Selenium Tutorial, WebDriver

- http://www.seleniumeasy.com/selenium-tutorials
- http://www.seleniumeasy.com/selenium-webdriver-tutorials

Selenium Python Bindings

- http://selenium-python.readthedocs.io/locating-elements.html

Testing Framework used with Selenium

Testing frameworks that can be used with Selenium:

- Python - Frameworks: unittest, pyunit, py.test, robot framework
- Ruby - Frameworks: RSpec, Test::Unit
- Java - Frameworks: JUnit, TestNG
- PHP - Frameworks: Behat + Mink, Yii
- C# - Frameworks: NUnit Haskell
- JavaScript
- Objective-C
- Perl
- R

Selenium POM - Page Object Model

guru99 POM

GURU99 Xpath

Selenium JavaScript

How engines work?

Engines are complicated. But the basics are easy.

1.The script is written and distributed as a plain text (can be compressed/optimized by so-called “javascript minifiers”).

2.The engine (embedded if it’s a browser) reads the script (“parses”) and converts (“compiles”) it to the machine language.

3.And then it runs, pretty fast.

The engine applies optimizations on every stage of the process. It even watches the script as it runs, analyzes the data which flows through it and applies optimizations to the machine-code basing on that knowledge. That’s why the code runs fast.

https://javascript.info/intro

Selenium with Python

- Python Django tips

Marionette is an automation driver for Mozilla's Gecko engine.

https://stackoverflow.com/questions/42956380/difference-between-geckodriver-and-marionette

Selenium uses W3C Webdriver protocol to send requests to Geckodriver, which translates them and uses Marionette protocol to send them to Firefox

Selenium<--(W3C Webdriver)-->Geckodriver<---(Marionette)--->Firefox

original discussion

Marionette

Marionette is an automation driver for Mozilla's Gecko engine. It is basically the the remote protocol of Gecko/Firefox. It can remotely control either the UI or the internal JavaScript of a Gecko platform, such as Firefox. Marionette consists of two parts: A server which takes requests and executes them in Gecko, and a client. The client sends commands to the server and the server executes the command inside the browser. Marionette combines a gecko component (the Marionette server) with an outside component (the Marionette client), which drives the tests. The Marionette server ships with Firefox, and to use it you will need to download a Marionette client.

GeckoDriver

With the release of Selenium 3 things changed. As Selenium 3 will not have any native implementation of Firefox, we have to direct all the driver commands through Gecko Driver. Gecko Driver is an executable file that you need to have in one of the system path before starting your tests. Firefox browser implements the WebDriver protocol using an executable called GeckoDriver.exe. This executable starts a server on your system. All your tests communicate to this server to run your tests. It translates calls into the Marionette automation protocol by acting as a proxy between the local and remote ends.

Up to version 45, the driver used to automate Firefox was an extension included with each client. But this extension was dropped, probably due to the change of policy which now requires all the extensions to be signed by Mozilla.

Marionette is the new driver that is shipped/included with Firefox. This driver has it's own protocol which is not directly compatible with the Selenium/WebDriver protocol.

The Gecko driver (previously named wires) is an application server implementing the Selenium/WebDriver protocol. It translates the Selenium commands and forwards them to the Marionette driver.

For the Java client, the default behavior is to use the Gecko driver, but it can be overridden to use the legacy extension as a driver with the webdriver.firefox.marionette property or with the marionette capability:

Gecko Driver

Mozilla runs on **Gecko browser engine**.

Gecko browser engine is an open source browser engine which can be used by anyone in there application. It can help applications render web pages. Just like other browsers, Chrome, Internet explorer and Edge Mozilla foundation has decided to introduce **Marionette driver** to control Firefox browser.

Moving forward, it is expected that to interact with Firefox Browser we will need to run an instance of Marionette driver. Marionette driver, some times loosely termed as Gecko driver, can drive both UI and Web Page in the Firefox browser. This gives an exceptional amount of control over the Web Page and the UI of the web browser to the tester.

Selenium 3 has upgraded itself to now launch Firefox driver using Marionette driver instead of the default initialisation supported earlier.

http://toolsqa.com/selenium-webdriver/how-to-use-geckodriver/

This program provides the HTTP API described by the WebDriver protocol to communicate with Gecko browsers, such as Firefox. It translates calls into the Firefox remote protocol by acting as a proxy between the local- and remote ends.

geckodriver github

Marionette Driver

Marionette is an automation driver for Mozilla's Gecko engine. It can remotely control either the UI or the internal JavaScript of a Gecko platform, such as Firefox. It can control both the chrome (i.e. menus and functions) or the content (the webpage loaded inside the browsing context), giving a high level of control and ability to replicate user actions. In addition to performing actions on the browser, Marionette can also read the properties and attributes of the DOM.

https://developer.mozilla.org/en-US/docs/Mozilla/QA/Marionette

In [ ]:
```
from selenium import webdriver
browser = webdriver.Firefox()
browser.get('http://inventwithpython.com')
#assert 'Django' in browser.title
```

Selenium and scripts in NodeJS

Node JS - Documentation for writing automate test scripts in Node JS.

https://www.browserstack.com/automate/node

Protractor JS and TypeScript Tutorial

http://www.diwebsity.com/2016/11/17/protractorjs-typescript-tutorial/

AngularJS HelloWorkld

https://dzone.com/articles/angularjs-my-first-hello-world

Karma Framework

https://stackoverflow.com/questions/32557644/testing-angularjs-application-with-selenium

Selenium AngularJS

http://qavalidation.com/2015/04/selenium-testing-for-angular-js-sites.html/

Clojure

http://www.compoundtheory.com/writing-angularjs-with-clojurescript-and-purnam/

IJavaScript

https://github.com/n-riesco/ijavascript#installation

Java Selenium

1. http://scraping.pro/how-to-use-selenium-webdriver-with-java/
2. http://www.software-testing-tutorials-automation.com/2014/01/learn-selenium-webdriver-online-free.html

Protractor

Testing AngularJS apps

Selenium Node JS

https://team.goodeggs.com/getting-started-with-selenium-webdriver-for-node-js-f262a00c52e1