**TestVagrant hiring challenge for applicants: Code Review and Changes**

* This project is designed to run on Mac, Linux and Window system, therefore in all three classes i.e FlightBookingTest.java, HotelBookingTest.java, SignInTest.java method setDriverPath()is defined.

Also Driver binaries are placed in project which is increasing project size and may require change when new version is available in market.

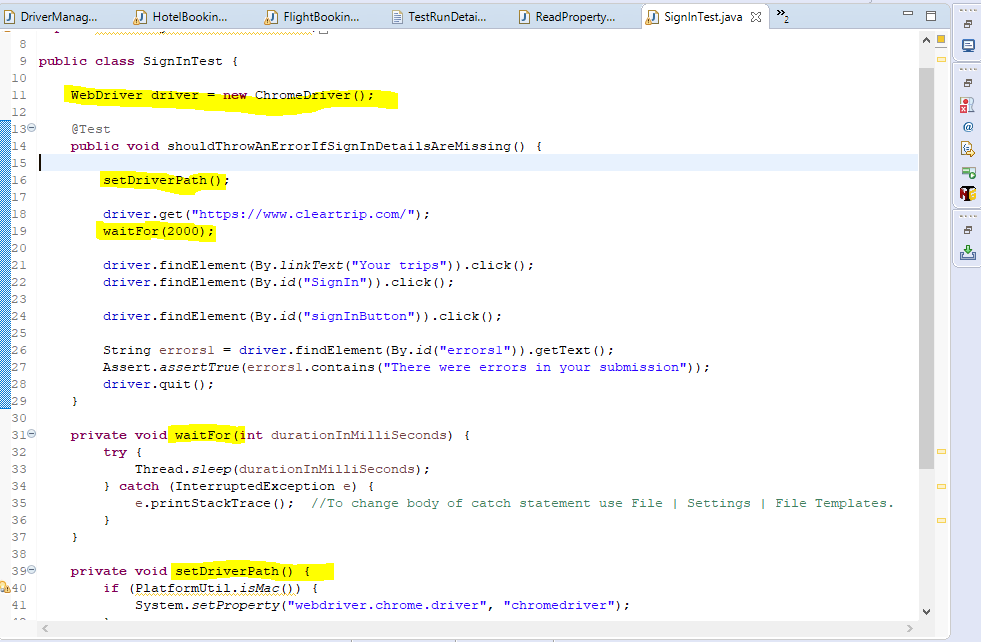
Changes done:

* All driver binaries are removed from project.
* Implemented Bonigracia’s webdriver manager, this will not only help to run program in multiple OS but it will support multiple browser without worrying about binary files.
* With the help of WebDriver manager we can remove setDriverPath()
* Change done to Framework structure: For Code reusability and to follow POM framework, creating a new folder src/main/test and moving all test files to src/main/test
* To improve code reusability DriverManager.java added to testVagrantCodingRound.driver
* All the input data is provided in code itself:
  + Changes done:
    - Input data moved to TestRunDetails.properties file in src folder
    - To read this property file ReadProperty.java is added in package testVagrantCodingRound.ReadProperty

**Modification in Test Files:**

1. SignInTest.java

Code that require improvement:

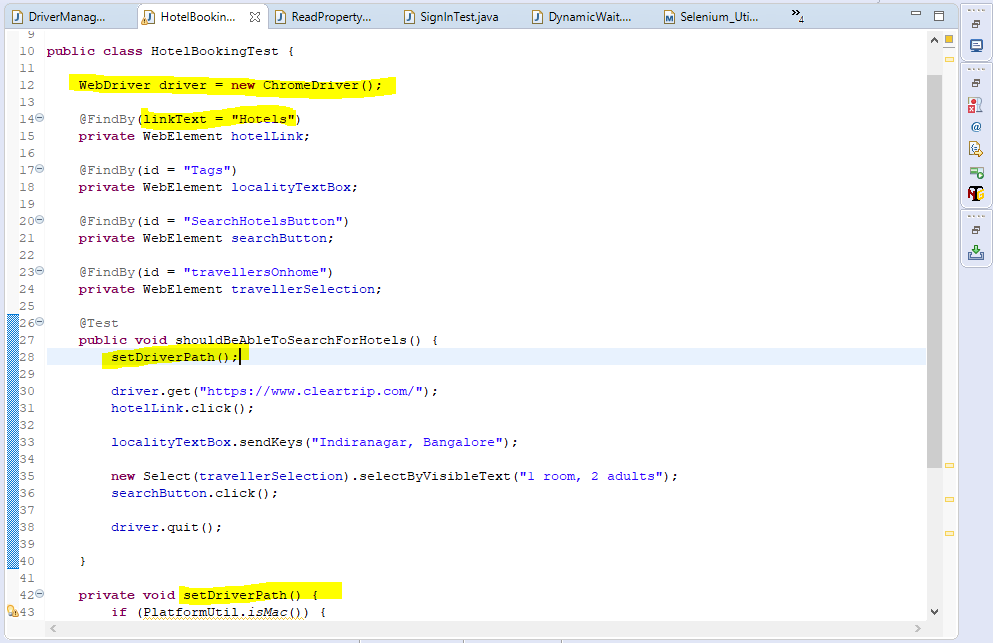


1. Method uses static wait.
2. setDriver Path is no longer required, even if we don’t use Webdrver manager, we can create separate utility for this method
3. POM is not followed, link text is used
4. Hardcoded values are used for url
5. SignIn and Error message is available in frame, frame is not switched

**Improvements done:**

1. Instead of Static wait, Explicit wait is added
2. Due to use of Webdriver manager setDriverPath() is removed
3. POM model is used.
4. Instead of link text, using xpath or id.
5. Hardcoded values are fetched from TestRunDetails.properties file
6. @AfterTest and @BeforeTest are added for test setup and clearance

2.HotelBookingTest.java



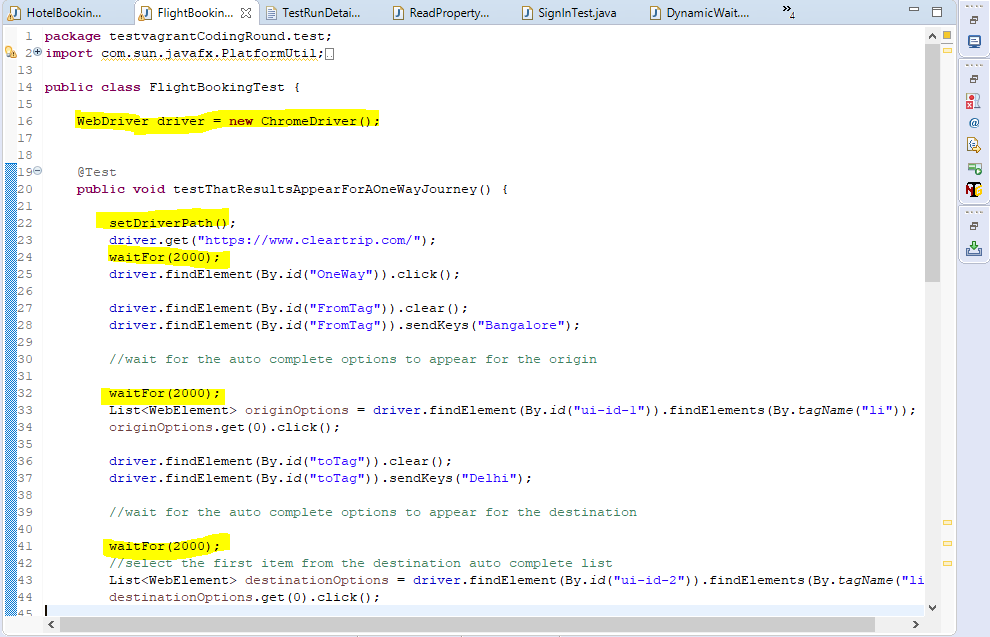
Code that require improvement:

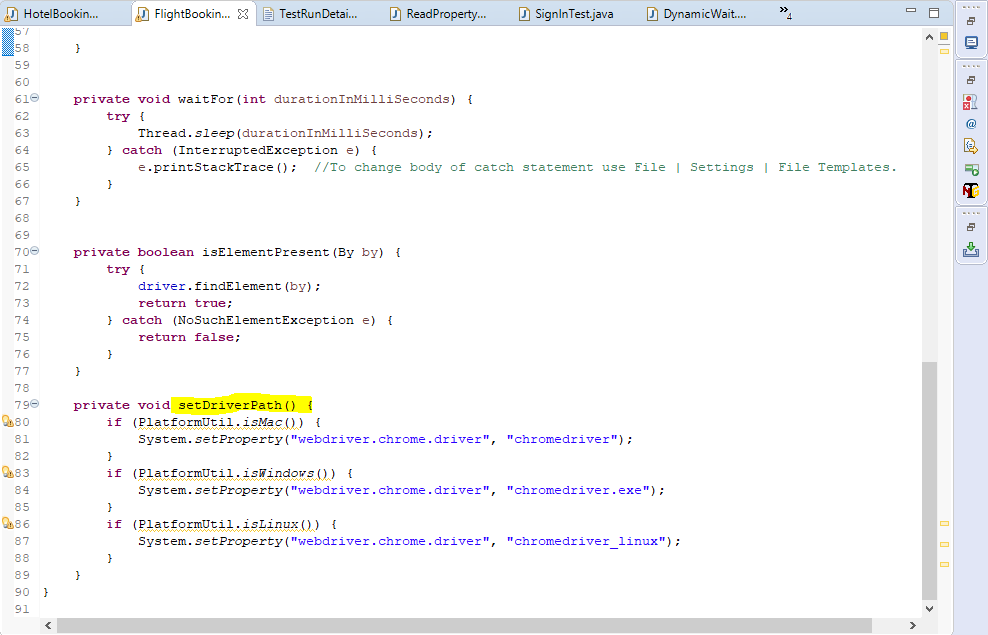
1. Method uses static wait.
2. setDriver Path is no longer required, even if we don’t use Webdrver manager, we can create separate utility for this method
3. Hardcoded values are used for url and location
4. link text is used

**Improvements done:**

1. Instead of Static wait, Explicit wait is added
2. Due to use of Webdriver manager setDriverPath() is removed
3. Instead of link text, using xpath or id.
4. Hardcoded values are fetched from TestRunDetails.properties file
5. @AfterTest and @BeforeTest are added for test setup and clearance
6. Assertion added to check if results are displayed in next page

3.FlightBookingTest.java





Code that require improvement:

1. Method uses static wait.
2. setDriver Path is no longer required, even if we don’t use Webdrver manager, we can create separate utility for this method
3. POM is not followed, link text is used
4. Hardcoded values are used for url and locations
5. SignIn and Error message is available in frame, frame is not switched

**Improvements done:**

1. Instead of Static wait, Explicit wait is added
2. Due to use of Webdriver manager setDriverPath() is removed
3. POM model is used.
4. Instead of link text, using xpath or id.
5. Hardcoded values are fetched from TestRunDetails.properties file
6. @AfterTest and @BeforeTest are added for test setup and clearance

**Package Sturcture:**

**Src/main/java**

This folder contains all the logic

Package **testvagrantCodingRound.driver:**

This package contains DriverManager.java, it is responsible for launching browser and supports google chrome, FireFox, IE and Edge browser.

Package **testvagrantCodingRound.ReadProperty:**

This package contains class and logic to read data from TestRunDetails.properties file

Package **testvagrantCodingRound.Utility:**

This package contains class file for custome functions

DatePicker.java: This file provides xpath to select dates

DyanamicWait.java: This file contains explicit wait programs

Package **testvagrantCodingRound.WebPages:**

This package contains class files for each webpage used for testing.

Page elements and Methods to perform on page are provided in the files.

**Src/test/java**

This folder contains only one package **testvagrantCodingRound.test**

This package contains files according to test cases, each test case has its class file

**Src/test/resources**

It contains only one file TestRunDetails.properties, This file contains data required for test execution, eg: browser name, filepath etc.

**Parallel Execution:**

All test cases are modified for parallel execution, **testing.xml** has been added to support parallel execution.

By changing thread-count number of classes to run parallel can be decided.

To run test case in parallel, run from testing.xml