Handling the Test Data and executing it in Multiple Environment

Source code

GetRequestTest.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 GetUserRequestTest {


      private static final String BASE_URL = "https://reqres.in/api";
      private static final Logger logger =
Logger.getLogger(GetUserRequestTest.class);

      @Test(description = "Test for executing GET request using rest
assured")

      public void getRequestTest() {
            logger.info("Start :: GET All users test.");
            int page = 2 , total_pages =2, per_page=6,  total=12;

            logger.info("GET : URL "+BASE_URL + "/users?page="+page);

            RestAssured.given().baseUri(BASE_URL)
            .when().get("/users?page="+page).then().statusCode(200)
            .and()
            .assertThat()
            .body("page", equalTo(page)).and()
            .body("total", equalTo(total)).and()
            .body("per_page", equalTo(per_page)).and()
            .body("total_pages", equalTo(total_pages)).and()
            .body("data[0].first_name", equalTo("Michael"));

            String response =
RestAssured.given().baseUri(BASE_URL).queryParam ("page",page)
            .when().get("/users?page="+page).getBody().asString();

            logger.info("Reponse :: "+response);
            logger.info("End :: GET All users test.");
      }

      @Test(description = "Test for executing GET request using query
parms")
```

```
      public void getRequestWithQueryParmsTest() {
            logger.info("Start :: GET All users test with query parms.");

            int page = 2 , total_pages =2, per_page=6,  total=12;

             String response =
RestAssured.given().baseUri(BASE_URL).queryParam ("page",page)
            .when().get("/users").getBody().asString();

             /** System.out.println(response);  **/
            logger.info("GET : URL "+BASE_URL + "/users?page="+page);

            RestAssured.given().baseUri(BASE_URL).queryParam ("page",page)
            .when().get("/users").then().statusCode(200)
            .and()
            .assertThat()
            .body("page", equalTo(page)).and()
            .body("total", equalTo(total)).and()
            .body("per_page", equalTo(per_page)).and()
            .body("total_pages", equalTo(total_pages)).and()
            .body("data[0].first_name", equalTo("Michael"))
            .body("data[0].last_name", equalTo("Lawson"))
            .body("data[0].email", equalTo("michael.lawson@reqres.in"));

            logger.info("Reponse :: "+response);
            logger.info("End :: GET All users test with query parms.");
      }

      @Test(description = "Test for executing GET /users request with
userId")

      public void getRequestWithUserIdTest() {
            logger.info("Start :: GET user by {userId} test.");
            int userId = 7 ;

            String response = RestAssured.given().baseUri(BASE_URL)
            .when().get("/users/"+userId).getBody().asString();
            logger.info("GET : URL "+BASE_URL + "/users/"+userId);

            RestAssured.given().baseUri(BASE_URL)
            .when().get("/users/"+userId).then().statusCode(200)
            .and()
            .assertThat()
            .body("data.first_name", equalTo("Michael")).and()
            .body("data.last_name", equalTo("Lawson")).and()
            .body("data.email", equalTo("michael.lawson@reqres.in"));

            logger.info("Reponse :: "+response);
            logger.info("End :: GET user by {userId} test.");
      }
}
```

LoginRequestTest.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 LoginRequestTest {
      private static final String BASE_URL = "https://reqres.in/api";
    private static final Logger logger =
Logger.getLogger(LoginRequestTest.class);

    @Test(description = "Test on Login request with Rest assured")

    public void testOnLoginRequest() {

        logger.info("Start :: Test on Login request with Rest assured");
        String response = null;
        try {

            response =
RestAssured.given().baseUri(BASE_URL).contentType(ContentType.JSON)
                    .body("{ \"email\": \"eve.holt@reqres.in\",
\"password\": \"pistol\" }").when().post("/login")
                    .then().statusCode(200).and()
                    .body("token", equalTo("QpwL5tke4Pnpja7X4"))
                    .extract().asString();
        } catch (Exception e) {
            logger.error("Exception Object :: " + e.toString());
            logger.error("End Exception :: " + e.getLocalizedMessage());
        }
        logger.info("Response Object ::" + response);
        logger.info("End :: Test on Login request with Rest assured");
    }
}
```

RegisterRequestTest.java

```
package com.simplilearn.restassuredtest;

import static org.hamcrest.CoreMatchers.notNullValue;

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



public class RegisterRequestTest {
	private static final String BASE_URL = "https://reqres.in/api";
	private static final Logger logger = 
Logger.getLogger(RegisterRequestTest.class);

	@Test(description = "Test Registration  with rest assured")

	public void testresgistrationWithToken() {

		logger.info("Start :: Test Registration  with rest assured");
		logger.info("POST :: URL "+ BASE_URL+"/register");
	User user=new User("eve.holt@reqres.in","pistol");
	logger.info("Request Object :: "+user);
	RestAssured.given().baseUri(BASE_URL).when()
	.contentType(ContentType.JSON)
	.body(user)
	.log().uri()
	.post("/register").then()
	.log().body()  // response logs
	.assertThat().statusCode(200).and()
	.assertThat().body ("id", notNullValue()).and()
	.assertThat().body ("token", notNullValue());

	String response = RestAssured.given().baseUri(BASE_URL).when()
	.contentType(ContentType.JSON)
	.body(user)
	.post("/register").getBody().asString();

	logger.info("Response Object:: "+response);
	logger.info("End :: Test Registration  with rest assured");;

}
}
class User {

	public String email;
	public String password;

	public User(String email, String password) {
		super();
		this.email = email;
		this.password = password;
	}

	@Override
	public String toString() {
		return "User [email=" + email + ", password=" + password + "]";
	}
}
```

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");
      }
}
```