
funkload-friendly Documentation

Release 0.2

Shinya Okano

March 15, 2016

1	Overview	3
1.1	Requirements	3
1.2	License	3
1.3	Cookiecutter	3
1.4	Examples	3
1.5	Documentaion	3
1.6	Develop	4
1.7	Author	4
2	Usage	5
2.1	Install	5
2.2	Getting Started	5
3	Friendly TestCase Class	7
3.1	Cookie Support	7
3.2	Configuration Support	7
3.3	Description Decorator	8
4	JSON data support	9
5	Reference	11
5.1	funkload_friendly.cookie	11
5.2	funkload_friendly.datatypes	11
5.3	funkload_friendly.test	11
6	Change logs	13
6.1	v0.2	13
6.2	v0.1	13
7	Indices and tables	15
	Python Module Index	17

Friendly wrapper of FunkLoad.

Contents:

Overview

funkload-friendly is a wrapper of [FunkLoad](#).

The FunkLoad is good tool for load testing of Web Application. But some rules are not friendly to user.

funkload-friendly resolve some frustration.

1.1 Requirements

- Python 2.7
- funkload>=1.17.1

1.2 License

This software is licensed under the MIT License.

1.3 Cookiecutter

There are Cookiecutter template for this project: <https://github.com/tokibito/cookiecutter-funkload-friendly>

1.4 Examples

There are some examples: <https://github.com/tokibito/funkload-friendly-example>

1.5 Documentaion

The latest documentation is hosted at Read The Docs.

<https://funkload-friendly.readthedocs.org/en/latest/>

1.6 Develop

This project is hosted at Github: <https://github.com/tokibito/funkload-friendly>

1.7 Author

- Shinya Okano <http://www.nullpobug.com/>

Usage

2.1 Install

Install funkload-friendly with easy_install into system python environment or virtualenv.

```
$ easy_install funkload-friendly
```

Note: This package uses the easy_install to install. Because it can not “pip install funkload”.

2.2 Getting Started

1. Writing your load testing code.

loadtest.py:

```
from funkload_friendly.test import TestCase, description

class MainTest(TestCase):
    @description("Load top_page")
    def test_top_page(self):
        self.get(self.site_url + "/")
```

2. create config file(e.g. funkload.conf).

funkload.conf:

```
[main]
title = My Project Loadtest
description = load testing for myproject
url = http://www.example.com

[fctest]
log_to = console file
log_directory = ./
result_directory = ./
sleep_time_min = 0
sleep_time_max = 0

[bench]
```

```
cycles = 1:3:10
duration = 10
startup_delay = 0.01
sleep_time = 0.01
cycle_time = 1
sleep_time_min = 0
sleep_time_max = 0.5
log_to = console file
log_directory = ./
result_directory = ./
```

3. run benchmark test with fl-run-bench.

```
$ fl-run-bench --config=funkload.conf loadtest MainTest.test_top_page
```

4. build the benchmark report.

```
$ fl-build-report --html test_top_page.xml
```

Note: There are some example codes: <https://github.com/tokibito/funkload-friendly-example>

Friendly TestCase Class

funkload-friendly provides a TestCase, it is extended FunkLoadTestCase.

This class features:

- Retain cookie data between multiple requests.
- Be able to configure log and result output directory, and some shortcut attribute.
- decorator for test description

3.1 Cookie Support

In FunkLoad, cookie data doesn't retain between request.

In funkload-friendly, cookie data retain between request. This means it easy to be able to test using the Cookie-based session.

Example:

```
class LoginTest(TestCase):
    def test_login(self):
        self.get(self.site_url + "/login/") # take the CSRF token into cookie
        # Login with cookie (CSRF token required)
        self.post(self.site_url + "/login/",
                  params=[
                      ['username', 'spam'],
                      ['password', 'P@ssw0rd'],
                      ['csrftoken', self.cookie['csrftoken']],
                  ])
        )
```

3.2 Configuration Support

funkload-friendly can be a configuration shown below.

3.2.1 Omit Filename

In FunkLoad, output destination of log and result is, you must specify the file name.

In funkload-friendly, specify a directory name, you can omit the file name.

Example:

```
[fctest]
log_to = console file
log_directory = fctest/
result_directory = fctest/

[bench]
log_to = console file
log_directory = bench/
result_directory = bench/
```

3.2.2 Shortcut Attribute

Some config value assign to the attribute of TestCase.

- section: [main], key: url map to TestCase.site_url.
Specify the url in http[s]://host[:port] format.

Example:

```
[main]
url = http://localhost:8000
```

3.3 Description Decorator

In FunkLoad, method description has must be described in config file.

In funkload-friendly, description can be described with the description descriptor.

Example:

```
from funkload_friendly.test import TestCase, description

class MainTest(TestCase):
    @description("Load top_page")
    def test_top_page(self):
        self.get(self.site_url + "/")
```

Result(log):

```
test_top_page: Starting -----
                Load top_page
test_top_page: GET: http://localhost:8000/
                Page 1: ...
test_top_page: Done in 0.005s
test_top_page: Load css and images...
test_top_page: Done in 0.000s
```

JSON data support

`funkload_friendly.datatypes.JSONData` class extends `funkload.utils.Data`. And support JSON parameter in HTTP POST.

Example:

```
>>> from funkload_friendly.datatypes import JSONData
>>> JSONData({'spam': 'egg', 'ham': 'bacon'})
[User data application/json]
```

```
from funkload_friendly.test import TestCase, description
from funkload_friendly.datatypes import JSONData

class APITest(TestCase):
    @description("Load REST API")
    def test_calculate_add(self):
        response = self.post(self.site_url + "/calculate/add/",
                             params=JSONData({
                                 'value1': 100,
                                 'value2': 50,
                             }))
        self.assertEqual(response.code, 200)
        self.assertEqual(response.data['result'], 150)
```


5.1 `funkload_friendly.cookie`

This module is processing Cookie Data.

```
class funkload_friendly.cookie.CookieDict (*args, **kwargs)
    Represent Cookie Data

    CookieDict class extends OrderedDict.

    from_cookie_string (cookie_string)
        update self with cookie_string.

    render_to_string ()
        Render to cookie strings.
```

5.2 `funkload_friendly.datatypes`

```
class funkload_friendly.datatypes.JSONData (data, content_type='application/json')
    Support JSON
```

5.3 `funkload_friendly.test`

```
class funkload_friendly.test.TestCase (methodName='runTest', options=None)
    funkload-friendly TestCase

    clear_session_id ()
        Clear session ID

    conf_get (section, key, default=[], quiet=False)
        Override FunkLoadTestCase.conf_get

    reset_cookie ()
        Reset cookie data.

    set_session_id (session_id)
        Set session ID to cookie

    funkload_friendly.test.description (description)
        Set description to test_method
```

Change logs

6.1 v0.2

- added document
- fix some bugs

6.2 v0.1

- first releasae

Indices and tables

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

f

`funkload_friendly.cookie`, [11](#)
`funkload_friendly.datatypes`, [11](#)
`funkload_friendly.test`, [11](#)

C

`clear_session_id()` (funkload_friendly.test.TestCase method), 11
`conf_get()` (funkload_friendly.test.TestCase method), 11
`CookieDict` (class in funkload_friendly.cookie), 11

D

`description()` (in module funkload_friendly.test), 11

F

`from_cookie_string()` (funkload_friendly.cookie.CookieDict method), 11
`funkload_friendly.cookie` (module), 11
`funkload_friendly.datatypes` (module), 11
`funkload_friendly.test` (module), 11

J

`JSONData` (class in funkload_friendly.datatypes), 11

R

`render_to_string()` (funkload_friendly.cookie.CookieDict method), 11
`reset_cookie()` (funkload_friendly.test.TestCase method), 11

S

`set_session_id()` (funkload_friendly.test.TestCase method), 11

T

`TestCase` (class in funkload_friendly.test), 11