
Test Automation & Advanced Selenium
Lesson 4: Selenium 2.0 – WebDriver
Capgemini

## Lesson Objectives

- Introduction To WebDriver
- Selenium WebDriver Architecture
- Selenium WebDriver Architecture - Components
- Web Driver Vs Selenium RC Vs Selenium IDE
- Benefits of Web Driver over Selenium IDE and RC
- Limitations of Web Driver

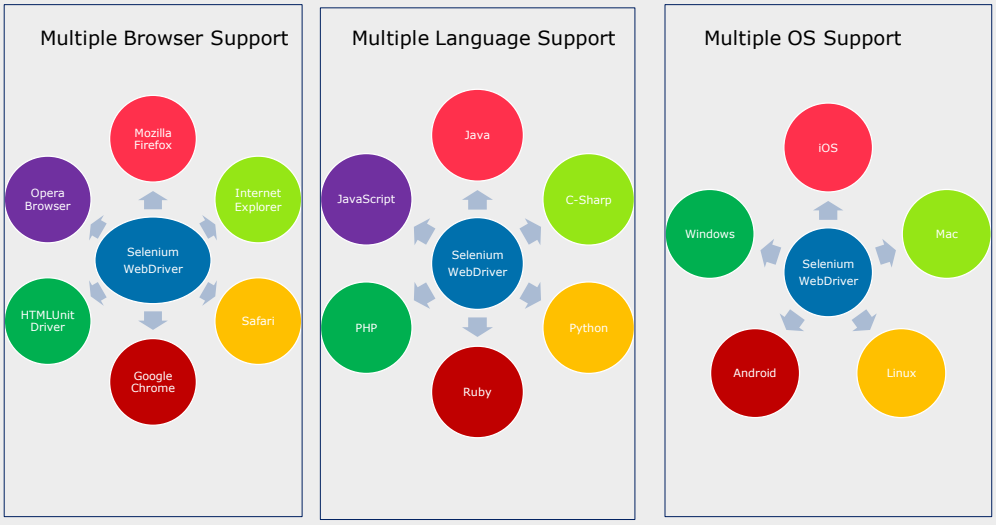
4.1: Selenium 2.0 – WebDriver

## Introduction To WebDriver

- Selenium WebDriver is one of the most important key component of Selenium releases
- It is a web automation framework that allows you to execute your tests against different browsers and not just in Firefox, unlike Selenium IDE
- It is said to be more advanced and sophisticated replacement for Selenium RC
- Selenium WebDriver is a well-designed Object Oriented API that provides improved support for web application testing
- Selenium WebDriver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded
- Selenium WebDriver supports many browsers including headless HTMLUnit browser and it supports multiple programming languages
- It works on different operating systems as well

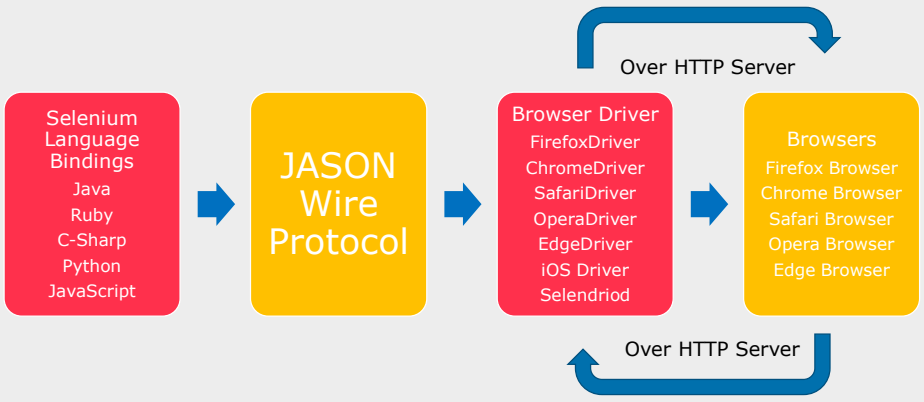
4.1: Selenium 2.0 – WebDriver

## Introduction To WebDriver (Cont.)



4.1: Selenium 2.0 – WebDriver

## Selenium WebDriver Architecture



4.1: Selenium 2.0 – WebDriver

## Selenium WebDriver Architecture - Components

- Language Bindings:
  - These are like glue code/wrapper libraries written in corresponding languages to communicate with the WebDriver API. Selenium Developers have developed language bindings to allow Selenium to support multiple languages. The language binding will send the commands across the common driver API.
- JSON Wire Protocol:
  - JSON stands for JavaScript Object Notation. It is used to transfer data between a server and a client on the web. It transfers information between HTTP server. Each browser driver having it's own HTTP server.
- Browser Drivers:
  - Browser drivers will do the important job of driving the corresponding browsers. Example: a Chrome browser driver knows how to drive Chrome browser to perform low level activities such as interacting with UI elements, navigating between pages etc. We mention the required driver in our code. When we run our tests, the driver server listens on a port on our local machine. It will interpret the commands received from the WebDriver API, execute on the actual browser and return the results back to our code through the API.

4.1: Selenium 2.0 – WebDriver

## Web Driver Vs Selenium RC Vs Selenium IDE

- WebDriver
  - Supports all browsers like Firefox, IE, Chrome, Safari, Opera etc.
  - Doesn't supports Record and playback
  - Doesn't required to start server before executing the test script
  - Interacts natively with browser application
  - As compared to RC, it is bit complex and large API.
- Selenium RC
  - Supports all browsers like Firefox, IE, Chrome, Safari, Opera etc.
  - Doesn't supports Record and playback
  - Required to start server before executing the test script.
  - Core engine is JavaScript based
  - It is easy and small API
- Selenium IDE
  - Only works in Mozilla browser
  - Supports Record and playback
  - Doesn't required to start server before executing the test script.
  - Core engine is JavaScript based
  - Very simple to use

4.1: Selenium 2.0 – WebDriver

## Benefits of Web Driver over Selenium IDE and RC

- Architecture is simpler than Selenium RC's
- Faster than Selenium RC
- Interacts with page elements in a more realistic way
- API is simpler than Selenium RC's
- Support the headless HtmlUnitDriver browser

1. **Architecture:** WebDriver's architecture is simpler than Selenium RC's. It controls the browser from the OS level. All you need are your programming language's IDE (which contains your Selenium commands) and a browser. You first need to launch a separate application called Selenium Remote Control (RC) Server before you can start testing. The Selenium RC Server acts as a "middleman" between your Selenium commands and your browser. When you begin testing, Selenium RC Server "injects" a Javascript program called Selenium Core into the browser. Once injected, Selenium Core will start receiving instructions relayed by the RC Server from your test program. When the instructions are received, Selenium Core will execute them as Javascript commands. The browser will obey the instructions of Selenium Core, and will relay its response to the RC Server. The RC Server will receive the response of the browser and then display the results to you. RC Server will fetch the next instruction from your test script to repeat the whole cycle.

2. **Speed:** WebDriver is faster than Selenium RC since it speaks directly to the browser uses the browser's own engine to control it. Selenium RC is slower since it uses a Javascript program called Selenium Core. This Selenium Core is the one that directly controls the browser, not you.

3. **Real-life Interaction:** WebDriver interacts with page elements in a more realistic way. For example, if you have a disabled text box on a page you were testing, WebDriver really cannot enter any value in it just as how a real person cannot. Selenium Core, just like other JavaScript codes, can access disabled elements. In the past, Selenium testers complain that Selenium Core was able to enter values to a disabled text box in their tests.

4.1: Selenium 2.0 – Web Driver

## Limitations of Web Driver

- Cannot Readily Support New Browsers
  - Since, WebDriver operates on the OS level
  - Different browsers communicate with the OS in different ways.
  - If a new browser comes out, it may have a different process of communicating with the OS as compared to other browsers
- Has no built-in command that automatically generates a Test Results File
  - Have to rely on your IDE's output window, or design the report yourself using the capabilities of your programming language and store it as text, html, etc.

## Summary


In this lesson, you have learnt

- WebDriver is a tool for testing web applications across different browsers using different programming languages
- You are now able to make powerful tests because WebDriver allows you to use a programming language of your choice in designing your tests
- WebDriver is faster than Selenium RC because of its simpler architecture
- WebDriver directly talks to the browser while Selenium RC needs the help of the RC Server in order to do so
- WebDriver's API is more concise than Selenium RC's
- WebDriver can support HtmlUnit while Selenium RC cannot
- The only drawbacks of WebDriver are:
  - It cannot readily support new browsers, but Selenium RC can
  - It does not have a built-in command for automatic generation of test results

Add the notes here.

## Review Question

Question 1

- Select which is NOT a feature of Web Driver
- Architecture is simpler
- Faster
- API is complex
- Supports all the browsers

Question 2: True/False

- WebDriver can Readily Support New Browsers.

Question 3: Fill in the Blanks

- WebDriver can support _____while Selenium RC cannot.