HW10

Using RestAssured to excursive GET (all), POST, PUT, GET (particular), DELETE API automation.

You can either subscribe https://github.com/james-willett/VideoGameDB to your local machine and run server, or select API from public websites, e.g. https://swagger.io/tools/swagger-ui/
https://demoqa.com/swagger/#/BookStore

1. Using template to fill in your test case details (excel);

2. Automate testing Rest APIs

3. Submit your source code and running results

- **RestAssuredAPIVideoGameTest_Hw10.java**

```java
package RestAPIProject;
import io.restassured.response.Response;
import org.apache.log4j.*;
import org.testng.Assert;
import org.testng.annotations.Test;

import java.util.HashMap;

import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.equalTo;

//This class will add new video game, edit the game & test and finally delete
public class RestAssuredAPIVideoGameTest_Hw10 {

    Logger logger = Logger.getLogger(RestAssuredAPIVideoGameTest_Hw10.class);
    // get All VideoGame
    @Test(priority = 1)
    public void test_getAllVideoGames() {
        //logger.info("add info");
        given()

                .when()
                .get("http://localhost:8080/app/videogames")
                .then()
                .statusCode(200);
    }
    // add a new VideoGame
    @Test(priority = 2)
    public void test_addNewVideoGame() {
        //logger.info("add info");
        HashMap data = new HashMap();
        data.put("id", "1000");
        data.put("name", "Minion");
```

```java
        data.put("releaseDate", "2022-12-04T07:54:39.518Z");
        data.put("reviewScore", "9");
        data.put("category", "Comedy");
        data.put("rating", "Sony Pictures");

        Response res =
                given()
                        .contentType("application/json")
                        .body(data)
                        .when()
                        .post("http://localhost:8080/app/videogames")
                        .then()
                        .statusCode(200)
                        .log().body()
                        .extract().response();

        String jsonString = res.asString();
        Assert.assertEquals(jsonString.contains("Record Added Successfully"),
true);


    }

    // get particular VideoGame
    @Test(priority = 3)
    public void test_getVideoGame() {
        given()

                .when()
                .get("http://localhost:8080/app/videogames/1000")

                .then()
                .statusCode(200)
                .log().body()
                .body("videoGame.id", equalTo("1000"))
                .body("videoGame.name", equalTo("Minion"));
    }
    // update existing VideoGame
    @Test(priority = 4)
    public void test_updatedVideoGame() {
        HashMap data = new HashMap();
        data.put("id", "1000");
        data.put("name", "Minion2");
        data.put("releaseDate", "2022-12-05T07:54:39.518Z");
        data.put("reviewScore", "7");
        data.put("category", "Comedy");
        data.put("rating", "Universal");

        given()
                .contentType("application/json")
                .body(data)
                .when()
                .put("http://localhost:8080/app/videogames/1000")

                .then()
```

```
                    .statusCode(200)
                    .log().body()
                    .body("videoGame.id", equalTo("1000"))
                    .body("videoGame.name", equalTo("Minion2"));
    }
    // delete a VideoGame
    @Test(priority = 5)
    public void test_DeleteVideoGame() {
        Response res =
                given()
                        .when()
                        .delete("http://localhost:8080/app/videogames/1000")
                        .then()
                        .statusCode(200)
                        .log().body()
                        .extract().response();
        String jsonString = res.asString();
        Assert.assertEquals(jsonString.contains("Record Deleted Successfully"),
true);

    }

}
```

- **Attached Log**