python-StatusPageio Documentation

Release 0.0.1

Luke Morfitt

February 09, 2016

Contents

StatusPage.oi API V1 library client for Python. Based on the documentaion from https://doers.statuspage.io/api/v1/

Provides most of the funcitonality for the http://statuspage.io api's via handy python code.

CHAPTER 1

Tests

Sorry. These need to be written.

CHAPTER 2

Thanks

Thank you to the BaseCRM development team who created the majority of the code for this project. We forked the code as the aritectural style worked really well for this project. Please see https://github.com/basecrm/basecrm-python for more details

Thank you so much!

CHAPTER 3

License

MIT

CHAPTER 4

Bug Reports

Report here.

Contents:

4.1 Installation

Statuspageio package can be installed either via pip or easy_install:

```
$ pip install --upgrade statuspageio
```

or

```
$ easy_install statuspageio
```

You can install from the source code as well. First clone the repo and then execute:

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

After installing, import `statuspageio` package:

```
import statuspageio
```

4.2 Usage

```
import statuspageio

# Then we instantiate a client (as shown below)
```

4.2.1 Build a client

Using this api without authentication gives an error

```
client = statuspageio.Client(api_key='<YOUR_PERSONAL_API_KEY>', page_id=<YOUR_PERSONAL_PAGE_ID')
```

or with User management enabled

```
client = statuspageio.Client(api_key='<YOUR_PERSONAL_API_KEY>',
    page_id=<YOUR_PERSONAL_PAGE_ID',organization_id=<YOUR_PERSONAL_ORGANIZATION_ID)
```

Note: The organization_id can be found by login in to https://manage.statuspage.io/ and clicking team members. It's part of the url https://manage.statuspage.io/organizations/REMOVED/team

4.2.2 Client Options

The following options are available while instantiating a client:

- **api_key**: Personal API Key
- **page_id**: Personal page id
- **organization_id**: Personal organization id, used for managing users.
- **base_url**: Base url for the api
- **user_agent**: Default user-agent for all requests
- **timeout**: Request timeout
- **verbose**: Verbose/debug mode

4.2.3 Architecture

The library follows few architectural principles you should understand before digging deeper. 1. Interactions with resources are done via service objects. 2. Service objects are exposed as properties on client instances. 3. Service objects expose resource-oriented actions. 4. Actions return dictionaries that support attribute-style access, a la JavaScript (thanks to Bunch and it's form Munch).

For example, to interact with components API you will use `statuspageio.ComponentsService`, which you can get if you call:

```
client = statuspageio.Client(api_key='<YOUR_PERSONAL_API_KEY>', page_id=<YOUR_PERSONAL_PAGE_ID')
client.components # statuspageio.ComponentsService
```

To retrieve list of resources and use filtering you will call `#list` method:

```
client = statuspageio.Client(api_key='<YOUR_PERSONAL_API_KEY>', page_id=<YOUR_PERSONAL_PAGE_ID')
client.components.list() # list(dict|Munch)
```

4.2.4 Resources and actions

Documentation for every action can be found in `statuspageio/services.py` file or in the following *statuspageio package*

In short we have implimented the folowing API functions at this time.

Page

Actions:

- Retrieve the page information - `client.page.get`
- Update the page information - `client.page.update`

Components

Actions:

- List all components - `client.components.list`
- Create a component - `client.components.create`
- Update a component - `client.components.update`
- Delete a component - `client.components.delete`

Incidents

- List all incidents - `client.incidents.list`
- List unresolved incidents - `client.incidents.list_unresolved`
- List scheduled incidents - `client.incidents.list_scheduled`
- Create a incident - `client.incidents.create`
- Update a incident - `client.incidents.update`
- Delete a incident - `client.incidents.delete`

Subscribers

- List all subscribers - `client.subscribers.list`
- Create a subscribers - `client.subscribers.create`
- Delete a subscribers - `client.subscribers.delete`

Metrics

- List all available providers - `client.metrics.list_available`
- List linked providers - `client.metrics.list_linked`
- List metrics for provider - `client.metrics.list_metrics_for_provider`
- Create a custom metric - `client.metrics.create`
- Delete a custom metric - `client.metrics.delete`
- Submit data for a custom metric - `client.metrics.submit_data`

Users

- List all Users - `client.users.list`
- Create a User - `client.users.create`
- Delete a User - `client.users.delete`

4.3 statuspageio package

4.3.1 Submodules

4.3.2 statuspageio.client module

class statuspageio.client.**Client**(***options*)
Bases: object

The *Client* is the entry point to all services and actions.

:attribute Configuration config: Current StatusPage.io client configuration. :attribute HttpClient http_client: Http client.

Copyright

3. 2016 by GameSparks Developers, and partial code by 2015, BaseCRM developers (developers@getbase.com).

License MIT, see LICENSE for more details.

components

incidents

metrics

pages

subscribers

users

4.3.3 statuspageio.configuration module

class statuspageio.configuration.**Configuration**(***options*)
Bases: object

validate()
Validates whether a configuration is valid.

Return type bool

Raises

- **ConfigurationError** – if no api_key provided.
- **ConfigurationError** – if no page_id provided.

:warns 'No organization_id provided.' if no organization_id provided

4.3.4 statuspageio.errors module

exception statuspageio.errors.**BaseError**(*http_status*, *errors_payload*)
Bases: exceptions.Exception

This is the base class for all the errors returned by StatusPage.io servers.

Classes derived from this class: * *RequestError* * *ResourceError* * *ServerError*

Attribute int http_status Http status code.

Attribute str logref Request unique identifier.

Attribute list errors List of `Munch` objects repsenting returned errors.

Each error object has following attributes: :attribute str code: The error code. :attribute str message: Human redable error description. :attribute str details: (optional) Detailed description. :attribute str resource: (optional) Resource name the error relates to. :attribute str field: (optional) Field name of the resource the error relates to, in the JSON pointer format.

exception `statuspageio.errors.`**`ConfigurationError`**
Bases: `exceptions.Exception`

Exception raised in case of invalid client configuration e.g. no access token provided, invalid access token, invalid base url etc.

exception `statuspageio.errors.`**`RateLimitError`**
Bases: `exceptions.Exception`

Exception raised when the rate limit was exceeded.

exception `statuspageio.errors.`**`RequestError`**(*http_status*, *errors_payload*)
Bases: *statuspageio.errors.BaseError*

Exception raised if the request was invalid e.g. unknown query parameter, invalid request's body envelope etc.

exception `statuspageio.errors.`**`ResourceError`**(*http_status*, *errors_payload*)
Bases: *statuspageio.errors.BaseError*

Exception raised in case of any resource related error.

exception `statuspageio.errors.`**`ServerError`**(*http_status*, *errors_payload*)
Bases: *statuspageio.errors.BaseError*

Exception raised if Base CRM's servers encountered an unexpected condition.

4.3.5 statuspageio.http_client module

class `statuspageio.http_client.`**`HttpClient`**(*config*)
Bases: `object`

Wrapper over **:module:'requests'** that understands StatusPage.io envelope, encoding and decoding schema.

`API_VERSION` = **'/v1'**

`delete`(*url*, *params=None*, ***kwargs*)
Send a DELETE request.

Parameters

- **`url`** (*str*) – Sub URL for the request. You MUST not specify neither base url nor api version prefix.
- **`params`** (*dict*) – (optional) Dictionary of query parameters.
- **`**kwargs`** (*dict*) – (optional) Other parameters which are directly passed to `requests.request()`.

Returns Tuple of three elements: (http status code, headers, response - either parsed json or plain text)

Return type tuple

`enable_logging`()

get(*url*, *params=None*, ***kwargs*)
Send a GET request.

Parameters

- **url** (*str*) – Sub URL for the request. You MUST not specify neither base url nor api version prefix.
- **params** (*dict*) – (optional) Dictionary of query parameters.
- ****kwargs** (*dict*) – (optional) Other parameters which are directly passed to `requests.request()`.

Returns Tuple of three elements: (http status code, headers, response - either parsed json or plain text)

Return type tuple

handle_error_response(*resp*)

patch(*url*, *body=None*, ***kwargs*)
Send a PATCH request.

Parameters

- **url** (*str*) – Sub URL for the request. You MUST not specify neither base url nor api version prefix.
- **body** (*dict*) – (optional) Dictionary of body attributes that will be wrapped with envelope and json encoded.
- ****kwargs** (*dict*) – (optional) Other parameters which are directly passed to `requests.request()`.

Returns Tuple of three elements: (http status code, headers, response - either parsed json or plain text)

Return type tuple

post(*url*, *body=None*, ***kwargs*)
Send a POST request.

Parameters

- **url** (*str*) – Sub URL for the request. You MUST not specify neither base url nor api version prefix.
- **body** (*dict*) – (optional) Dictionary of body attributes that will be wrapped with envelope and json encoded.
- ****kwargs** (*dict*) – (optional) Other parameters which are directly passed to `requests.request()`.

Returns Tuple of three elements: (http status code, headers, response - either parsed json or plain text)

Return type tuple

put(*url*, *body=None*, ***kwargs*)
Send a PUT request.

Parameters

- **url** (*str*) – Sub URL for the request. You MUST not specify neither base url nor api version prefix.

- **body** (*dict*) – (optional) Dictionary of body attributes that will be wrapped with envelope and json encoded.
- ****kwargs** (*dict*) – (optional) Other parameters which are directly passed to `requests.request()`.

Returns Tuple of three elements: (http status code, headers, response - either parsed json or plain text)

Return type tuple

request(*method*, *url*, *params=None*, *body=None*, ***kwargs*)

Send an HTTP request.

The **:param:'params'** will be properly encoded, as well as **:param:'body'** which will be wrapped with envelope the API expects and json encoded.

When you get a reponse the method will try to json decode the response, if the media type represents json, unwrap the envelope and munchify what has left, for JavaScript like access.

Parameters

- **url** (*str*) – Sub URL for the request. You MUST not specify neither base url nor api version prefix.
- **params** (*dict*) – (optional) Dictionary of query parameters.
- **body** (*dict*) – (optional) Dictionary of body attributes that will be wrapped with envelope and json encoded.
- ****kwargs** (*dict*) – (optional) Other parameters which are directly passed to `requests.request()`.

Raises

- **RequestError** – if authentication failed, invalid query parameter etc.
- **RateLimitError** – if rate limit exceeded.
- **ResourceError** – if requests payload included invalid attributes or were missing.
- **ServerError** – if StatusPage.io backend servers encounterered an unexpected condition.

Returns Tuple of three elements: (http status code, headers, response - either parsed json or plain text)

Return type tuple

Keyword Arguments

- **param dict headers** (optional) Dictionary of headers. Default: `{}`.
- **param bool raw** (optional) Whether to wrap and uwrap the envelope. Default: `False`.

static unwrap_envelope(*body*)

static wrap_envelope(*container*, *body*)

Wrap the body with the correct container to match the API

4.3.6 statuspageio.services module

class `statuspageio.services.`**ComponentsService**(*http_client*, *page_id*)

Bases: `object`

`statuspageio.ComponentsService` is used by `statuspageio.Client` to make actions related to Components resource.

Normally you won't instantiate this class directly.

OPTS_KEYS_TO_PERSIST = ['name', 'description', 'group_id', 'status']

create(***kwargs*)
Create a component

Creates component If the specified contact does not exist, the request will return an error

Calls `post pages/{page_id}/components.json`

Parameters ****kwargs** (*dict*) – component attributes to update.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

delete(*component_id*)
Delete a component

Deletes a component If the specified contact does not exist, the request will return an error

Calls `delete pages/{page_id}/components/{component_id}.json`

Parameters **component_id** (*int*) – Unique identifier of a component.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

http_client

list()
List components

Lists components and their information If the specified contact does not exist, the request will return an error

Calls `get pages/{page_id}/components/{component_id}.json`

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

update(*component_id*, ***kwargs*)
Update a component

Updates component information If the specified contact does not exist, the request will return an error

Calls `patch pages/{page_id}/components/{component_id}.json`

Parameters

- **component_id** (*int*) – Unique identifier of a component.
- ****kwargs** (*dict*) – component attributes to update.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

class statuspageio.services.**IncidentsService**(*http_client*, *page_id*)
Bases: object

statuspageio.IncidentsService is used by statuspageio.Client to make actions related to Incidents resource.

Normally you won't instantiate this class directly.

OPTS_KEYS_TO_PERSIST = ['name', 'description', 'group_id', 'status']

create(***kwargs*)
Create a incident

Calls post pages/{page_id}/incidents.json

Parameters ****kwargs** (*dict*) – incident attributes to update.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

create_scheduled(***kwargs*)
Create a scheduled incident

Calls post pages/{page_id}/incidents.json

Parameters ****kwargs** (*dict*) – incident attributes to update.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

delete(*incident_id*)
Remove a incident

Calls delete pages/{page_id}/incidents.json

Returns status code

Return type int

http_client

list()
List all incidents

Calls get pages/{page_id}/incidents.json

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

list_scheduled()
List scheduled incidents

Calls get pages/{page_id}/incidents/scheduled.json

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

list_unresolved()
List unresolved incidents

Calls `get pages/{page_id}/incidents/unresolved.json`

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

update(*incident_id*, ***kwargs*)
Update a incident

Updates incident information

NOTE: if either of status or message is modified, a new incident update will be generated. You should update both of these attributes at the same time to avoid two separate incident updates being generated. :param dict **kwargs: incident attributes to update. :calls: `patch /pages/[page_id]/incidents/[incident_id].json` :return: Status code :rtype: string

class `statuspageio.services.`**MetricsService**(*http_client*, *page_id*)
Bases: `object`

`statuspageio.MetricsService` is used by `statuspageio.Client` to make actions related to Metrics resource.

Normally you won't instantiate this class directly.

create(*provider_id=None*, ***kwargs*)
Create a custom metric

Calls `post /pages/[page_id]/metrics_providers/[metrics_provider_id]/metrics.json`

Parameters

- **provider_id** – The id of the custom provider or 'self' from the available providers list
- ****kwargs** (*dict*) – metic attributes to create.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

delete(*metric_id=None*)
Delete Custom Metric

Calls `delete /pages/[page_id]/metrics/[metric_id].json`

Parameters **metric_id** – The id of the custom metric.

Returns status code.

Return type int

delete_all_data(*metric_id=None*)
Delete All Metric Data

Calls `delete /pages/[page_id]/metrics/[metric_id]/data.json`

Parameters **metric_id** – The id of the custom metric.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

http_client

list_available()
List available metric providers :calls: `get /metrics_providers.json` :return: Dictionary that support attriubte-style access and represents updated Component resource. :rtype: dict

list_linked()
List linked metric providers :calls: `get /pages/[page_id]/metrics_providers.json` :return: Dictionary that support attriubte-style access and represents updated Component resource. :rtype: dict

list_metrics_for_provider(*provider_id=None*)
List metrics for a linked metric provider :params provider_id This is the ID from the provider you are looking up :calls: `/pages/{page_id}/metrics_providers/{metrics_provider_id}/metrics.json` :return: Dictionary that support attriubte-style access and represents updated Component resource. :rtype: dict

submit_data(*metric_id=None*, ***kwargs*)
Create a custom metric

Calls `post /pages/{page_id}/metrics/{metric_id}/data.json`

Parameters

- **metric_id** – The id of the custom metric.
- ****kwargs** (*dict*) – metic attributes to create.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

class statuspageio.services.**PageService**(*http_client*, *page_id*)
Bases: `object`

`statuspageio.PageService` is used by `statuspageio.Client` to make actions related to Page resource.

Normally you won't instantiate this class directly.

OPTS_KEYS_TO_PERSIST = ['name', 'url', 'notifications_from_email']

get()
Get page details

Gets page information If the specified page does not exist, the request will return an error

Calls `get pages/{page_id}.json`

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

http_client

update(***kwargs*)
Update page details

Updates page information If the specified page does not exist, the request will return an error

Calls `patch pages/{page_id}.json`

Parameters ****kwargs** (*dict*) – component attributes to update.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

class statuspageio.services.**SubscribersService**(*http_client*, *page_id*)
Bases: object

statuspageio.SubscribersService is used by statuspageio.Client to make actions related to Subscriber resource.

Normally you won't instantiate this class directly.

OPTS_KEYS_TO_PERSIST = ['name', 'description', 'group_id', 'status']

create(***kwargs*)
Create a subscriber

Calls post pages/{page_id}/subscribers.json

Parameters ****kwargs** (*dict*) – subscriber attributes to update.

Returns Dictionary that support attriubte-style access and represents updated Component resource.

Return type dict

delete(*subscriber_id=None*)
Create a subscriber

Calls delete pages/{page_id}/subscribers.json

:param subscriber_id :return: status code :rtype: int

http_client

list()
List subscribers

Lists all of the current subscribers :calls: get /pages/[page_id]/subscribers.json :return: Dictionary that support attriubte-style access and represents updated Component resource. :rtype: dict

class statuspageio.services.**UsersService**(*http_client*, *organization_id*)
Bases: object

statuspageio.UsersService is used by statuspageio.Client to make actions related to Users resource.

Normally you won't instantiate this class directly.

create(***kwargs*)
Create a user

Calls post /organizations/[organization_id]/users.json

Parameters ****kwargs** (*dict*) – Users attributes to create.

Returns Dictionary that support attriubte-style access and represents updated User resource.

Return type dict

delete(*user_id=None*)
Delete a User

Calls delete organizations/[organization_id]/users/[user_id].json

Parameters **user_id** – The id of the user to delete.

Returns status code.

Return type int

http_client

list()
List all users :calls: `get organizations/[organization_id]/users.json` :return: Dictionary that support attriubte-style access and represents User resource. :rtype: dict

4.3.7 statuspageio.version module

Current client version.

4.3.8 Module contents

StatusPage.io API V1 library client for Python.

Usage::

```
>>> import statuspageio
>>> client = statuspageio.Client(api_key=os.environ.get('STATUSPAGE_API_KEY')
>>> status = client.components.list()
>>> print status
```

copyright

3. 2016 by GameSparks TechOps (techops@gamesparks.com).

license MIT, see LICENSE for more details.

S

A

B

C

D

E

G

H

I

L

M

O

P

R

S

U

V

W