sprockets.mixins.correlation

Release 3.0.0

Dec 20, 2019

Contents

This sprocket provides a single mix-in that imbues your `RequestHandler` with a unique correlation ID. If a correlation ID is present upon input then it will be preserved in the output. It is also available for your use as the `correlation_id` property.

CHAPTER 1

Installation

sprockets.mixins.correlation is available on the Python Package Index and can be installed via pip:

```
$ pip install sprockets.mixins.correlation
```

CHAPTER 2

Example

```
from sprockets.mixins import correlation
from tornado import ioloop, web

class Handler(correlation.HandlerMixin, web.RequestHandler):
   def get(self):
      self.finish('my id is {0}'.format(self.correlation_id)

if __name__ == '__main__':
   application = web.Application([('/', Handler)])
   application.listen(8888)
   ioloop.IOLoop.instance().start()
```

2.1 Generated Correlation ID

```
GET / HTTP/1.1
Host: localhost:8888
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Correlation-ID: 0a2b6080-e4da-43bf-a2a5-38d861846cb9
Content-Length: 44

my id is 0a2b6080-e4da-43bf-a2a5-38d861846cb9
```

2.2 Relayed Correlation ID

```
GET / HTTP/1.1
Host: localhost:8888
```

(continues on next page)

(continued from previous page)

```
Connection: keep-alive
Correlation-Id: 4676922073c4c59b1f5e6b4a18894bd46f867316
```

```
HTTP/1.1 200 OK
Correlation-ID: 4676922073c4c59b1f5e6b4a18894bd46f867316
Connection: close
Content-Length: 48

my id is 4676922073c4c59b1f5e6b4a18894bd46f867316
```

CHAPTER 3

API Documentation

3.1 correlation.HandlerMixin

class sprockets.mixins.correlation.**HandlerMixin**(**args*, ***kwargs*)

Mix this in over a `RequestHandler` for a correlating header.

Parameters **correlation_header** (*str*) – the name of the header to use for correlation. If this keyword is omitted, then the header is named `Correlation-ID`.

This mix-in ensures that responses include a header that correlates requests and responses. If there header is set on the incoming request, then it will be copied to the outgoing response. Otherwise, a new UUIDv4 will be generated and inserted. The value can be examined or modified via the `correlation_id` property.

The MRO needs to contain something that resembles a standard `tornado.web.RequestHandler`. Specifically, we need the following things to be available:

- `prepare()` needs to be called appropriately
- `set_header()` needs to exist in the MRO and it needs to overwrite the header value
- `set_default_headers()` should be called to establish the default header values
- `self.request` is a object that has a `headers` property that contains the request headers as a `dict`.

correlation_id

Correlation header value.

get_request_header(*name*, *default*)

Retrieve the value of a request header.

Parameters

- **name** (*str*) – the name of the header to retrieve
- **default** – the value to return if the header is not set

This method abstracts the act of retrieving a header value out from the implementation. This makes it possible to implement a *RequestHandler* that is something other than a `tornado.web.RequestHandler`

by simply implementing this method and `set_header` over the underlying implementation, for example, say AMQP message properties.

CHAPTER 4

Contributing to this Library

The easiest way to start working with this code is to set up a virtual environment and run `env/bin/pip -r dev-requirements.txt`. That will install the necessary testing tools. Then you can run everything else using `env/bin/python setup.py`:

- *setup.py nosetests* will run the tests using nose to test against the and generate a coverage report to stdout.
- *setup.py build_sphinx* will generate HTML documentation into *build/doc/html*. This is the doc set that is uploaded to Read The Docs.
- *setup.py flake8* will run the `flake8` utility and report on any static code analysis failures.

This library follows the standard fork, modify, and pull request flow for contributing.

CHAPTER 5

Version History

5.1 3.0.0 (18-Dec-2019)

- Dropped support for Tornado 4
- Fixed support for async prepare in superclasses of `HandlerMixin`

5.2 2.0.1 (18-Mar-2019)

- Add support for Tornado 6
- Move requirements files to requires/
- Increase test coverage

5.3 2.0.0 (26-Nov-2018)

- Drop support for Python 2.7, 3.3, 3.4
- Drop support for Tornado < 4.2
- Add support for Tornado 5.1 and async with `AsyncIOHandlerMixin`

5.4 1.0.2 (20-Jun-2016)

- Add support for async prepare in superclasses of `HandlerMixin`

5.5 1.0.1 (31-Mar-2015)

- Adds `sprockets.mixins.correlation.HandlerMixin`

Index