**Playwright Automation Tool**

**1)Playwright Support on Cross-Browser, Cross-Platform and Cross-Language**

**2)It has auto wait mechanism system (No need to write any wait methods how we write in selenium which avoids flaky Tests)**

**PlayWright Setup:**

1. **Install Eclipse**
2. **Create a Maven Project**
3. **Now Add below dependencies in POM.xml**

<dependency>

<groupId>com.microsoft.playwright</groupId>

<artifactId>playwright</artifactId>

<version>1.28.0</version>

</dependency>

Initial Script with Playwright:

Graphical user interface, text, application

Description automatically generated

**PlayWright playwright=Playwright.create()--->Will start the playwright server**

playwright.close(); 🡪 Close the server

**How to launch the chromium Browser in playwright?**

Browser browser = playwright.chromium().launch(**new** BrowserType.LaunchOptions().setHeadless(**false**));

**How to launch the chromium Browser in playwright?**

**Need to add SetChannel method and need to pass the Browser name**

Browser browser = playwright.chromium().launch(**new** BrowserType.LaunchOptions().setHeadless(**false**).setChannel("chrome"));

**How to launch the safari Browser in playwright?**

Browser browser = playwright.firefox().launch(**new** BrowserType.LaunchOptions().setHeadless(**false**));

**How to launch the safari Browser in playwright?**

Browser browser = playwright.webkit().launch(**new** BrowserType.LaunchOptions().setHeadless(**false**));

**How to get title of the page?**

page.title()

**How to get the html content ?**

page.content()

**How to check element is enabled?**

page.isEnabled("//a[text()='Sign in securely']");



**How to check element is disabled?**

page.isDisabled("//a[text()='Sign in securely']");

Graphical user interface, text, application

Description automatically generated

**How to check element is displayed?**

page.isVisible("//a[text()='Sign in securely']");

Graphical user interface, text, application

Description automatically generated

**Locaters in Playwright:**

**In locators we can pass xpath, id,css,name**

For entering text we can either use fill or type method

1.page.locator("#twotabsearchtextbox").fill("hello world");

2.page.locator("#twotabsearchtextbox").type("hello world");

3.page.getByLabel(selector);

**Return multiple elements similar to find elements in selenium**

Locator multipleElements=page.locator(selector);

List<String> list=multipleElements.allTextContents();

**for**(String listOfElements : list)

System.***out***.println(listOfElements);

**CheckBox and Radio Buttons in PlayWright**

page.locator(selector).check();

**To check whether the checkbox or radio button is enabled or not**

page.locator(selector).isChecked;

**Dropdown Handles in PlayWright**

Select dropdown By Value

page.locator(selector).selectOption(value);

Select dropdown By Label

page.getByLabel(selector).selectOption(labelName);

Select Multiple dropdown Values

dropdownValues values should be in list of array

page.locator(selector).selectOption(dropdownValues);

**Double click :**

page.locator(selector).dblclick();

**Generic Click;**

page.locator(selector).click();

**Right click in Mouse**

page.locator(selector).click(**new** Locator.ClickOptions().setButton(MouseButton.***RIGHT***));

**Mouse hover on element**

page.locator(selector).hover();

**Force click on element**

page.locator(selector).click(**new** Locator.ClickOptions().setForce(**true**));

**Upload a file:**

page.locator(selector).setInputFiles(Paths.*get*(fileName));

**Remove all the upload files**

page.locator (selector).setInputFiles(**new** Path[0]);

**Drag and Drop:**

page.locator(source).dragTo(page.locator(Destination));

**Some of the AssertionMethods**

A picture containing table

Description automatically generated

**alert(), confirm(), prompt() dialogs**

By default, dialogs are auto-dismissed by Playwright, so you don't have to handle them. However, you can register a dialog handler before the action that triggers the dialog to either Dialog.accept([promptText]) or Dialog.dismiss() it.

page.onDialog(dialog -> dialog.accept());

**Handling Frames :**

Selector is frameid or frameName

Selector2 is the element which you want click inside the frame

Graphical user interface, text, email

Description automatically generated