

---

# **python-simple-rest-client**

## **Documentation**

***Release 0.1.0***

**Allisson Azevedo**

**Oct 12, 2020**



---

## Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Quickstart . . . . .	3



Simple REST client for python 3.6+, supports sync (with requests) and asyncio (with aiohttp) requests



# CHAPTER 1

## Contents

### 1.1 Installation

Install the latest stable release via pip:

```
pip install simple-rest-client
```

python-simple-rest-client runs with Python 3.6+ .

### 1.2 Quickstart

Let's start building a client for users resource in <https://reqres.in/> service:

```
>>> # import API
>>> from simple_rest_client.api import API
>>> # create api instance
>>> api = API(
...     api_root_url='https://reqres.in/api/', # base api url
...     params={}, # default params
...     headers={}, # default headers
...     timeout=2, # default timeout in seconds
...     append_slash=False, # append slash to final url
...     json_encode_body=True, # encode body as json
... )
>>> # add users resource
>>> api.add_resource(resource_name='users')
>>> # show resource actions
>>> api.users.actions
{'list': {'method': 'GET', 'url': 'users'}, 'create': {'method': 'POST', 'url': 'users
↪'}, 'retrieve': {'method': 'GET', 'url': 'users/{}'.format(id)}, 'update': {'method': 'PUT',
↪'url': 'users/{}'.format(id)}, 'partial_update': {'method': 'PATCH', 'url': 'users/{}'.format(id)},
↪'destroy': {'method': 'DELETE', 'url': 'users/{}'.format(id)}}
```

(continues on next page)

(continued from previous page)

```

>>> # list action
>>> response = api.users.list(body=None, params={}, headers={})
>>> response.url
'https://reqres.in/api/users'
>>> response.method
'GET'
>>> response.body
{'page': 1, 'per_page': 3, 'total': 12, 'total_pages': 4, 'data': [{ 'id': 1, 'first_
↳ name': 'george', 'last_name': 'bluth', 'avatar': 'https://s3.amazonaws.com/uifaces/
↳ faces/twitter/calebogden/128.jpg'}, { 'id': 2, 'first_name': 'lucille', 'last_name':
↳ 'bluth', 'avatar': 'https://s3.amazonaws.com/uifaces/faces/twitter/josephstein/128.
↳ jpg'}, { 'id': 3, 'first_name': 'oscar', 'last_name': 'bluth', 'avatar': 'https://s3.
↳ amazonaws.com/uifaces/faces/twitter/olegpogodaev/128.jpg'}]}
>>> response.headers
{'Date': 'Sat, 15 Apr 2017 21:39:46 GMT', 'Content-Type': 'application/json;
↳ charset=utf-8', 'Transfer-Encoding': 'chunked', 'Connection': 'keep-alive', 'X-
↳ Powered-By': 'Express', 'Access-Control-Allow-Origin': '*', 'ETag': 'W/"1be-
↳ q96WkDv6JqflvIPiRhZWJQ"', 'Server': 'cloudflare-nginx', 'CF-RAY': '35020f33aaf04a9c-
↳ GRU', 'Content-Encoding': 'gzip'}
>>> response.client_response.cookies
<RequestsCookieJar[Cookie(version=0, name='__cfduid', value=
↳ 'd85187baf0752d02c6836abf7fb0c426c1506866165', port=None, port_specified=False,
↳ domain='.reqres.in', domain_specified=True, domain_initial_dot=True, path='/', path_
↳ specified=True, secure=False, expires=1538402165, discard=False, comment=None,
↳ comment_url=None, rest={'HttpOnly': None}, rfc2109=False)]>

>>> response.status_code
200
>>> # create action
>>> body = {'name': 'morpheus', 'job': 'leader'}
>>> response = api.users.create(body=body, params={}, headers={})
>>> response.status_code
201
>>> # retrieve action
>>> response = api.users.retrieve(2, body=None, params={}, headers={})
>>> response.status_code
200
>>> # update action
>>> response = api.users.update(2, body=body, params={}, headers={})
>>> response.status_code
200
>>> # partial update action
>>> response = api.users.partial_update(2, body=body, params={}, headers={})
>>> response.status_code
200
>>> # destroy action
>>> response = api.users.destroy(2, body=None, params={}, headers={})
>>> response.status_code
204

```

Building async client for users resource in <https://reqres.in/> service:

```

>>> # import asyncio, API and AsyncResource
>>> import asyncio
>>> from simple_rest_client.api import API
>>> from simple_rest_client.resource import AsyncResource
>>> # create api instance

```

(continues on next page)



(continued from previous page)

```
>>> api = API(
...     api_root_url='https://reqres.in/api/', # base api url
...     params={}, # default params
...     headers={}, # default headers
...     timeout=2, # default timeout in seconds
...     append_slash=False, # append slash to final url
...     json_encode_body=True, # encode body as json
... )
>>> # add users resource
>>> api.add_resource(resource_name='users', resource_class=AsyncResource)
>>> async def main():
...     print(await api.users.list())
...
>>> loop = asyncio.get_event_loop()
>>> loop.run_until_complete(main())
Response(url='https://reqres.in/api/users', method='GET', body={'page': 1, 'per_page': 3, 'total': 12, 'total_pages': 4, 'data': [{'id': 1, 'first_name': 'george', 'last_name': 'bluth', 'avatar': 'https://s3.amazonaws.com/uifaces/faces/twitter/calebogden/128.jpg'}, {'id': 2, 'first_name': 'lucille', 'last_name': 'bluth', 'avatar': 'https://s3.amazonaws.com/uifaces/faces/twitter/josephstein/128.jpg'}, {'id': 3, 'first_name': 'oscar', 'last_name': 'bluth', 'avatar': 'https://s3.amazonaws.com/uifaces/faces/twitter/olegpogodaev/128.jpg'}]}, headers={'Date': 'Mon, 26 Jun 2017 19:03:04 GMT', 'Content-Type': 'application/json; charset=utf-8', 'Transfer-Encoding': 'chunked', 'Connection': 'keep-alive', 'Set-Cookie': '__cfduid=d0412e4ebb0c5c29b09c0f756408d6ccb1498503784; expires=Tue, 26-Jun-18 19:03:04 GMT; path=/; domain=reqres.in; HttpOnly', 'X-Powered-By': 'Express', 'Access-Control-Allow-Origin': '*', 'ETag': 'W/"1be-q96WkDv6JqfLvIPiRhZWJQ"', 'Server': 'cloudflare-nginx', 'CF-RAY': '37526cadd214af1-GRU', 'Content-Encoding': 'gzip'}, status_code=200)
```

Now, building a client for github events resource (<https://developer.github.com/v3/activity/events/>):

```
>>> # import API and Resource
>>> from simple_rest_client.api import API
>>> from simple_rest_client.resource import Resource
>>> # create EventResource with custom actions
>>> class EventResource(Resource):
...     actions = {
...         'public_events': {'method': 'GET', 'url': 'events'},
...         'repository_events': {'method': 'GET', 'url': '/repos/{}/{}/events'},
...         'repository_issues_events': {'method': 'GET', 'url': '/repos/{}/{}/issues/
... events'},
...         'public_network_events': {'method': 'GET', 'url': '/networks/{}/{}/events'
... },
...         'public_organization_events': {'method': 'GET', 'url': '/orgs/{}/events'},
...         'user_received_events': {'method': 'GET', 'url': '/users/{}/received_
... events'},
...         'public_user_received_events': {'method': 'GET', 'url': '/users/{}/
... received_events/public'},
...         'user_events': {'method': 'GET', 'url': '/users/{}/events'},
...         'public_user_events': {'method': 'GET', 'url': '/users/{}/events/public'},
...         'organization_events': {'method': 'GET', 'url': '/orgs/{}/events/orgs/{
... }'},
...     }
...
>>> # set default params
>>> default_params = {'access_token': 'valid-token'}
```

(continues on next page)

(continued from previous page)

```

>>> # create api instance
>>> github_api = API(
...     api_root_url='https://api.github.com', params=default_params,
...     json_encode_body=True
... )
>>> # add events resource with EventResource
>>> github_api.add_resource(resource_name='events', resource_class=EventResource)
>>> # show resource actions
>>> github_api.events.actions
{'public_events': {'method': 'GET', 'url': 'events'}, 'repository_events': {'method':
↪ 'GET', 'url': '/repos/{}/events'}, 'repository_issues_events': {'method': 'GET',
↪ 'url': '/repos/{}/issues/events'}, 'public_network_events': {'method': 'GET',
↪ 'url': '/networks/{}/events'}, 'public_organization_events': {'method': 'GET',
↪ 'url': '/orgs/{}/events'}, 'user_received_events': {'method': 'GET', 'url': '/users/
↪ {}/received_events'}, 'public_user_received_events': {'method': 'GET', 'url': '/
↪ users/{}/received_events/public'}, 'user_events': {'method': 'GET', 'url': '/users/
↪ {}/events'}, 'public_user_events': {'method': 'GET', 'url': '/users/{}/events/public
↪ '}, 'organization_events': {'method': 'GET', 'url': '/users/{}/events/orgs/{}}'}
>>> # public_events action
>>> response = github_api.events.public_events(body=None, params={}, headers={})
>>> response.url
'https://api.github.com/events?access_token=valid-token'
>>> response.method
'GET'
>>> # repository_events action
>>> response = github_api.events.repository_events('allisson', 'python-simple-rest-
↪ client', body=None, params={}, headers={})
>>> response.url
'https://api.github.com/repos/allisson/python-simple-rest-client/events?access_
↪ token=valid-token'
>>> response.method
'GET'

```

### Create API without certificate validation

```

>>> # import API
>>> from simple_rest_client.api import API
>>> # create api instance
>>> api = API(
...     api_root_url='https://develop-environment-with-self-signed-certificate.in/api/
↪ ', # base api url
...     params={}, # default params
...     headers={}, # default headers
...     timeout=2, # default timeout in seconds
...     append_slash=False, # append slash to final url
...     json_encode_body=True, # encode body as json
...     ssl_verify=False # ignore verifying the ssl
... )

```

Check <https://github.com/allisson/python-simple-rest-client/tree/master/examples> for more code examples.