# Playwright Typescript Automation framework

This is a test automation framework around [**Playwright**](https://playwright.dev/) - for end-to-end testing. This is Playwright Automation framework has all wrapper actions methods and assertions method along with logger. This is plug and play Playwright Test Automation framework in Typescript.

**Overview**

I have created playwright automation framework using typescript. In this framework user can use the Playwright automation framework with very less knowledge about playwright and typescript and start automation. This Framework has inbuild wrapper Actions methods, Logger, Assertions and Configuration. Locator writing style for this framework different. We write the framework locator in the form of Locator and control description. It is written in form of Object of ‘WebControl’ Object.

**Project Setup:**

1. Create a project folder at C drive and open the same folder using visual studio code.
2. Open Terminal >> New Terminal
3. Type Command ‘npm init playwright’ and hit enter.
4. Enter all information in terminal asked in terminal and press enter such as language as Typescript going to use in this project, Test case folder folder.
5. Project structure will look like below.

A screenshot of a computer program

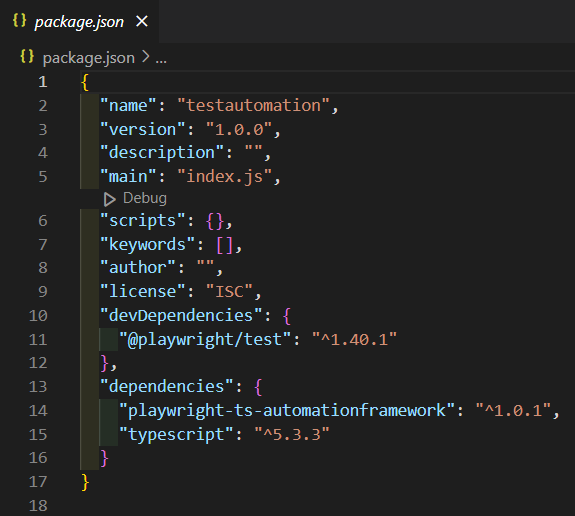
Description automatically generated

1. Again Open terminal. In terminal type below command and hit enter button.

‘**npm install playwright-ts-automationframework’**.

It will download the playwright typescript framework library and save in node modules folder. Also you will see entry is added in package.json file.

1. Install other necessary libraries using below commands in same way.
2. **npm install typescript**
3. After installation all the libraries name with version will get added in package.json file.

****

**Page folder:**

1. Create a folder with name ‘pages’ in the project level.
2. Create a file with name MethodBase.pom.ts and class name as MethodBase in page folder which extends the Assertion class from framework library.
3. Add constructor with parameter as page and pass this page instance to the parent class.

A screen shot of a computer program

Description automatically generated

1. Now you can create each page class which extends MethodBase Class. This means methodBase class will be parent of each page class.
2. Create LoginPage.pom.ts class which extends methodBase class. Add constructor to it with page object.

A screen shot of a computer

Description automatically generated

1. Let us add some controls for login page and method to it. In this framework we write in the format of Object of WebControl which accepts first parameter as playwright locator and second parameter as control description. Let us try to automate one basic test case for OrangeHRM and try to create objects and method for it.
2. URL: <https://opensource-demo.orangehrmlive.com/web/index.php/auth/login>

usernameTxtbx = new WebControl(this.page.locator("xpath=//input[@name='username']"), "Username textbox");

    passwordTxtbx = new WebControl(this.page.locator("xpath=//input[@name='password']"), "Password textbox");

    loginBtn = new WebControl(this.page.locator("xpath=//button[@type='submit']"), "Login button");

    dashboardBtn = new WebControl(this.page.locator("xpath=//span[text()='Dashboard']"), "Dashboard tab button");

1. Now Let us try to add two methods which does doLogin and verifyLoginSuccessful page. As Assertion and Actions method extends method base you get access of all wrapper action mthods and assertion methods which are present in framework. All framework methods accept WebControl object as parameter.

async doLogin(username: string, password: string)

    {

        await this.type(this.usernameTxtbx, username);

        await this.type(this.passwordTxtbx, password);

        await this.click(this.loginBtn);

    }

    async verifyLoginSuccessful()

    {

        await this.verifyIsDisplayed(this.dashboardBtn);

    }

1. Code will look like below.

A screen shot of a computer program

Description automatically generated

1. Let us create test file with name ‘login.spec.ts’ test cases in tests folder. We will create beforeEach and afterEach in which we will initialize browser, SetTCID and Execution Completed logging method.
2. We will Add Test cases and call the method from page objects and will send parameters from test cases.
3. import { test } from '@playwright/test';
4. import { LoginPage } from '../pages/loginPage.pom';
5. import { BasePage } from 'playwright-ts-automationframework';
6. let loginPage: LoginPage;
7. test.beforeEach( async ({ page, baseURL }, testinfo) => {
8. loginPage = new LoginPage(page);
9. BasePage.setTestCaseID(testinfo);
10. await loginPage.initializeBrowser(baseURL);
11. })
12. test('Verify login with valid credentials', async ({ page }) => {
13. await loginPage.doLogin("Admin", "admin123");
14. await loginPage.verifyLoginSuccessful();
15. });
16. test.afterEach( async ({ page, baseURL }, testinfo) => {
17. BasePage.executionCompleted(testinfo);
18. })

12. Last Step we need to perform, we need to add two json files at project level

a. ExecutionSettings.json

Execution Settings will contain all the information related to the execution of application such as environment, emailRecivers, senderEmail, senderPassword.

{

    "environment": "#{environment}#",

    "emailRecievers": "#{emailRecievers}#",

    "senderEmail": "#{senderEmail}#",

    "senderPassword": "#{senderPassword}#"

}

b. AppConfigurations.json

AppConfigurations.json file will contain all the configurations related to the QA, PROD or Staging. JSON file will look like below.

{

"QA":

    {

        "URL": "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login",

        "DB\_Name": "QA\_Database\_Name",

        "DB\_URL": "QA\_Database\_URL",

        "Username": "QA\_Database\_Username",

        "Password": "QA\_Database\_Password",

        "API\_BaseURL": "https://gorest.co.in/public/v2",

        "Authorization": "Bearer "

    },

"STAGE":

    {

        "URL": "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login",

        "DB\_Name": "STAGE\_Database\_Name",

        "DB\_URL": "STAGE\_Database\_URL",

        "DB\_Username": "STAGE\_Database\_Username",

        "DB\_Password": "STAGE\_Database\_Password",

        "API\_BaseURL": "https://gorest.co.in/public/v2",

        "API\_Authorization": "Bearer "

    },

"PROD":

    {

        "URL": "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login",

        "DB\_Name": "PROD\_Database\_Name",

        "DB\_URL": "PROD\_Database\_URL",

        "DB\_Username": "PROD\_Database\_Username",

        "DB\_Password": "PROD\_Database\_Password",

        "API\_BaseURL": "https://gorest.co.in/public/v2",

        "API\_Authorization": "Bearer "

    }

}

13. Open to playwright.config.ts and navigate to use section. Set BaseURL and add headless flag as false in file.

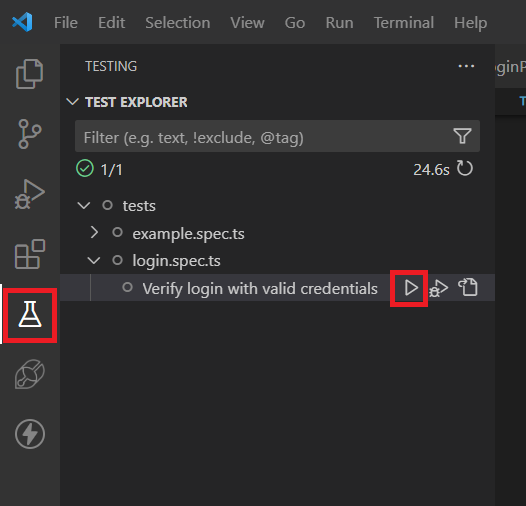
A screen shot of a computer

Description automatically generated

14. Make sure you have installed below extension in your visual Studio code.

<https://marketplace.visualstudio.com/items?itemName=ms-playwright.playwright>

15. Navigate to Testing Tab in Left side tree. Expand login.spec.ts, select run button in front of test case.



16. This will execute the test case and result will be appeared there.

Methods Available in Actions and Assertions Class within framework.

|  |  |
| --- | --- |
| **Actions Class Methods** | |
| **MethodName** | **Description** |
| click(control: WebControl) | /\*\*  \* Click on an element.  \*   \* @param WebControl Control on which user want to click.  \*   \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* click(loginBtn);  \*/ |
| mouseHover(control: WebControl) | /\*\*  \* Mouse hover on an element.  \*   \* @param WebControl Control on which user want to mouse hover.  \*   \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* mouseHover(loginBtn);  \*/ |
| focusAndPressKeyboardEvent(control: WebControl, action: string) | /\*\*  \* Focus on element and press keyboard button.  \*   \* @param WebControl Control on which user want to focus and perform the keyboard action.  \* @param action Keyboard action which needs to perform.  \*   \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* focusAndPressSpace(loginBtn, "Enter");  \*/ |
| type(control: WebControl, textTobeEntered: string) | /\*\*  \* Enter value in textbox control.  \*   \* @param WebControl Textbox control on which user want to enter value.  \* @param textTobeEntered Value user want to enter in textbox.  \* Example:   \*   \* usernameTxtbx = new WebControl(this.page.locator("#username"), 'Username textbox');  \*   \* type(usernameTxtbx);  \*/ |
| pressKeyboardEvent(control: WebControl, key: string) | /\*\*  \* Press Keyboard event.  \*   \* @param WebControl Control on which user want perform the keyboard action.  \* @param key Keyboard action which needs to perform.  \*   \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* pressKeyboardEvent(loginBtn, "Enter");  \*/ |
| selectFromDropdownByValue(control: WebControl, optionTobeSelcted: string) | /\*\*  \* Select dropdown element by value.  \*   \* @param WebControl Dropdown Control of which value need to be selected.  \* @param optionTobeSelcted Dropdown value which needs to be selected.  \*   \* Example:   \*   \* cityDrpdwn = new WebControl(this.page.locator('#city'), 'City dropdown');  \*   \* selectFromDropdownByValue(cityDrpdwn, "Sydney");  \*/ |
| selectDropDownByText(control: WebControl, optionTobeSelcted: string) | /\*\*  \* Select dropdown element by text.  \*   \* @param WebControl Dropdown Control.  \* @param optionTobeSelcted Dropdown text which needs to be selected.  \*   \* Example:   \*   \* cityDrpdwn = new WebControl(this.page.locator('#city'), 'City dropdown');  \*   \* selectDropDownByText(cityDrpdwn, "Sydney");  \*/ |
| selectFromDropdownByIndex(control: WebControl, index: number) | /\*\*  \* Select dropdown element by index.  \*   \* @param WebControl Dropdown Control.  \* @param index Item number which needs to be selected.  \*   \* Example:   \*   \* cityDrpdwn = new WebControl(this.page.locator('#city'), 'City dropdown');  \*   \* selectFromDropdownByIndex(cityDrpdwn, 1);  \*/ |
| selectCheckbox(control: WebControl, expectedCheckboxValue: boolean = true) | /\*\*  \* Select checkbox element.  \*   \* @param WebControl Checkbox Control .  \* @param expectedCheckboxValue Expected checkbox value. If user want to select checkbox then provide 'true' else 'false'  \*   \* Example:   \*   \* isMinorChkbx = new WebControl(this.page.locator('#Minor'), 'Minor Checkbox');  \*   \* selectCheckbox(isMinorChkbx, true);  \*/ |
| isSelected(control: WebControl) | /\*\*  \* Info to get weather checkbox/Radio element is selected or not.  \* Returns true if element is selected else false  \*   \* @param WebControl Checkbox Control of which selected value need to retrived.  \*   \* Example:   \*   \* isMinorChkbx = new WebControl(this.page.locator('#Minor'), 'Minor Checkbox');  \*   \* let checkboxValue = isSelected(isMinorChkbx);  \*/ |
| isEnabled(control: WebControl) | /\*\*  \* Info to get wheather element is enabled or not.  \* Returns true if element is enabled else false  \*   \* @param WebControl Control of which enabled or disabled value need to be retrived.  \*   \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* let enabledValue = isEnabled(loginBtn);  \*/ |
| isDisplayed(control: WebControl) | /\*\*  \* Info to get wheather element is displayed or not.  \* Returns true if element is displayed else false  \*   \* @param WebControl Control of which is displayed value need to be retrived.  \*   \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* let displayedValue = isDisplayed(loginBtn);  \*/ |
| getText(control: WebControl) | /\*\*  \* Read text of an element  \*   \* @param WebControl Control of which text need to be retrieved.  \*   \* Example:   \*   \* errorMsg = new WebControl(this.page.locator('#Alert'), 'Invalid Credentials Error message');  \*   \* let errorMessage = getText(errorMsg);  \*/ |
| getTextboxValue(control: WebControl) | /\*\*  \* Read textbox value of an element  \*   \* @param WebControl textbox Control.  \*   \* Example:   \*   \* usernameTxtbx = new WebControl(this.page.locator("#username"), 'Username textbox');  \*   \* let textboxValue = getTextboxValue(usernameTxtbx);  \*/ |
| getSelectedItemFromDropdown(control: WebControl) | /\*\*  \* Read selected dropdown item.  \*   \* @param WebControl Dropdown Control.  \*   \* Example:   \*   \* cityDrpdwn = new WebControl(this.page.locator('#City'), 'City dropdown');  \*   \* let selectedItem = getSelectedItemFromDropdown(cityDrpdwn);  \*/ |
| getAttributeValue(control: WebControl, attributeName: string) | /\*\*  \* Read attribute value of element.  \*   \* @param WebControl Control of which attriubute value need to retrieved.  \* @param attributeName Name of attribute.  \*   \* Example:   \*   \* usernameTxtbx = new WebControl(this.page.locator('#username'), 'Username textbox');  \*   \* let attributeValue = getAttributeValue(usernameTxtbx, "title");  \*/ |
| scrollToControl(control: WebControl) | /\*\*  \* Scroll to an element.  \*   \* @param WebControl Control on which scroll need to perform.  \*   \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* scrollToControl(loginBtn);  \*/ |
| navigateBack() | /\*\*  \* Navigate back on browser.  \*   \* Example:   \*   \* navigateBack();  \*/ |
| handleAlert(action: string) | /\*\*  \* Handle web Alert by accepting or dismissing it   \*   \* @param action If send parameter as "accept" it accept alert else dismiss it.  \*   \* Example:   \*   \* handleAlert("Accept");  \*/ |
| getAlertDialogMessage() | /\*\*  \* Read the web alert message   \*   \* Example:   \*   \* let alertMsg = getAlertDialogMessage();  \*/ |
| findAll(control: WebControl) | /\*\*  \* Return all elements for given search criteria.  \*   \* @param WebControl Control on which user want to click.  \*   \* Example:   \*   \* dateColumn = new WebControl(this.page.locator('#date'), 'Date Column values');  \*   \* findAll(dateColumn);  \*/ |
| getURL() | /\*\*  \* Read the web URL   \*   \* Example:   \*   \* let url = getURL();  \*/ |
| getTitle() | /\*\*  \* Read the title of web Page   \*   \* Example:   \*   \* let title = getTitle();  \*/ |
| waitTillPageURLContains(url: string) | /\*\*  \* Wait till URL contains value   \*   \* @param url Wait for URL.  \*   \* Example:   \*   \* waitTillPageURLContains("https://www.google.com/");  \*/ |
| waitTillElementIsPresent(control: WebControl) | /\*\*  \* Wait till element is present   \*   \* @param control Wait for element to be present.  \*   \* Example:   \*   \* usernameTxtbx = new WebControl(this.page.locator('#username'), 'Username textbox');  \*   \* waitTillElementIsPresent(usernameTxtbx);  \*/ |
| waitTillElementIsAttached(control: WebControl) | /\*\*  \* Wait till element is attached to the dom  \*   \* @param control Wait for element to be attached in dom.  \*   \* Example:   \*   \* usernameTxtbx = new WebControl(this.page.locator('#username'), 'Username textbox');  \*   \* waitTillElementIsAttached(usernameTxtbx);  \*/ |
| waitTillElementIsNotPresent(control: WebControl) | /\*\*  \* Wait till element is not displayed  \*   \* @param control WebControl for which wait to get hidden.  \*   \* Example:   \*   \* usernameTxtbx = new WebControl(this.page.locator('#username'), 'Username textbox');  \*   \* waitTillElementIsNotPresent(usernameTxtbx);  \*/ |
| sleep(milliseconds: number = 5) | /\*\*  \* Wait for execution to stop for given time  \*   \* @param milliseconds Time for which executed should halt in seconds  \*   \* Example:   \*   \* sleep(10);  \*/ |
| closeBrowser() | /\*\*  \* Close the browser instance  \*   \* closeBrowser();  \*/ |

|  |  |
| --- | --- |
| **Assertions Class Methods** | |
| **MethodName** | **Description** |
| verifyIsEquals(object1: any, object2: any) | /\*\*  \* Verify two objects are equals.   \* If both objects are equal test case gets passed else fails.  \*   \* @param object1 First object.  \* @param object2 Second object.  \* Example:   \*   \* verifyIsEquals("Invalid username", "Invalid username");  \*/ |
| verifyIsContains(actual: string, expected: string) | /\*\*  \* Verify first value contains substring as second.   \* If expected value is present in actual value then test case gets passsed else failed.  \*   \* @param expected First object which has .  \* @param actual Second object.  \* Example:   \*   \* verifyIsContains("Invalid username Test", "Invalid username");  \*/ |
| verifyIsDisplayed(control: WebControl, expectedIsDisplayed: boolean = true) | /\*\*  \* Verify element's displayed/not displayed status as per expected value  \* If expected element's displayed status is as per expected then test case gets passsed else failed.  \*   \* @param control Control of which enable or disable status.  \* @param expValue Expected value enabled as true or disabled as false.  \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* verifyIsEnabled(loginBtn, true);  \*/ |
| verifyIsNotDisplayed(control: WebControl, isAlreadyHidden = false) | /\*\*  \* Verify element is not displayed  \* If expected element's displayed status is false the test case gets passed else failed.  \*   \* @param control Control for which not displayed status need to verify.  \* @param isAlreadyHidden Is control already hidden or need to wait to get hidden.  \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* verifyIsNotDisplayed(loginBtn, true);  \*/ |
| verifyAttributeValue(control: WebControl, attributeName: string, expectedAttributeValue: string) | /\*\*  \* Verify attribute value of the control  \* If elements attribute value is as per expected value then test case gets passed else failed.  \*   \* @param control Control for which attribute value need to verify.  \* @param attributeName Attribute name.  \* @param attributeValue Expected attribute value    \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* verifyAttributeValue(loginBtn, "value", "Submit");  \*/ |
| verifyIsEnabled(control: WebControl, expValue: boolean = true) | /\*\*  \* Verify element's enabled/disabled status as per expected value  \* If expected element's enabled status is as per expected then test case gets passsed else failed.  \*   \* @param control Control of which enable or disable status.  \* @param expValue Expected value enabled as true or disabled as false.  \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* verifyIsEnabled(loginBtn, true);  \*/ |
| verifyDisplayedText(control: WebControl, expectedText: string) | /\*\*  \* Verify element's text as per expected value  \* If expected element's text is as per expected value then test case gets passsed else failed.  \*   \* @param control Control of which text need to verify.  \* @param expectedText Expected value.  \* Example:   \*   \* errorMsg = new WebControl(this.page.locator('#error'), 'Login Error message');  \*   \* verifyDisplayedText(loginBtn, "Invalid Username Test User");  \*/ |
| verifyDisplayedTextContains(control: WebControl, expectedText: string) | /\*\*  \* Verify element's text contains partial expected value  \* If expected element's text contains expected value then test case gets passsed else failed.  \*   \* @param control Control of which text need to verify.  \* @param expectedText Expected partial value.  \* Example:   \*   \* errorMsg = new WebControl(this.page.locator('#error'), 'Login Error message');  \*   \* verifyDisplayedTextContains(loginBtn, "Invalid Username");  \*/ |
| verifyDisplayedTextDoesNotContains(control: WebControl, expectedText: string) | /\*\*  \* Verify element's text does not contains expected value  \* If expected element's text does not contains expected value then test case gets passsed else failed.  \*   \* @param control Control of which text need to verify.  \* @param expectedText Expected partial value.  \* Example:   \*   \* errorMsg = new WebControl(this.page.locator('#error'), 'Login Error message');  \*   \* verifyDisplayedTextDoesNotContains(loginBtn, "Enter User");  \*/ |
| verifyTextboxValue(control: WebControl, expectedText: string) | /\*\*  \* Verify element's textbox is equal to expected value  \* If expected element's textbox value to be expected value then test case gets passsed else failed.  \*   \* @param control Textbox control.  \* @param expectedText Expected value.  \* Example:   \*   \* usernameTxtbx = new WebControl(this.page.locator('#username'), 'Username textbox');  \*   \* verifyTextboxValue(usernameTxtbx, "Test@Adactin");  \*/ |
| verifyCheckboxValue(control: WebControl, value: boolean) | /\*\*  \* Verify element's checkbox/radio value is equal to expected value  \* If expected element's checkbox/radio value to be expected value then test case gets passsed else failed.  \*   \* @param control Checkbox/radio control.  \* @param expectedText Expected checkbox/radio value.  \* Example:   \*   \* isMinor = new WebControl(this.page.locator('#isMinor'), 'Is Minor checkbox');  \*   \* verifyCheckboxValue(isMinor, true);  \*/ |
| verifyAttributeValueContains(locator: WebControl, attributeName: string, attributeValue: string) | /\*\*  \* Verify element's attribute value contains expected value  \* If element's attribute value contains expected value then test case gets passsed else failed.  \*   \* @param control Control of which attribute value need to verify.  \* @param attributeName Attribute name  \* @param expectedText Expected partial value.  \*   \* Example:   \*   \* loginBtn = new WebControl(this.page.locator('#login'), 'Login button');  \*   \* verifyAttributeValueContains(loginBtn, "value", "Submit");  \*/ |
| verifyTagName(control: WebControl, expectedTagName: string) | /\*\*  \* Verify element's tagname is equal to expected value  \* If expected element's Tagname is equal to expected value then test case gets passsed else failed.  \*   \* @param control Checkbox/radio control.  \* @param expectedText Expected checkbox/radio value.  \* Example:   \*   \* isMinor = new WebControl(this.page.locator('#isMinor'), 'Is Minor checkbox');  \*   \* verifyTagName(isMinor, "a");  \*/ |
| verifyListContainsValue(control: WebControl, valueToVerify: string) | /\*\*  \* Verify list of elements text values contains expected text at any point in list it gets passed else failed.  \* If expected text is present in anywhere in list it gets passed else failed.  \*   \* @param control List of elements  \* @param valueToVerify Expected text present in list.  \* Example:   \*   \* dateList = new WebControl(this.page.locator('#dates'), 'dates column');  \*   \* verifyListContainsValue(dateList, "12-Dec-2023");  \*/ |
| verifyListContainsMultipleValues(control: WebControl, listOfValueToVerify: string[], waitTillElementIsDisplayed = true) | /\*\*  \* Verify list of elements text values contains multiple values  \* If expected multiple values is present in anywhere in list it gets passed else failed.  \*   \* @param control List of elements  \* @param valueToVerify Expected text present in list.  \*/ |
| verifyListForMultipleValues(control: WebControl, listOfValueToVerify: string[], waitTillElementIsDisplayed = true) | /\*\*  \* Verify list of elements text values contains multiple values in exact sequence  \* If expected multiple values is present in sequence in list it gets passed else failed.  \*   \* @param control List of elements  \* @param valueToVerify Expected text present in list.  \*/ |
| verifyAlertText(expectedText: string) | /\*\*  \* Verify Alert text is equal to expected value  \* If alert text is equal to expected value then test case gets passsed else failed.  \*   \* @param expectedText Expected alert text.  \* Example:   \*   \* verifyAlertText("Password should be 8 digit.")  \*/ |
| verifyURLContains(subString: string) | /\*\*  \* Verify URL contains expected value  \* If URL contains expected value then test case gets passsed else failed.  \*   \* @param expectedText Expected partial URL.  \* Example:   \*   \* verifyURLContains("google.com");  \*/ |
| verifyCountOfElements(control: WebControl, expectedCount: number) | /\*\*  \* Verify count of elements which matched the control  \* If count of elements is equal to expected value then test case will get passsed else failed.  \*   \* @param control Control element.  \* @param expectedCount Expected count of elements matching criteria  \* Example:   \*   \* checkboxes = new WebControl(this.page.locator("xpath=//a"), 'checkboxes');  \* verifyCountOfElements(isMinor, 5);  \*/ |
| verifyPageTitleContains(expectedText: string) | /\*\*  \* Verify title of web page contains expected value  \* If title contains expected value then test case will get passsed else failed.  \*   \* @param expectedText Expected partial Title.  \* Example:   \*   \* verifyPageTitleContains("Google");  \*/ |