Hello Fresh QA Automation Assignment

Task1 - UI Automated test using Java, Selenium WebDriver

For this task initial code structure was provided and I need to improve the code design/structure by adding logging, reporting, screenshots of failed tests, multiple browser's support, encapsulation and configurator etc.

Solution includes following:

Framework: Cucumber framework (BDD)

- Behavior driven development framework
- It is free tool
- Plugin in cucumber works faster
- It provides best encapsulation by its feature file.
- It is very easy to understand because of its feature file.
- Feature file contains high level description of test scenarios in simple language (gherkin language).
- It also provides built in reporting feature and you don't need to add any plugin/tool for the reporting.
- For logging, log4j tool is also integrable with cucumber framework easily by just adding its dependency in pom.xml
- Single testcase can also be run easily with this framework.

Web Driver: Selenium

- **WebDriver** is a web automation framework that allows you to execute your tests against different browsers, not just Firefox, Chrome (unlike **Selenium** IDE). **WebDriver** also enables you to use a programming language in creating your test scripts (not possible in **Selenium** IDE).

Maven:

- Maven is a build automation tool used primarily for Java projects. Maven addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies.

Configurator:

- I have created **Config>ReadBasicConfig.java** class which read path of configurations property file from **enums>ConfigPath.java** class to read **Hello-fresh.properties** where all the configuration properties are mentioned. The purpose of creating the enum was if we want to read data from multiple file then just have to give path of file in enum and it will read the file automatically.

Reporting:

- Cucumber provides built-in reporting feature, so I have used this feature in **RunITTest.java** class as following:

```
format = { "pretty",
                    "html:target/site/cucumber-pretty",
                         "json:target/cucumber.json" }
```

- It will create html and json reports on the mentioned paths in the project.

Screenshots:

- Cucumber itself doesn't provide functionality for taking screenshots, but it does make it possible to embed them in the report.

- But we can do it by **Selenium** webdriver with Cucumber, Following is the command to take screenshot in S**elenium**:

```
// Returns
byte[]((TakesScreenshot)driver).getScreenshotAs(OutputTyp
e.BYTES)
```

Multiple Browser's support:

- I have added multiple browsers support in PageStepDefs.java class to use multiple browsers like chrome and firefox etc.
- By default, I have set the chrome browser.
- Cucumber best supports the firefixe version 21.0.

Instructions to run the code:

- *Import the project as existing maven project*
- *Open command line terminal window*
- *Go to the directory where your project exists e.g cd /eclipse-workspace/hello-fresh-assignment*
- *Run the command* `mvn test -Dbrowser="googlechrome"`
- *It will open the chrome browser and testcases will start to execute.*
- *You can also change the browser by changing the* `-Dbrowser="firefox"` *in the above command.*
- *You can execute all the testcases or single testcase by RunITTest.java class by giving tags in the tags parameter e.g* `tags={"@UserSignIn,@UserlogIn,@UserPurchasing"}`
- *After the test execution you can see the logged info on the terminal about the testcases execution status.*
- *You can also see the log information in the file created in the project named* ***Cucumber-Automation.log*** *also this project will show proper logs in console as well.*
- *You can check the generated json and html reports as well.*

- *You can also check the screenshots of failed tests in the reports as it will embedded into the reports.*

Task 2: API Automated test using Java

Task is to test a Restful booking API, it consists of a simple *booking-controller* that allows you to create, update, delete and view existing bookings.

Solution includes following:

Framework: TestNG framework

For this task I have chosen TestNG framework

- TestNG is an automation testing framework
- Annotations are easier to use and understand.
- Test cases can be grouped more easily.
- TestNG allows us to create parallel tests.
- TestNG is capable of generating HTML-based reports

API: Rest Assured API

A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

- Implemented getBookings, getBooking{id}, createBooking(POST) calls.
- Also tried to implement PUT and DELETE to update and delete booking but I was constantly getting 403 and 405 error code of forbidden and unauthorized access respectively.

Instruction to run the code:

- *Import the project as existing maven project*
- *Run the **testing.xml** file as testing suite*