

Selenium

Testing web applications

Michal Hořejšek, horejsekmichal@gmail.com



SEZNAM.CZ

```
def test_list_clients_contracts(api, client, contract):  
    res = api.client.list_contracts(client.id)  
    assert len(res) == 1  
    assert res[0] == contract
```

```
def test_client_persons_suggest(test_app, client):  
    data = {  
        'query': client.surname,  
    }  
    response = test_app.json('/client/search', data)  
  
    assert response.status_code == 200  
    assert response['clients'][0] == client
```

```
describe 'Restaurant', () ->
  restaurant = new models.Restaurant
    name: 'Delicious dream'
    lastUpdate: '2013-01-01 18:30'

describe '#getPrintalbeLastUpdate()', () ->
  it 'should return date as pretty string', () ->
    res = restaurant.getPrintalbeLastUpdate()
    assert.equal res, '1. ledna v 18:30'
```

** Actually this is test of our web app with restaurant's lunch menus at Prague Anděl, lunchtimeandel.cz. Don't forget to use it when you will be around. :-)*

Selenium

„Selenium automates browsers. That's it!“

```
from selenium import webdriver

driver = webdriver.Chrome()
driver.get('http://www.seznam.cz')
print driver.title
driver.quit()
```

```
from selenium.webdriver.common.keys import Keys

input = driver.find_element_by_name('q')
input.send_keys('selenium' + Keys.TAB)

submit = driver.find_element_by_id('submit')
submit.click()
```

```
driver.get_elm('f').fill_out_and_submit({  
    'q': 'selenium',  
})
```



```
>>> driver.go_to('http://www.seznam.cz')
>>> driver.get_elm('resultCount')
NoSuchElementException: Message: u'No
element <* id=resultCount> found at
http://www.seznam.cz/'
```

```
f = driver.get_elm('link').download_file()  
f.method  
f.status_code  
f.headers  
f.data
```

```
elm = driver.get_elm('q')  
elm.send_keys('aa')
```

```
# JavaScript changes DOM  
time.sleep(1)
```

```
# Raises StaleElementException  
elm.send_keys('bb')
```

```
driver.get_elm('q').send_keys('aa')
```

```
# JavaScript changes DOM
```

```
time.sleep(1)
```

```
# OK now
```

```
driver.get_elm('q').send_keys('bb')
```

```
from selenium.webdriver.support.wait import WebDriverWait

def callback(driver):
    return driver.find_element_by_id('id')

wait = WebDriverWait(driver, timeout=10)
wait.until(callback)
```

```
driver.wait_for_element(id_='id')
```

```
driver.contains_text('sherlock')  
driver.find_element_by_text('sherlock')
```

```
//div[contains(text(), "xxx")]
```

```
//select[@name="country"]/option[@value="cr"]
```

http://www.w3schools.com/xpath/xpath_intro.asp


```
class TestCase(WebdriverTestCase):
    domain = 'https://www.google.com'
    instances_of_driver = ONE_INSTANCE_PER_TESTCASE

    def _get_driver(self):
        return Chrome()

    def test_doodle(self):
        self.click('gbqfsb')
        res = self.contains_text('Doodles')
        self.assertTrue(res)
```

```
@expected_error_page(403)
def test_no_permission_for_admin(self):
    self.go_to('/admin')

@expected_error_messages('exists')
def test_already_exists(self):
    self.get_elm('form').fill_out_and_submit({
        'item.username': 'admin',
    })
```

```
<script type="text/javascript">
    window.jsErrors = [];
    window.onerror = function(errorMessage) {
        window.jsErrors.push(errorMessage);
    }
</script>
```

```
def test_doodle(driver):  
    driver.click('gbqfsb')  
    assert driver.contains_text('Doodles')  
  
def test_search(driver):  
    driver.get_elm('gbqf').fill_out_and_submit({  
        'q': 'hello',  
    })  
    driver.wait_for_element(id_='resultStats')
```

Virtual X server

Choices: Xvfb, Xephyr, Xvnc

```
# apt-get install xvfb  
# pip install pyvirtualdisplay
```

```
from pyvirtualdisplay import Display

display = Display(size=(800, 400))
display.start()
# ...
display.stop()
```

Selenium grid

- **More browsers**
- **More operating systems**
- **Faster tests**

```
url = 'http://127.0.0.1:4444/wd/hub'  
desire = webdriver.DesiredCapabilities.FIREFOX.copy()  
  
driver = webdriver.Remote(  
    command_executor=url,  
    desired_capabilities=desire,  
)
```


Check documentation for more info:

<http://horejsek.github.io/python-webdriverwrapper>



Thank you!

Michal Hořejšek, horejsekmichal@gmail.com