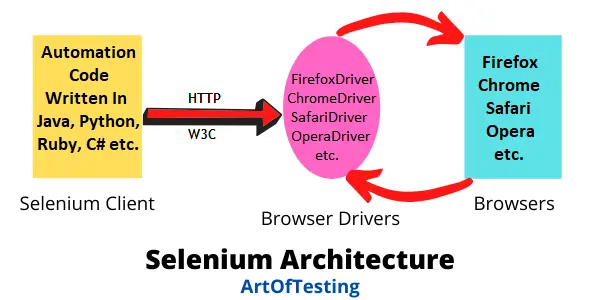
**Selenium Class**

# Chapter#2 – Selenium WebDriver Architecture Explained

1. **Selenium WebDriver Architecture**
2. **Selenium Client Library**
3. **Browser Drivers**
4. **JSON Wire Protocol over HTTP**
5. **Browsers**
6. **Automation using Selenium WebDriver Explained**

## Selenium WebDriver Architecture

The knowledge of Selenium WebDriver architecture is required for an in-depth understanding of Web-UI automation using Selenium WebDriver. In order to understand the architecture, we first need to know about the four components that comprise this framework.

**Selenium WebDriver Architecture Diagram**

## Selenium Client Library

The developers of Selenium WebDriver have made it possible to support different programming languages. The Selenium Client Library or the language bindings component of the Selenium WebDriver architecture allows us to write the Selenium automation scripts in the language of our choice – Java, Python, C#, Ruby, JavaScript, etc.

## Browser Drivers

For each of the supported browsers in Selenium, we have a separate browser driver. These drivers take commands from the Selenium scripts and pass them to the respective browsers.

In this way, these browser drivers allow communication between the Selenium script with the different browsers. Also, the communication happens without revealing the internal logic of those browsers.

Some of the browser drivers in Selenium are – FirefoxDriver, ChromeDriver, InternetExplorerDriver, SafariDriver, OperaBrowser, EdgeDriver, and HtmlUnitDriver.

## JSON Wire Protocol over HTTP

The JSON or JavaScript Object Notation is a lightweight data format for storing and transferring data. The JSON wire protocol provides the capability of transferring data between the client and the server over the web.

In the WebDriver architecture, the JSON wire protocol is used for communication between the Selenium scripts and the Browser Drivers.

## Browsers

This component of the WebDriver architecture in Selenium is pretty straightforward. The browsers receive the command and call the respective method to perform the desired automation task. The different browsers supported by Selenium are – Chrome, Firefox, IE, Edge, Safari, Opera, etc.

## Automation using Selenium WebDriver Explained

Time to see the internal working of the automation of web applications in the different browsers using Selenium WebDriver.

Let’s understand this with the help of the below commands to perform a simple automation task-

***WebDriver driver = new FirefoxDriver();***

***driver.get("https://www.google.com ");***

***driver.findElement(By.id("q")).click();***

Now, the script is written in Java. So, the Selenium Client Library component helps us in creating the scripts in the Java language. Using the JSON wire protocol, the Selenium client communicates with the browser driver (Firefox in the case of our example). It will perform the get() operation to launch the Google website and then click on a button.

The Firefox browser driver will then send the commands to the respective browser. On successful completion of the automation tasks, the response is sent back to the automation script.