httpsrv Documentation

Release 1.0b1

Alexander Nyrkov

Sep 14, 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`**(*expectation*)
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`(*request*)
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

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_any(*method*, *path=None*, *headers=None*, *always=False*)
Request with any content. Path is optional if omitted any path will match.

Server will respond to matching parameters one time and remove the rule from list unless `always` flag is set to `True`

Parameters

- **method** (*str*) – request method: `'GET'`, `'POST'`, etc. can be some custom string
- **path** (*str*) – request path including query parameters. If omitted any path will do
- **headers** (*dict*) – dictionary of headers to expect. If omitted any headers will do
- **always** (*bool*) – if `True` this rule will not be removed after use

Return type *Rule*

Returns newly created expectation rule

on_files(*method*, *path*, *files*, *form=None*, *headers=None*, *always=False*)
File upload with optional form data.

Server will respond to matching parameters one time and remove the rule from list unless `always` flag is set to `True`

Parameters

- **method** (*str*) – request method: `'GET'`, `'POST'`, etc. can be some custom string
- **path** (*str*) – request path including query parameters

- **json** – expected form data. **Please note that filed values must always BaseException of collection type e.g. ‘‘{‘user’: [’Dude’]}‘‘.** this restriction is caused by possible multi-value fields and may be lifted in future releases
- **headers** (*dict*) – dictionary of headers to expect. If omitted any headers will do
- **always** (*bool*) – if `True` this rule will not be removed after use

Return type *Rule*

Returns newly created expectation rule

on_form(*method*, *path*, *form*, *headers=None*, *always=False*)

Request with form data either in requets body or query params.

Server will respond to matching parameters one time and remove the rule from list unless `always` flag is set to `True`

Parameters

- **method** (*str*) – request method: `'GET'`, `'POST'`, etc. can be some custom string
- **path** (*str*) – request path including query parameters
- **json** – expected form data. **Please note that filed values must always be of collection type** e.g. `{'user': ['Dude']}`. this restriction is caused by possible multivalue fields and may be lifted in future releases
- **headers** (*dict*) – dictionary of headers to expect. If omitted any headers will do
- **always** (*bool*) – if `True` this rule will not be removed after use

Return type *Rule*

Returns newly created expectation rule

on_json(*method*, *path*, *json*, *headers=None*, *always=False*)

Request with JSON body. This will not check for `Content-Type` header. Instead we’ll try to parse whatever request body contains.’

Server will respond to matching parameters one time and remove the rule from list unless `always` flag is set to `True`

Parameters

- **method** (*str*) – request method: `'GET'`, `'POST'`, etc. can be some custom string
- **path** (*str*) – request path including query parameters
- **json** (*dict*) – expected json data
- **headers** (*dict*) – dictionary of headers to expect. If omitted any headers will do
- **always** (*bool*) – if `True` this rule will not be removed after use

Return type *Rule*

Returns newly created expectation rule

on_text(*method*, *path*, *text*, *headers=None*, *always=False*)

Request with generic text data. Can be used if nothing else matches.

Server will respond to matching parameters one time and remove the rule from list unless `always` flag is set to `True` :type method: str :param method: request method: `'GET'`, `'POST'`, etc. can be some custom string

Parameters

- **path** (*str*) – request path including query parameters
- **text** (*str*) – expected text sent with request
- **headers** (*dict*) – dictionary of headers to expect. If omitted any headers will do
- **always** (*bool*) – if `True` this rule will not be removed after use

Return type *Rule*

Returns newly created expectation rule

reset()
Clears the server expectations. Useful for resetting the server to its default state in `tearDown()` 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_any('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.on_any('OPTIONS', always=True).status(200)
        res = requests.get('http://localhost:8080')
        assert res.status_code == 200

    def test_should_respond_to_json(self):
        # this means that server will respond to the POST request
        # containing target json document in its body
        server.on_json('POST', '/users', {'name': 'John Doe'}).json(
            {'id': 1, 'name': 'John Doe'}, status=201)
        res = requests.post('http://localhost:8080/users', json={'name': 'John Doe'})
        assert res.status_code == 201
```

For more details and full list of `on_*` methods see API documentation

1.2 Installation

```
pip install httpsrv
```

1.3 Documentation

http://httpsrv.readthedocs.org

h

A

H

J

M

O

P

R

S

T