WebDriver Installation and Initialization for Different Browsers

Here we will see how to install driver for different browsers and use it for creating WebDriver session.

WE'LL COVER THE FOLLOWING

Before beginning

Most WebDriver implementations require an additional executable for Selenium to communicate with the browser.

We can manually add the executable before starting WebDriver session, but this can make the framework less portable as the executables are platform-dependent and need to be in the same place on every machine or include the executables with the framework.

By adding the driver to the system classpath, selenium will be able to locate the executables without bundling them with the framework.

Downloading & installation of WebDriver

Download WebDriver

For downloading driver executables of different browsers:

Chrome: https://sites.google.com/a/chromium.org/chromedriver/downloads

Firefox: https://github.com/mozilla/geckodriver

Edge: https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/

Internet Explorer: https://selenium-release.storage.googleapis.com/index.html

Opera: https://github.com/operasoftware/operachromiumdriver/releases

---

Adding WebDriver to classpath

Windows

```
setx /m path "%path%;C:\selenium-drivers\bin"
```

Linux/Macintosh

```
sudo mkdir -p $HOME/selenium-drivers

export PATH=$PATH:$HOME/selenium-drivers >> ~/.profile
```

Setting driver path and creating WebDriver instance

The path of the driver can be set explicitly, and WebDriver instance of different browsers can be created using the following code.

Chrome driver

```
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeDriverService;
import org.openqa.selenium.WebDriver;

// `executable_path` can be excluded if the driver is in the classpath already
```

```java
System.setProperty(ChromeDriverService.CHROME_DRIVER_EXE_PROPERTY, "/path/to/chromedriver");


WebDriver driver = new ChromeDriver();
```

Firefox driver

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.Firefox.FirefoxDriver;

// `executable_path` can be excluded if the driver is in the classpath already
System.setProperty(GeckoDriverService.GECKO_DRIVER_EXE_PROPERTY, "/path/to/geckodriver");

WebDriver driver = new FirefoxDriver();
```

Edge driver

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.edge.EdgeDriverService;

System.setProperty(EdgeDriverService.EDGE_DRIVER_EXE_PROPERTY, "/path/to/MicrosoftWebDriver.exe")

WebDriver driver = new EdgeDriver();
```

Internet Explorer driver

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.ie.InternetExplorerDriverService;

System.setProperty(InternetExplorerDriverService.IE_DRIVER_EXE_PROPERTY, "/path/to/IEDriver.exe");

WebDriver driver = new InternetExplorerDriver();
```

Opera driver

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.opera.OperaDriver;
import org.openqa.selenium.opera.OperaDriverService;
```

```
System.setProperty(OperaDriverService.OPERA_DRIVER_EXE_PROPERTY, "/path/to/operadriver");

WebDriver driver = new OperaDriver();
```

Safari driver

Unlike the other browsers, for Safari we do not have any executable. Rather, enable the automation control on Safari. For enabling, follow the steps below:

For High Sierra and later

Run `safaridriver --enable` once.

Sierra and earlier

If not done already, make the Develop menu available.

Open Safari Browser → ***Preferences***, and on the Advanced tab, select ***"Show Develop menu in menu bar"***.

Navigate to ***Develop → Allow Remote Automation***.

Authorize SafariDriver to launch the XPC service that hosts the local webserver. To permit this, manually run ***/usr/bin/safaridriver*** once and follow the authentication prompt.

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.safari.SafariDriver;

WebDriver driver = new SafariDriver();
```

Safari is no longer available for Windows, hence making it a poor choice for an automation platform.

After installing WebDriver on a machine, the next step is learning the configuration of a specific instance of the browser. See the next lesson to learn how it’s done.