pytest-{{cookiecutter.plugin_name}} Documentation

Release 0.0.3

{{cookiecutter.full_name}}

Jun 26, 2019

Contents

pytest-play plugin driving browsers using Selenium/Splinter under the hood. Selenium grid compatible and implicit auto wait actions for more robust scenarios with less pain.

More info and examples on:

- pytest-play, documentation
- cookiecutter-qa, see `pytest-play` in action with a working example if you want to start hacking

CHAPTER 1

Browser based commands

Browser based commands here. `play_selenium` supports by default browser interactions. For example it can be used for running selenium splinter scenarios driving your browser for your UI test or system tests.

`play_selenium` is also your friend when page object approach (considered best practice) is not possible. For example:

- limited time, and/or
- lack of programming skills

Instead if you are interested in a page object pattern have a look at pypom_form or pypom.

`play_selenium` supports automatic waiting that should help to keep your tests more reliable with implicit waits before moving on. By default it waits for node availability and visibility but it supports also some wait commands and wait until a given Javascript expression is ok. So it is at the same time user friendly and flexible.

1.1 Conditional commands (Javascript)

Based on a browser level expression (Javascript):

```
- type: clickElement
  provider: selenium
  locator:
    type: css
    value: body
  condition: "'$foo' === 'bar'"
```

1.2 Supported locators

Supported selector types:

- css

- xpath
- tag
- name
- text
- id
- value

1.3 Open a page

With parametrization:

```
- type: get
  provider: selenium
  url: "$base_url"
```

or with a regular url:

```
- type: get
  provider: selenium
  url: https://google.com
```

1.4 Pause

This command invokes a javascript expression that will pause the execution flow of your commands:

```
- type: pause
  provider: selenium
  waitTime: 1500
```

If you need a pause/sleep for non UI tests you can use the `sleep` command provided by the play_python plugin.

1.5 Click an element

```
- type: clickElement
  provider: selenium
  locator:
    type: css
    value: body
```

1.6 Fill in a text

```
- type: setElementText
  provider: selenium
  locator:
    type: css
```

(continues on next page)

(continued from previous page)

```
    value: input.title
  text: text value
```

1.7 Interact with select input elements

Select by label:

```
- type: select
  provider: selenium
  locator:
    type: css
    value: select.city
  text: Turin
```

or select by value:

```
- type: select
  provider: selenium
  locator:
    type: css
    value: select.city
  value: '1'
```

1.8 Eval a Javascript expression

```
- type: eval
  provider: selenium
  script: alert('Hello world!')
```

1.9 Create a variable starting from a Javascript expression

The value of the Javascript expression will be stored in `play.variables` under the name `count`:

```
- type: storeEval
  provider: selenium
  variable: count
  script: document.getElementById('count')[0].textContent
```

1.10 Assert if a Javascript expression matches

If the result of the expression does not match an `AssertionError` will be raised and the test will fail:

```
- type: verifyEval
  provider: selenium
  value: '3'
  script: document.getElementById('count')[0].textContent
```

1.11 Verify that the text of one element contains a string

If the element text does not contain the provided text an `AssertionError` will be raised and the test will fail:

```
- type: verifyText
  provider: selenium
  locator:
    type: css
    value: ".my-item"
  text: a text
```

1.12 Send keys to an element

All `selenium.webdriver.common.keys.Keys` are supported:

```
- type: sendKeysToElement
  provider: selenium
  locator:
    type: css
    value: ".confirm"
  text: ENTER
```

Supported keys:

```
KEYS = [
    'ADD', 'ALT', 'ARROW_DOWN', 'ARROW_LEFT', 'ARROW_RIGHT',
    'ARROW_UP', 'BACKSPACE', 'BACK_SPACE', 'CANCEL', 'CLEAR',
    'COMMAND', 'CONTROL', 'DECIMAL', 'DELETE', 'DIVIDE',
    'DOWN', 'END', 'ENTER', 'EQUALS', 'ESCAPE', 'F1', 'F10',
    'F11', 'F12', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8',
    'F9', 'HELP', 'HOME', 'INSERT', 'LEFT', 'LEFT_ALT',
    'LEFT_CONTROL', 'LEFT_SHIFT', 'META', 'MULTIPLY',
    'NULL', 'NUMPAD0', 'NUMPAD1', 'NUMPAD2', 'NUMPAD3',
    'NUMPAD4', 'NUMPAD5', 'NUMPAD6', 'NUMPAD7', 'NUMPAD8',
    'NUMPAD9', 'PAGE_DOWN', 'PAGE_UP', 'PAUSE', 'RETURN',
    'RIGHT', 'SEMICOLON', 'SEPARATOR', 'SHIFT', 'SPACE',
    'SUBTRACT', 'TAB', 'UP',
]
```

1.13 Wait until a Javascript expression matches

Wait until the given expression matches or raise a `selenium.common.exceptions.TimeoutException` if takes too time.

At this time of writing there is a global timeout (20s) but in future releases you will be able to override it on command basis:

```
- type: waitUntilCondition
  provider: selenium
  script: document.body.getAttribute('class') === 'ready'
```

1.14 Wait for element present in DOM

Present:

```
- type: waitForElementPresent
  provider: selenium
  locator:
    type: css
    value: body
```

or not present:

```
- type: waitForElementPresent
  provider: selenium
  locator:
    type: css
    value: body
  negated: true
```

1.15 Wait for element visible

Visible:

```
- type: waitForElementVisible
  provider: selenium
  locator:
    type: css
    value: body
```

or not visible:

```
- type: waitForElementVisible
  provider: selenium
  locator:
    type: css
    value: body
  negated: true
```

1.16 Assert element is present in DOM

An `AssertionError` will be raised if assertion fails.

Present:

```
- type: assertElementPresent
  provider: selenium
  locator:
    type: css
    value: div.elem
```

or not present:

```
- type: assertElementPresent
  provider: selenium
  locator:
    type: css
    value: div.elem
  negated: true
```

1.17 Assert element is visible

An `AssertionError` will be raised if assertion fails.

Present:

```
- type: assertElementVisible
  provider: selenium
  locator:
    type: css
    value: div.elem
```

or not present:

```
- type: assertElementVisible
  provider: selenium
  locator:
    type: css
    value: div.elem
  negated: true
```

CHAPTER 2

Twitter

play_selenium tweets happens here:

- @davidemoro

2.1 Welcome to play_selenium's documentation!

Contents:

2.1.1 Changelog

0.0.3 (2019-06-26)

- according to new pytest-play versions (>= 2.0) the self.engine.parametrize should be used instead of accessing self.engine.parametrizer

0.0.2 (2019-02-18)

- Remove pytest version constraint (added compatibility with pytest>=4)

0.0.1 (2019-01-25)

- Supports new pytest-play >= 2.0 YAML based syntax (json no more supported)

2.2 Indices and tables

- genindex

- modindex
- search