PLAYWRIGHT AUTOMATION

1. Introduction to Playwright and Java

Playwright is a cross-browser automation framework. With Playwright for Java, you can write reliable end-to-end tests across multiple browsers like Chrome, Firefox, and Safari.

2. Prerequisites

- **Java Development Kit (JDK)** version 8 or higher.
- **Maven** for dependency management.
- An IDE like **IntelliJ IDEA** or **Eclipse**.

3. Setting Up Playwright in a Java Project

Step 1: Create a New Maven Project

1. Open your IDE and create a new Maven project.
2. Define your pom.xml with the necessary dependencies.

Step 2: Add Playwright Dependency to pom.xml

In your pom.xml, add the Playwright dependency:

```
<dependencies>
  <dependency>
    <groupId>com.microsoft.playwright</groupId>
    <artifactId>playwright</artifactId>
    <version>1.48.0</version>
  </dependency>
</dependencies>
```

P.S. It must be noted that at the time of document creation latest version for Playwright is **1.48.0** respectively. Further versions can be altered from here accordingly.

4. Basic Playwright Structure

Core Playwright Components in Java:

1. **Playwright**: Main class to create and launch browsers.
2. **Browser**: Represents the browser instance.

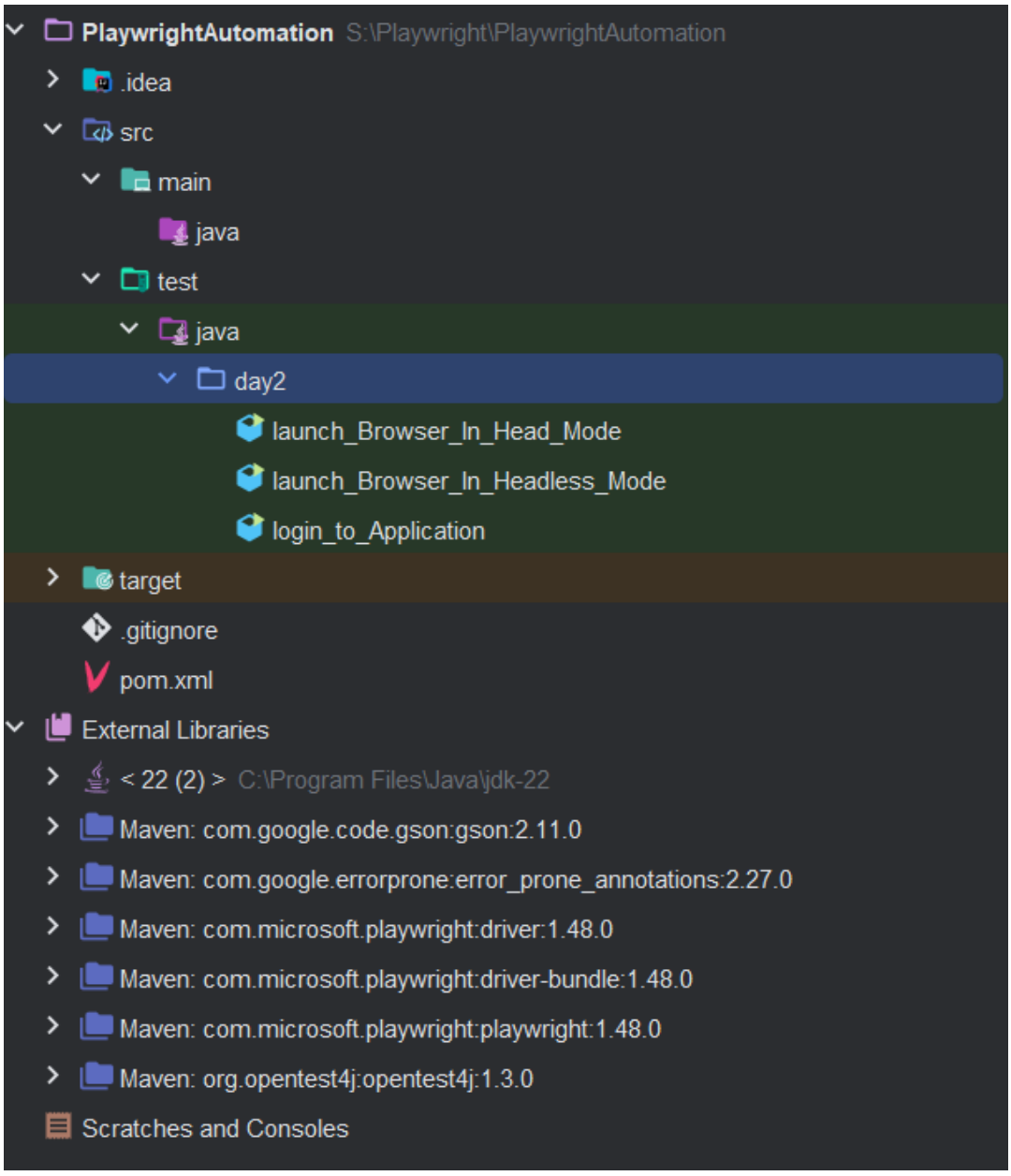
3. **BrowserContext**: Isolated environments for each test.
4. **Page**: Represents a tab or a popup

5. Writing Your First Test

Step 1: Create a Package firstly

Create a package name of your choice respectively under src/test/java directory as shown in the figure below

The Snippet for the Folder Structure is shared below : -



Step 2: Now, Create the First ever test script

Create a test class named **launch_Browser_In_Headless_Mode.java** in the src/test/java directory.

The Snippet for the code is shared below : -

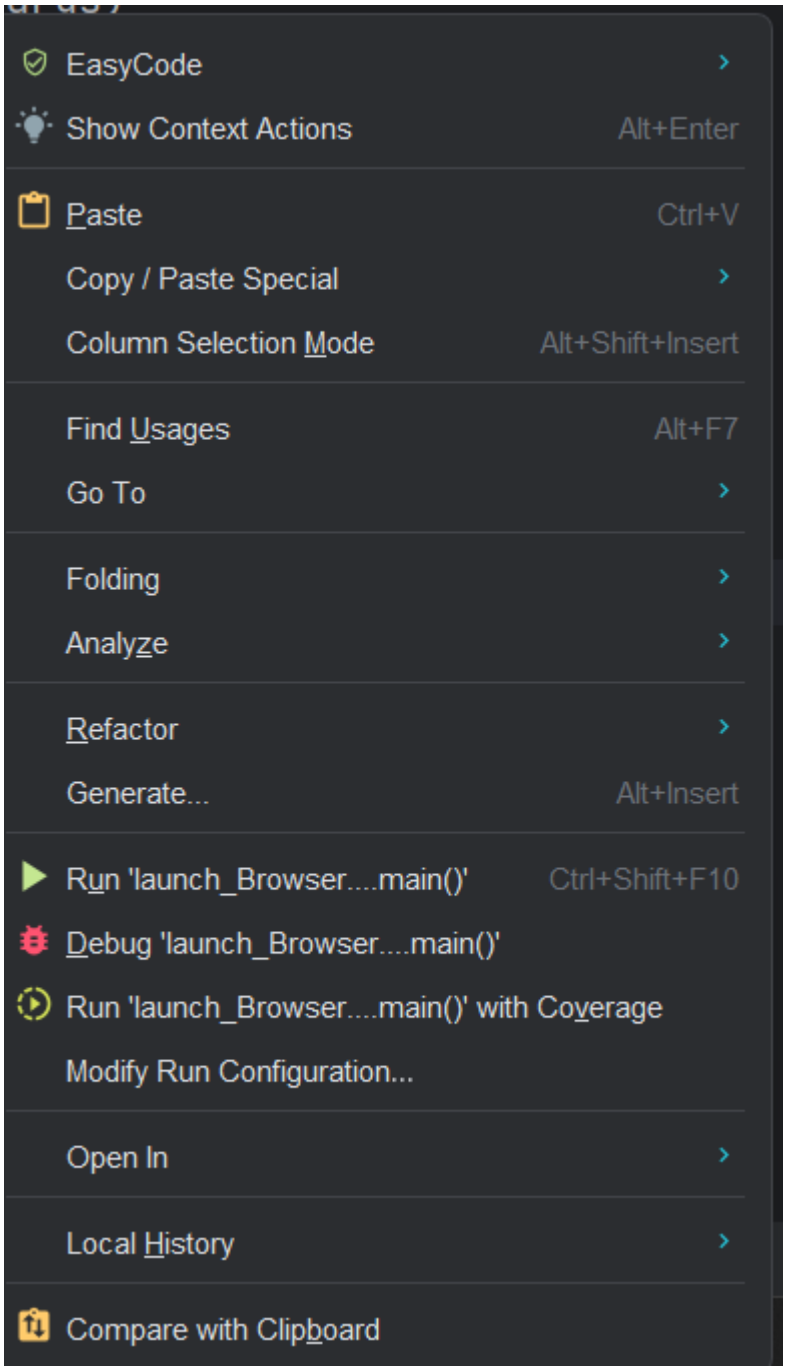
```
package day2;

import com.microsoft.playwright.Browser;
import com.microsoft.playwright.Page;
import com.microsoft.playwright.Playwright;

public class launch_Browser_In_Headless_Mode
{
    public static void main(String[] args)
    {
       Playwright pw = Playwright.create();
       Browser brw = pw.chromium().launch();
        Page pg = brw.newPage();
        pg.navigate( url: "https://www.youtube.com/");
        brw.close();
        pw.close();
    }
}
```

Step 3: Next, Execute the First ever test script

Now execute your automation code successful and observe the screen though automation will not be shown or happen in UI like Selenium



Step 4: Next, Execute the First ever test script in Head Mode

Now to execute your script and make them appear them on GUI just setHeadless as False and watch out for the changes on screen to load respectively.

```
package day2;

import com.microsoft.playwright.Browser;
import com.microsoft.playwright.BrowserType;
import com.microsoft.playwright.Page;
import com.microsoft.playwright.Playwright;

public class launch_Browser_In_Head_Mode
{
    public static void main(String[] args)
    {
        Playwright pw = Playwright.create();
        Browser brw = pw.chromium().launch(
                new BrowserType.LaunchOptions().setHeadless(false)
        );
        Page pg = brw.newPage();
        pg.navigate( url: "https://www.youtube.com/");
        brw.close();
        pw.close();
    }
}
```

Once the browser headless option is set false, the automation window will open up just like Selenium automation and you will be able to view results on GUI as per your comfort.