This framework is developed to get 5 days weather forecast through openweathermap API.

**Specifications**

* Gradle: v5.6.2
* TestNg: v7.1.0
* RestAssured: v4.3.0
* Extent Reports: v2.41.2
* JSON Simple: v1.1.1
* Log4j: v1.2.17

**Features of JARs used to develop the framework**

* Gradle as build tool (A basic building block for CI/CD)
* TestNg for developing exhaustive tests
* RestAssured as the interface to automate Rest endpoints and related tests
* Extent Reports for HTML reporting
* JSON Simple to read and write JSON Object during REST call
* Log4j for textual logging

**Project Structure**

The code base of above project is divided into two folders as per gradle division, **src/main/** & **src/test/**. Let's understand the working of each class written within the folder:

**src/main**

It contains package within **java folder**: com.project.freecharge.utilities. There are three classes available within this package: **Configuration.java, NoRainyException.java** and **RestUtility.java**. There is a resource folder within src/main where config.properties are kept.

**Configuration.java**

This class is used to read value of all the properties written inside config.properties. It uses encapsulation and gives getters to read the properties values. These proeprties values are used throughout the project wherever required.

**RESTUtility.java**

This class contains all the static methods which is used to perform REST calls, like GET, POST, PUT, PATCH, and DELTE. Apart from these operations, there are methods to perform other operations like getting response, status code, setting and resetting BASE URL, etc.

**NoRainyException.java**

This class extends Exception class. This is designed to explicitly throw the exception when weather is not Rainy.

**resources/config.properties**

This file contains all the properties like access token, media types, etc which are used at project levels. This file is accessed by Configuration.java class.

**src/test**

It also contains package within **java folder**: com.project.freecharge. This package have two sub-packages which are- controller & test. Controller sub-package contains **APITestBed.java, BaseTest.java, RetryFailedTests.java, RetryListener.java & MyListeners.java** classes, whereas, test sub-package contains all the test cases built with TestNG. So it mainly contains TestNg class like **WeatherForecastTest.java.**

Further, resources folder contains two sub-folders which are-

* JsonBody
* Suite

Let's understand each classes and folders.

**BaseTest.java**

This class contains all the common methods which are used by all the test cases. It contains methods to set up report generation and closing, tear down methods, Json read and write, etc.

**APITestBed.java**

This class contains @BeforeSuite & @AfterSuite annotations, which mainly sets & resets the BASE URI. It extents BaseTest.java class to re-use the test level common methods.

**MyListeners.java**

This class extends ITestListener interface which is used for auto logging and reporting the testing events like failure, success, test startups, test end process, etc.

**RetryFailedTest.java**

This class implements IRetryAnalyzer interface which is used retrying failed test cases.

**RetryListener.java**

This class extends IAnnotationTransformer interface which is used for triggering RetryFailedTest.java class.

**WeatherForecast.java**

This is a TestNg class which contains test for validating rainy weather for cities defined withing @DataProvider annotation..

**resources/JsonBody folder**

This folder contains Json files which are mainly gathered by using GET call of REST API. The json files are used to get weather reports.

**resources/Suite folder**

This folder contains testng.xml file, **apitsuite.xml** contains test class for Open Weather Map API testing. The configuration of this xml is done inside build.gradle file as test task which uses useTestNg() method.

**ExecutionLog.log**

This file is stored inside the root directory of the project. It stores all the logs generated during test exeuction and it is directly configured with log4j.xml file.

**TestReport Folder**

This folder contains all the HTML reports generated by Extent Report and it is stored in root directly of the project.

**How to execute API Tests?**

Go to src/test/resources/Suite/apisuite.xml, Right click and Run. So, API tests are executed either from apisuite.xml file or directly running that Testng class file which is stored inside test sub-package of src/test/java folder.

**Alternative way to execute the tests?**

1. Go to build.gradle file and directly run **runApi()** task to execute API test suite.
2. Use terminal and run command**gradle runApi** for executing API test suite.