

# Selenium



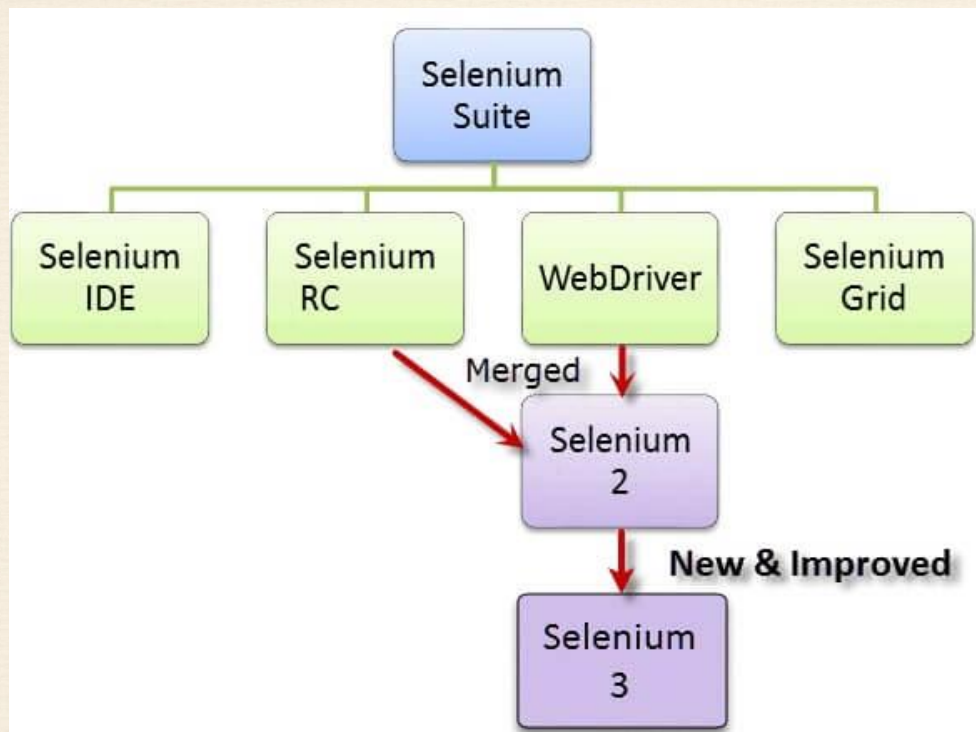
# What is it?

- . Open source web based automation testing tool
- . Can be used to test web applications across different platforms and web browsers
- . Supports multiple programming languages like Java, C#, Ruby, Python etc.
- . We will be sticking to Java



# Components

- . Selenium IDE
- . Selenium RC
- . WebDriver
- . Selenium Grid



# Selenium IDE

- . Integrated Development Environment
- . Easy to use
- . Available as Chrome and Firefox extension
- . Easy to get started and saves time



# Selenium WebDriver

- . It is a third party library which can be downloaded and used
- . Does it do all the automation?
- . It uses the browser automation APIs provided by the browser vendors to control the browser
- . WebDriver interface defines the methods that are required for performing various actions on the browser
- . Implementation has to be provided by specific driver types

- . `AndroidDriver`
- . `ChromeDriver`
- . `FirefoxDriver`
- . `InternetExplorerDriver`
- . `EdgeDriver`
- . `SafariDriver` etc. etc.
- . You can write your own driver as well



- . Using the above APIs, WebDriver creates an instance of the browser
- . So it has complete control over the browser object
- . The actions performed by WebDriver are as if a real user is performing on the target website
- . It is not intrusive
- . So the site you are testing is the same site that may go live



# Selenium Grid

- . Was called Standalone Server earlier
- . As we said that the purpose of automation is to run the tests across multiple envs (Windows, OS, Linux etc.) and across multiple browsers
- . So we need a way to run the tests we are developing using WebDriver in multiple machines
- . This is where Selenium Grid comes to our rescue

- . It is a smart proxy server
- . Routes selenium commands to remote web browser instances
- . Supports running tests in parallel on multiple machines
- . Works in a hub – node model
- . So there s a server acting as hub with multiple nodes registered with it

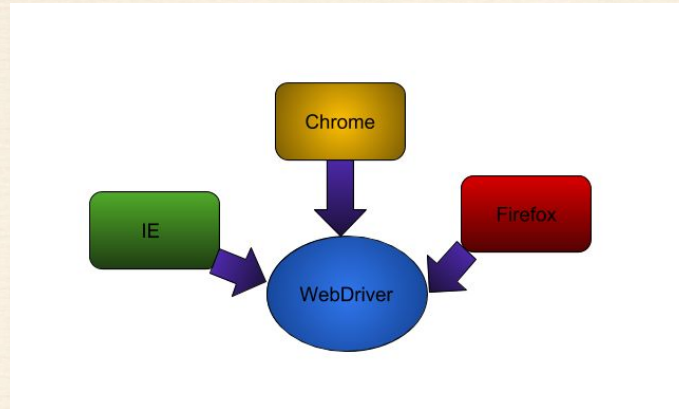


# Selenium Language Bindings

- . Simply put these are the different flavours of Selenium WebDriver to support scripting in different languages
- . Had bindings for Java, Javascript, Python, C#, Ruby, Kotlin etc. to name a few
- . This list is not exhaustive as anyone can develop a new language binding and make it available to us

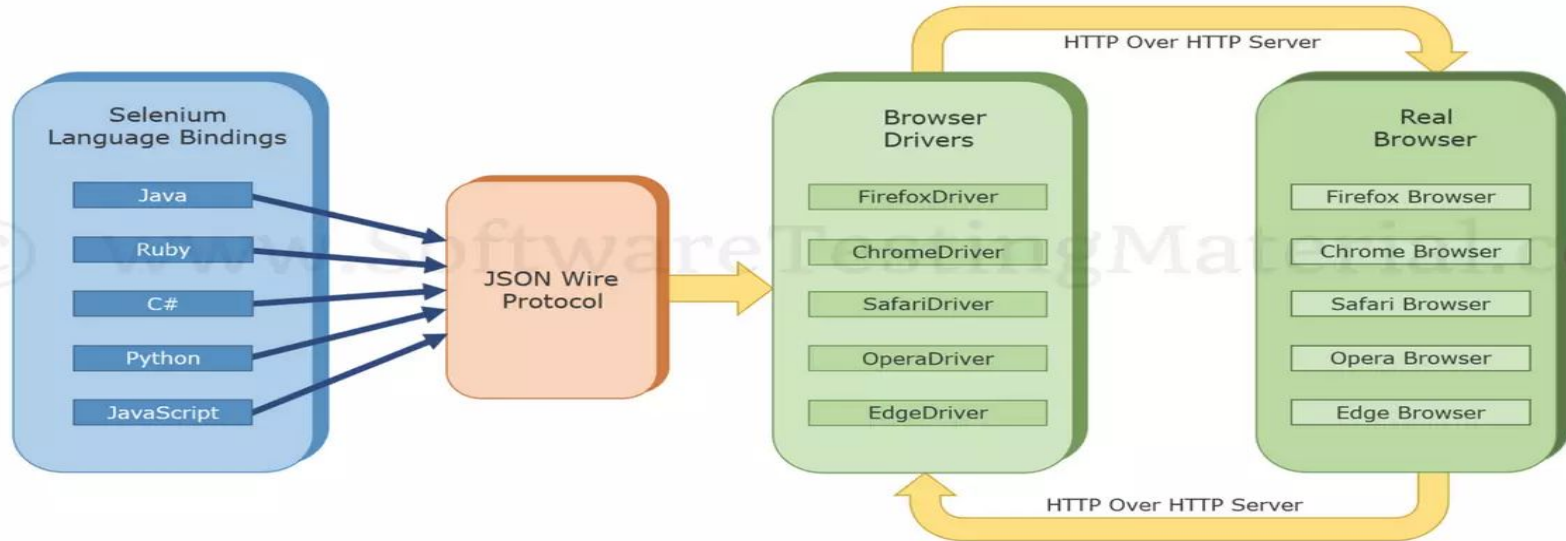
# WebDriver





- . Its main functionality is:
  - . To control the browser
  - . To help us select various HTML elements and perform actions on them

# Selenium WebDriver Architecture





- . WebDriver interface has some abstract methods to interact with the browser
  - . `get(String url)`
  - . `quit()`
  - . `close()`
  - . `getWindowHandle()`, `getWindowHandles()`
  - . `getTitle()` etc.

- . It also has some nested interfaces like Window, Navigation, Timeouts
- . These interfaces contain methods to control the browser and perform various actions



# Common Elements

- . Text Box, Radio Button, Check Box,
- . Select Drop Downs, Multi-select Box,
- . Button, Links, File Upload, Download links

# No so common

- . Tables, Dynamic Tables
- . Pop up windows, Alerts, Dialog boxes



# WebDriver Hands On

- . Download driver exes required for chrome and firefox
- . Open the base project we created yesterday and add two methods
- . One for opening chrome and navigating to google.com
- . Other for opening firefox and navigating to google.com
- . Try `driver.close()` and `driver.quit()` methods and note the difference

# Things you are not supposed to do

- CAPTCHA automation
  - CAPTCHA stands for “*Completely Automated Public Turing test to tell Computers and Humans Apart*”
  - Explicitly designed to prevent automation
  - So DO NOT try
- Workaround
  - Disable CAPTCHA in test env
  - Add hooks to allow tests to bypass CAPTCHA



- . File download automation
  - . Using selenium we can go and click the link to start file download
  - . But the API does not support monitoring the download progress
  - . So we have to use some other crude mechanisms if we are trying to test file download

- . Do not use it to write scripts to automate logging in to your social accounts like gmail, facebook etc. or frequently used websites
- . This is against their policy and your account may get blocked



- . Your tests should not be dependent on one another
- . They should run in any order
- . They should not rely on any other test case to be completed

- . Performance testing using selenium is not advised
- . It can still be used but it is not optimized for the job