

Source Code

DeleteCallRequest.java

```
package com.simplilearn.restassuredtest.RestAssured_Assignment_001;
```

```
import static org.testng.Assert.assertEquals;
```

```
import static org.testng.Assert.assertTrue;
```

```
import org.testng.annotations.Test;
```

```
import io.restassured.RestAssured;
```

```
import io.restassured.response.Response;
```

```
public class DeleteCallTest {
```

```
    private static String baseUrl = "https://petstore.swagger.io/v2/pet";
```

```
    @Test(description="Test on delete call request")
```

```
    public void testDeletePet() {
```

```
        int petId=2003;
```

```
        Response response = RestAssured.given()
```

```
            .when()
```

```
            .delete(baseUrl + "/" + petId)
```

```
            .then()
```

```
            .statusCode(200)
```

```
            .extract().response();
```

```
        assertTrue(response.getBody().asString().contains("code"));
        assertTrue(response.getBody().asString().contains("message"));
        assertEquals(response.jsonPath().get("message"), String.valueOf(petId));
    }
}
```

GetCallTest.java

```
package com.simplilearn.restassuredtest.RestAssured_Assignment_001;
```

```
import static org.testng.Assert.assertEquals;
```

```
import static org.testng.Assert.assertTrue;
```

```
import org.testng.annotations.Test;
```

```
import io.restassured.RestAssured;
```

```
import io.restassured.response.Response;
```

```
public class GetCallTest {
```

```
    private static String baseUrl = "https://petstore.swagger.io/v2/pet";
```

```
    @Test(description="Test on Get call request")
```

```
    public void testGetPet() {
```

```
        int petId=2003;
```

```
        Response response = RestAssured.given()
```

```
            .when()
```

```

        .get(baseUrl + "/" + petId)

        .then()

        .statusCode(200)

        .extract().response();

    assertTrue(response.getBody().asString().contains("status"));

    assertTrue(response.getBody().asString().contains("id"));

    assertEquals(response.jsonPath().get("status"), "available");

}

}

```

PostCallTest.java

```

package com.simplilearn.restassuredtest.RestAssured_Assignment_001;

```

```

import org.testng.annotations.Test;
import static org.hamcrest.CoreMatchers.equalTo;
import io.restassured.RestAssured;
import io.restassured.http.ContentType;

```

```

public class PostCallTest {

```

```

    private static String baseUrl =
    "https://petstore.swagger.io/v2/pet";

```

```

    @Test(description="Test on post call request")

```

```

    public void testPostPet() {
        String requestBody = "{\n" +
            "\"id\": 2003,\n" +
            "\"category\": {\n" +
            "\"id\": 0,\n" +
            "\"name\": \"string\"\n" +
            "},\n" +
            "\"name\": \"Doggie\",\n" +

```

```

        "\"photoUrls\": [\n" +
        "\"string\"\n" +
        "],\n" +
        "\"tags\": [\n" +
        "{\n" +
        "\"id\": 0,\n" +
        "\"name\": \"string\"\n" +
        "}\n" +
        "],\n" +
        "\"status\": \"available\"\n" +
        "}";

```

```

RestAssured.given().baseUrl(baseUrl)
    .contentType(ContentType.JSON)
    .body(requestBody)
    .when()
    .post()
    .then()
    .statusCode(200)
    .body("id", equalTo(2003))
    .extract().response();

```

```

    }

```

```

}

```

PutCallRequest.java

```

package com.simplilearn.restassuredtest.RestAssured_Assignment_002;

```

```

import java.util.HashMap;

```

```

import org.testng.annotations.Parameters;

```

```

import org.testng.annotations.Test;

```

```

import static org.hamcrest.CoreMatchers.equalTo;

```

```

import io.restassured.RestAssured;

```

```

import io.restassured.http.ContentType;

```

```
import io.restassured.response.Response;
```

```
public class PutCallRequest {
```

```
    private static String baseUrl = "https://petstore.swagger.io/v2/pet";
```

```
    private static HashMap<String, String> statusMap = new HashMap<>();
```

```
    static {
```

```
        statusMap.put("DEV", "available_DEV");
```

```
        statusMap.put("QA", "available_QA");
```

```
        statusMap.put("PROD", "available_PROD");
```

```
    }
```

```
    @Test
```

```
    @Parameters("environment")
```

```
    public void putCallTesting(String environment) {
```

```
        String status = statusMap.get(environment);
```

```
        String requestBody = "{ \r\n"
```

```
            + "\"id\": 9223372016900013000, \"category\": { \r\n"
```

```
            + "\"id\": 20021,    \r\n"
```

```
            + "\"name\": \"string\",    \r\n"
```

```
            + "\"name\": \"doggie\", \"photoUrls\": [    \r\n"
```

```
            + "\"string\"    \r\n"
```

```
            + "], \"tags\": [    \r\n"
```

```
            + "{ \r\n"
```

```

+ "\"id\": 0, \r\n"
+ "\"name\": \"string\" \r\n"
+ "}], \r\n"
+ "\"status\": \""+status+"\" \r\n"
+ "} ";

```

```

Response response = RestAssured.given()

```

```

    .contentType(ContentType.JSON)
    .body(requestBody)
    .when()
    .put(baseUrl)
    .then()
    .statusCode(200)
    .body("category.id", equalTo(20021))
    .body("status", equalTo(status))
    .extract().response();

```

```

System.out.println(response);

```

```

}

```

```

}

```

TestNG.xml

```

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

<suite name="PetStorePUTTestSuite">
    <test name="PetStorePUTTest_DEV">
        <parameter name="environment" value="DEV"/>
        <classes>
            <class
name="com.simplilearn.restassuredtest.RestAssured_Assignment_002.PutCallReq
uest"/>

```

```

        </classes>
    </test>

    <test name="PetStorePUTTest_QA">
        <parameter name="environment" value="QA"/>
        <classes>
            <class
name="com.simplilearn.restassuredtest.RestAssured_Assignment_002.PutCallReq
uest"/>
        </classes>
    </test>

    <test name="PetStorePUTTest_PROD">
        <parameter name="environment" value="PROD"/>
        <classes>
            <class
name="com.simplilearn.restassuredtest.RestAssured_Assignment_002.PutCallReq
uest"/>
        </classes>
    </test>
</suite>

```

GetUserTest.java

```

package com.simplilearn.restassuredtest.RestAssured_Assignment_003;
import static org.hamcrest.CoreMatchers.equalTo;

import org.testng.annotations.Test;

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

public class GetUserTest {

    private static String baseUrl =
"https://petstore.swagger.io/v2/user/Username001";

    @Test

    public void getUserDetails() {
        RestAssured.given()
            .contentType(ContentType.JSON)
            .when()
            .get(baseUrl)
            .then()
            .statusCode(200)
            .body("username", equalTo("Username001"))
    }
}

```

```

        .body("email", equalTo("Positive@Attitude.com"))
        .body("userStatus", equalTo(1))
        .extract().response();
    }
}

```

LoginRequestTest.java

```

package com.simplilearn.restassuredtest.RestAssured_Assignment_004;
import static org.hamcrest.Matchers.*;

import org.testng.annotations.Test;

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

public class LoginRequestTest {

    private static String baseURL =
"https://petstore.swagger.io/v2/user/login";
    private static String username = "Uname001";
    private static String password = "@tt!tude";

    @Test(description="Test with login request")

    public void loginAndGetSession() {
        RestAssured.given()
            .auth()
            .basic(username, password)
            .contentType(ContentType.JSON)
            .when()
            .get(baseURL)
            .then()
            .statusCode(200)
            .body("code", equalTo(200))
            .body("message", containsString("logged in user session"))
            .extract().response();
    }
}

```

GetStatusTest.java


```

package com.simplilearn.restassuredtest.RestAssured_Assignment_005;

import org.hamcrest.Matchers;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

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

public class GetStatusTest {

    private static final String BASE_URL =
"https://petstore.swagger.io/v2/pet/findByStatus";

    @DataProvider(name = "statusProvider")
    public Object[][] statusProvider() {
        return new Object[][] {
            {"available"},
            {"pending"},
            {"sold"}
        };
    }

    @Test(dataProvider = "statusProvider")

    public void testPetStatus(String status) {
        Response response = RestAssured.given()
            .param("status", status)
            .when()
            .get(BASE_URL)
            .then()
            .statusCode(200)
            .contentType(ContentType.JSON)
            .extract()
            .response();

        // Validate that all pet details have the correct status
        response.then().assertThat().body("status",
Matchers.everyItem(Matchers.equalTo(status)));
    }
}

```

LogRequestTest.java

```

package com.simplilearn.restassuredtest.RestAssured_Assignment_006;

```

```

import org.testng.annotations.Test;

import io.restassured.RestAssured;
import static org.hamcrest.Matchers.*;

public class LogoutRequestTest {

    @Test(description="Test with logout scemario")

    public void testLogoutEndpoint() {

        RestAssured.given()
            .baseUrl("https://petstore.swagger.io/v2/user")
            .when()
                .get("/logout")
            .then()
                .assertThat()
                    .statusCode(200) // Validate response status code
                    .body("code", equalTo(200)) // Validate 'code' field in
the response
                    .body("message", equalTo("ok")); // Validate 'message'
field in the response
            }
    }
}

```

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>Phase3_EndProject_Restassured</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <name>Phase3_EndProject_Restassured</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>

```

```
        </plugins>
    </build>
</project>
```

TestNg.xml

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

<suite name="RestAssuredTest">
<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.RestAssured_Assignment_001.PostCallTe
st"></class>
        <class
name="com.simplilearn.restassuredtest.RestAssured_Assignment_001.GetCallTes
t"></class>
        <class
name="com.simplilearn.restassuredtest.RestAssured_Assignment_001.DeleteCall
Test"></class>

    </classes>
</test>
</suite>
```
