**ACTION\_ITEMS**

Group2\_QEA20QE006

**Introduction:**

It is an Automation related Project. We have Automated the **Action\_Item** Module of Mainspring platform **(https://pratesting.cognizant.com)** using Selenium which is an open-source tool used for automation, Java technologies, Page Object Model, Maven, TestNG have been used.

**System Requirements:**

* Java
* Maven
* Selenium Jars
* Apache POI
* TestNG
* Excel

**Please read the document below before executing:**

1. **Enter your Email (Cognizant) in config.properties against Email key.**
2. **Enter your Password (Cognizant) in config.properties against Password key.**
3. **The OTP verification needs to be entered manually. The wait time for OTP authentication is 20 seconds. Do not click submit. The submit button is automated.**
4. **Don’t change the path for browser drivers BaseSetup class.**
5. **Don’t change the path for Excel.**

**Script Description**

**Packages Used:**

1. baseSetup

2. MainProject.MainSpring

3. testClass

4. Utilities

**baseSetup package description:**

**BaseSetup:** This class invokes all the base drivers, browser drivers, generate report, method to get excel data, open browser, redirect to URL. This class is extended by all classes of MainProject.MainSpring package, testClass.

**Some of the Methods Used:-**

* **invokebrowser()** - Used for invoking the driver that is used for the particular browser ie. either edge or chrome and for maximizing the window.
* **Properties readProperties()** - Used for loading the config.properties file.
* **getExcelData(String fileName, String sheetName,int totalNoOfRows,int totalNoOfCols)** - This method is of 2-D String type consists of 4 arguments and is used to read the excel data.
* **addValues(String LocatorValue, String value)** - This method is basically assigning the values to the TestEvent module.
* **clearField(String LocatorValue)** - This method is used to clear the input fields, this is used before every addValue method.
* **isAlertPresent()** **-** This method of boolean return type is checking whether an Alert is present or not.
* **openUrl()** - Used for navigating to the URL
* **close()** - This method is used for closing the driver.

**MainProject.MainSpring package description:**

**LoginMail:** This class locates the email element, reads email id from config.properties and press submit button. Explicit wait is used. This page redirects to the LoginPassword page.

**LoginPassword:** This class locates the password element, reads password from config.properties and press submit button. Explicit wait is used.This page redirects to the LoginOTP page.

**LoginOTP**: This class waits for the OTP that is generated and which is entered manually. The static wait time is 50 seconds. The Submit button is automated. This page redirects to the LoginRem page.

**LoginRem**: This class is the reminder class where the submit button is clicked. Explicit wait is used. This page redirects to the HomePage class.

**HomePage**: This class loads on Pratesting homepage. After loading, it Clicks on the hamburger icon present on the top left side and selects the project name Common Agile Exe Proj\_9th Oct 2020. This page redirects to the ProjectPage class.

**ProjectPage**: This page is loaded on our project where we hover on the Execute option and click on TestEvent ie, the module assigned to us. This page redirects to the TestEventPage class.

**TestEventPage**:This page consists of all the list of TestEvents already created.Our main aim is to assign a new Test Event. This is possible by Clicking on **+** icon.This page redirects to the TestEvent creation page.

**Testpage**: It performs Frame switching. It Checks the functionality of Save ,Save & add new and Reset button.

**testClass package description**

**TestInvalid\_1**: Instantiate all the classes of MainProject.MainSpring, Executes all the classes created. Provides the invalid data to ActionItem form using dataprovider from excel sheet named MainSpring Testing.xlsx comprising Sheet 1. This class uses the Test annotation for TestNG and DataProvider annotation for reading all the invalid test cases from the sheet created by Abhishek Choudhary.

**TestInvalid\_2**: Instantiate all the classes of MainProject.MainSpring, Executes all the classes created. Provides the invalid data to ActionItem form using dataprovider from excel sheet named MainSpring Testing.xlsx comprising Sheet 1. This class uses the Test annotation for TestNG and DataProvider annotation for reading all the invalid test cases from the sheet created by Shashank Thakur.

**TestValid\_1**: Instantiate all the classes of MainProject.MainSpring, Executes all the classes created. Provides the valid data to ActionItem form using dataprovider from excel sheet named MainSpring Testing.xlsx comprising Sheet 1. This class uses the Test annotation for TestNG and DataProvider annotation for reading all the valid test cases from the sheet created by R.K.Barath.

**TestValid\_2**: Instantiate all the classes of MainProject.MainSpring, Executes all the classes created. Provides the valid data to ActionItem form using dataprovider from excel sheet named MainSpring Testing.xlsx comprising Sheet 1. This class uses the Test annotation for TestNG and DataProvider annotation for reading all the invalid test cases from the sheet created by Manisha Kumari

**TestInvalid**: Instantiate all the classes of MainProject.MainSpring, Executes all the classes created. Provides the invalid data to ActionItem form using dataprovider from excel sheet named MainSpring Testing.xlsx comprising Sheet 1. This class uses the Test annotation for TestNG and DataProvider annotation for reading all the invalid test cases from the sheet.

**TestValid**: Instantiate all the classes of MainProject.MainSpring, Executes all the classes created. Provides the valid data to ActionItem form using dataprovider from excel sheet named MainSpring Testing.xlsx comprising Sheet 1. This class uses the Test annotation for TestNG and DataProvider annotation for reading all the valid test cases from the sheet.

**utilities package description**

**DateUtil:** It is used to create timestamp for logs.

**ExtentReportManager:** It is used to create a HTML report.

**MainSpring Testing.xlsx:**

It is the excel sheet where all the raw data that is to be passed through the form is kept. It consists of six sheets.

**TestNG.xml:**

The information about our test (e.g. the class name, the groups,suites etc...) is present inside this TestNg.xml file.

**Config.properties:**

This property file is added which consists of all the global properties/values that is required throughout the project.

**Project Description**

**The following key concepts are handled:**

1. **Exception Handling** - A mechanism used to handle the runtime errors so that normal flow of the application can be maintained. try, catch and throws concept of Exception Handling has been used.

2. **Wait** -The wait functions are essential when it comes to executing Selenium tests. They help to observe and troubleshoot issues that may occur due to variation in time lag. Explicit, Implicit and thread.sleep are its ways has been used in this project.

3. **iframe** - iFrame in Selenium Webdriver is a web page or an inline frame which is embedded in another web page or an HTML document embedded inside another HTML document. For switching between iframe we have used switchTo.frame().

4. **xpath** - XPath in Selenium is an XML path that is used for navigation through the HTML structure of the page. It is a syntax or language for finding any element on a web page using XML path. implementation using id , class , Relative and Absolute xpaths has been used.

5. **Reading data from Excel** - Reading data from [Excel](https://www.guru99.com/excel-tutorials.html) file is possible with the help of [Java](https://www.guru99.com/java-tutorial.html) IO package and [Apache](https://www.guru99.com/apache.html) POI library.The data for Test Event form is taken using data provider.

6. **Data Provider** - It is an annotation in TESTNG and returns a two-dimensional JAVA object to the test method and the test method. The data for Test Event form is provided using this.

6. **alert()** - An Alert in Selenium is a small message box that appears on screen to give the user some information or notification. It notifies the user with some specific information or error, asks for permission to perform certain tasks and it also provides warning messages as well. All the alerts in the Test event module are handled.

7. **Page Object Model (POM)** - The entire project is built around this pattern. It is basically a design pattern, popularly used in test automation which creates Object Repository for web UI elements. The advantage of the model is that it reduces code duplication and improves test maintenance. By the help of Page Factory in Selenium which is an inbuilt Page Object Model framework concept for Selenium WebDriver initialization of Page objects and redirecting is possible.