

---

# **FoxPuppet Documentation**

*Release latest*

**Mozilla**

**Oct 08, 2017**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Requirements . . . . .	3
1.2	Install FoxPuppet . . . . .	3
<b>2</b>	<b>User Guide</b>	<b>5</b>
2.1	Example . . . . .	5
<b>3</b>	<b>Development</b>	<b>7</b>
3.1	Automated Testing . . . . .	7
3.2	Running Tests . . . . .	7
<b>4</b>	<b>Release Notes</b>	<b>9</b>
<b>5</b>	<b>Developer Interface</b>	<b>11</b>
5.1	FoxPuppet . . . . .	11
5.2	Windows . . . . .	11
	<b>Python Module Index</b>	<b>13</b>



FoxPuppet is a library for automating user interactions in [Firefox](#) using [Selenium](#).



### Requirements

FoxPuppet will work with python 2.7, and 3.6

### Install FoxPuppet

To install FoxPuppet using `pip`:

```
$ pip install FoxPuppet
```

To install from source:

```
$ python setup.py develop
```





FoxPuppet is a library for automating user interactions in Firefox using Selenium. This section give some example use cases of FoxPuppet.

### Example

This is an example for setting up FoxPuppet and opening a private window:

```
from foxpuppet import FoxPuppet
from selenium.webdriver import Firefox

selenium = Firefox()
foxpuppet = FoxPuppet(selenium)

window = foxpuppet.browser.open_window(private=True)
selenium.quit()
```

This example opens first a non-private browser window, then proceeds to open a private browser window. Finally it quits via `selenium.quit()`.

**Note** The initial browser window is automatically assigned to the `browser` attribute and is always available.



### Automated Testing

All pull requests and merges are tested in [Travis CI](#) based on the `.travis.yml` file.

Usually, a link to your specific travis build appears in pull requests, but if not, you can find it on the [pull requests page](#)

The only way to trigger Travis CI to run again for a pull request, is to submit another change to the pull branch.

Test coverage is done by [Coveralls](#).

### Running Tests

You will need [Tox](#) installed to run the tests against the supported Python versions.

You also need to have [Geckodriver](#) installed so the tests can run Firefox.

```
$ pip install tox
$ tox
```



## CHAPTER 4

---

Release Notes

---



This part of the documentation describes the interfaces for using FoxPuppet.

### FoxPuppet

**class** `foxpuppet.foxpuppet.FoxPuppet` (*selenium*)

Set up the interface for interacting with the Firefox browser.

**Parameters** `selenium` – (`WebDriver`): Firefox `WebDriver` object.

### Windows

This module contains all of the window types as well as the window manager.

#### Window Manager

**class** `foxpuppet.windows.WindowManager` (*selenium*)

A window manager that controls the creation of window objects.

**Parameters** `selenium` – (`WebDriver`): Firefox `WebDriver` object.

#### Browser Window

**class** `foxpuppet.windows.browser.window.BrowserWindow` (*selenium, handle*)

Representation of a browser window.

**close** ()

Close the window.

**document\_element**

Return the inner DOM window element.

**Returns** WebDriver element object for the DOM window element.

**Return type** `WebElement`

**firefox\_version**

Major version of Firefox in use.

**Returns** Major component of the Firefox version.

**Return type** `int`

**is\_private**

Property that checks if the specified window is private or not.

**Returns** True if this is a Private Browsing window.

**Return type** `bool`

**navbar**

Provide access to the Navigation Bar.

**Returns** FoxPuppet NavBar object.

**Return type** `NavBar`

**notification**

Provide access to the currently displayed notification.

**Returns** FoxPuppet BaseNotification object.

**Return type** `BaseNotification`

**open\_window** (*private=False*)

Open a new browser window.

**Parameters** **private** (*bool*) – Optional parameter to open a private browsing window. Defaults to False.

**Returns** Opened window.

**Return type** `BrowserWindow`

**switch\_to** ()

Switch focus for Selenium commands to this window.

**wait\_for\_notification** (*notification\_class=<class 'foxpuppet.windows.browser.notifications.base.BaseNotification'>*)

Wait for the specified notification to be displayed.

**Parameters** **notification\_class** (`BaseNotification`, optional) – The notification class to wait for. If *None* is specified it will wait for any notification to be closed. Defaults to `BaseNotification`.

**Returns** Firefox notification.

**Return type** `BaseNotification`



**f**

`foxpuppet.foxpuppet`, 11  
`foxpuppet.windows`, 11  
`foxpuppet.windows.browser.window`, 11



## B

BrowserWindow (class in foxpuppet.windows.browser.window), 11

## C

close() (foxpuppet.windows.browser.window.BrowserWindow method), 11

## D

document\_element (foxpuppet.windows.browser.window.BrowserWindow attribute), 11

## F

firefox\_version (foxpuppet.windows.browser.window.BrowserWindow attribute), 12

FoxPuppet (class in foxpuppet.foxpuppet), 11

foxpuppet.foxpuppet (module), 11

foxpuppet.windows (module), 11

foxpuppet.windows.browser.window (module), 11

## I

is\_private (foxpuppet.windows.browser.window.BrowserWindow attribute), 12

## N

navbar (foxpuppet.windows.browser.window.BrowserWindow attribute), 12

notification (foxpuppet.windows.browser.window.BrowserWindow attribute), 12

## O

open\_window() (foxpuppet.windows.browser.window.BrowserWindow method), 12

## S

switch\_to() (foxpuppet.windows.browser.window.BrowserWindow method), 12

## W

wait\_for\_notification() (foxpuppet.windows.browser.window.BrowserWindow method), 12

WindowManager (class in foxpuppet.windows), 11