Last updated: Oct 4, 2021

Selenium Java Training - Session 34 - Developing a Framework and Automating Live Project (Part 2)

- Configuring log4j for logs
    - Copy paste the log4j2.xml file here into the resources package - View here
    - Open the LoginTest and write the logs - View here
    - Add the tags to pom.xml file to specify where exactly log4j file is available in the project
        - Search for 'Maven filtering' in google
        - Add the tags under <build> tag
    - Run the code and observe that the logs will be printed in the logs file
- Enable Parallel execution using TestNG
    - Adding sample automation code in all the Test Classes and Run them as a group to see whether they are running one after the other
    - Modify the testng.xml file as below:
        - Separate individual classes to separate tags
        - Add parallel=tests
    - Run the code and observe that all the tests will run in parallel
- Taking screenshots for failing Tests
    - Create a listeners package under src/main/java
    - Create a Listeners class under it and make it implement ITestListener interface
        - Add TestNG library if unable to resolve the errors
    - Select 'Source' menu in Eclipse IDE < Override/Implement Methods and select all the check boxes of ITestListerner
    - Create reusable method for taking screenshots in Base class with two parameters - View here
    - Extend the Listeners class with Base class
    - Update the onTestFailure() method of Listeners class - View here
    - Add the Listeners tags in testng.xml file - View here
    - Make the driver of the Test Classes to global and public
    - Intentionally fail a test and run the testng.xml file
- Integrating ExtentReports to the framework
    - Create a package say 'utilities' under 'src/main/java'
    - Create a class under this package say 'ExtentReporter'
    - Create a method say 'getExtentReport' with this code - View here
    - Make the getExtentReport method static
    - Write extent report code into different Listeners methods (onTestStart, onTestSuccess, onTestFailure and onFinish) - View here
    - Remove parallel execution from testng.xml file and Run
    - Make ExtentReports thread-safe by adding this code to Listeners class

- Add parallel execution to testng.xml file
- Add this line to the Listeners class
  - ThreadLocal<ExtentTest> extentTestThread = new ThreadLocal<ExtentTest>();
- Add this line inside onTestStart
  - extentTestThread.set(extentTest);
- Replace existing line with this line inside onTestSuccess
  - extentTestThread.get().log(Status.PASS,"Test Passed");
- Replace existing line with this link inside onTestFailure
  - extentTestThread.get().fail(result.getThrowable());

- Adding the screenshot to the ExtentReports
  - Make the takeScreenshot() method return the destination file path - View here
  - Update the onTestFailure method
    - String screenshotFilePath = takeScreenshot(testMethodName,driver);
    - extentTestThread.get().addScreenCaptureFromPath(screenshotFilePath, testMethodName);
- Run the testng.xml file

- Applying Encapsulation in the framework
  - Make all the Page Objects as private - So far Project Download Here
- Integrating Cucumber BDD into this Framework
  - Add the dependencies for Cucumber
    - Cucumber-Java
    - Cucumber-testng
  - Verify whether the Cucumber Eclipse IDE plugin is installed
  - Create a package say 'features' under 'src/test/java'
  - Create a feature file say 'Login.feature' under this package - View here
  - Create a package say 'stepdefinitions' under 'src/test/java'
  - Create a Class say 'Login.java' under this package
  - Auto-generate the Step definitions using TinyGerkhin Chrome Plugin
  - Move the automation code to respective step definitions methods - View here
  - Create a runner package
  - Create a Runner class under it - View here
    - Extend AbstractTestNGCucumberTests
  - Generate a testng2.xml file and modify like this - View here
  - Run the testng2.xml file
  - Mention this testng2.xml file in maven surefire plugin of pom.xml and run using the Maven command (mvn test)

By,
Arun Motoori