Swagger Tester Documentation

Release 0.2.8

Cyprien Guillemot

May 20, 2018

Contents

Contents:

CHAPTER 1

swagger-tester

Swagger-tester will test automatically your swagger API. Swagger API made with connexion (https://github.com/zalando/connexion) are supported directly without running the API server. In the case you use connexion it will automatically run a test server from your swagger file.

To run the test, swagger-tester will detect every path and actions of your API. And for each, it will send a request and check if the response match the swagger file specification.

1.1 Related Libraries

You may find related libraries to this one:

- https://github.com/Trax-air/swagger-stub: A stub you can use in your client's unit tests. All the HTTP calls to your swagger API are mocked by default. You can also add your own mocked_calls in your test functions.
- https://github.com/Trax-air/swagger-aggregator: Aggregate several swagger specs into one. Useful for your API gateways!
- https://github.com/Trax-air/swagger-parser: A helper that parses swagger specs. You can access the HTTP actions / paths and some example data

1.2 Example Usage

```
from swagger_tester import swagger_test

 # Dict containing the error you don't want to raise.
 # By default, every status_code over other than 1xx, 2xx or 3xx
 # will be considered as an error.
 authorize_error = {
       'post': {
           '/pet/{petId}': [200],
           '/pet': [200]
```

(continues on next page)

(continued from previous page)

```
        },
        'put': {
            '/user/{username}': [200],
            '/pet': [200]
        },
        'delete': {
            '/pet/{petId}': [200],
            '/store/order/{orderId}': [200],
            '/user/{username}': [200]
        }
    }

# Run the test with connexion
# An AssertionError will be raise in case of error.
swagger_test('path_to_your_swagger.yaml', authorize_error=authorize_error)

# Or if you have a running API
swagger_test(app_url='http://petstore.swagger.io/v2', authorize_error=authorize_
↪error)
```

1.3 Documentation

More documentation is available at https://swagger-tester.readthedocs.org/en/latest/.

1.4 Setup

make install or *pip install swagger-tester*

1.5 License

swagger-tester is licensed under http://opensource.org/licenses/MIT.

CHAPTER 2

Installation

At the command line:

```
$ easy_install swagger_tester
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv swagger_tester
$ pip install swagger_tester
```

CHAPTER 3

Usage

To use Swagger Tester in a project:

```
from swagger_tester import swagger_test

# Dict containing the error you don't want to raise.
# By default, every status_code over other than 1xx, 2xx or 3xx
# will be considered as an error.
authorize_error = {
  'get': {
    '/pet/': ['400', '404']
  }
}

# Run the test
# An AssertionError will be raise in case of error.
swagger_test('path_to_your_swagger.yaml', authorize_error=authorize_error)
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/cyprieng/swagger_tester/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

Swagger Tester could always use more documentation, whether as part of the official Swagger Tester docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/cyprieng/swagger_tester/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *swagger_tester* for local development.

1. Fork the *swagger_tester* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/swagger_tester.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv swagger_tester
$ cd swagger_tester/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 swagger_tester tests
$ python setup.py test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/cyprieng/swagger_tester/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_swagger_tester
```

CHAPTER 5

Credits

5.1 Development Lead

- Cyprien Guillemot <cyprien.guillemot@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.2.7 (2016-11-22)

- TODO

6.2 0.2.6 (2016-5-20)

- Fix repeated base path bug

6.3 0.2.5 (2016-3-25)

- Add support for headers parameters.

6.4 0.2.4 (2016-3-23)

- Improve the check of status code when 'default' is in the specification.

6.5 0.2.3 (2016-2-10)

- Fix some errors (like file upload).

6.6 0.2.2 (2016-2-3)

- Fix validation of standard types.

6.7 0.2.1 (2016-1-31)

- Change license to MIT.

6.8 0.2.0 (2016-1-31)

- Now support swagger APIs not made with connexion.

6.9 0.1 (2016-1-29)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search