}

SourceCode-RestAssured-TestNG-Project

TestLogs.java

```
package com.simplilearn.logtesting;


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");
        }
}
```

AddProduct.java

```
package com.simplilearn.restassuredtest;

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 ");
      int id = 101;
      String image_url = "image_url";
      String name = "sampleShoe";
      String category = "Runnings";
```

```
            int sizes = 9;
            int price = 1000;

            RestAssured.given().baseUri(BASE_URL)
                    .queryParam("id",id).queryParam("image",image_url)
                    .queryParam("name",name).queryParam("category",category)
                    .queryParam("sizes",sizes).queryParam("price",price)
                    .when().post("/add-shoe")
                    .then().assertThat().statusCode(200)
                    .body("message",equalTo("sampleShoe Added Successfully."));

            response = RestAssured.given().baseUri(BASE_URL)
                          .queryParam("id",id).queryParam("image",image_url)
                    .queryParam("name",name).queryParam("category",category)
                    .queryParam("sizes",sizes).queryParam("price",price)
                    .when().post("/add-shoe")
                    .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");
    }
}
```

DeleteProduct.java

```
package com.simplilearn.restassuredtest;
import static org.hamcrest.CoreMatchers.equalTo;
import org.apache.log4j.Logger;
import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class DeleteProduct {

	private static final String BASE_URL="http://localhost:9010";
	private static Logger logger=Logger.getLogger(DeleteProduct.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=201;
		RestAssured.given().baseUri(BASE_URL)
		.when().delete("/delete-shoe?id="+id).then().statusCode(200).and()
		.body("message",equalTo("Shoe with ID 201 Deleted Successfully."));


		response = RestAssured.given().baseUri(BASE_URL)
		.when().delete("/delete-shoe?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");
        }
}
```

RetrieveAllProducts.java

```
package com.simplilearn.restassuredtest;

import static org.hamcrest.CoreMatchers.equalTo;

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

import io.restassured.RestAssured;


public class RetrieveAllProducts {

      private static final String BASE_URL="http://localhost:9010";
      private static Logger 
logger=Logger.getLogger(RetrieveAllProducts.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-shoes").then().statusCode(200).and()
            .body("code",equalTo(101));


            response = RestAssured.given().baseUri(BASE_URL)
            .when().get("/get-shoes").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");
      }
}
```

RetrieveAllUsers.java

```
package com.simplilearn.restassuredtest;
import static org.hamcrest.CoreMatchers.equalTo;
import org.apache.log4j.Logger;
import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class RetrieveAllUsers {

        private static final String BASE_URL="http://localhost:9010";
        private static Logger logger=Logger.getLogger(RetrieveAllProducts.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("3 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");
        }

}
```

UpdateProduct.java

```
package com.simplilearn.restassuredtest;
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 UpdateProduct {

    private static final String BASE_URL="http://localhost:9010";
    private static Logger logger=Logger.getLogger(UpdateProduct.class);
    String response=null;
    @Test(description="Test to update the product in the store")
    public void testGetProduct() {
            try {
                int id = 101;
```

```
        String image= "update image";
        String name = "updated Shoe";
        String category = " updated Runnings";
        int sizes = 12;
        int price = 1099;
                logger.info("Start :: Test to update the product in the store");

                RestAssured.given().baseUri(BASE_URL)
                .queryParam("id",id).queryParam("image",image)
       .queryParam("name",name).queryParam("category",category)
       .queryParam("sizes",sizes).queryParam("price",price)
                .when().put("/update-shoe").then()
                .assertThat().statusCode(200)
                .body("message", equalTo("updated Shoe Updated Successfully."));



                response = RestAssured.given().baseUri(BASE_URL)
                        .queryParam("id",id).queryParam("image",image)
                  .queryParam("name",name).queryParam("category",category)
                  .queryParam("sizes",sizes).queryParam("price",price)
                  .when().put("/update-shoe").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");
        }
}
```

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.restassuredtest</groupId>
  <artifactId>PhaseEnd_RestAssured_Project</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>PhaseEnd_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>
```

```xml
		<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.restassuredtest.AddProduct"></class>
      <class name="com.simplilearn.restassuredtest.DeleteProduct"></class>
      <class
name="com.simplilearn.restassuredtest.RetrieveAllProducts"></class>
      <class
name="com.simplilearn.restassuredtest.RetrieveAllUsers"></class>
      <class name="com.simplilearn.restassuredtest.UpdateProduct"></class>
   </classes>
</test>
</suite>
```