ghostly Documentation

Release 0.2.4

Alex Hayes

April 22, 2016

Contents

1 Contents 3

2 License 9

3 History 11

4 Author 13

Python Module Index 15

Lightweight API around Selenium Webdriver and helpers for end to end testing with Django.

This package is alpha, the API will most likely change!

It supports Django 1.6+ for Python versions 2.7, 3.3, 3.4, 3.5 and pypy (where the Django version supports the Python version).

CHAPTER 1

Contents

1.1 Installation

You can install ghostly either via the Python Package Index (PyPI) or from github.

To install using pip;

```
$ pip install ghostly
```

From github;

```
$ pip install git+https://github.com/alexhayes/ghostly.git
```

1.2 Usage

You can use use this package outside of Django however it has limited use.

Essentially there are two components, as follows;

- `Ghostly` - A lightweight wrapper and helper methods for Selenium Webdriver. Presently it provides a handful of methods that utilise xpath to deal with a page, such as `xpath`, `xpath_wait` et al.
- `GhostlyDjangoTestCase` - A lightweight test case that extends StaticLiveServerTestCase and sets up an instance of `Ghostly`. It provides methods such as `assertCurrentUrl`, `assertXpathEqual` et al.

1.2.1 GhostlyDjangoTestCase

`GhostlyDjangoTestCase` inherits StaticLiveServerTestCase and thus fires up a WSGI server that handles requests.

Given you have a named URL `home` with a `<h1>Hello World</h1>` visible in the source, you can do the following;

```
class MyTestCase(GhostlyDjangoTestCase):

    def test_homepage(self):
        self.goto(reverse('home'))

        # Assert that an element is equal to something
        self.assertXpathEqual('//h1', 'Hello World')
```

```
        # Assert the current url, relative or absolute
        self.assertCurrentUrl('/home')
```

1.2.2 Working with SVG

To traverse SVG with Selenium web driver you must use xpath.

```
class MyTestCase(GhostlyDjangoTestCase):

    def test_homepage(self):
        self.goto(reverse('home'))

        # Click on an element, or example, in an SVG.
        self.ghostly.xpath_click('//*[@id="refresh"]')

        # Assert that an Xpath is equal to something
        self.assertXpathEqual('//h2', 'Hello World')

        # Wait for xpath to exist
        self.ghostly.xpath_wait('//div[@id="something"]')
```

1.3 Developer Documentation

1.3.1 Contributions

Contributions are more than welcome!

To get setup do the following;

```
mkvirtualenv --python=/usr/bin/python3.5 ghostly
git clone https://github.com/alexhayes/ghostly.git
cd ghostly
pip install -r requirements/dev.txt
pip install Django>=1.8,<1.9
```

1.3.2 Running Tests

Once you've checked out you should be able to run the tests.

```
tox
```

1.3.3 Creating Documentation

```
cd docs
make clean html
```

1.4 Internal Module Reference

Release 0.2.4

Date April 22, 2016

1.4.1 ghostly.ghostly

Lightweight wrapper and helpers around Selenium Webdriver.

class `ghostly.ghostly.`**`Ghostly`**(*driver*, *maximise_window=True*)

Bases: `object`

Lightweight wrapper and helper utilities around Selenium webdriver.

`driver` = **None**

:type : webdriver.Chrome

`end`()

`form_fill`(*xpath*, ***data*)

Fill a simple form with data.

Parameters

- **`xpath`** – The xpath locator of the form.
- **`data`** – A dict of data on the form to fill. The key of each item should equal the name of an input field.

Returns selenium.webdriver.remote.webelement.WebElement

`form_submit`(*xpath*, ***data*)

Submit a form optionally setting (simple) data on the form.

Parameters

- **`xpath`** – The xpath locator of the form.
- **`data`** – A dict of data to supply to *`Ghostly.form_fill()`*

Returns selenium.webdriver.remote.webelement.WebElement

`get`(*url*)

Load the provided URL in the web driver

`wait`(*seconds*)

Wait for a specified number of seconds

`xpath`(*xpath*)

Finds an element by xpath.

This simply passes through to `WebDriver.find_element_by_xpath`.

Parameters **`xpath`** – The xpath locator of the element to find.

Returns selenium.webdriver.remote.webelement.WebElement

`xpath_click`(*xpath*, *wait=0.1*, *move_to=True*)

Click an element selected using xpath.

Parameters

- **`xpath`** – The xpath locator of the element to be clicked or an WebElement
- **`wait`** – Wait after the click - set to None for no wait.

- **move_to** – If True (default) then an ActionChains is created and move_to_element called - this approach works well for elements that respond to clicks such as a/span/div tags. If False, click is called on the element - this approach works well for choosing items in a select tag.

xpath_wait(*xpath*, *visible=True*, *timeout=5*, *sleep=0.25*, *click=False*, *click_move_to=True*, *click_wait=0.1*)

Wait for timeout seconds for xpath to exist and optionally be visible.

Parameters

- **xpath** – The xpath locator of the element to find.
- **visible** – If True, also wait for the element to become visible.
- **timeout** – Timeout in seconds before GhostlyTimeoutError is raised.
- **sleep** – How long to sleep for between each check to see if
- **click** – If True, then `Ghostly.xpath_click()` is called upon completion.
- **click_move_to** – Passed onto `Ghostly.xpath_click()` if click is True.
- **click_wait** – Passed onto `Ghostly.xpath_click()` if click is True.

Returns selenium.webdriver.remote.webelement.WebElement

1.4.2 ghostly.errors

All local errors that ghostly raises.

exception `ghostly.errors.`**GhostlyError**

Bases: `Exception`

Ghostly error, which all Ghostly errors inherit.

exception `ghostly.errors.`**DriverDoesNotExistError**

Bases: `ghostly.errors.GhostlyError`

Raised when `Ghostly.__init__()` can't load a specific driver.

exception `ghostly.errors.`**GhostlyTestFailed**

Bases: `ghostly.errors.GhostlyError`

1.4.3 ghostly.django.testcase

Module containing GhostlyDjangoTestCase.

class `ghostly.django.testcase.`**GhostlyDjangoTestCase**(*methodName='runTest'*)

Bases: `django.contrib.staticfiles.testing.StaticLiveServerTestCase`

Django TestCase that allows you to define your Ghostly tests pragmatically.

This class is mostly a light weight wrapper around Ghostly.

assertCurrentUrl(*expected*)

Assert the current URL is equal to expected.

Parameters **expected** – Expected URL, if relative the test servers URL is prepended.

assertXpathEqual(*xpath*, *expected*)

assertXpathNotVisible(*xpath*)

assertXpathVisible(*xpath*)

driver = 'PhantomJS'

goto(*url*)

Helper method to perform a HTTP GET with support for relative URLs.

Parameters

- **url** (*str*) – The URL to retrieve, if relative the test servers URL is prepended.
- **assert_statuses** (*list*) – A list of acceptable status codes.

maximise_window = True

setUp()

CHAPTER 2

License

This software is licensed under the *MIT License*. See the LICENSE.

CHAPTER 3

History

This package started out as a simple way to construct browser tests using YAML, written by Brenton Cleeland.

The focus of this fork is to allow the developer to write programmatic unit tests in the style of unittest.

Currently this fork does not contain any of the CSS selector style methods that were originally available as the focus has been on xpath only support until a more robust CSS selector toolkit can be provided.

CHAPTER 4

Author

- Alex Hayes <alex@commoncode.com>
- Brenton Cleeland <brenton@commoncode.com>

g

ghostly.django.testcase, 6
ghostly.errors, 6
ghostly.ghostly, 5

A

assertCurrentUrl() (ghostly.django.testcase.GhostlyDjangoTestCase method), 6
assertXpathEqual() (ghostly.django.testcase.GhostlyDjangoTestCase method), 6
assertXpathNotVisible() (ghostly.django.testcase.GhostlyDjangoTestCase method), 6
assertXpathVisible() (ghostly.django.testcase.GhostlyDjangoTestCase method), 6

D

driver (ghostly.django.testcase.GhostlyDjangoTestCase attribute), 7
driver (ghostly.ghostly.Ghostly attribute), 5
DriverDoesNotExistError, 6

E

end() (ghostly.ghostly.Ghostly method), 5

F

form_fill() (ghostly.ghostly.Ghostly method), 5
form_submit() (ghostly.ghostly.Ghostly method), 5

G

get() (ghostly.ghostly.Ghostly method), 5
Ghostly (class in ghostly.ghostly), 5
ghostly.django.testcase (module), 6
ghostly.errors (module), 6
ghostly.ghostly (module), 5
GhostlyDjangoTestCase (class in ghostly.django.testcase), 6
GhostlyError, 6
GhostlyTestFailed, 6
goto() (ghostly.django.testcase.GhostlyDjangoTestCase method), 7

M

maximise_window (ghostly.django.testcase.GhostlyDjangoTestCase attribute), 7

S

setUp() (ghostly.django.testcase.GhostlyDjangoTestCase method), 7

W

wait() (ghostly.ghostly.Ghostly method), 5

X

xpath() (ghostly.ghostly.Ghostly method), 5
xpath_click() (ghostly.ghostly.Ghostly method), 5
xpath_wait() (ghostly.ghostly.Ghostly method), 6