module), 19
<code>cachetclient.v1.components</code>	(module), 14
<code>cachetclient.v1.enums</code>	(module), 7
<code>cachetclient.v1.incident_updates</code>	(module), 27
<code>cachetclient.v1.incidents</code>	(module), 23
<code>cachetclient.v1.metric_points</code>	(module), 33
<code>cachetclient.v1.metrics</code>	(module), 30
<code>cachetclient.v1.ping</code>	(module), 8

cachetclient.v1.schedules (module), 36
 cachetclient.v1.subscribers (module), 11
 cachetclient.v1.version (module), 9
 calc_type (cachetclient.v1.metrics.Metric attribute), 31
 calculated_value (cachetclient.v1.metric_points.MetricPoint attribute), 34
 Client() (in module cachetclient), 7
 collapsed (cachetclient.v1.component_groups.ComponentGroup attribute), 20
 completed_at (cachetclient.v1.schedules.Schedule attribute), 37
 COMPONENT_GROUP_COLLAPSED_FALSE (in module cachetclient.v1.enums), 8
 COMPONENT_GROUP_COLLAPSED_NOT_OPERATIONAL (in module cachetclient.v1.enums), 8
 COMPONENT_GROUP_COLLAPSED_TRUE (in module cachetclient.v1.enums), 8
 component_id (cachetclient.v1.incidents.Incident attribute), 23
 COMPONENT_STATUS_LIST (in module cachetclient.v1.enums), 7
 COMPONENT_STATUS_MAJOR_OUTAGE (in module cachetclient.v1.enums), 7
 COMPONENT_STATUS_OPERATIONAL (in module cachetclient.v1.enums), 7
 COMPONENT_STATUS_PARTIAL_OUTAGE (in module cachetclient.v1.enums), 7
 COMPONENT_STATUS_PERFORMANCE_ISSUES (in module cachetclient.v1.enums), 7
 count() (cachetclient.v1.component_groups.ComponentGroupManager method), 21
 count() (cachetclient.v1.components.ComponentManager method), 18
 count() (cachetclient.v1.incident_updates.IncidentUpdatesManager method), 29
 count() (cachetclient.v1.incidents.IncidentManager method), 26
 count() (cachetclient.v1.metric_points.MetricPointsManager method), 35
 count() (cachetclient.v1.metrics.MetricsManager method), 32
 count() (cachetclient.v1.schedules.ScheduleManager method), 38
 count() (cachetclient.v1.subscribers.SubscriberManager method), 13
 counter (cachetclient.v1.metric_points.MetricPoint attribute), 34
 create() (cachetclient.v1.component_groups.ComponentGroupManager method), 21
 create() (cachetclient.v1.components.ComponentManager method), 17
 create() (cachetclient.v1.incident_updates.IncidentUpdatesManager method), 28
 create() (cachetclient.v1.incidents.IncidentManager method), 25
 create() (cachetclient.v1.metric_points.MetricPointsManager method), 35
 create() (cachetclient.v1.metrics.MetricsManager method), 32
 create() (cachetclient.v1.schedules.ScheduleManager method), 37
 create() (cachetclient.v1.subscribers.SubscriberManager method), 13
 created_at (cachetclient.v1.component_groups.ComponentGroup attribute), 20
 created_at (cachetclient.v1.components.Component attribute), 17
 created_at (cachetclient.v1.incident_updates.IncidentUpdate attribute), 28
 created_at (cachetclient.v1.incidents.Incident attribute), 24
 created_at (cachetclient.v1.metric_points.MetricPoint attribute), 34
 created_at (cachetclient.v1.metrics.Metric attribute), 31
 created_at (cachetclient.v1.subscribers.Subscriber attribute), 12

D

default_value (cachetclient.v1.metrics.Metric attribute), 31
 default_view (cachetclient.v1.metrics.Metric attribute), 31
 del_tag() (cachetclient.v1.components.Component method), 15
 delete() (cachetclient.v1.component_groups.ComponentGroup method), 20
 delete() (cachetclient.v1.component_groups.ComponentGroupManager method), 22
 delete() (cachetclient.v1.components.Component method), 15
 delete() (cachetclient.v1.components.ComponentManager method), 18
 delete() (cachetclient.v1.incident_updates.IncidentUpdate method), 27
 delete() (cachetclient.v1.incident_updates.IncidentUpdatesManager method), 29
 delete() (cachetclient.v1.incidents.Incident method), 26
 delete() (cachetclient.v1.incidents.IncidentManager method), 26
 delete() (cachetclient.v1.metric_points.MetricPoint attribute), 33

`delete()` (*cachetclient.v1.metric_points.MetricPointsManager method*), 35
`delete()` (*cachetclient.v1.metrics.Metric method*), 30
`delete()` (*cachetclient.v1.metrics.MetricsManager method*), 32
`delete()` (*cachetclient.v1.schedules.Schedule method*), 36
`delete()` (*cachetclient.v1.schedules.ScheduleManager method*), 38
`delete()` (*cachetclient.v1.subscribers.Subscriber method*), 12
`delete()` (*cachetclient.v1.subscribers.SubscriberManager method*), 13
`deleted_at` (*cachetclient.v1.incidents.Incident attribute*), 24
`description` (*cachetclient.v1.components.Component attribute*), 16
`description` (*cachetclient.v1.metrics.Metric attribute*), 31
`display_chart` (*cachetclient.v1.metrics.Metric attribute*), 31

E

`email` (*cachetclient.v1.subscribers.Subscriber attribute*), 12
`enabled` (*cachetclient.v1.components.Component attribute*), 16
`enabled_components` (*cachetclient.v1.component_groups.ComponentGroup attribute*), 20

G

`get()` (*cachetclient.v1.component_groups.ComponentGroupManager method*), 19
`get()` (*cachetclient.v1.component_groups.ComponentGroupManager method*), 22
`get()` (*cachetclient.v1.components.Component method*), 15
`get()` (*cachetclient.v1.components.ComponentManager method*), 18
`get()` (*cachetclient.v1.incident_updates.IncidentUpdate method*), 27
`get()` (*cachetclient.v1.incident_updates.IncidentUpdatesManager method*), 29
`get()` (*cachetclient.v1.incidents.Incident method*), 23
`get()` (*cachetclient.v1.incidents.IncidentManager method*), 26
`get()` (*cachetclient.v1.metric_points.MetricPoint method*), 33
`get()` (*cachetclient.v1.metrics.Metric method*), 30
`get()` (*cachetclient.v1.metrics.MetricsManager method*), 32
`get()` (*cachetclient.v1.ping.PingManager method*), 9
`get()` (*cachetclient.v1.schedules.Schedule method*), 36
`get()` (*cachetclient.v1.schedules.ScheduleManager method*), 38
`get()` (*cachetclient.v1.subscribers.Subscriber method*), 12
`get()` (*cachetclient.v1.version.Version method*), 9
`get()` (*cachetclient.v1.version.VersionManager method*), 10
`group_id` (*cachetclient.v1.components.Component attribute*), 16
H
`has_tag()` (*cachetclient.v1.components.Component method*), 15
`human_status` (*cachetclient.v1.incident_updates.IncidentUpdate attribute*), 28
`human_status` (*cachetclient.v1.incidents.Incident attribute*), 24
I
`id` (*cachetclient.v1.component_groups.ComponentGroup attribute*), 20
`id` (*cachetclient.v1.components.Component attribute*), 16
`id` (*cachetclient.v1.incident_updates.IncidentUpdate attribute*), 27
`id` (*cachetclient.v1.incidents.Incident attribute*), 23
`id` (*cachetclient.v1.metric_points.MetricPoint attribute*), 34
`id` (*cachetclient.v1.metrics.Metric attribute*), 31
`id` (*cachetclient.v1.schedules.Schedule attribute*), 36
`id` (*cachetclient.v1.subscribers.Subscriber attribute*), 12
`INCIDENT_FIXED` (in module *cachetclient.v1.enums*), 8
`incident_id` (*cachetclient.v1.incident_updates.IncidentUpdate attribute*), 28
`INCIDENT_IDENTIFIED` (in module *cachetclient.v1.enums*), 8
`INCIDENT_INVESTIGATING` (in module *cachetclient.v1.enums*), 8
`INCIDENT_SCHEDULED` (in module *cachetclient.v1.enums*), 8
`incident_status_human()` (in module *cachetclient.v1.enums*), 8
`INCIDENT_WATCHING` (in module *cachetclient.v1.enums*), 8
`instance_from_dict()` (*cachetclient.v1.component_groups.ComponentGroupManager method*), 22
`instance_from_dict()` (*cachetclient.v1.components.ComponentManager method*), 18

<code>instance_from_dict()</code>	(cachet-client.v1.incident_updates.IncidentUpdatesManager method), 29	<code>instance_list_from_json()</code>	(cachet-client.v1.incident_updates.IncidentUpdatesManager method), 30
<code>instance_from_dict()</code>	(cachet-client.v1.incidents.IncidentManager method), 26	<code>instance_list_from_json()</code>	(cachet-client.v1.incidents.IncidentManager method), 27
<code>instance_from_dict()</code>	(cachet-client.v1.metric_points.MetricPointsManager method), 35	<code>instance_list_from_json()</code>	(cachet-client.v1.metric_points.MetricPointsManager method), 35
<code>instance_from_dict()</code>	(cachet-client.v1.metrics.MetricsManager method), 32	<code>instance_list_from_json()</code>	(cachet-client.v1.metrics.MetricsManager method), 33
<code>instance_from_dict()</code>	(cachet-client.v1.schedules.ScheduleManager method), 38	<code>instance_list_from_json()</code>	(cachet-client.v1.schedules.ScheduleManager method), 38
<code>instance_from_dict()</code>	(cachet-client.v1.subscribers.SubscriberManager method), 13	<code>instance_list_from_json()</code>	(cachet-client.v1.subscribers.SubscriberManager method), 14
<code>instance_from_dict()</code>	(cachet-client.v1.version.VersionManager method), 11	<code>instance_list_from_json()</code>	(cachet-client.v1.version.VersionManager method), 10
<code>instance_from_json()</code>	(cachet-client.v1.component_groups.ComponentGroupManager method), 22	<code>is_collapsed</code>	(cachet-client.v1.component_groups.ComponentGroup attribute), 20
<code>instance_from_json()</code>	(cachet-client.v1.components.ComponentManager method), 18	<code>is_global</code>	(cachetclient.v1.subscribers.Subscriber attribute), 12
<code>instance_from_json()</code>	(cachet-client.v1.incident_updates.IncidentUpdatesManager method), 30	<code>is_open</code>	(cachetclient.v1.component_groups.ComponentGroup attribute), 20
<code>instance_from_json()</code>	(cachet-client.v1.incidents.IncidentManager method), 26	<code>is_operational</code>	(cachet-client.v1.component_groups.ComponentGroup attribute), 20
<code>instance_from_json()</code>	(cachet-client.v1.metric_points.MetricPointsManager method), 35	L	
<code>instance_from_json()</code>	(cachet-client.v1.metrics.MetricsManager method), 33	<code>latest</code>	(cachetclient.v1.version.Version attribute), 10
<code>instance_from_json()</code>	(cachet-client.v1.schedules.ScheduleManager method), 38	<code>link</code>	(cachetclient.v1.components.Component attribute), 16
<code>instance_from_json()</code>	(cachet-client.v1.subscribers.SubscriberManager method), 14	<code>list()</code>	(cachetclient.v1.component_groups.ComponentGroupManager method), 21
<code>instance_from_json()</code>	(cachet-client.v1.version.VersionManager method), 11	<code>list()</code>	(cachetclient.v1.components.ComponentManager method), 18
<code>instance_list_from_json()</code>	(cachet-client.v1.component_groups.ComponentGroupManager method), 22	<code>list()</code>	(cachetclient.v1.incident_updates.IncidentUpdatesManager method), 29
<code>instance_list_from_json()</code>	(cachet-client.v1.components.ComponentManager method), 19	<code>list()</code>	(cachetclient.v1.incidents.IncidentManager method), 26
		<code>list()</code>	(cachetclient.v1.metric_points.MetricPointsManager method), 35
		<code>list()</code>	(cachetclient.v1.metrics.MetricsManager method), 32
		<code>list()</code>	(cachetclient.v1.schedules.ScheduleManager method), 38
		<code>list()</code>	(cachetclient.v1.subscribers.SubscriberManager method), 13
		<code>lowest_human_status</code>	(cachet-client.v1.component_groups.ComponentGroup attribute), 20

attribute), 20

M

message (cachetclient.v1.incident_updates.IncidentUpdate attribute), 28

message (cachetclient.v1.incidents.Incident attribute), 24

message (cachetclient.v1.schedules.Schedule attribute), 37

metric_id (cachetclient.v1.metric_points.MetricPoint attribute), 34

N

name (cachetclient.v1.component_groups.ComponentGroup attribute), 20

name (cachetclient.v1.components.Component attribute), 16

name (cachetclient.v1.incidents.Incident attribute), 24

name (cachetclient.v1.metrics.Metric attribute), 31

name (cachetclient.v1.schedules.Schedule attribute), 36

notify (cachetclient.v1.incidents.Incident attribute), 24

O

occurred_at (cachetclient.v1.incidents.Incident attribute), 24

on_latest (cachetclient.v1.version.Version attribute), 10

order (cachetclient.v1.component_groups.ComponentGroup attribute), 20

order (cachetclient.v1.components.Component attribute), 16

order (cachetclient.v1.metrics.Metric attribute), 31

P

path (cachetclient.v1.component_groups.ComponentGroupManager attribute), 23

path (cachetclient.v1.components.ComponentManager attribute), 19

path (cachetclient.v1.incident_updates.IncidentUpdatesManager attribute), 30

path (cachetclient.v1.incidents.IncidentManager attribute), 27

path (cachetclient.v1.metric_points.MetricPointsManager attribute), 36

path (cachetclient.v1.metrics.MetricsManager attribute), 33

path (cachetclient.v1.ping.PingManager attribute), 9

path (cachetclient.v1.schedules.ScheduleManager attribute), 39

path (cachetclient.v1.subscribers.SubscriberManager attribute), 14

path (cachetclient.v1.version.VersionManager attribute), 11

permalink (cachetclient.v1.incident_updates.IncidentUpdate attribute), 28

places (cachetclient.v1.metrics.Metric attribute), 31

points (cachetclient.v1.metrics.Metric attribute), 31

R

resource_class (cachetclient.v1.component_groups.ComponentGroupManager attribute), 23

resource_class (cachetclient.v1.components.ComponentManager attribute), 19

resource_class (cachetclient.v1.incident_updates.IncidentUpdatesManager attribute), 30

resource_class (cachetclient.v1.incidents.IncidentManager attribute), 27

resource_class (cachetclient.v1.metric_points.MetricPointsManager attribute), 36

resource_class (cachetclient.v1.metrics.MetricsManager attribute), 33

resource_class (cachetclient.v1.ping.PingManager attribute), 9

resource_class (cachetclient.v1.schedules.ScheduleManager attribute), 39

resource_class (cachetclient.v1.subscribers.SubscriberManager attribute), 14

resource_class (cachetclient.v1.version.VersionManager attribute), 11

S

SCHEDULE_STATUS_COMPLETE (in module cachetclient.v1.enums), 8

SCHEDULE_STATUS_IN_PROGRESS (in module cachetclient.v1.enums), 8

SCHEDULE_STATUS_UPCOMING (in module cachetclient.v1.enums), 8

scheduled_at (cachetclient.v1.incidents.Incident attribute), 24

scheduled_at (cachetclient.v1.schedules.Schedule attribute), 37

set_tags () (cachetclient.v1.components.Component method), 15

status (cachetclient.v1.components.Component attribute), 16

status (cachetclient.v1.incident_updates.IncidentUpdate attribute), 28

status (cachetclient.v1.incidents.Incident attribute), 24

`status` (*cachetclient.v1.schedules.Schedule* attribute), 37

`status_name` (*cachetclient.v1.components.Component* attribute), 16

`stickied` (*cachetclient.v1.incidents.Incident* attribute), 24

`suffix` (*cachetclient.v1.metrics.Metric* attribute), 31

T

`tag_names` (*cachetclient.v1.components.Component* attribute), 16

`tag_slugs` (*cachetclient.v1.components.Component* attribute), 17

`tags` (*cachetclient.v1.components.Component* attribute), 16

`threshold` (*cachetclient.v1.metrics.Metric* attribute), 31

U

`update()` (*cachetclient.v1.component_groups.ComponentGroup* method), 19

`update()` (*cachetclient.v1.component_groups.ComponentGroupManager* method), 21

`update()` (*cachetclient.v1.components.Component* method), 15

`update()` (*cachetclient.v1.components.ComponentManager* method), 17

`update()` (*cachetclient.v1.incident_updates.IncidentUpdate* method), 27

`update()` (*cachetclient.v1.incident_updates.IncidentUpdatesManager* method), 29

`update()` (*cachetclient.v1.incidents.Incident* method), 23

`update()` (*cachetclient.v1.incidents.IncidentManager* method), 25

`update()` (*cachetclient.v1.metric_points.MetricPoint* method), 33

`update()` (*cachetclient.v1.metrics.Metric* method), 30

`update()` (*cachetclient.v1.schedules.Schedule* method), 36

`update()` (*cachetclient.v1.schedules.ScheduleManager* method), 37

`update()` (*cachetclient.v1.subscribers.Subscriber* method), 11

`updated_at` (*cachetclient.v1.component_groups.ComponentGroup* attribute), 20

`updated_at` (*cachetclient.v1.components.Component* attribute), 17

`updated_at` (*cachetclient.v1.incident_updates.IncidentUpdate* attribute), 28

`updated_at` (*cachetclient.v1.incidents.Incident* attribute), 24

`updated_at` (*cachetclient.v1.metric_points.MetricPoint* attribute), 34

`updated_at` (*cachetclient.v1.metrics.Metric* attribute), 31

`updated_at` (*cachetclient.v1.subscribers.Subscriber* attribute), 12

`updates()` (*cachetclient.v1.incidents.Incident* method), 23

`user_id` (*cachetclient.v1.incident_updates.IncidentUpdate* attribute), 28

V

`value` (*cachetclient.v1.metric_points.MetricPoint* attribute), 34

`value` (*cachetclient.v1.version.Version* attribute), 10

`verified_at` (*cachetclient.v1.subscribers.Subscriber* attribute), 12

`verify_code` (*cachetclient.v1.subscribers.Subscriber* attribute), 12

`visible` (*cachetclient.v1.component_groups.ComponentGroup* attribute), 21

`visible` (*cachetclient.v1.incidents.Incident* attribute), 24

`visible` (*cachetclient.v1.metrics.Metric* attribute), 31