**Verifications/Tests:**

Below Scenarios are in scope:

1. Validated all the countries and total count
2. Validated search by 2 characters, 3 characters or Any character and validated the response message
3. Validated the negative scenarios
4. Validated if any duplicate entry is there or not

TestCase1.class,// REST web-service to get a list of all Countries and also will validate the duplicates

TestCase2.class,// This testcase is to Validate REST web-service to search country by 2 character alphanumeric ISO code

TestCase3.class,//This testcase is to Validate REST web-service to search country by 3 character alphanumeric ISO code

TestCase4.class,//This testcase is to Validate REST web-service to search country by 3 character alphanumeric ISO code Negative Testcase

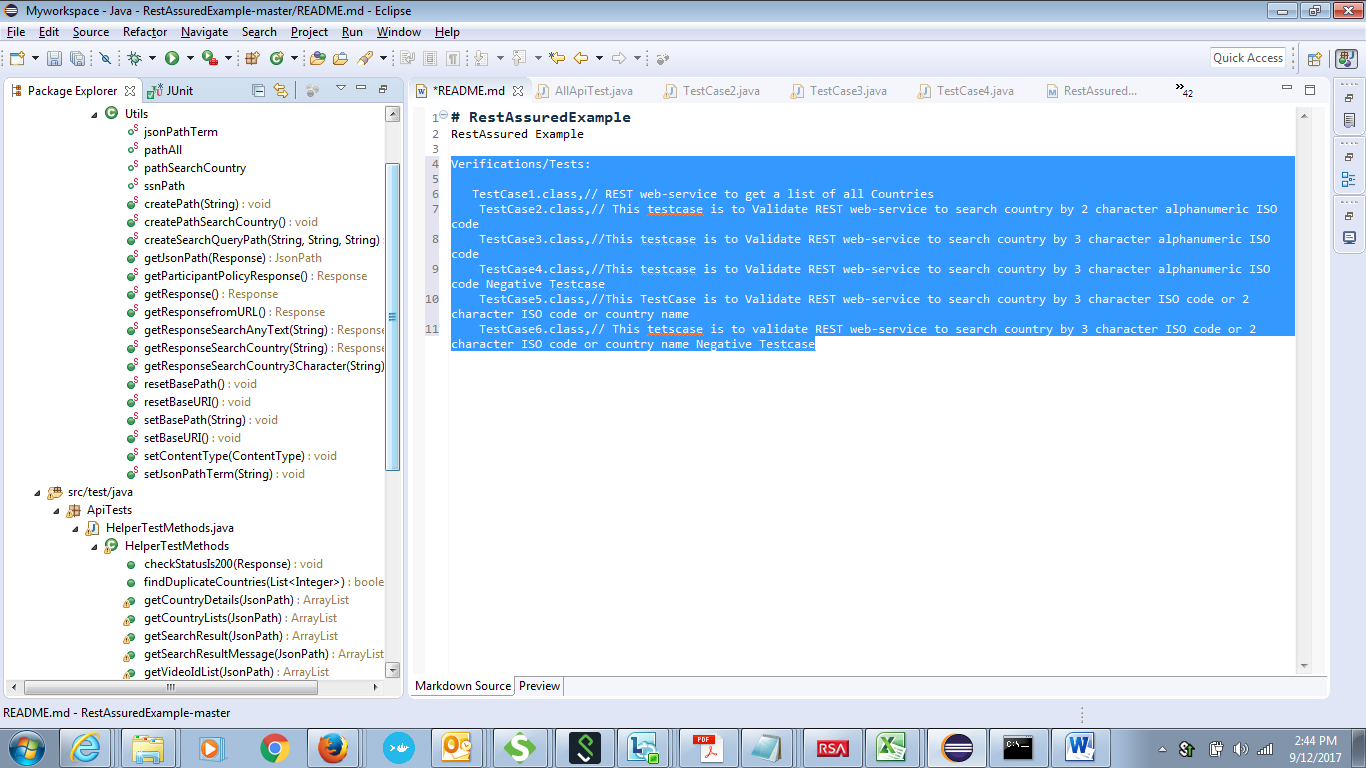
TestCase5.class,//This TestCase is to Validate REST web-service to search country by 3 character ISO code or 2 character ISO code or country name

TestCase6.class,// This tetscase is to validate REST web-service to search country by 3 character ISO code or 2 character ISO code or country name Negative Testcase

First, it is very reasonable to use a framework/library which provides us testing an API easily in a short period of time and I chose the **Rest-assured library**

**Project Structure:**

**1)** I wrote **HelperMethods** class and **RestUtil** classes for **reusability** and **clean project structure**. **– HelperMethods** class comprises of **test related functions**. **– RestUtil** class contains **Rest Assured related methods**.



**2)** **TestCase1** and **TestCase2** etc. contains our **tests and assertions**.

**3)** **AllApiTest** is the test runner **suite**. We can **run all the test with the AllApiTest test sui**t.

In this project I used **Maven** and **pom.xml** is our dependency and profile xml file. We will also go into details of this file further.

Now we can start to examine the test code. We start with pom.xml.

**POM.xml**

In pom.xml we should add:

– **JUnit library** – It is our test framework

– **Hamcrest library** – For assertion methods

– Jayway **Rest Assured library** – It is our REST API testing library

– For running the test from **command prompt** with maven I also added **AllApiTests profile** at the end of the pom.xml. We can run all tests with “**mvn test –PallApiTests**” command.

pom.xml

XHTML



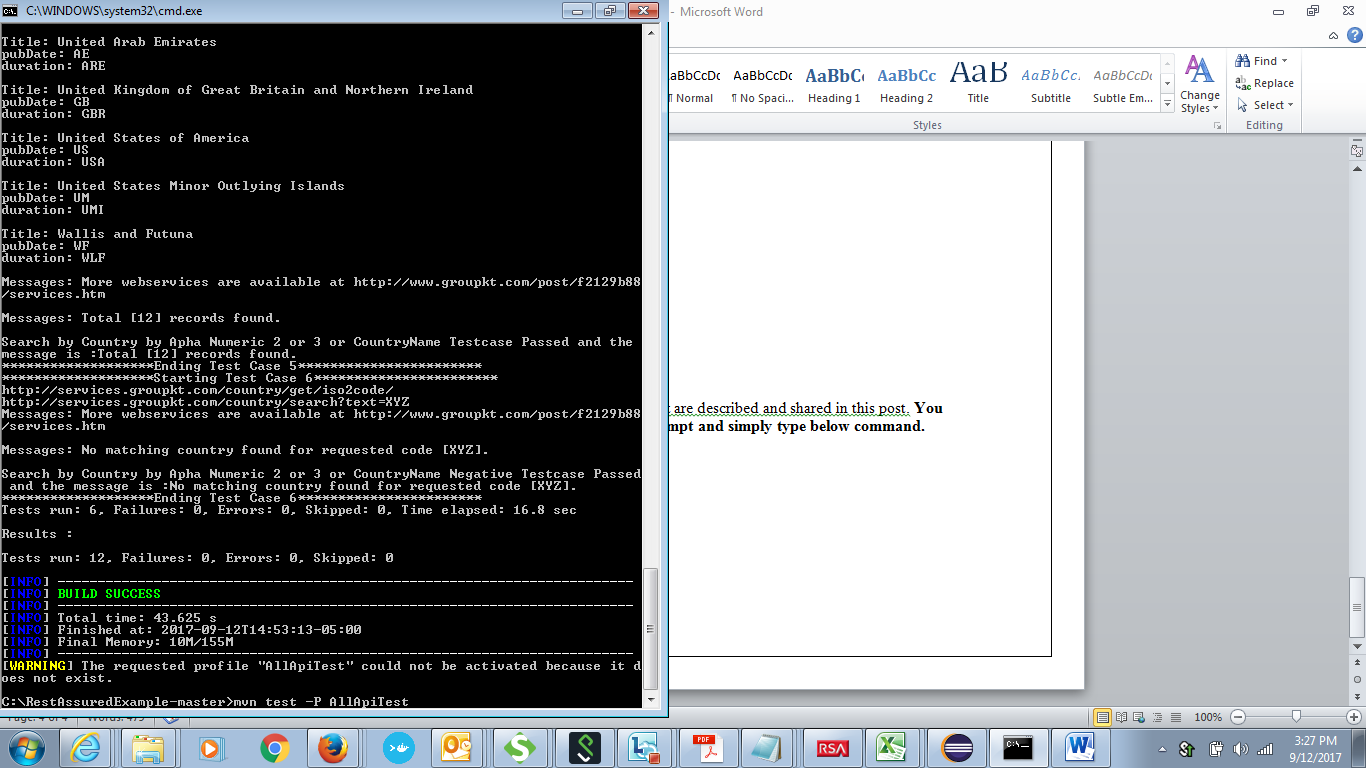
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70 | <?xml version="1.0" encoding="UTF-8"?>  <project xmlns="http://maven.apache.org/POM/4.0.0"           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">      <modelVersion>4.0.0</modelVersion>        <groupId>com.5min.apitest</groupId>      <artifactId>5min-apitest</artifactId>      <version>1.0-SNAPSHOT</version>        <!--Dependencies-->      <dependencies>          <dependency>              <groupId>org.hamcrest</groupId>              <artifactId>hamcrest-all</artifactId>              <version>1.3</version>          </dependency>            <dependency>              <groupId>org.hamcrest</groupId>              <artifactId>hamcrest-junit</artifactId>              <version>2.0.0.0</version>          </dependency>            <dependency>              <groupId>junit</groupId>              <artifactId>junit</artifactId>              <version>4.12</version>          </dependency>            <dependency>              <groupId>com.jayway.restassured</groupId>              <artifactId>json-schema-validator</artifactId>              <version>2.8.0</version>          </dependency>            <dependency>              <groupId>com.jayway.restassured</groupId>              <artifactId>rest-assured</artifactId>              <version>2.8.0</version>          </dependency>      </dependencies>        <!--Profiles-->      <profiles>          <profile>              <id>AllApiTests</id>              <activation>                  <activeByDefault>true</activeByDefault>              </activation>              <build>                  <pluginManagement>                      <plugins>                          <plugin>                              <groupId>org.apache.maven.plugins</groupId>                              <artifactId>maven-surefire-plugin</artifactId>                              <version>2.19.1</version>                              <configuration>                                  <includes>                                      <include>\*\*/AllApiTest.class</include>                                  </includes>                              </configuration>                          </plugin>                      </plugins>                  </pluginManagement>              </build>          </profile>      </profiles>    </project> |

**Test Execution**

When you create your project with the files that are described and shared in this post. **You should go project directory in command prompt and simply type below command.**

**mvn test -P AllApiTest**

Then, our **tests will run** as shown below.



Also, you can **run the tests in Eclipse**.

Select AllApiTest.Java under Test Suite and click run all the testcases will run together.

