SourceCode-RestAssured-TestNg-Project

AddProduct.java

```
package com.simplilearn.restassured;


import static org.hamcrest.CoreMatchers.equalTo;


import org.apache.log4j.Logger;
import org.testng.annotations.Test;


import io.restassured.RestAssured;
import io.restassured.http.ContentType;


public class AddProduct {
          private static String BASE_URL = "http://localhost:9010";
   private static Logger logger = Logger.getLogger(AddProduct.class);
   String response = null;


   @Test(description = "Test to create a product")
   public void testAddProduct() {
      try {
         logger.info("Start :: Test to create a product ");
         String requestBody="{\r\n"
                    + "       \"id\": 999,\r\n"
                    + "       \"image\": \"2.png\",\r\n"
                    + "       \"name\": \"Disprin+\",\r\n"
                    + "       \"category\": \"medicine\",\r\n"
                    + "       \"brand\": \"BZ Medico\",\r\n"
                    + "       \"status\": 1,\r\n"
                    + "       \"price\": 120\r\n"
```

```
              + "}\r\n"
              + "";

      RestAssured.given().baseUri(BASE_URL)
          .contentType(ContentType.JSON)
          .body(requestBody)
          .when().post("/add-product")
          .then().assertThat().statusCode(200)
          .body("code",equalTo(101));

      response = RestAssured.given().baseUri(BASE_URL)
              .contentType(ContentType.JSON)
          .body(requestBody)
          .when().post("/add-product")
          .getBody().asString();
    } catch (Exception e) {
      logger.error("Exception Object :: " + e.toString());
      logger.error("End Exception :: " + e.getLocalizedMessage());
    }

    logger.info("Response :: " + response);
    logger.info("End :: Test to create a product");
  }

}
```

DeleteMedicine.java

```
package com.simplilearn.restassured;


import static org.hamcrest.CoreMatchers.equalTo;


import org.apache.log4j.Logger;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class DeleteMedicine {


	private static final String BASE_URL="http://localhost:9010";
	private static Logger logger=Logger.getLogger(DeleteMedicine.class);
	String response=null;
	@Test(description="Test to delete product from the store")
	public void testDeleteProduct() {
		try {
		logger.info("Start :: Test to delete product from the store");
		int id=999;
		RestAssured.given().baseUri(BASE_URL)
		.when().delete("/delete-product?id="+id).then().statusCode(200).and()
		.body("message",equalTo("Product with ID "+id+" Deleted Successfully."));



		response = RestAssured.given().baseUri(BASE_URL)
		.when().delete("/delete-product?id="+id).getBody().asString();
	}catch(Exception e) {
		logger.error("Exception Object :: "+e.toString());
		logger.error("End Exception :: "+e.getLocalizedMessage());
	}
		logger.info("Reponse :: "+response);
		logger.info("End :: Test to delete product from the store");
	}

}
```

GetMedicine.java

```
package com.simplilearn.restassured;


import static org.hamcrest.CoreMatchers.equalTo;


import org.apache.log4j.Logger;
import org.testng.annotations.Test;
import io.restassured.RestAssured;


public class GetMedicine {
	private static final String BASE_URL="http://localhost:9010";
	private static Logger logger=Logger.getLogger(GetMedicine.class);
	String response=null;
	@Test(description="Test to get all the products from the store")
	public void testGetProduct() {
		try {
		logger.info("Start :: Test to get all the products from the store");
		RestAssured.given().baseUri(BASE_URL)
		.when().get("/get-products").then().statusCode(200).and()
		.body("code",equalTo(101));
```

```
        response = RestAssured.given().baseUri(BASE_URL)
        .when().get("/get-prodcuts").getBody().asString();
    }catch(Exception e) {
        logger.error("Exception Object :: "+e.toString());
        logger.error("End Exception :: "+e.getLocalizedMessage());
    }
        logger.info("Reponse :: "+response);
        logger.info("End :: Test to get all the products from the store");
    }

}
```

GetUsers.java

```
package com.simplilearn.restassured;


import static org.hamcrest.CoreMatchers.equalTo;


import org.apache.log4j.Logger;
import org.testng.annotations.Test;




import io.restassured.RestAssured;


public class GetUsers {
    private static final String BASE_URL="http://localhost:9010";
    private static Logger logger=Logger.getLogger(GetUsers.class);
    String response=null;
    @Test(description="Test to get all the register users from the store")
    public void testGetusers() {
        try {
        logger.info("Start :: Test to get all the register users from the store");
        RestAssured.given().baseUri(BASE_URL)
        .when().get("/get-users").then().statusCode(200).and()
        .body("message",equalTo("4 Users Fetched Successfully.")).and()
        .body("code",equalTo(101));



     response = RestAssured.given().baseUri(BASE_URL)
        .when().get("/get-users").getBody().asString();



        }catch(Exception e) {
            logger.error("Exception Object :: "+e.toString());
            logger.error("End Exception :: "+e.getLocalizedMessage());
        }

        logger.info("Reponse :: "+response);
        logger.info("End :: Test to get all the register users from the store");
    }
}
```

UpdateMedicine.java

```
package com.simplilearn.restassured;

import static org.hamcrest.CoreMatchers.equalTo;

import org.apache.log4j.Logger;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;

public class UpdateMedicine {


    private static final String BASE_URL="http://localhost:9010";
    private static Logger logger=Logger.getLogger(UpdateMedicine.class);
    String response=null;
    @Test(description="Test to update the product in the store")
    public void testGetProduct() {
        try {
            String requestBody="{\r\n"
                    + "    \"id\": 999,\r\n"
                    + "    \"image\": \"2.png\",\r\n"
                    + "    \"name\": \"Disprin+\",\r\n"
                    + "    \"category\": \"medicine\",\r\n"
                    + "    \"brand\": \"BZ Medico\",\r\n"
                    + "    \"status\": 1,\r\n"
                    + "    \"price\": 120\r\n"
                    + "}\r\n"
                    + "";
        logger.info("Start :: Test to update the product in the store");

        RestAssured.given().baseUri(BASE_URL)
        .contentType(ContentType.JSON).body(requestBody)
        .when().put("/update-product").then()
        .assertThat().statusCode(200)
        .body("message", equalTo("Disprin+ Updated Successfully."));



        response = RestAssured.given().baseUri(BASE_URL)
                .contentType(ContentType.JSON).body(requestBody)
           .when().put("/update-product").getBody().asString();
    }catch(Exception e) {
        logger.error("Exception Object :: "+e.toString());
        logger.error("End Exception :: "+e.getLocalizedMessage());
    }
        logger.info("Reponse :: "+response);
        logger.info("End :: Test to update the product in the store");
    }
}
```

UpdateMedicineStatus.java

```
package com.simplilearn.restassured;

import static org.hamcrest.CoreMatchers.equalTo;
import org.apache.log4j.Logger;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;

public class UpdateMedicineStatus {

    private static final String BASE_URL="http://localhost:9010";
    private static Logger logger=Logger.getLogger(UpdateMedicine.class);
    String response=null;
    @Test(description="Test to update the product in the store")
    public void testGetProduct() {
        try {
            String requestBody="{\r\n"
                    + "    \"id\": 999,\r\n"
                    + "    \"image\": \"2.png\",\r\n"
                    + "    \"name\": \"Disprin+\",\r\n"
                    + "    \"category\": \"medicine\",\r\n"
                    + "    \"brand\": \"BZ Medico\",\r\n"
                    + "    \"status\": 0,\r\n"
                    + "    \"price\": 120\r\n"
                    + "}\r\n"
                    + "";
        logger.info("Start :: Test to update the product in the store");

        RestAssured.given().baseUri(BASE_URL)
        .contentType(ContentType.JSON).body(requestBody)
        .when().put("/update-product-status").then()
```

```
                .assertThat().statusCode(200)
                .body("message", equalTo("Disprin+ Status Updated Successfully."));



                response = RestAssured.given().baseUri(BASE_URL)
                                .contentType(ContentType.JSON).body(requestBody)
                        .when().put("/update-product-status").getBody().asString();
        }catch(Exception e) {
                logger.error("Exception Object :: "+e.toString());
                logger.error("End Exception :: "+e.getLocalizedMessage());
        }
                logger.info("Reponse :: "+response);
                logger.info("End :: Test to update the product in the store");
        }
}
```

TestLogs.java

```
package com.simplilearn.testlogs;

import org.testng.annotations.Test;

import org.apache.log4j.LogManager;
import org.apache.log4j.Logger;

public class TestLogs {
```

```
          static final Logger logger = Logger.getLogger(TestLogs.class);
   @Test
          public void testInfoLogs() {
                    logger.info("This is info logs");
          }

   @Test
          public void testWarnLogs() {
                    logger.info("This is Warning logs");
          }

   @Test
          public void testDebugLogs() {
                    logger.info("This is debug logs");
          }

   @Test
          public void testErrorLogs() {
                    logger.info("This is error logs");
          }
}
```

Log4j.properties

```
# Root Logger option
log4j.rootLogger=INFO, file, stdout

# Direct log messages to stdout (Print log on console add appender as
ConsoleAppender)
#log4j.appender.stdout=org.apache.log4j.ConsoleAppender
```

```
#log4j.appender.stdout.Target=System.out
#log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
#log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-
5p %c{1}:%L %m%n

## Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=logs/rest-api.log
log4j.appender.file.MaxFileSize=10MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
%c{1}:%L %m%n
log4j.appender.file.Append=true
```

Pom.xml

```
<?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.simplilearn.cucumber</groupId>
  <artifactId>ATE-Capstone_RestAssured_Project</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>ATE-Capstone_RestAssured_Project</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>8</maven.compiler.source>
    <maven.compiler.target>8</maven.compiler.target>
    <rest.assured.version>5.4.0</rest.assured.version>
    <testng.version>7.8.0</testng.version>
  </properties>

  <dependencies>
   <!-- testng -->
          <dependency>
                <groupId>org.testng</groupId>
                <artifactId>testng</artifactId>
                <version>${testng.version}</version>
                <scope>test</scope>
          </dependency>
```

```
<!-- rest-assured -->
<dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>rest-assured</artifactId>
      <version>${rest.assured.version}</version>
      <scope>test</scope>
</dependency>

<!-- rest-assured/json-path -->
<dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>json-path</artifactId>
      <version>${rest.assured.version}</version>
      <scope>test</scope>
</dependency>

<!-- rest-assured/json-schema-validator -->
<dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>json-schema-validator</artifactId>
      <version>${rest.assured.version}</version>
</dependency>

<!-- io.rest-assured/xml-path -->
<dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>xml-path</artifactId>
      <version>${rest.assured.version}</version>
</dependency>

<!-- reportng -->
<dependency>
      <groupId>org.uncommons</groupId>
      <artifactId>reportng</artifactId>
      <version>1.1.4</version>
      <scope>test</scope>
      <exclusions>
            <exclusion>
                  <groupId>org.testng</groupId>
                  <artifactId>testng</artifactId>
            </exclusion>
      </exclusions>
</dependency>

<!-- com.google.inject/guice -->
<dependency>
      <groupId>com.google.inject</groupId>
      <artifactId>guice</artifactId>
      <version>3.0</version>
</dependency>

<!-- com.google.inject/guice -->
<dependency>
      <groupId>velocity</groupId>
      <artifactId>velocity-dep</artifactId>
      <version>1.4</version>
</dependency>

<!-- log4j/log4j -->
<dependency>
      <groupId>log4j</groupId>
```

```
                  <artifactId>log4j</artifactId>
                  <version>1.2.17</version>
            </dependency>

      </dependencies>

      <build>
            <plugins>
                  <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-surefire-plugin</artifactId>
                        <version>2.18.1</version>
                        <configuration>
                              <properties>
                                    <property>
                                          <name>usedefaultlisteners</name>
                                          <value>false</value>
                                    </property>
                                    <property>
                                          <name>listener</name>

      <value>org.uncommons.reportng.HTMLReporter,

      org.uncommons.reportng.JUnitXMLReporter</value>
                                    </property>
                              </properties>
                              <workingDirectory>target/</workingDirectory>
                        </configuration>
                  </plugin>
            </plugins>
      </build>
</project>
```

TestNG.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<suite name="RestAssuredTestNGTest">
<Listeners>
          <Listener class-name="org.uncommons.reportng.HTMLReporter" />
          <Listener class-name="org.uncommons.reportng.JUnitXMLReporter" />
</Listeners>
<test name="REST APIs">
   <classes>
      <class name="com.simplilearn.restassured.AddProduct"></class>
      <class name="com.simplilearn.restassured.DeleteMedicine"></class>
      <class name="com.simplilearn.restassured.GetMedicine"></class>
      <class name="com.simplilearn.restassured.GetUsers"></class>
      <class name="com.simplilearn.restassured.UpdateMedicine"></class>
      <class
name="com.simplilearn.restassured.UpdateMedicineStatus"></class>
   </classes>
</test>
</suite>
```