

6

INSTALL SELENIUM WEBDRIVER & FEW BABY STEPS



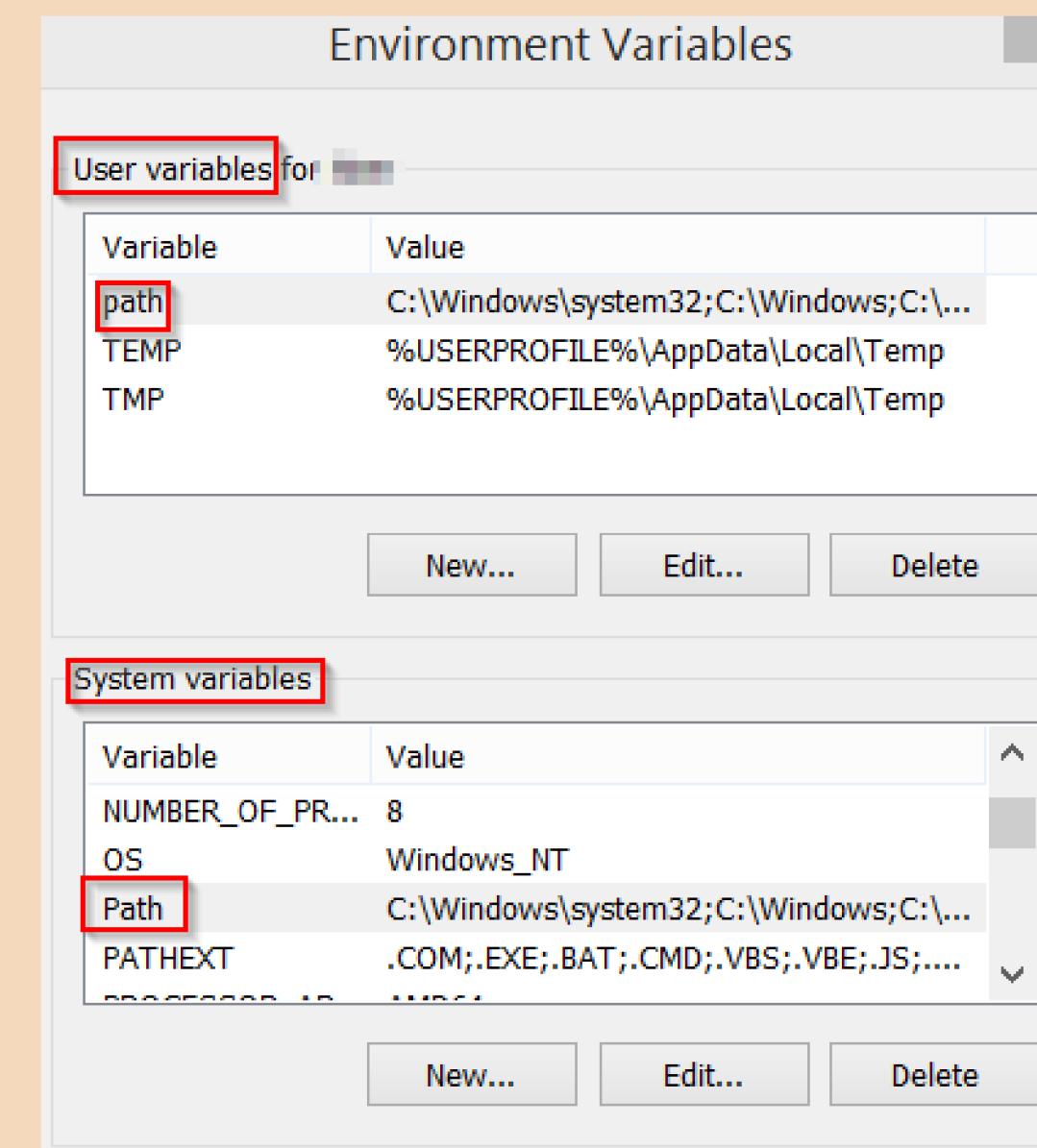
WHAT TO EXPECT



- 1 Creating a project with npm init
- 2 Install Selenium Webdriver

PATH

The drivers for Chrome, Firefox, and Microsoft's IE and Edge web browsers are all standalone executables that should be placed on your system PATH





npm init

Stands for **Node Package Manager**. It is the default package manager for Node.js, used to manage and install packages (libraries, frameworks, tools) for Node.js projects.

The **npm init** command will prompt you with a series of questions.

After you answer all the questions, **npm init** will summarize your responses and ask if you want to proceed. If you're satisfied with the information provided, you can confirm, and it will generate the **package.json** file in the current directory.



npm install selenium.webdriver

Stands for **Node Package Manager**. It is the default package manager for Node.js, used to manage and install packages (libraries, frameworks, tools) for Node.js projects.

This is the command used to install packages. When you run npm **install**, npm looks in the project's **package.json** file for dependencies and installs them.

This is the name of the package being installed. **selenium.webdriver** is the official **JavaScript WebDriver** implementation provided by the Selenium project. It allows you to automate web browsers for testing purposes.

 index.js

This keyword is used to declare a constant variable named selenium. Once a value is assigned to selenium, it cannot be reassigned.

This is the variable name chosen by the developer. You could use any valid variable name here.

This part of the line is using the require function, which is a built-in Node.js function used to import external modules or files. In this case, it is importing the selenium-webdriver module.

```
const selenium = require("selenium-webdriver");
```

```
let driver = await new selenium.Builder().forBrowser(selenium.Browser.FIREFOX).build();
```



```
const selenium = require("selenium-webdriver");

let driver = await new selenium.Builder().forBrowser(selenium.Browser.FIREFOX).build();
```

This declares a variable named driver using the let keyword. The driver variable will hold the instance of the WebDriver.

The await keyword is used to wait for the asynchronous creation of the WebDriver instance. selenium.Builder() is a class provided by the Selenium WebDriver module that is used to configure and build a WebDriver instance.

This part of the line configures the WebDriver to work with the Firefox browser. The forBrowser() method is used to specify the browser type. selenium.Browser.FIREFOX is an enumeration constant representing the Firefox browser in the Selenium WebDriver module.

This method is called to finalize the configuration and build the WebDriver instance. It returns a promise, and using await ensures that the script waits until the WebDriver is fully initialized before proceeding.

</> index.js

The await keyword is used within an asynchronous function to wait for the completion of a promise. It ensures that the script doesn't proceed to the next line until the asynchronous operation (in this case, navigating to the URL) is completed.

The driver variable holds an instance of the Selenium WebDriver. This instance is created earlier using the

This is the `get` method provided by the Selenium WebDriver. It instructs the browser to navigate to the specified URL, in this case, "<https://bookstore.qacurry.com>".

```
await driver.get("https://bookstore.qacurry.com");
```

```
driver.quit()
```



```
await driver.get("https://bookstore.qacurry.com");
```

driver.quit()

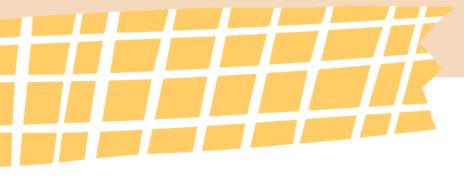
The `driver` variable holds an instance of the Selenium WebDriver. This instance is created earlier using the

This method is called on the `driver` instance to terminate the WebDriver session. It closes all browser windows or tabs associated with that WebDriver instance and releases any resources used by the browser driver.



or





“

See you in Next Video

