
Spectacles Documentation

Release 0.1

Christo Crampton

Jul 12, 2017

Contents

1	Getting Started	3
1.1	Installation	3
1.2	Quickstart	3
1.3	Next Steps	4
2	Spectacles Command Line Interface	5
2.1	Basic usage	5
2.2	Specify a browser driver	5
2.3	Spec discovery	5
3	Outputting Messages	7
4	Interacting with the browser	9
5	Indices and tables	11

Contents:

CHAPTER 1

Getting Started

Installation

```
pip install python-spectacles
```

Quickstart

The default project layout is like this:

```
.
|_ specs/ # put your yaml specs here
|_ reports/
    |_ screenshots # any screenshots you take will be saved here
    |_ specs # spec reports will be saved here
```

You can create that layout with the following commands:

```
mkdir specs
mkdir -p reports/screenshots
mkdir -p reports/specs
```

Let's create a quick spec:

file: specs/im_feeling_lucky.yml

```
---
- scenario: I'm feeling lucky
  steps:
    - goto: /
    - screenshot: {}
    - expect_elements :
      - "#1st-ib": "search input"
```

```
- fill_fields:
  - q: "testing"
- wait: 1
- click: "[name='btnG']"
- wait: 5
- wait_for_element : "#rcnt"
- screenshot:
  widths: [375, 768, 990, 1200, 1600]
```

Run your spec:

```
spectacles https://www.google.com
```

You should see a result something like the following:

```
##I'm feeling lucky

* Go to: /
* ✓ Check that search input exists
* enter django-spectacles into field:q
* ✓ Check that [name='btnG'] exists
* Click button:
* waiting for #rcnt to load
* ✓ Element has loaded: #rcnt
* Click the first link
* ✓ Check that h3.r a exists
* Click button: GitHub - toast38coza/django-spectacles: Write Integration tests in ...
```

Next Steps

Using includes

There are a lot of things that you might want to do repeatedly. It doesn't make sense to have to rewrite these the whole time. That's where the *include* command comes in.

Create a new folder inside specs:

Now you can create some sub-specs in "code:specs/includes". You can then include them from a top-level spec like so:

```
---
- scenario: Do some stuff
  steps:
    - include: specs/includes/login.yml
    - include: specs/includes/do-something.yml
    - include: specs/includes/do-something-else.yml
```

Note: by default spectacles will look for specs with the following glob `specs/*.yml`. Therefore the specs in includes won't be run by the spec runner by default, but they can be included with the include directive

Note: the include path is relative to the location from which spectacles is being run.

Spectacles Command Line Interface

Basic usage

Specify a browser driver

Spec discovery

CHAPTER 3

Outputting Messages

```
class spectacles.printer.Printer
```

```
    step (message)
```

```
        Print out a descriptive message explaining what we're doing
```


CHAPTER 4

Interacting with the browser

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

P

Printer (class in `spectacles.printer`), [7](#)

S

`step()` (`spectacles.printer.Printer` method), [7](#)