* 1. **What is Rest Assured?**
  2. *Rest-Assured is a Java based library that is used to test RESTful Web Services. This library behaves like a headless Client to access REST web services*
  3. Rest Assured simple java libraries for testing of Rest WebServcie API
  4. *Rest-Assured library also provides ability to validate the Rest-HTTP Responses received from server. For e.g. we can verify the Status code, Status message, Headers and even the Body of the response. This makes Rest-Assured a very flexible library that can be used for testing.*
  5. It support both XML and JSON, JS, html, text

1. **Why RestAssured is very Popular**

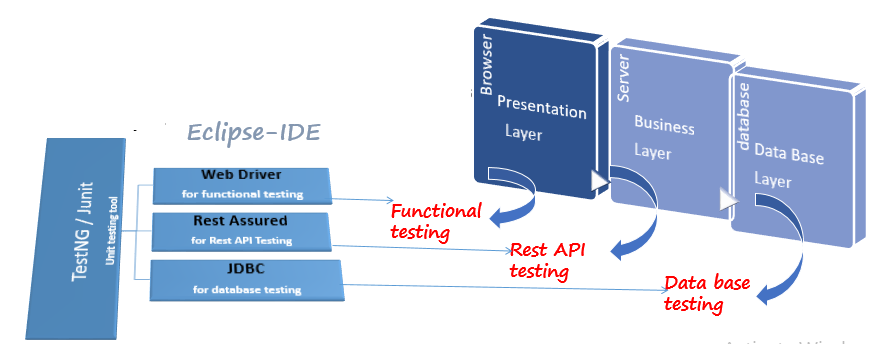
* Integrate seamlessly with existing java based framework like

Testing/junit/BDD

Selenium Webdriver

JDBC

* We can Automate E2E business workflow which includes

 all the layers

1. **What all the Class/ API used in RestAssured?**

RestAssured: it’s a main class, where we can perform CRUD operation without precondition

Request Specification: it’s used to set precondition for CRUD request

Response: all crud operation return the object of Response, which is used for capture response header & body

Response Validator: it’s used to validate the response using assertion statements & also we log the response

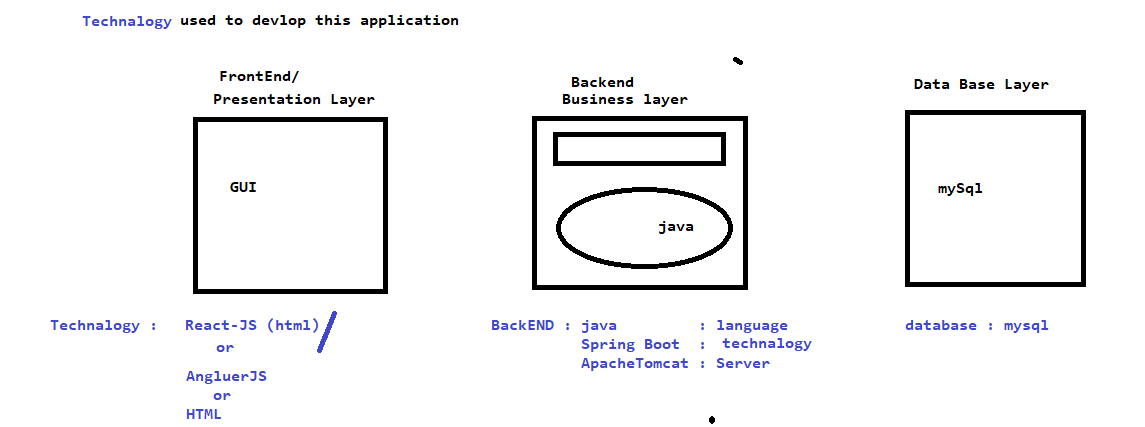
ResponseSpecBuilder:

RequestSpecBuilder :

1. Rest Assured Required libraries?

|  |  |
| --- | --- |
| **Tool** | **Description** |
| rest-assured-3.0.2 | Rest api headless client |
| json-simple-1.1.1 | Used to create Json object |
| jackson-mapper-asl-1.9.13 | Used to create pojo object & parser |
| testng-6.8 | Unit testing tool for batch execution , reports |

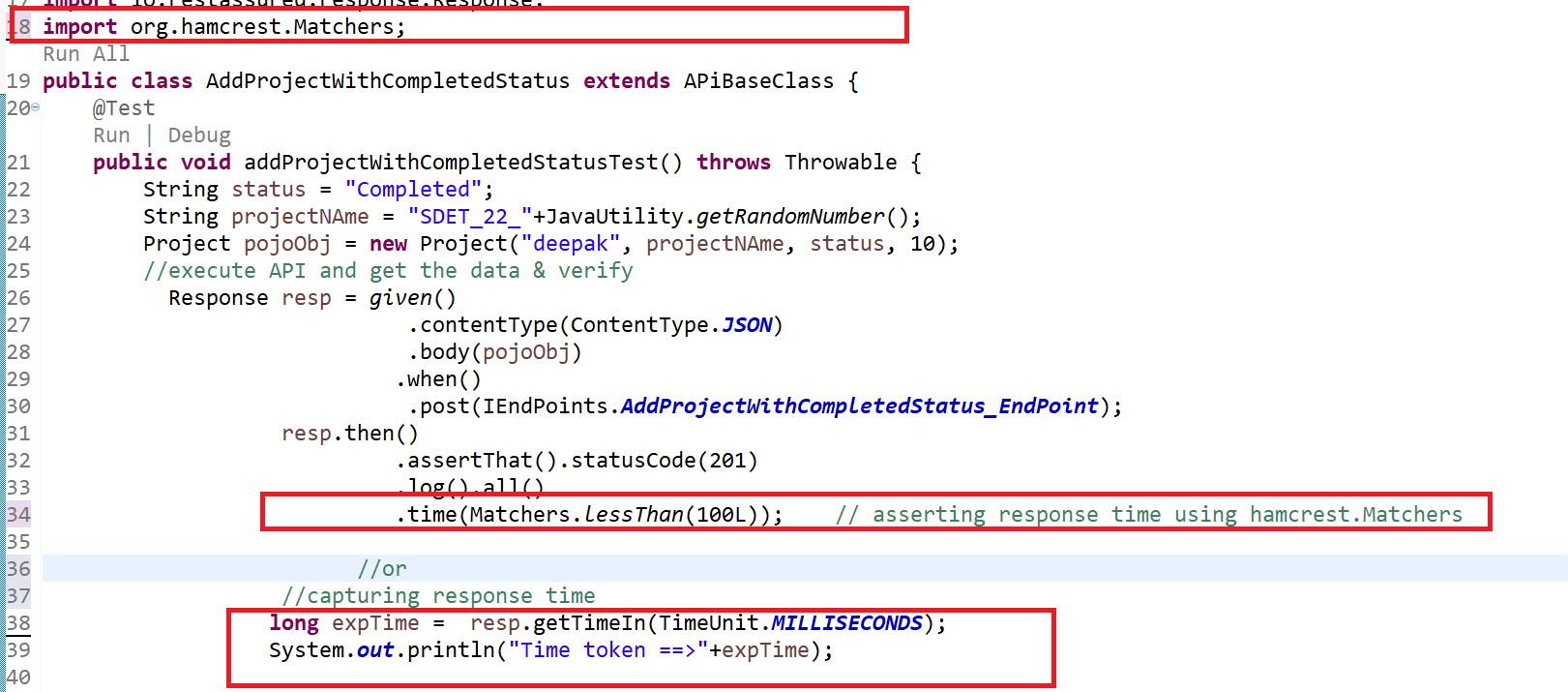
1. **What technology your application built on/ what is your application stack**

****

1. **What is the Advantages of restAssured ?**

* Support for http methods
* Support for BDD
* Use of Hamcrest matches for validation /Assertions
* Provide inbuilt method to create a request header & body
* Provide inbuilt method to validate response header & body
* Handles various authentication like Token AUTH, Basic auth , oauth1.0 , oauth2.0
* Send request over the network using existing protocol like http
* Integrate seamlessly with existing java based framework like selenium, testing , jdbc , poi etc
* Used for complete backend automation
* Framework can integrate with CI/CD pipeline
* Open source headless client

1. **How do you verify Response time in RestAssured ?**

****

1. **WAP to create POST request using rest Assured ?**

@Test

**public** **void** getallProjectsTest() {

JSONObject obj = **new** JSONObject();

obj.put("createdBy", "Manoj3");

obj.put("projectName", "SCAnamiras");

obj.put("status", "On Going");

obj.put("teamSize", 12);

*given*()

.contentType(ContentType.***JSON***)

.body(obj)

.when()

.put("http://localhost:8084/projects/TY\_PROJ\_1603")

.then()

.log().all()

.assertThat().statusCode(200);

}

OR

@Test

**public** **void** getallProjectsTest() {

HashMap obj = **new** HashMap ();

obj.put("createdBy", "Manoj3");

obj.put("projectName", "SCAnamiras");

obj.put("status", "On Going");

obj.put("teamSize", 12);

*given*()

.contentType(ContentType.***JSON***)

.body(obj)

.when()

.put("http://localhost:8084/projects/TY\_PROJ\_1603")

.then()

.log().all()

.assertThat().statusCode(200);

}

Or

@Test

**public** **void** sampleTest() {

File fis = **new** File("./sampleProject.json");

*given*()

.contentType(ContentType.***JSON***)

.body(fis)

.when()

.post("http://localhost:8084/addProject")

.then()

.log().all()

.assertThat().statusCode(201);

} OR

@Test

**public** **void** sampleTest() {

Random ram = **new** Random();

**int** randomNum = ram.nextInt(1000);

Project obj = **new** Project("Muni\_"+randomNum, "Swagger", "Completed", 10);

*given*()

.contentType(ContentType.***JSON***)

.body(obj, ObjectMapperType.***JACKSON\_1***)

.when()

.post("http://localhost:8084/addProject")

.then()

.log().all()

.assertThat().statusCode(201);

}

====================Pojo class====================

**package** com.rmgyantra.api.pojoclass;

**public** **class** Project {

String createdBy;

String projectName;

String status;

**int** teamSize;

**public** Project(String createdBy, String projectName, String status, **int** teamSize) {

**this**.createdBy = createdBy;

**this**.projectName = projectName;

**this**.status = status;

**this**.teamSize = teamSize;

}

**public** String getCreatedBy() {

**return** createdBy;

}

**public** **void** setCreatedBy(String createdBy) {

**this**.createdBy = createdBy;

}

**public** String getProjectName() {

**return** projectName;

}

**public** **void** setProjectName(String projectName) {

**this**.projectName = projectName;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** **int** getTeamSize() {

**return** teamSize;

}

**public** **void** setTeamSize(**int** teamSize) {

**this**.teamSize = teamSize;

}

Project(){}

}

1. **How to post Complex Request in Rest Assured?**

@Test

**public** **void** sampleTest() {

File fis = **new** File("./sampleProject.json");

*given*()

.contentType(ContentType.***JSON***)

.body(fis)

.when()

.post("http://localhost:8084/addProject")

.then()

.log().all()

.assertThat().statusCode(201);

}

1. **What is Pojo class in Rest API ?**

Pojo class is a java service engine where Rest API properties are defined making use of annotations, using pojo class we can achieve sterilization & dieselization(Parsing)

1. How any ways we can post a Request in RestAssured ?
2. Using JSONObject class
3. Using HashMap
4. Using POJO class
5. Using JSONFile
6. How to Post Complex Request body in Rest Assured?
7. Using POJO class
8. Using JSONFile
9. How to Validate Complex Response in Rest Assured

Using JSONPATH() we can write jsonXpath to navigate to specific node in Response , it will written Object if single match else return List<Object> if multiple match found

Object obj = Resp.jsonPath(JsonXpath) : if one matching found

List<Object> list = Resp.jsonPath(JsonXpath) : if multiple matching found

1. **Write a Program to check Specific Data available in complex Response**

**package** practice;

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

**import** com.rmgyantra.api.genericUtils.BaseAPIClass;

**import** io.restassured.http.ContentType;

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

**import** **static** io.restassured.RestAssured.\*;

**import** java.util.ArrayList;

**public** **class** GetAndFindProject {

@Test

**public** **void** getAndVerifyProject() {

Response resp = *given*()

.contentType(ContentType.***JSON***)

.when()

.get("http://localhost:8084/projects");

resp.then()

.assertThat().statusCode(200);

ArrayList<String> lst = resp.jsonPath().get("projectName");

Boolean flag = false;

**for**(**int** i=0; i<lst.size();i++) {

**if**(lst.get(i).equals("HDFC")) {

System.***out***.println("PASS");

Flag = true;

**break**;

}

}

Assert.assertTrue(flag);

}

}

1. **What type of Authentication you have used in previous Project ?**

**We have used Bearer Token & Oauth2 Autehtication**

@Test

**public** **void** getAllProject() {

given()

.auth().oauth2("BearerTokenID")

.contentType(ContentType.JSON)

.when()

.get("http://localhost:8084/projects")

.then()

.log().all()

}

OR

Basic Auth

*given*()

.auth().basic("username", "password")

.contentType(ContentType.***JSON***)

.body(fis)

.when()

.post("http://localhost:8084/addProject")

.then()

.log().all()

.assertThat().statusCode(201);

}

1. **Can you write a program to POST a request?**

**Import static io.restassured.RestAussured.\*;**

**Class CreateCustomer{**

**@test**

**Public void createCustomer(){**

**HashMAp map = new HAshMAp()**

**map.put(“customerNAme” , “hdfc”);**

**map.put(“add”, “bangalore”);**

**map.put(“id” , 123);**

**given()**

**.contentType(ContentType.JSON)**

**.auth().oath2(“barereTokenID”)**

**.body(map)**

**.when()**

**.post(“http://rmgyantra.com/projects”)**

**.then()**

**.assertThat().statusCode(200)**

**}**

**}**

1. **How execute same API multiple times with Different data**

**public** **class** CreateProject\_BDD **extends** BaseAPIClass{

@Test(dataProvider = "getData")

**public** **void** createPRoject(String prjectNAme , String status) {

Project pObj = **new** Project("deepak", prjectNAme, status, 10);

*given*()

.spec(requestSpec)

.body(pObj)

.when()

.post("/addProject")

.then()

.assertThat().statusCode(201)

.log().all();

}

@DataProvider

**public** Object[][] getData() {

Object[][] objArr = **new** Object[4][2];

objArr[0][0] = "HDFC-11";

objArr[0][1] = "Completed";

objArr[1][0] = "ICICI";

objArr[1][1] = "Completed";

objArr[2][0] = "YES";

objArr[2][1] = "Completed";

objArr[3][0] = "SMB";

objArr[3][1] = "Completed";

**return** objArr;

}

}

1. **Types of parameter available in RestAssured ?**
2. **Path parameter**
3. **Query Parameter**
4. **Form parameter**
5. **Param parameter**
6. **What is Request Chaining?**

**Get the data from one response & pass it to another Request**

**package** practice;

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

**import** com.rmgyantra.api.genericUtils.BaseAPIClass;

**import** io.restassured.http.ContentType;

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

**import** **static** io.restassured.RestAssured.\*;

**import** java.util.ArrayList;

**public** **class** GetAndFindProject {

@Test

**public** **void** getAndVerifyProject() {

Response resp = *given*()

.contentType(ContentType.***JSON***)

.when()

.get("http://localhost:8084/projects");

String projectId = resp.jsonPath().get("projectId[0]");

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

*given*()

.contentType(ContentType.***JSON***)

.pathParam("PID", projectId)

.when()

.delete("http://localhost:8084/projects/{PID}")

.then()

.assertThat().statusCode(204);

}

}