

Basic Selenium

INDEX

1. Introduction to Selenium
2. Selenium Architecture
3. Selenium WebDriver Architecture
4. Installation of Selenium
5. Launch An Empty Browser
6. WebDriver Methods
7. HTML
8. SearchContext Methods
9. Locators & Xpath
10. Relative Locators
11. WebElement Methods
12. Synchronization
13. Select Class
14. Auto Suggestion
15. Actions Class
16. TakesScreenshot
17. JavascriptExecutor
18. Frames
19. PopUps
20. Handling New Windows
21. Data Driven Testing
22. TestNG
23. POM

DISADVANTAGES OF MANUAL TESTING

- Time Consuming
- Requires more manpower
- Tedious and monotonous job
- It is susceptible to human errors

What is Automation?

Testing a software with the help of any tool is called as automation Testing

Advantages of Automation

- It is faster
- Saves Time
- Reduces Human Effort
- More accurate

Disadvantages of Automation

- Initial investment is High
- Required skilled manpower

DISADVANTAGES OF MANUAL TESTING

- Time Consuming
- Requires more manpower
- Tedious and monotonous job
- It is susceptible to human errors

What is Automation?

Testing a software with the help of any tool is called as automation Testing

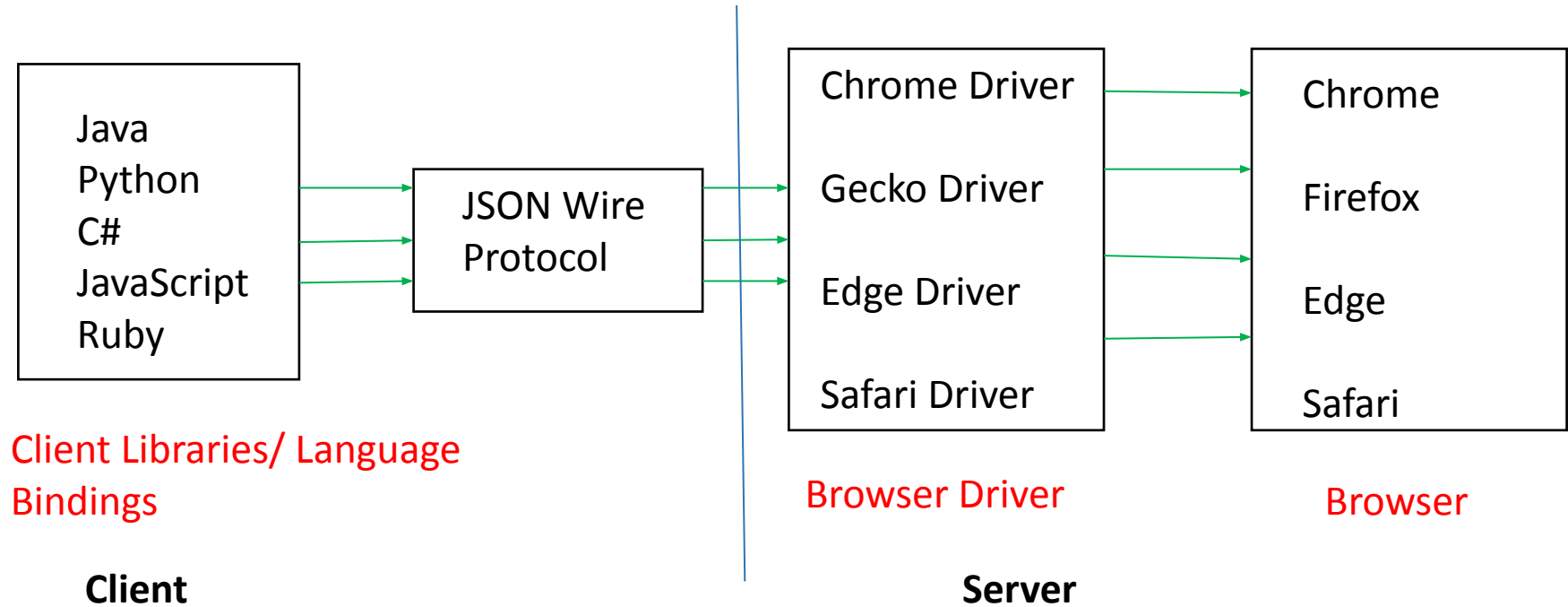
Advantages of Automation

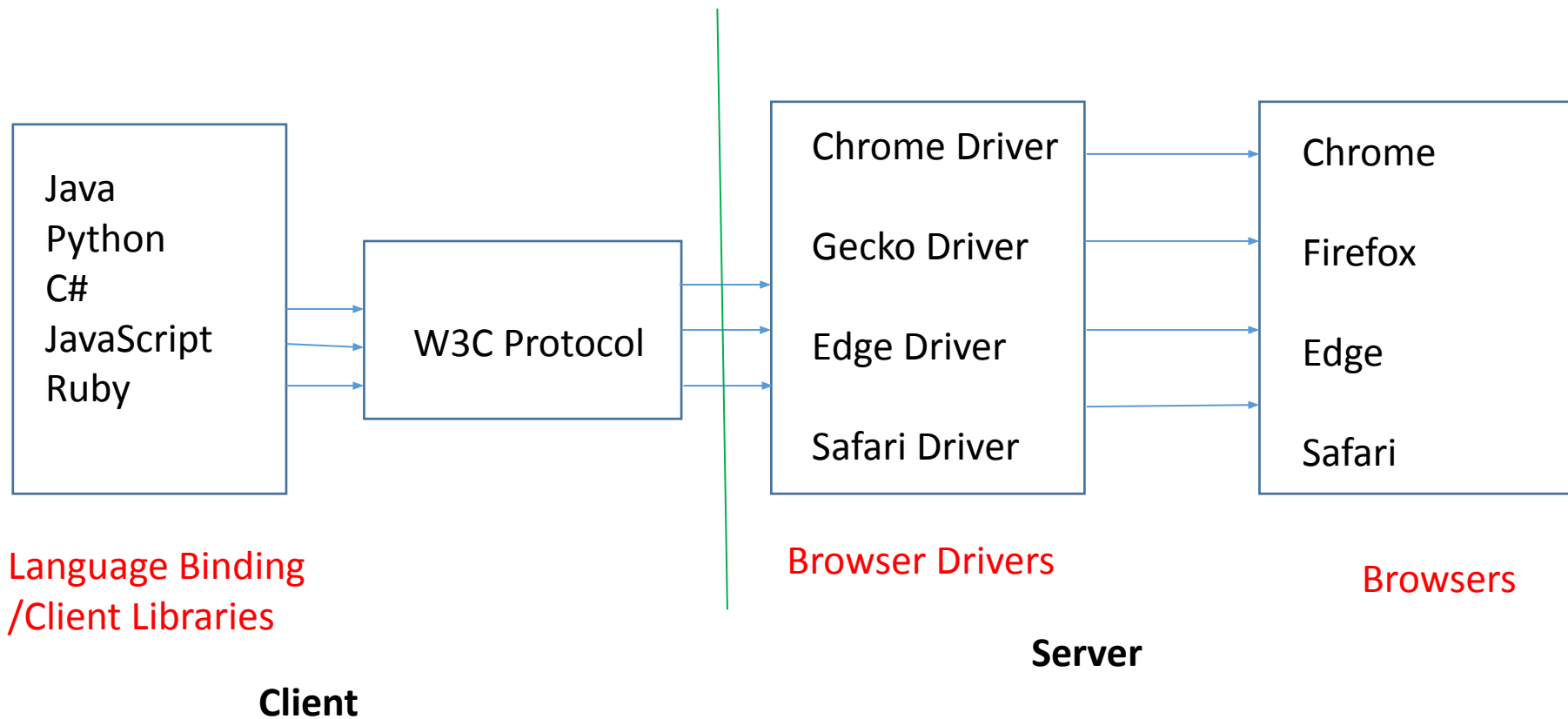
- It is faster
- Saves Time
- Reduces Human Effort
- More accurate

Disadvantages of Automation

- Initial investment is High
- Required skilled manpower

SELENIUM ARCHITECTURE



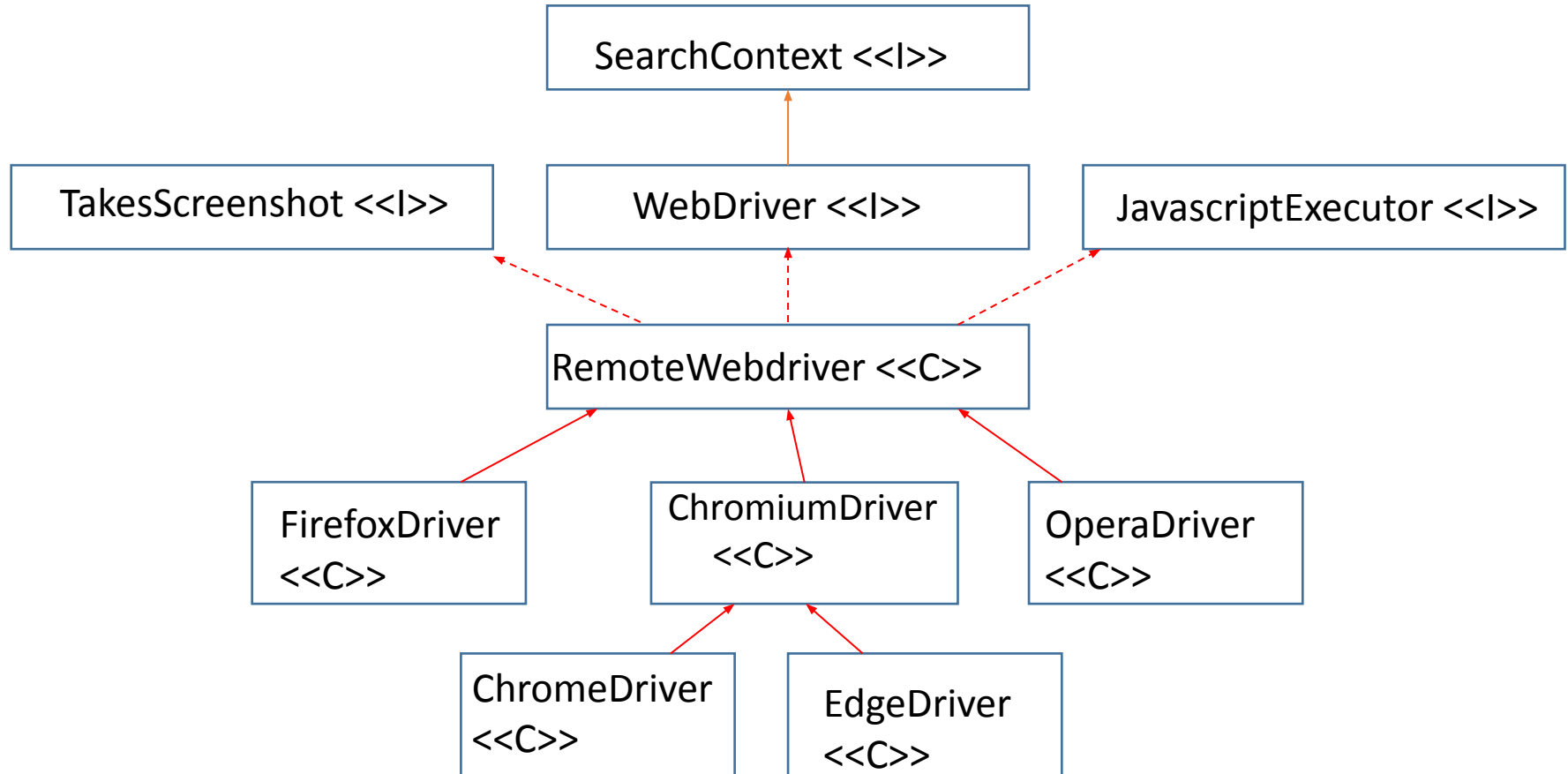


Selenium 4 onwards

- Selenium Architecture follows Client-Server Architecture and depends on request and response.
- On Client side,
 - Selenium supports Language Bindings or Client Libraries.
 - The client libraries are Java, Python, C#, JavaScript, etc. up to 13+ different programming languages.
 - We make use of these client libraries to write automation script on IDE.
 - We need to download Java client library to work on Java-Selenium(Similarly for other languages). Here, we write selenium commands.
 - Once you execute the automation script, the selenium commands are mapped to particular web services at run time and sent as a request to Browser Drivers via JSON Wire Protocol.
- JSON Wire Protocol is a standard through which all the mandatory services are implemented to the Browser Driver.
- This Protocol acts like a translating medium between Language Bindings and Browser Drivers.

- In Server side,
 - Every Browsers have their own dedicated Browser Drivers as Browsers will not expose their services to Selenium
 - Browser Driver receives and understand the Request in the form of http.
 - Browser Driver takes control over respective Browsers and corresponding actions are performed.
- Once after actions are performed response comes back to Browser Driver in the form of http.
- From the Browser Driver, the response is sent across to user's console in a readable format.
 - It can be the status of the Browser, commands that have been executed, status code, etc.
- From Selenium 4 onwards JSON Wire Protocol was replaced by W3C Protocol.
 - It Provides InterWebOperability, how a web application should communicate.
 - Using this, language and driver could communicate more efficiently and provides stability.

SELENIUM WEBDRIVER ARCHITECTURE



TakesScreenshot

It provides mechanism to take screenshot of entire web page

It provides 1 abstract method

1. `getScreenshotAs()`

JavascriptExecutor

It provides mechanism to write a script in browser understandable language (JavaScript)

It provides 2 abstract Methods

1. `executeScript()`
2. `executeAsyncScript()`

Note:

All the implementation of these methods are given in **RemoteWebDriver** class.

Respective Browser specified classes extends **RemoteWebDriver** Class.

SearchContext

It provides a mechanism to search web element in a web page

It provides 2 abstract methods

1. findElement(By by)
2. findElements(By by)

WebDriver

It provides mechanism to perform browser related actions.

It provides 11 abstract methods(direct) and 2 inherited methods.

1. get(String url)
2. getTitle()
3. getCurrentUrl()
4. getPageSource()
5. close()
6. quit()
7. manage()
8. navigate()
9. getWindowHandle()
10. getWindowHandles()
11. switchTo()

INSTALLATION STEPS OF SELENIUM

1. Create a new Project in Eclipse
2. Open Browser and search “selenium dev”.
3. Click on “downloads” in top navigation bar.
4. Scroll down till “Previous Releases” and select the version.
5. Once after it is downloaded, open Eclipse and right click on your Project.
6. Select “Build Path” and then “Configure Build Path”.
7. Click on “Libraries” major tab and click on “class path”.
8. Select the thus enabled “Add External jar file” button in Right hand Navigation.
9. Choose the downloaded selenium jar file and click on “open”.
10. Click on “Apply” and then “Apply and Close”.

Note:

Always select the latest and stable version of selenium while downloading.

TO LAUNCH AN EMPTY BROWSER

ChromeDriver driver= new ChromeDriver();

EdgeDriver driver= new EdgeDriver();

FirefoxDriver driver= new FirefoxDriver(); ...etc.

Upcasting the Browser driver

WebDriver driver= new ChromeDriver();

= new EdgeDriver();

= new FirefoxDriver(); ...etc.

Here,

- WebDriver is a class and driver is a reference variable.
- new is a keyword and chromeDriver() is constructor calling statement.
- Constructor has 3 jobs here
 - Load, Register and Reinitialize the non static members
 - Start the server
 - Launch an empty Browser

Advantages of Upcasting

- We can achieve runtime polymorphism
- We can achieve Cross Browser Testing

1. **get(String url)**

It is used to navigate to an application specified by the URL

URL should be fully qualified path

Return Type is **void**

2. **getTitle()**

It is used to capture the title of the web page

It is used for Validation Purpose.

Return type is **String**.

3. **getCurrentUrl()**

It is used to capture the URL of the web page.

Return type is **String**.

4. **getPageSource()**

It is used to capture the source code (HTML, CSS, JS) of the web page

Return type is **String**.