

AUTOMATION TESTING WITH CUCUMBER

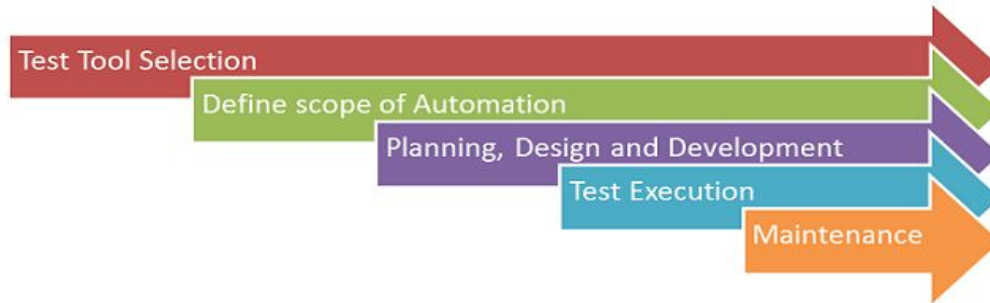
Developing Cucumber-Based Automation

Why Automation

- Increased Test Speed, Test Efficiency and Software Quality
- Decreased Costs and shorter Time to Market
- Regression test after debugging or further development of software
- Greater Test Coverage

Automation Process

Following steps are followed in an Automation Process



Cucumber

Cucumber is a tool that executes plain-text functional descriptions as automated tests. The language that Cucumber understands is called [Gherkin](#).

What are the benefits?

1. It is helpful to involve business stakeholders who can't easily read code
2. Cucumber focuses on end-user experience
3. Style of writing tests allow for easier reuse of code in the tests
4. Quick and easy setup and execution
5. Instead of writing your tests purely in code, with Cucumber you start by writing a human-readable user story. Then, you write code to run the story and perform test(s) based on it.

Feature Introduction

Every .feature file conventionally consists of a single feature. A line starting with the keyword Feature followed by free indented text starts a feature. A feature usually contains a list of scenarios. You can write whatever you want up until the first scenario, which starts with the word Scenario (or localized equivalent; Gherkin is localized for dozens of languages) on a new line. You can use tagging to group features and scenarios together independent of your file and directory structure.



Given The login page is opening

When I input username, password and click Login button

Then I am on the Home page

Identifying the Elements by using Xpaths

Add Firefox Plugin: FirePath and FireBug

Absolute Xpath: To get the Absolute location of the element

Start selection from the document node

Starts with “/” Selects from the root node

Starts with “//” Selects nodes in the document from the current node that match the selection no matter where they are

Eg: Google Search Box :

```
html/body/div[1]/div[2]/div[1]/div[1]/div[3]/div/div/div/form/fieldset[2]/div/div/div/div/div[3]/div/input[3]
```

Relative Xpath :

Instead of starting at the root element, you can simply reference the element you want and go from there.

Starts with “./”*”

Eg: Google Search Box : `./*[@id='gs_htif0']`

Watir WebDriver

Watir WebDriver: the most elegant way to use WebDriver with ruby.

Watir (pronounced Water) stands for **Web Application Testing in Ruby**.

Getting Started

```
gem install watir-webdriver
```

Cucumber + Watir-WebDriver work really well together to automate your web app testing with a business domain specific specification layer.

Examples

Including **Watir-WebDriver** gem to drive Firefox/Chrome on Windows/Mac/Linux

```
require 'watir-webdriver'
```

Starting a new browser & and going to our site

```
browser = Watir::Browser.new
```

```
browser.goto 'http://bit.ly/watir-example'
```

Setting a text field

What is your name? *

A text field

```
browser.text_field(:name => 'entry.0.single').set 'Watir'
```

Setting a multi-line text box

What is your story?

A text box



```
browser.text_field(:name => 'entry.1.single').set "I come here from Australia. \n The weather is great here."
```

Setting and clearing a radio button

What testing tool do you like?

Radio buttons

- ☐ Watir
- ☐ Selenium

```
browser.radio(:value => 'Watir').set  
browser.radio(:value => 'Watir').clear
```


Setting and clearing check boxes

What language do you like?

Checkboxes

☐ Ruby

☐ Java

☐ Python

```
browser.checkbox(:value => 'Ruby').set  
browser.checkbox(:value => 'Python').set  
browser.checkbox(:value => 'Python').clear
```

Clearing, getting and selecting selection list values

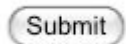
What browser do you use?

Drop down box

Firefox

```
browser.select_list(:name => 'entry.6.single').clear  
puts browser.select_list(:name => 'entry.6.single').options  
browser.select_list(:name => 'entry.6.single').select 'Chrome'
```

Clicking a button



```
browser.button(:name => 'submit').click
```

Checking for text in a page

```
puts browser.text.include? 'Your response has been recorded.'
```

Checking the title of a page

```
puts browser.title == 'Thanks!'
```

By id : we can get the element location by giving its unique Id.

Eg: [`@id`, "text_search_query"]

Example

- 1) Open up google.com
- 2) Enter "Qwinix Technologies"
- 3) click on Search button
- 4) The result page should have the first result - www.qwinixtech.com/

Setting up/Installing Cucumber

Install Ruby



***Install Ruby Development
Kit***



Install Cucumber



Install IDE



Install Ansicon



Install Watir



Install rspec

