aiohttp$_validateDocumentation$

Release 1.0.0

Dmitry Chaplinsky

May 12, 2020

Contents

Contents:

CHAPTER 1

aiohttp_validate

Simple library that helps you validate your API endpoints requests/responses with jsonschema. Documentation is also available here at https://aiohttp-validate.readthedocs.io.

1.1 Installation

Install from PyPI:

```
pip install aiohttp_validate
```

1.2 Usage

Complete example of validation for text tokenization microservice:

```
from aiohttp_validate import validate

@validate(
    request_schema={
        "type": "object",
        "properties": {
            "text": {"type": "string"},
        },
        "required": ["text"],
        "additionalProperties": False
    },
```

(continues on next page)

(continued from previous page)

```
    response_schema={
        "type": "array",
        "items": {
            "type": "array",
            "items": {
                "type": "array",
                "items": {"type": "string"}
            }
        }
    }
)
async def tokenize_text_handler(request, *args):
    return tokenize_text(request["text"])
```

1.3 Features

- The decorator to (optionally) validate the request to your aiohttp endpoint and it's response.
- Easily integrates with aiohttp_swaggerify to automatically document your endpoints with swagger.
- Validation errors are standardized and can be easily parsed by the clients of your service and also human-readable.

1.4 Developing

Install requirement and launch tests:

```
pip install -r requirements-dev.txt
py.test
```

1.5 Credits

That package is influenced by Tornado-JSON written by Hamza Faran Code to parse errors is written by Ruslan Karalkin

1.6 Versioning

This software follows Semantic Versioning

1.7 License

- Free software: MIT license

CHAPTER 2

Installation

2.1 Stable release

To install aiohttp_validate, run this command in your terminal:

```
$ pip install aiohttp_validate
```

This is the preferred method to install aiohttp_validate, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for aiohttp_validate can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/dchaplinsky/aiohttp_validate
```

Or download the tarball:

```
$ curl  -OL https://github.com/dchaplinsky/aiohttp_validate/tarball/master
```

Once you have a copy of the source, you can install it with:

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

CHAPTER 3

Usage

To use aiohttp_validate in a project:

```
import aiohttp_validate
```

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/dchaplinsky/aiohttp_validate/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” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

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

4.1.4 Write Documentation

aiohttp_validate could always use more documentation, whether as part of the official aiohttp_validate 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/dchaplinsky/aiohttp_validate/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 *aiohttp_validate* for local development.

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

   ```
   $ git clone git@github.com:your_name_here/aiohttp_validate.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 aiohttp_validate
   $ cd aiohttp_validate/
   $ 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 aiohttp_validate tests
   $ python setup.py test or 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/dchaplinsky/aiohttp_validate/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_aiohttp_validate
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search