httpsrv Documentation

Release 0.1.6

Alexander Nyrkov

Nov 20, 2016

Contents

Httpsrv is a simple HTTP server for API mocking during automated testing

exception `httpsrv.`**`PendingRequestsLeftException`**
Raises when server has pending reques expectations by calling the *`Server.assert_no_pending()`* method

class `httpsrv.`**`Rule`**(*method*, *path*, *headers*, *text*, *json*)
Expectation rule — defines expected request parameters and response values

Parameters

- **`method`** (*`str`*) – expected request method: `'GET'`, `'POST'`, etc. Can take any custom string
- **`path`** (*`str`*) – expected path including query parameters, e.g. `'/users?name=John%20Doe'` if ommited any path will do
- **`headers`** (*`dict`*) – dictionary of expected request headers
- **`text`** (*`str`*) – expected request body text
- **`json`** (*`dict`*) – request json to expect. If ommited any json will match, if present text param will be ignored

`json`(*json_doc*, *status=200*, *headers=None*)
Respond with given status and JSON content. Will also set `'Content-Type'` to `'applicaion/json'` if header is not specified explicitly

Parameters

- **`json_doc`** (*`dict`*) – dictionary to respond with converting to JSON string
- **`status`** (*`int`*) – status code to return
- **`headers`** (*`dict`*) – dictionary of headers to add to response

`matches`(*method*, *path*, *headers*, *bytes=None*)
Checks if rule matches given request parameters

Parameters

- **`method`** (*`str`*) – HTTP method, e.g. `'GET'`, `'POST'`, etc. Can take any custom string
- **`path`** (*`str`*) – request path including query parameters, e.g. `'/users?name=John%20Doe'`
- **`bytes`** (*`bytes`*) – request body

Returns `True` if this rule matches given params

Return type bool

`method`
Method name this rule will respond to

Returns epected method name

Return type str

`status`(*status*, *headers=None*)
Respond with given status and no content

Parameters

- **`status`** (*`int`*) – status code to return
- **`headers`** (*`dict`*) – dictionary of headers to add to response

Returns itself

Return type *Rule*

text (*text*, *status=200*, *headers=None*)
Respond with given status and text content

Parameters

- **text** (*str*) – text to return
- **status** (*int*) – status code to return
- **headers** (*dict*) – dictionary of headers to add to response

Returns itself

Return type *Rule*

class httpsrv.**Server** (*port*)
Tunable HTTP server running in a parallel thread.

Please note that *this server is not thread-safe* which should not cause any troubles in common use-cases due to python single-threaded nature.

Parameters **port** (*int*) – port this server will listen to after *Server.start()* is called

always (*method*, *path=None*, *headers=None*, *text=None*, *json=None*)
Sends response every time matching parameters are found util *Server.reset()* is called

Parameters

- **method** (*str*) – request method: 'GET', 'POST', etc. can be some custom string
- **path** (*str*) – request path including query parameters
- **headers** (*dict*) – dictionary of headers to expect. If omitted any headers will do
- **text** (*str*) – request text to expect. If ommited any text will match
- **json** (*dict*) – request json to expect. If ommited any json will match, if present text param will be ignored

Return type *Rule*

Returns newly created expectation rule

assert_no_pending (*target_rule=None*)
Raises a *PendingRequestsLeftException* error if server has target rule non-resolved.

When target_rule argument is ommitted raises if server has any pending expectations.

Useful in tearDown() test method to verify that test had correct expectations

Parameters **target_rule** (Rule) – will raise if this rule is left pending

Raises *PendingRequestsLeftException*

on (*method*, *path=None*, *headers=None*, *text=None*, *json=None*)
Sends response to matching parameters one time and removes it from list of expectations

Parameters

- **method** (*str*) – request method: 'GET', 'POST', etc. can be some custom string
- **path** (*str*) – request path including query parameters
- **headers** (*dict*) – dictionary of headers to expect. If omitted any headers will do

- **text** (*str*) – request text to expect. If ommited any text will match
- **json** (*dict*) – request json to expect. If ommited any json will match, if present text param will be ignored

Return type *Rule*

Returns newly created expectation rule

reset()
Clears the server expectations. Useful for resetting the server to its default state in `teardDown()` test method instead of time-consuming restart procedure

start()
Starts a server on the port provided in the *Server* constructor in a separate thread

Return type *Server*

Returns server instance for chaining

stop()
Shuts the server down and waits for server thread to join

CHAPTER 1

Httpsrv

Simple http server for API mocking during automated testing Plays nicely with httpsrvvcr library for automated request recording

1.1 Example usage

A typical usage pattern would probably look like the one below.

Using requests library:

```
import unittest
import requests
from httpsrv import Server

server = Server(8080).start()

class MyTestCase(unittest.TestCase):
    def setUp(self):
        server.reset()

    def test_should_get_hello(self):
        # this means that server will respond once upon GET request
        # further GET requests on this path will get 500
        server.on('GET', '/').text('hello')
        res = requests.get('http://localhost:8080')
        assert res.text == 'hello'

    def test_should_always_respond_to_options(self):
        # this means that any OPTIONS request will get status 200
        # such behavior is particulary useful when mocking preflight queries
        server.always('OPTIONS').status(200)
        res = requests.get('http://localhost:8080')
        assert res.status_code == 200
```

1.2 Installation

```
pip install httpsrv
```

1.3 Documentation

http://httpsrv.readthedocs.org

h

A

H

J

M

O

P

R

S

T