**RestAssured**

REST Api or RESTful Api is Representational State Transfer.

* RestAssured is an Java library that is designed to test/automate REST api. It is available in jar files.
* It supports only REST services and can’t support SOAP services.

For installation <https://www.youtube.com/watch?v=yDdBOspPp_c&t=844s>

* For RestAssured testing, we use Maven project as it downloads all the necessary jar files by just mentioning them in dependencies of pom.xml file.
* Dependencies that we need to add in pom.xml for Rest api testing.
  + RestAssured
  + TestNG
  + Json-simple
  + Apache POI
* All these dependencies (code) can be downloaded from maven repository official site.
* APIs are available in the form of JAR files.

For requesting data from server. GET response 🡪

**package** asd;

**import** org.testng.annotations.Test;

**import** io.restassured.RestAssured;

**import** io.restassured.response.Response;

**import** io.restassured.specification.RequestSpecification;

**import** junit.framework.Assert;

**public** **class** getRe {

@Test

**public** **void** weatherData() {

// Request object creation

RequestSpecification re = RestAssured.*given*();

// Response object

Response response = re.get("http://restapi.demoqa.com/utilities/weather/city/Pune");

// Printing response

String responseBody = response.getBody().asString();

System.***out***.println(responseBody);

}

}

Steps for any RestAssured API testing 🡪

* Create request object (RequestSpecification)
* Create response object
* Extract response body from it

For posting on server, POST request 🡪

RequestSpecification re1 = RestAssured.*given*();

re1.header("Content-Type", "application/json");

JSONObject json = **new** JSONObject();

json.put("id", "25");

json.put("title", "VJ");

json.put("author", "Vijay");

re1.body(json.toJSONString());

re1.post("http://localhost:3000/posts");

For deleting something from server, DELETE request 🡪

RequestSpecification re2 = RestAssured.*given*();

re2.delete("http://localhost:3000/posts/21");//need to provide exact URL

For modifying existing data, PUT request (same as POST just write PUT request with modified data)🡪

RequestSpecification re3 = RestAssured.*given*();

re1.header("Content-Type", "application/json");

JSONObject json = **new** JSONObject();

json.put("id", "25");

json.put("title", "VJ");

json.put("author", "Vijay");

re3.body(json.toJSONString());

re3.put("http://localhost:3000/posts/25");

Different Validation points in RestAssured 🡪

* Status code validation
* Status line validation
* Validating Headers
* Success code validation
* Validating node values from JSON response
* Basic Preemptive Authentication

Note – For more details, please check compete programs

RequestSpecification re = RestAssured.*given*();

Response response = re.get("http://restapi.demoqa.com/utilities/weather/city/Pune");

// Status code validation

**int** statusCode = response.getStatusCode();

System.***out***.println("Status code is " + statusCode);

Assert.*assertEquals*(200, statusCode);

// Status Line validation

String statusLine = response.getStatusLine();

System.***out***.println("Status line is " + statusLine);

Assert.*assertEquals*("HTTP/1.1 200 OK", statusLine);

// Success Code validation

String successCode = response.jsonPath().get("successCode");

Assert.*assertEquals*("OPERATION\_SUCCESS", successCode);

// Validating Headers

String contentType = response.header("Content-Type"); // ‘Content-Type’ is one of the headers

Assert.*assertEquals*("application/xml; charset=UTF-8", contentType);

// Validating node value

JsonPath jsonpath = response.jsonPath();

Assert.*assertEquals*(jsonpath.get("Temperature"), "31 Degree celsius");

// Basic Authentication

PreemptiveBasicAuthScheme authscheme = **new** PreemptiveBasicAuthScheme();

authscheme.setUserName("ToolsQA");

authscheme.setPassword("TestPassword");

RestAssured.*authentication*=authscheme;

**Authentication in REST\_Api**

* Authentication is a process to prove that you are authentic person or not.
* Rest Assured supports several authentication schemes like OAuth, digest, certificate, form and preemptive basic authentication.
* Basic Authentication can be classified as Preemptive and Challenged.
* In Preemptive Authentication, along with login request credentials (username and password) are also send. So it reduces one extra call from server.
* This is not the case with Challenged Authentication.
* Mostly Preemptive Authentication is used.

@Test

**public** **void** test5() {

**int** code = RestAssured.*given*()

.auth().preemptive()

.basic("ToolsQA", "TestPassword")

.when()

.get("http://restapi.demoqa.com/authentication/CheckForAuthentication")

.getStatusCode();

System.***out***.println("Response code from server is " + code);

}