
pyrate Documentation

Release 0.3.5

Kim Thoenen

September 04, 2013

CONTENTS

1	Modules	1
1.1	Project Modules	1
1.2	pyrate Package	5
1.3	scripts Package	8
1.4	services Package	9
1.5	tests Package	11
2	Indices and tables	13
	Python Module Index	15

MODULES

1.1 Project Modules

1.1.1 pyrate Package

pyrate Package

main Module

class `pyrate.main.Pyrate`

Bases: `object`

This is the main class

Parameters

- **`http_methods`** (*list*) – List of available HTTP methods for this service
- **`return_formats`** (*list*) – List of available return formats for this service
- **`default_header_content`** – Default content for the request header
- **`default_body_content`** – Default content for the request body
- **`default_http_method`** (*string*) – Default HTTP method (will be used if none else is specified in request)
- **`default_return_format`** (*string*) – Default return format (will be used if none else is specified in request)
- **`connection_check_method`** (*string*) – Used by `check_connection()`
- **`auth_type`** (*string*) – The authentication type. Obsolete.
- **`base_url`** (*string*) – The base url for all api requests
- **`send_json`** (*bool*) – Whether the request body should be encoded with json

`auth_type` = `None`

`base_url` = `None`

`build_content` (*args*)

`check_connection` ()

`check_response_success` (*response*)

`connection_check_method` = `None`

create_basic_auth (*user, password*)

Creates the header content for HTTP Basic Authentication.

Parameters

- **user** – Username
- **password** – Password

Return type Base64-encoded auth string

default_body_content = None

default_header_content = None

default_http_method = None

default_return_format = None

do (*method, content=None, headers=None, http_method=None, return_format=None*)

do_request (*http_method, url, headers, body, return_format*)

get_oauth ()

handle_response (*response, return_format*)

http_methods = ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS']

parse_errors (*response*)

return_formats = ['json']

send_json = False

Subpackages

scripts Package

cliutils Module

`pyrate.scripts.cliutils.main()`

`pyrate.scripts.cliutils.setup_twitter_oauth(oauth_consumer_key,
oauth_consumer_secret)`

`pyrate.scripts.cliutils.twitter_oauth()`

services Package

github Module

`class pyrate.services.github.GithubPyrate(auth_user, auth_pass, default_http_method=None,
default_return_format=None)`

Bases: `pyrate.main.Pyrate`

auth_pass = ''

auth_type = 'BASIC_AUTH'

auth_user = ''

base_url = 'https://api.github.com/'

connection_check_method = ('GET', '#')

create_repo (*name, description=False, org_name=False, private=False*)

```
default_body_content = {}
delete_repo (name, org_name=False)
get_my_orgs ()
http_methods = ['GET', 'POST', 'PATCH', 'DELETE']
return_formats = []

send_json = True
exception pyrate.services.github.OrganisationNotFoundError
Bases: exceptions.Exception
```

harvest Module

```
class pyrate.services.harvest.HarvestPyrate (auth_user, auth_pass, organisation_name, default_http_method=None, default_return_format=None)

Bases: pyrate.main.Pyrate

auth_pass = ''
auth_type = 'BASIC_AUTH'
auth_user = ''
connection_check_method = ('GET', 'account/who_am_i')
default_body_content = {}
http_methods = ['GET', 'POST']
organisation_name = ''
return_formats = ['json']
```

mailchimp Module

```
exception pyrate.services.mailchimp.ListNotFoundError
Bases: exceptions.Exception
class pyrate.services.mailchimp.MailchimpPyrate (apikey, default_http_method=None, default_return_format=None)

Bases: pyrate.main.Pyrate

api_key = ''
auth_type = 'API_KEY'
check_response_success (response)
connection_check_method = ('POST', 'helper/ping')
default_header_content = {}
default_http_method = 'POST'
getListByName (list_name)
getLists (filters=None, start=None, limit=None, sort_field=None, sort_dir=None)
http_methods = ['POST']
parse_errors (response)
return_formats = ['JSON', 'XML', 'PHP']
```

```
send_json = True

subscribeToList(list_name, user_email, merge_vars=None, email_type=None, double_optin=None, update_existing=None, replace_interests=None, send_welcome=None)

unsubscribeFromList(list_name, user_email, delete_member=None, send_goodbye=None, send_notify=None)
```

twitter Module

```
class pyrate.services.twitter.TwitterPyrate(oauth_consumer_key, oauth_consumer_secret,
                                             oauth_token,          oauth_token_secret,
                                             default_http_method=None,          default_return_format=None)
```

Bases: `pyrate.main.Pyrate`

```
auth_type = 'OAUTH1'

base_url = 'https://api.twitter.com/1.1/'

check_response_success(response)

connection_check_method = ('GET', 'account/verify_credentials')

default_body_content = {}

default_header_content = {}

default_http_method = 'GET'

default_return_format = 'json'

get_oauth()

http_methods = ['GET', 'POST']

oauth_consumer_key = ''

oauth_consumer_secret = ''

oauth_token = ''

oauth_token_secret = ''

parse_errors(response)

return_formats = ['json']

tweet(status, in_reply_to_status_id=None, loc_lat=None, loc_long=None, place_id=None, display_coordinates=None, trim_user=None, include_entities=None)
```

tests Package

credentials Module

tests Module

1.2 pyrate Package

1.2.1 pyrate Package

1.2.2 main Module

class `pyrate.main.Pyrate`

Bases: `object`

This is the main class

Parameters

- **`http_methods`** (*list*) – List of available HTTP methods for this service
- **`return_formats`** (*list*) – List of available return formats for this service
- **`default_header_content`** – Default content for the request header
- **`default_body_content`** – Default content for the request body
- **`default_http_method`** (*string*) – Default HTTP method (will be used if none else is specified in request)
- **`default_return_format`** (*string*) – Default return format (will be used if none else is specified in request)
- **`connection_check_method`** (*string*) – Used by `check_connection()`
- **`auth_type`** (*string*) – The authentication type. Obsolete.
- **`base_url`** (*string*) – The base url for all api requests
- **`send_json`** (*bool*) – Whether the request body should be encoded with json

`auth_type` = `None`

`base_url` = `None`

`build_content` (*args*)

`check_connection` ()

`check_response_success` (*response*)

`connection_check_method` = `None`

`create_basic_auth` (*user, password*)

Creates the header content for HTTP Basic Authentication.

Parameters

- **`user`** – Username
- **`password`** – Password

Return type Base64-encoded auth string

`default_body_content` = `None`

`default_header_content` = `None`

`default_http_method` = `None`

`default_return_format` = `None`

`do` (*method, content=None, headers=None, http_method=None, return_format=None*)

```
do_request (http_method, url, headers, body, return_format)
get_oauth ()
handle_response (response, return_format)
http_methods = ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS']
parse_errors (response)
return_formats = ['json']
send_json = False
```

1.2.3 Subpackages

scripts Package

cliutils Module

```
pyrate.scripts.cliutils.main()
pyrate.scripts.cliutils.setup_twitter_oauth (oauth_consumer_key,
                                              oauth_consumer_secret)
pyrate.scripts.cliutils.twitter_oauth()
```

services Package

github Module

```
class pyrate.services.github.GithubPyrate (auth_user, auth_pass, default_http_method=None,
                                           default_return_format=None)
    Bases: pyrate.main.Pyrate
    auth_pass = ''
    auth_type = 'BASIC_AUTH'
    auth_user = ''
    base_url = 'https://api.github.com/'
    connection_check_method = ('GET', '#')
    create_repo (name, description=False, org_name=False, private=False)
    default_body_content = {}
    delete_repo (name, org_name=False)
    get_my_orgs ()
    http_methods = ['GET', 'POST', 'PATCH', 'DELETE']
    return_formats = []
    send_json = True

exception pyrate.services.github.OrganisationNotFoundError
    Bases: exceptions.Exception
```

harvest Module

```
class pyrate.services.harvest.HarvestPyrate(auth_user, auth_pass, organisation_name, default_http_method=None, default_return_format=None)

Bases: pyrate.main.Pyrate

auth_pass = ''
auth_type = 'BASIC_AUTH'
auth_user = ''
connection_check_method = ('GET', 'account/who_am_i')
default_body_content = {}
http_methods = ['GET', 'POST']
organisation_name = ''
return_formats = ['json']
```

mailchimp Module

```
exception pyrate.services.mailchimp.ListNotFoundError
Bases: exceptions.Exception

class pyrate.services.mailchimp.MailchimpPyrate(apikey, default_http_method=None, default_return_format=None)

Bases: pyrate.main.Pyrate

api_key = ''
auth_type = 'API_KEY'
check_response_success(response)
connection_check_method = ('POST', 'helper/ping')
default_header_content = {}
default_http_method = 'POST'
getListByName(list_name)
getLists(filters=None, start=None, limit=None, sort_field=None, sort_dir=None)
http_methods = ['POST']
parse_errors(response)
return_formats = ['JSON', 'XML', 'PHP']
send_json = True
subscribeToList(list_name, user_email, merge_vars=None, email_type=None, double_optin=None, update_existing=None, replace_interests=None, send_welcome=None)
unsubscribeFromList(list_name, user_email, delete_member=None, send_goodbye=None, send_notify=None)
```

twitter Module

```
class pyrate.services.twitter.TwitterPyrate(oauth_consumer_key, oauth_consumer_secret,
                                             oauth_token,          oauth_token_secret,
                                             default_http_method=None,      de-
                                             fault_return_format=None)
```

Bases: `pyrate.main.Pyrate`

auth_type = 'OAUTH1'

base_url = 'https://api.twitter.com/1.1/'

check_response_success (*response*)

connection_check_method = ('GET', 'account/verify_credentials')

default_body_content = {}

default_header_content = {}

default_http_method = 'GET'

default_return_format = 'json'

get_oauth ()

http_methods = ['GET', 'POST']

oauth_consumer_key = ''

oauth_consumer_secret = ''

oauth_token = ''

oauth_token_secret = ''

parse_errors (*response*)

return_formats = ['json']

tweet (*status, in_reply_to_status_id=None, loc_lat=None, loc_long=None, place_id=None, display_coordinates=None, trim_user=None, include_entities=None*)

tests Package

credentials Module

tests Module

1.3 scripts Package

1.3.1 cliutils Module

`pyrate.scripts.cliutils.main()`

`pyrate.scripts.cliutils.setup_twitter_oauth` (*oauth_consumer_key,*
oauth_consumer_secret)

`pyrate.scripts.cliutils.twitter_oauth()`

1.4 services Package

1.4.1 github Module

```
class pyrate.services.github.GithubPyrate(auth_user, auth_pass, default_http_method=None,
                                          default_return_format=None)

    Bases: pyrate.main.Pyrate
    auth_pass = ''
    auth_type = 'BASIC_AUTH'
    auth_user = ''
    base_url = 'https://api.github.com/'
    connection_check_method = ('GET', '#')
    create_repo(name, description=False, org_name=False, private=False)
    default_body_content = {}
    delete_repo(name, org_name=False)
    get_my_orgs()
    http_methods = ['GET', 'POST', 'PATCH', 'DELETE']
    return_formats = []
    send_json = True

exception pyrate.services.github.OrganisationNotFoundError
    Bases: exceptions.Exception
```

1.4.2 harvest Module

```
class pyrate.services.harvest.HarvestPyrate(auth_user, auth_pass, organisation_name,
                                             default_http_method=None, default_return_format=None)

    Bases: pyrate.main.Pyrate
    auth_pass = ''
    auth_type = 'BASIC_AUTH'
    auth_user = ''
    connection_check_method = ('GET', 'account/who_am_i')
    default_body_content = {}
    http_methods = ['GET', 'POST']
    organisation_name = ''
    return_formats = ['json']
```

1.4.3 mailchimp Module

exception `pyrate.services.mailchimp.ListNotFoundError`

Bases: `exceptions.Exception`

class `pyrate.services.mailchimp.MailchimpPyrate` (*apikey, default_http_method=None, default_return_format=None*)

Bases: `pyrate.main.Pyrate`

api_key = ''

auth_type = 'API_KEY'

check_response_success (*response*)

connection_check_method = ('POST', 'helper/ping')

default_header_content = {}

default_http_method = 'POST'

getListByName (*list_name*)

getLists (*filters=None, start=None, limit=None, sort_field=None, sort_dir=None*)

http_methods = ['POST']

parse_errors (*response*)

return_formats = ['JSON', 'XML', 'PHP']

send_json = True

subscribeToList (*list_name, user_email, merge_vars=None, email_type=None, double_optin=None, update_existing=None, replace_interests=None, send_welcome=None*)

unsubscribeFromList (*list_name, user_email, delete_member=None, send_goodbye=None, send_notify=None*)

1.4.4 twitter Module

class `pyrate.services.twitter.TwitterPyrate` (*oauth_consumer_key, oauth_consumer_secret, oauth_token, oauth_token_secret, default_http_method=None, default_return_format=None*)

Bases: `pyrate.main.Pyrate`

auth_type = 'OAUTH1'

base_url = 'https://api.twitter.com/1.1/'

check_response_success (*response*)

connection_check_method = ('GET', 'account/verify_credentials')

default_body_content = {}

default_header_content = {}

default_http_method = 'GET'

default_return_format = 'json'

get_oauth ()

```
http_methods = ['GET', 'POST']
oauth_consumer_key = ''
oauth_consumer_secret = ''
oauth_token = ''
oauth_token_secret = ''
parse_errors(response)
return_formats = ['json']
tweet(status, in_reply_to_status_id=None, loc_lat=None, loc_long=None, place_id=None, display_coordinates=None, trim_user=None, include_entities=None)
```

1.5 tests Package

1.5.1 credentials Module

1.5.2 tests Module

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

p

- `pyrate.__init__`, 5
- `pyrate.main`, 5
- `pyrate.scripts.cliutils`, 8
- `pyrate.services.github`, 9
- `pyrate.services.harvest`, 9
- `pyrate.services.mailchimp`, 10
- `pyrate.services.twitter`, 10