Selenium WebDriver

**Selenium-WebDriver was developed to better support dynamic web pages**

**What is WebDriver:**

Web automation framework that allows you to execute your tests against different browsers



**Why Webdriver:**

1. Support dynamic web pages
2. Supplies a well-designed object oriented API
3. Can handle keyboard events
4. Support for modern advanced web-app testing problems
5. Interacts with page elements in more realistic way
6. Supports all major Browsers like Mozilla, Chrome, IE, Safari etc.
7. Supports all major languages like java, C#, Perl, Ruby, Python etc.
8. Supports all major Operating System like Windows, Mac, IOS, Android etc.

|  |  |  |
| --- | --- | --- |
| **Selenium IDE** | **Selenium RC** | **Selenium WebDriver** |
| It only works in Mozilla browser. | It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc. | It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc. |
| It supports Record and playback | It doesn’t supports Record and playback | It doesn’t supports Record and playback |
| Doesn’t required to start server before executing the test script. | Required to start server before executing the test script. | Doesn’t required to start server before executing the test script. |
| It is a GUI Plug-in | It is standalone java program which allow you to run Html test suites. | It actual core API which has binding in a range of languages. |
| Core engine is Javascript based | Core engine is Javascript based | Interacts natively with browser application |
| Very simple to use as it is record & playback. | It is easy and small API | As compared to RC, it is bit complex and large API. |
| It is not object oriented | API’s are less Object oriented | API’s are entirely Object oriented |

**Configure Selenium Webdriver:**

Step 1: Download Selenium Webdriver jar files from [www.seleniumhq.org](http://www.seleniumhq.org) (selenium-server-standalone-3.0.1)

Step 2: Import above jar files in Eclipse java project (Right click on java project -> Build Path -> Configure Build Path -> Libraries -> Add External JARs…

Step 3: Create normal Java Class and write a script

**Sample script:**

System.*setProperty*("webdriver.chrome.driver", *driverPath*+"chromedriver.exe");

// To Launch browser

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

// To navigate to the application

driver.navigate().to("https://enterprise-demo.orangehrmlive.com/auth/login");

// To maximize the window

driver.manage().window().maximize();

// To find the element and perform the action

driver.findElement(By.*xpath*("//input[@type='text']")).sendKeys("Admin");

driver.findElement(By.*id*("txtPassword")).sendKeys("admin");

driver.findElement(By.*className*("button")).sendKeys(Keys.***ENTER***);

// To close the browser

driver.close();