Author: Vishnu Godse: Test Automation Developer []

20/03/2022

Test Automation: BDD Framework Documentation

Completed

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

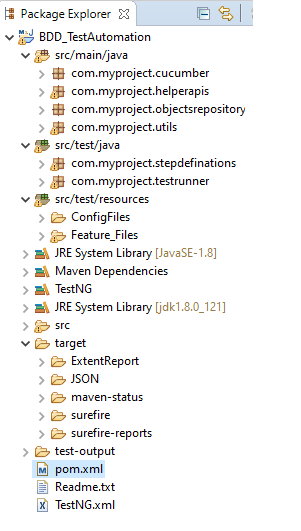
1. **Framework Structure**

**Below image shows the overall basic structure of designed BDD framework which mainly**

**Involved below components:**

* 1. **Java [JDK 1.8.0.321]**
  2. **Selenium 3.14.0**
  3. **Edge Browser v 99+**
  4. **Cucumber**
  5. **Junit**
  6. **TestNG**

**Snippet:**

****

1. **Detailed Overview of Framework**

**Detailed structure overview and workflow of this BDD framework is as mentioned below: the information of every package and usage of classes designed under each package as given below:**

* 1. **SRC/MAIN/JAVA**

**Under this directory we will be having below packages and classes: Please refer description provided under each**

* + 1. **com.myproject.cucumber:**

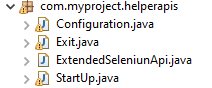
****

\* This is Customised Cucumber test runner class.

\* This has been customised to perform all our configurations before the framework test execution starts.

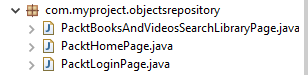
\* Runs each cucumber scenario found in the features as separated test

* + 1. **com.myproject.helperapis:**

****

**This package has 04 classes as shown in image above refer below description of each one of those**

* **Configiration.java :** This class has initial one time configuration required to launch automated test cases: Such as --- Set web driver [Browser,timeout..Etc.] , Configure Project properties files , configure reports and set other required paths.
* **ExtendedSeleniumAPI.java:** This class has user defined r- usables for each action that needs to be performed on web page such as – Input text, Click element, get xpath of element, Mouse hover, Scroll page , get color of element …. Etc.
* **StartUp.java:** This class has intial one time start up or pre requisites activities such as – Login on the basis of URL, User name and Password provided in config file. This action is stored in StartUp method as re-usable and it is being called in TestRunner class for each and every test case.
* **Exit.java:** This class is more similar like Startup – here we do written post requisites such as - Logout , Close driver, Quit Driver.. Etc.  
  + 1. **com.myproject.objectrepository**

****

Under this package we have different classes based on the module to store page objects and its related getter , Setter methods, for example : Login page.java class will have all web elements address in this class and will be access in respective step definition file by extending this class.

**Page Object sample Code snippet for technical understanding purpose:**

**final** String USERNAME\_INPUT\_ID = "login-input-email";

**final** String PASSWORD\_INPUT\_ID = "login-input-password";

**final** String SIGNIN\_BUTTON\_XPATH = "//button//span[contains(text(),'Sign In')]";

**public** **void** inputUserName(String userName) {

inputText(getWebElementByID(USERNAME\_INPUT\_ID), userName, "User name");

}

**public** **void** inputPassword(String password) {

**try** {

getWebElementByID(PASSWORD\_INPUT\_ID).sendKeys(password);

ReportGenerator.*getReportGenerator*().info("Input \*\*\*\*\*\*\*\*\*\*\*\* as Password",**false**);

} **catch** (StaleElementReferenceException e) {

} **catch** (Exception e) {

ReportGenerator.*getReportGenerator*().fail(e.getMessage());

}

}

**public** **void** clickSignInButton() {

clickOnElement(getWebElementByXpath(SIGNIN\_BUTTON\_XPATH), "Sign In Button");

}

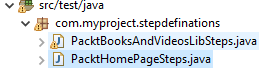
* + 1. **com.myproject.utils**

****

* **Assertions.java:** This Designed separately to maintain customized assert methods along with Fail message and screen capture functionality – For Example below methods : assertEquals, assertFalse, assertNotEquals, assertTrue
* **ReportGenerator.java:** This a singleton class to provide reporting utility to the framework. Basically this is used for to Generate Extent Report using below methods: Setup report, initialize report, Start Test , PASS,FAIL,INFO log, Flush report.
  1. **SRC/TEST/JAVA**

**Under this directory we will be having below packages and classes: Please refer description provided under each**

* + 1. **com.myproject.stepdefinations**

****

This package will have separate classes on the basis of module, each class will have implementation of feature declared in feature files.

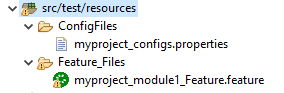
This can be also called as business layer and actual selenium automation steps coding.

* + 1. **Com.myproject.testrunner**

****

* This class is used to run test feature files and has intial configuration of cucumber things
* This class will have Methods those will be executed before each scenario in feature file.
* This class will have Methods those set up the reporting based on Feature and Scenario name.
* This class will have Methods those executed after each scenario in feature file.
* This class will have Methods those flush all the reporting of the scenario to html file and logout.
  1. **SRC/TEST/RESOURCES**

**Under this directory we will be having below packages and classes: Please refer description provided under each**

****

* + 1. **Config Files:** This directory will have Config.properties file which contains the ENV variables such as URL, USER NAME , PASSWORD and as an when required if any other Variables

****

* + 1. **Feature Files: This class will have declaration of features in the format of shown below:**

****

#Feature: List of scenarios.

#Scenario: Business rule through list of steps with arguments.

#Given: Some precondition step

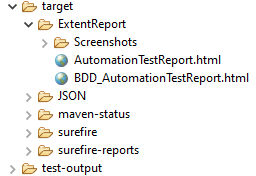
#When: Some key actions

#Then: To observe outcomes or validation

#And,But: To enumerate more Given,When,Then steps

#Scenario Outline: List of steps for data-driven as an Examples and <placeholder>

1. **Report: This directory will be used to store the result output of test executed**

****

1. **POM.xml :** This file will have records of dependencies that needs to be downloaded automatically and as an when we require new dependencies we can add those.

****

1. **TestNG.xml:** This file is being used to run test suite as TestNG .

****

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END of Document \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***