WATIR

Testing Web Applications

Overview

WATIR - Web Application Testing in Ruby

- Provides a Ruby interface for interacting with a browser
  - Supports common browsers
- Tests are scripted, not recorded

Basic Example

```ruby
require 'watir-webdriver'

browser = Watir::Browser.new :firefox

browser.goto 'bit.ly/watir-webdriver-demo'
browser.text_field(:id => 'entry_1000000').set 'your name'
browser.select_list(:id => 'entry_1000001').select 'Ruby'
browser.button(:name => 'submit').click
sleep 2
puts browser.url
puts browser.text
expect(browser.text).to include 'Thank you'
browser.close
```

More Basics

- Exposes all HTML elements and element collections
- Allows control of the browser (e.g. forward, backward, url, title)

Integrating with Cucumber

```
Feature: Platform Registration
    Scenario: Basic Registration
        Given I visit the platform registration system
        When I submit my information
        Then I get a big thanks

Before do |scenario|
    @browser = Watir::Browser.new 'safari'
  end
  After do |scenario|
    @browser.close
  end

Given(/^I visit the platform registration system$/) do
    @browser.goto 'bit.ly/watir-webdriver-demo'
  end
  When(/^I submit my information$/) do
    @browser.text_field(:id => 'entry_1000000').set 'your name'
    @browser.select_list(:id => 'entry_1000001').select 'Ruby'
    @browser.button(:name => 'submit').click
    sleep 2
  end
  Then(/^I get a big thanks$/) do
   expect(@browser.text).to include 'Thank you'
  end
```

Problems with WATIR

- Interacts with browser, we usually want to interact with a page
- Steps have knowledge of the implementation of a page
- May need to change many steps if an element id changes

Page Objects

- Page implementation abstracted from its presentation
- 1 class per page
- Only change the page class if the page implementation changes, not every test

Page Objects

```ruby
class Survey
    include PageObject

    page_url "bit.ly/watir-webdriver-demo"

    text_field(:name, :id => 'entry_1000000')
    select(:language, :id => 'entry_1000001')
    button(:submit, :name => 'submit')
  end

class SurveyResult
    include PageObject

    div(:response, :xpath => '//div[@class="ss-custom-resp"]')
  end
```

Page Objects

```ruby
include PageObject::PageFactory

  Given(/^I visit the platform registration system$/) do
    visit_page Survey
  end
  When(/^I submit my information$/) do
    on_page Survey do |page|
      page.name = 'your name'
      page.language = 'Ruby'
      page.submit
    end
    sleep 2
  end
  Then(/^I get a big thanks$/) do
    on_page SurveyResult do |page|
      expect(page.response).to include 'Thank you'
    end
  end
```