

---

# **BrowserMob Proxy Documentation**

*Release 0.6.0*

**David Burns**

**Oct 08, 2017**



---

## Contents

---

<b>1</b>	<b>How to install</b>	<b>3</b>
<b>2</b>	<b>How to use with selenium-webdriver</b>	<b>5</b>
<b>3</b>	<b>How to Contribute</b>	<b>7</b>
3.1	Getting Started . . . . .	7
3.2	Making Changes . . . . .	7
3.3	Submitting Changes . . . . .	8
	<b>Python Module Index</b>	<b>13</b>



Python client for the BrowserMob Proxy 2.0 REST API.



# CHAPTER 1

---

## How to install

---

BrowserMob Proxy is available on [PyPI](#), so you can install it with `pip`:

```
$ pip install browsermob-proxy
```

Or with *easy\_install*:

```
$ easy_install browsermob-proxy
```

Or by cloning the repo from [GitHub](#):

```
$ git clone git://github.com/AutomatedTester/browsermob-proxy-py.git
```

Then install it by running:

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





---

### How to use with selenium-webdriver

---

Manually:

```
from browsermobproxy import Server
server = Server("path/to/browsermob-proxy")
server.start()
proxy = server.create_proxy()

from selenium import webdriver
profile = webdriver.FirefoxProfile()
profile.set_proxy(proxy.selenium_proxy())
driver = webdriver.Firefox(firefox_profile=profile)

proxy.new_har("google")
driver.get("http://www.google.co.uk")
proxy.har # returns a HAR JSON blob

server.stop()
driver.quit()
```



### Getting Started

- Fork the repository on GitHub - well... duh :P
- Create a virtualenv: `virtualenv venv`
- Activate the virtualenv: `. venv/bin/activate`
- Install the package in develop mode: `python setup.py develop`
- Install requirements: `pip install -r requirements.txt`
- Run the tests to check that everything was successful: `py.test tests`

### Making Changes

- Create a topic branch from where you want to base your work. \* This is usually the master branch. \* Only target release branches if you are certain your fix must be on that branch.
  - To quickly create a topic branch based on master; `git checkout -b /my_contribution master`. Please avoid working directly on the *master* branch.
- Make commits of logical units.
- Check for unnecessary whitespace with `git diff -check` before committing.
- Make sure you have added the necessary tests for your changes.
- Run `_all_` the tests to assure nothing else was accidentally broken.

## Submitting Changes

- Push your changes to a topic branch in your fork of the repository.
- Submit a pull request to the main repository
- After feedback has been given we expect responses within two weeks. After two weeks will may close the pull request if it isn't showing any activity

Contents:

### client Package

**class** browsermobproxy.**Client** (*url, params=None, options=None*)

Initialises a new Client object

#### Parameters

- **url** – This is where the BrowserMob Proxy lives
- **params** – URL query (for example httpProxy and httpsProxy vars)
- **options** – Dictionary that can contain the port of an existing proxy to use (for example 'existing\_proxy\_port\_to\_use')

**add\_to\_capabilities** (*capabilities*)

Adds an 'proxy' entry to a desired capabilities dictionary with the BrowserMob proxy information

**Parameters capabilities** – The Desired capabilities object from Selenium WebDriver

**basic\_authentication** (*domain, username, password*)

This add automatic basic authentication

#### Parameters

- **domain** (*str*) – domain to set authentication credentials for
- **username** (*str*) – valid username to use when authenticating
- **password** (*str*) – valid password to use when authenticating

**blacklist** (*regexp, status\_code*)

Sets a list of URL patterns to blacklist

#### Parameters

- **regex** (*str*) – a comma separated list of regular expressions
- **status\_code** (*int*) – the HTTP status code to return for URLs that do not match the blacklist

**clear\_all\_rewrite\_url\_rules** ()

Clears all URL rewrite rules :return: status code

**clear\_dns\_cache** ()

Clears the DNS cache associated with the proxy instance

**close** ()

shuts down the proxy and closes the port

**har**

Gets the HAR that has been recorded

**headers** (*headers*)

This sets the headers that will set by the proxy on all requests

**Parameters** **headers** (*dict*) – this is a dictionary of the headers to be set

**limits** (*options*)

Limit the bandwidth through the proxy.

**Parameters** **options** (*dict*) – A dictionary with all the details you want to set. `downstream_kbps` - Sets the downstream kbps `upstream_kbps` - Sets the upstream kbps latency  
- Add the given latency to each HTTP request

**new\_har** (*ref=None, options=None, title=None*)

This sets a new HAR to be recorded

**Parameters**

- **ref** (*str*) – A reference for the HAR. Defaults to None
- **options** (*dict*) – A dictionary that will be passed to BrowserMob Proxy with specific keywords. Keywords are:
  - `captureHeaders`: Boolean, capture headers
  - `captureContent`: Boolean, capture content bodies
  - `captureBinaryContent`: Boolean, capture binary content
- **title** (*str*) – the title of first har page. Defaults to ref.

**new\_page** (*ref=None, title=None*)

This sets a new page to be recorded

**Parameters**

- **ref** (*str*) – A reference for the new page. Defaults to None
- **title** (*str*) – the title of new har page. Defaults to ref.

**proxy\_ports**

Return a list of proxy ports available

**remap\_hosts** (*address=None, ip\_address=None, hostmap=None*)

Remap the hosts for a specific URL

**Parameters**

- **address** (*str*) – url that you wish to remap
- **ip\_address** (*str*) – IP Address that will handle all traffic for the address passed in
- **\*\*hostmap** – Other hosts to be added as keyword arguments

**request\_interceptor** (*js*)

Executes the java/js code against each response `HttpRequest request`, `HttpMessageContents contents`, `HttpMessageInfo messageInfo` are available objects to interact with. `:param str js: the js/java code to execute`

**response\_interceptor** (*js*)

Executes the java/js code against each response `HttpRequest request`, `HttpMessageContents contents`, `HttpMessageInfo messageInfo` are available objects to interact with. `:param str js: the js/java code to execute`

**retry** (*retry\_count*)

Retries. No idea what its used for, but its in the API...

Parameters **retry\_count** (*int*) – the number of retries

**rewrite\_url** (*match, replace*)

Rewrites the requested url.

Parameters

- **match** – a regex to match requests with
- **replace** – unicode a string to replace the matches with

**selenium\_proxy** ()

Returns a Selenium WebDriver Proxy class with details of the HTTP Proxy

**timeouts** (*options*)

Configure various timeouts in the proxy

Parameters **options** (*dict*) – A dictionary with all the details you want to set. request - request timeout (in seconds) read - read timeout (in seconds) connection - connection timeout (in seconds) dns - dns lookup timeout (in seconds)

**wait\_for\_traffic\_to\_stop** (*quiet\_period, timeout*)

Waits for the network to be quiet

Parameters

- **quiet\_period** (*int*) – number of milliseconds the network needs to be quiet for
- **timeout** (*int*) – max number of milliseconds to wait

**webdriver\_proxy** ()

Returns a Selenium WebDriver Proxy class with details of the HTTP Proxy

**whitelist** (*regexp, status\_code*)

Sets a list of URL patterns to whitelist

Parameters

- **regex** (*str*) – a comma separated list of regular expressions
- **status\_code** (*int*) – the HTTP status code to return for URLs that do not match the whitelist

## server Package

**class** browsermobproxy.**Server** (*path='browsermob-proxy', options=None*)

Initialises a Server object

Parameters

- **path** (*str*) – Path to the browsermob proxy batch file
- **options** (*dict*) – Dictionary that can hold the port. More items will be added in the future. This defaults to an empty dictionary

**start** (*options=None*)

This will start the browsermob proxy and then wait until it can interact with it

Parameters **options** (*dict*) – Dictionary that can hold the path and filename of the log file with resp. keys of *log\_path* and *log\_file*

**stop** ()

This will stop the process running the proxy

## Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)





**b**

browsermobproxy, 10



## A

add\_to\_capabilities() (browsermobproxy.Client method), 8

## B

basic\_authentication() (browsermobproxy.Client method), 8

blacklist() (browsermobproxy.Client method), 8

browsermobproxy (module), 8, 10

## C

clear\_all\_rewrite\_url\_rules() (browsermobproxy.Client method), 8

clear\_dns\_cache() (browsermobproxy.Client method), 8

Client (class in browsermobproxy), 8

close() (browsermobproxy.Client method), 8

## H

har (browsermobproxy.Client attribute), 8

headers() (browsermobproxy.Client method), 8

## L

limits() (browsermobproxy.Client method), 9

## N

new\_har() (browsermobproxy.Client method), 9

new\_page() (browsermobproxy.Client method), 9

## P

proxy\_ports (browsermobproxy.Client attribute), 9

## R

remap\_hosts() (browsermobproxy.Client method), 9

request\_interceptor() (browsermobproxy.Client method), 9

response\_interceptor() (browsermobproxy.Client method), 9

retry() (browsermobproxy.Client method), 9

rewrite\_url() (browsermobproxy.Client method), 10

## S

selenium\_proxy() (browsermobproxy.Client method), 10

Server (class in browsermobproxy), 10

start() (browsermobproxy.Server method), 10

stop() (browsermobproxy.Server method), 10

## T

timeouts() (browsermobproxy.Client method), 10

## W

wait\_for\_traffic\_to\_stop() (browsermobproxy.Client method), 10

webdriver\_proxy() (browsermobproxy.Client method), 10

whitelist() (browsermobproxy.Client method), 10