FrameWork Documentation

=====================================================

Table Of Contents

1. Design Pattern
2. Tools Used
3. Project Hierarchy
4. Cross Browser
5. Reporting

=====================================================

**DESIGN PATTERN - Page Object Model**

**Page Object Model:**

* **Test Method level:** This is where the steps of a test case are executed and where we handle all inputs and validations. Data-source binding should be done at this level
* **Page Object Classes:** Each logical division in an application should have its own page class. User Actions will be done in this layer
* **Driver Factory:** Where the initialization of all types of drivers took place. Which will be helpful for cross browser/cross platform testing.
* **Object Repository:** Where the webelements get stored. Abstract is from the page object classes and save it in some data file it could be json/xml/.spec or even properties file. This adds another layer of modularity to the framework. In future if the UI changes then QA Team just need to focus only upon this layer.
* **Helper/Wrapper Class:** It's very useful to create a class where you write wrappers around the common Selenium functionalities such as FindElement, Click or SendKeys. By implementing your own wrappers, you can do error handling, smarter waits, timeouts, fail safe or retries.
* **Test Data Source:** Define a data source for the test method where the test will be performed with a different set of data. We can use .xls , .yml or even sql databases for this.

**TOOLS USED:**

1. **Selenium :** To automate web browsers.
2. **TestNG:** As a testing framework
3. **Java:** Programming language
4. **Maven:** build and dependencies management tool.
5. **Eclipse:** IDE

**Project Hierarchy:**

How FRIDAY Test Automation Task Relates to Above Points:

Explaining Test automation architecture implemented for FRIDAY QA Task:

**Folder: src/main/java**

1. **package com.friday.qatask.pageactions :** This package is related to page object classes where user actions happen. It contains the java classes for the pages coming for Quote selection till Entering Birth date page. By going through pages we came to know that Entering details of a car and its model become dynamic according to data for example: For FORD, FOCUS application asks for selecting body type but for other models it doesn’t. So, Encapsulating this behavior into CarDetailController.java class using SIMPLE FACTORY PATTERN.
2. **package com.friday.qatask.PageDSL:** In this package core selenium functions are being wrapped around by pageHelper.java class which is further extended by the pageAction classes.
3. **package com.friday.qatask.TestInitiator:** Driverfactory.java class returning the browser driver according to the demand from the TestNG.xml file. ScreenobjectProvider.java is a facade class to provide objects of screen actions. Testinitiator.java class is the class where prerequisites to run a test are happening.
4. **package com.friday.qatask.utilities:** This is the package where all the utilities are being done like file reading. YAML READER for reading test data. OBJECT FILE READER for reading .spec files where webelements and locating strategy is being stored. TakeScreenShot class Where screenshot will be taken and at the time of any test failure and put it into screenShots folder present inside src/test/resources. DATAPROVIDER class which is responsible for fetching the test data for a particular test and running it accordingly.

**Folder: src/test/java**

**→ package com.friday.qatask.tests:** This package consists all the test classes.

The test class “ThreeCarBrands\_EachBrandThreeDifferentCar\_Test” will be responsible for testing at least three different car brands, each brand with three different cars.”

**Folder: src/test/resources**

**→ Folder PageObjectRepository:** This folder consists of a .spec file where page locators and locating strategy is stored.

→ **Folder TestData:** This folder consists of QA\_Data.yml file. All the test data used for the test cases has been stored in this file.

→ **Folder TestNGXml:** This folder consists of the TestNG xml file. This file will be run as a testNG suite to execute all the test cases.

**Cross Browser:**

Browser parameter is set in the FridayAssignment\_TestNG.xml . To run in ChromeBrowser , set the browser parameter value as “chrome”. And to run it in FireFoxBrowser, set the browser parameter value as “firefox”.

**REPORTING:**

TestNG in built reporting is used. After running FridayAssignment\_TestNG.xml as a testng suite, reports will be present in test-output folder with the name as “emailable-report.html”