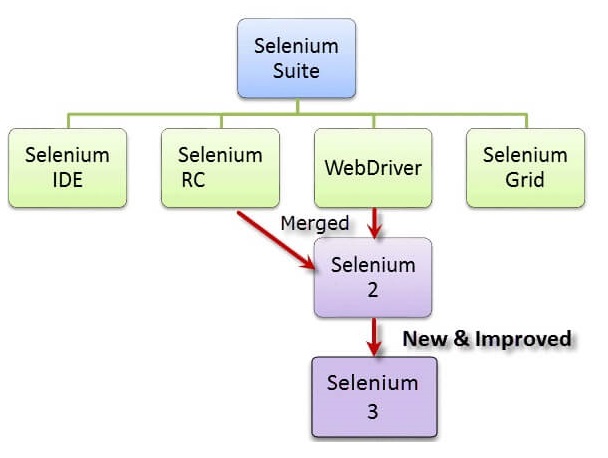
1. **Khái quát:**

* **Khái niệm :**
  + Selenium là công cụ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau.
  + Selenium tập trung vào việc tự động hoá các ứng dụng dựa trên web. Kiểm thử được thực hiện bằng cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium.
  + Selenium không chỉ là một công cụ mà còn là một bộ phần mềm, mỗi bộ đều đáp ứng nhu cầu kiểm tra khác nhau của một tổ chức.
* **Các tính năng nổi bật của Selenium là gì?**

Dưới đây là những tính năng nổi bật mà Selenium giới thiệu.

* + Bạn có thể kết hợp với nhiều ngôn ngữ lập trình như: Java,.Net, Python, Ruby một cách dễ dàng.
  + Selenium sẽ giả cho các thao tác người dùng ở trên web page và các web element.
  + Script được base dựa theo HTML nên rất dễ học và việc ứng dụng cũng sẽ rất nhanh.
  + Bạn có thể tạo ra một bộ test suite có chứa nhiều test case khác.
  + Bạn có thể run các test suite thông qua Selenium IDE hoặc Selenium command line
  + Selenium API sẽ hỗ trợ rất nhiều cho các loại trình duyệt, do vậy mà sẽ rất dễ dàng khi test ứng dụng web với nhiều loại trình duyệt khác nhau.
* **Nó gồm bốn thành phần bên dưới:**
  + Selenium Integrated Development Environment (IDE)
  + Selenium Remote Control (RC) (hiện nay đã gộp với webdriver)
  + WebDriver
  + Selenium Grid



1. **Giới thiệu về Selenium IDE**

Môi trường phát triển tích hợp Selenium (IDE) là framework đơn giản nhất trong bộ Selenium và là phần dễ nhất để học. Nó là một plugin Firefox mà bạn có thể cài đặt dễ dàng như các plugin khác. Tuy nhiên, vì sự đơn giản của nó, nên chỉ sử dụng Selenium IDE làm công cụ tạo mẫu. Nếu bạn muốn tạo thêm các trường hợp kiểm thử nâng cao, bạn sẽ cần phải sử dụng hoặc là Selenium RC hoặc WebDriver.

**Điểm mạnh của Selenium IDE**

* Dẽ dàng cài đặt và sử dụng.
* Không yêu cầu kinh nghiệm lập trình, chỉ cần hiểu về HTML và DOM.
* Có thể xuất ra các kiểm thử có định dạng có thể sử dụng được trong Selenium RC và WebDriver.
* Có module built-in help và báo cáo kết quả kiểm thử.
* Cung cấp sự hỗ trợ cho phần mở rộng.

**Điểm yếu của Selenium IDE**

* Chỉ có sẵn cho FireFox.
* Được thiết kế để chị tạo prototypes của kiểm thử.
* Không hỗ trợ cho hoạt động lặp lại và điều kiện.
* Thực thi kiểm thử chậm hơn Selenium RC và WebDriver.

1. **Giới thiệu về Selenium Remote Control (RC)**

Selenium RC là **framework kiểm thử hàng đầu** của toàn bộ dự án Selenium trong một thời gian dài. Đây là công cụ kiểm thử web tự động đầu tiên **cho phép người dùng sử dụng đa dạng ngôn ngữ lập trình để làm việc với nó**. Kể từ phiên bản 2.25.0, RC có thể hỗ trợ các ngôn ngữ lập trình sau:

* [Java](https://viettuts.vn/java)
* C#
* PHP
* Python
* Perl
* Ruby

**Điểm mạnh của Selenium RC**

* Chạy trên các trình duyệt và các hệ điều hành khác nhau.
* Hỗ trợ các hoạt động lặp lại và điều kiện.
* Có thể hỗ trợ data-driven testing.
* Có API hoàn thiện.
* Hỗ trợ các trình duyệt mới.
* Thực thi nhanh hơn Selenium IDE.

**Điểm yếu của Selenium RC**

* Cài đặt phức tạp hơn Selenium IDE.
* Phải có kiến thức về lập trinh.
* Yêu cầu Selenium RC Server để thực run.
* Sự tương tác trình duyệt ít thực tế hơn.
* Kết quả không phù hợp và sử dụng JavaScript.
* Thời gian thực thi chậm hơn WebDriver.

1. **Giới thiệu về WebDriver**

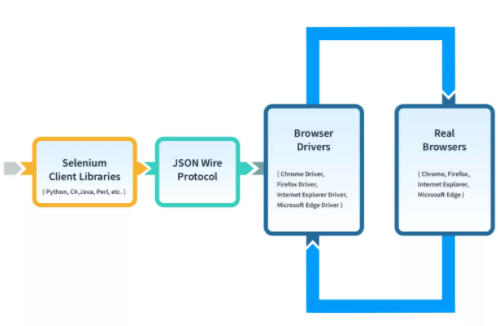
 WebDriver chứng tỏ rằng nó tốt hơn cả Selenium IDE và Selenium RC về nhiều mặt. Nó thực hiện một cách hiện đại và ổn định hơn trong tự động hoá các hành động của trình duyệt. WebDriver, không giống như Selenium RC, không dựa vào JavaScript dành cho tự động hóa. Nó điều khiển trình duyệt bằng cách trực tiếp liên lạc với nó. Các ngôn ngữ được hỗ trợ giống với những ngôn ngữ trong Selenium RC.

Các ngôn ngữ được hỗ trợ giống như ngôn ngữ trong Selenium RC.

* [Java](https://viettuts.vn/java)
* C#
* PHP
* Python
* Perl
* Ruby

**Cấu trúc: Gồm 4 component quan trọng:**

* Selenium Client library
* JSON wire protocol over HTTP
* Browser Drivers
* Browser



\* **Selenium Client Libraries/Language Bindings:** **cung cấp hỗ trợ cho nhiều thư viện như Ruby, Python, Java**

**\* JSON wire protocol over HTTP**

**JSON là từ viết tắt của JavaScript Object Notation. Nó là một tiêu chuẩn mở cung cấp một cơ chế vận chuyển để truyền dữ liệu giữa máy khách và máy chủ trên web. Nó cung cấp hỗ trợ cho các cấu trúc dữ liệu khác nhau như mảng và đối tượng, giúp đọc và ghi dữ liệu từ JSON dễ dàng hơn.**

**JSON phục vụ như một API REST (Chuyển trạng thái đại diện) trao đổi thông tin giữa các máy chủ HTTP.**

**\* Browser Drivers**

**Selenium cung cấp trình điều khiển cụ thể cho từng trình duyệt và không tiết lộ logic bên trong của chức năng trình duyệt, trình điều khiển trình duyệt tương tác với trình duyệt tương ứng bằng cách thiết lập kết nối an toàn. Các trình điều khiển trình duyệt này cũng dành riêng cho ngôn ngữ được sử dụng để tự động hóa trường hợp thử nghiệm như C #, Python, Java, v.v.**

**\* Browser: như đã nói, selenium webdriver hỗ trợ nhiều trình duyệt khác nhau.**

**Điểm mạnh của WebDriver**

* Đây là một trong những công cụ Mã nguồn mở phổ biến nhất và rất dễ bắt đầu để thử nghiệm các ứng dụng dựa trên web. Nó cũng cho phép bạn thực hiện kiểm tra khả năng tương thích của nhiều trình duyệt.
* Hỗ trợ nhiều hệ điều hành như Windows, Mac, Linux, Unix, v.v.
* Nó cung cấp khả năng tương thích với một loạt các ngôn ngữ bao gồm Python, Java, Perl, Ruby, v.v.
* Cung cấp hỗ trợ cho các trình duyệt hiện đại như Chrome, Firefox, Opera, Safari và Internet Explorer.
* Selenium WebDriver hoàn thành việc thực thi các tập lệnh thử nghiệm nhanh hơn khi so sánh với các công cụ khác
* API ngắn gọn hơn (giao diện lập trình ứng dụng) so với Selenium RC
* Nó cũng cung cấp khả năng tương thích với iPhoneDriver, HtmlUnitDriver và AndroidDriver
* Cài đặt đơn giản hơn Selenium RC.
* Không yêu cầu các thành phần khác, chẳng hạn như RC Server.
* Thời gian thực thi nhanh hơn Selenium IDE và RC.

**Điểm yếu của WebDriver**

* Yêu cầu kiến thức lập trình.
* Không sẵn sàng hộ trợ các trình duyệt mới. Cần phải thiết lập driver cho trình duyệt tương ứng.
* Không có cơ chế built-in cho việc ghi lại các thông điệp trong thời gian chạy và sinh ra kết quả kiểm thử.

Các hoạt động của selenium webdriver:

* Các lệnh kiểm tra được chuyển đổi thành một yêu cầu HTTP bằng giao thức JSON.
* Trước khi thực hiện bất kỳ trường hợp thử nghiệm nào, mọi trình duyệt đều có trình điều khiển riêng để khởi tạo máy chủ.
* Sau đó, trình duyệt bắt đầu nhận yêu cầu thông qua trình điều khiển của nó

Ví dụ:

WebDriver driver = new ChromeDriver ();

driver. get (https://www.browserstack.com)

Đoạn code này khởi chạy trình duyệt Chrome sẽ điều hướng đến trang web BrowserStack.

Mọi dòng code / tập lệnh sẽ được chuyển đổi thành một URL thông qua Giao thức JSON Wire qua HTTP. Sau đó, URL này được chuyển đến trình điều khiển trình duyệt (ChromeDriver). Tại thời điểm này, thư viện client (Python) dịch code sang định dạng JSON và tương tác với ChromeDriver.

Kết quả:

https://localhost:8080/{"url":https://www.browserstack.com"}

Loại request phổ biến: GET – POST

Cách sử dụng:

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

public class LoginAutomation {

@Test

public void login() {

System.setProperty("webdriver.chrome.driver", "path of driver");

WebDriver driver=new ChromeDriver();

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

driver.get("https://www.browserstack.com/users/sign\_in");

WebElement username=driver.findElement(By.id("user\_email\_Login"));

WebElement password=driver.findElement(By.id("user\_password"));

WebElement login=driver.findElement(By.name("commit"));

username.sendKeys("abc@gmail.com");

password.sendKeys("your\_password");

login.click();

String actualUrl="https://live.browserstack.com/dashboard";

String expectedUrl= driver.getCurrentUrl();

Assert.assertEquals(expectedUrl,actualUrl);

}

}

Các công việc được thực hiện:

* Tạo một phiên bản Selenium WebDriver
* Định cấu hình trình duyệt nếu được yêu cầu
* Điều hướng đến trang web được yêu cầu và tìm phần tử web có liên quan
* Thực hiện hành động trên phần tử web
* Xác minh và xác thực hành động

1. **Giới thiệu Selenium Grid**

Selenium Grid là một công cụ **được sử dụng cùng với Selenium RC để chạy kiểm thử song song** trên các máy khác nhau và các trình duyệt khác nhau cùng một lúc. Thực thi song song có nghĩa là chạy nhiều kiểm thử cùng một lúc.

**Tính năng, đặc điểm**:

* Cho phép **chạy đồng thời các kiểm thử** trong nhiều trình duyệt và môi trường.
* **Tiết kiệm rất nhiều thời gian**.
* Sử dụng khái niệm **hub-and-nodes**. Hub hoạt động như một nguồn trung tâm của các lệnh Selenium cho mỗi nút được kết nối với nó.

1. **Lưu ý về Trình duyệt và hỗ trợ Môi trường**

Do sự khác biệt về kiến trúc, Selenium IDE, Selenium RC và WebDriver hỗ trợ các bộ trình duyệt và môi trường hoạt động khác nhau.

* Selenium IDE
  + Hỗ trợ trình duyệt: Mozilla Firefox
  + Hệ điều hành: Windows, Mac OS X, Linux
* WebDriver
  + Hỗ trợ trình duyệt:
    - Internet Explorer phiên bản 6 đến 11, cả 32 và 64-bit.
    - Microsoft Edge phiên bản 12.10240 trở lên (một phần hỗ trợ một số chức năng đang được phát triển).
    - Firefox 3.0 trở lên.
    - Google Chrome 12.0. trở lên.
    - Opera 11.5 trở lên.
    - Android - 2,3 trở lên dành cho điện thoại và máy tính bảng (thiết bị & bộ mô phỏng).
    - IOS 3+ cho điện thoại (thiết bị & người giả lập) và 3,2+ dành cho máy tính bảng (thiết bị và trình giả lập).
    - HtmlUnit 2.9 trở lên
  + Hệ điều hành: Tất cả các hệ điều hành mà các trình duyệt ở trên có thể chạy.

# 7. **Selenium WebDriver**

Cấu trúc gồm 4 component quan trọng:

- Selenium Client library

- JSON wire protocol over HTTP

- Browser Drivers

- Browser