**SoapUi Project**

**SBE CODING (done)**

* **GO TO LOGIN UNDER SBE AND CLICK ON ASSERTION(GET AND STORE RESPONSE)**

import com.eviware.soapui.support.XmlHolder

//i am going to write a groovy script to capture the response and fetch the sessionID and stor it in test case/test suite level

//i have cretead a varibale called response

//i want to fetch the entire response and store it in this variable

def response = context.testCase.getTestStepByName('Login').getPropertyValue("Response")

//holder

//i want to store this entire response in a holder

//so that i can get a specify data and storeit again

def holder = new XmlHolder(response)

//get node value is used when i want to fetch a specific response data

def sess = holder.getNodeValue("//\*:sessionId")

//i want to store this session id in test suite level sothat i can use it in all my test case

context.testCase.testSuite.setPropertyValue("sessionId",sess)

//can i store it in the test case

context.testCase.setPropertyValue("sessionId",sess)

//i want to access this acroos project level

context.testCase.testSuite.project.setPropertyValue("sessionId",sess)

log.info sess

**2) SYSTEM TESTING CODING PART**

import com.eviware.soapui.model.testsuite.TestRunner.Status

//From here i want to execute Script Based Execution

//if i want to run script based execution from here i have to use Test Runner

//coz it is in different test suite

def tc = testRunner.testCase.testSuite.project.testSuites["Shopping Service Functional Test"].testCases["Script Based Execution"]

tc.testSteps["username and password properties"].setPropertyValue("username","Aishwarya")

tc.testSteps["username and password properties"].setPropertyValue("password","Joush12")

//i am running the entire test case by changing the data

//if you are running more then one step in testcase

//you have to say null means there is no context and false is coz i am not checking any condition

tc.run(null,false)

//i want to check what all the steps got executed

def teststeplist = tc.getTestStepList()

log.info teststeplist.size()

//log.info TSL

teststeplist.each

{

def tsname = it.name

log.info tsname

//fetch the currect step which i am executing

log.info tc.getCurrentStep().name

}

**3) PELL GROOVY script based code for TO PURCHASE OR NOT TO PURCHASE.**

//if the user is Aishwarya do the purchase

//first step to fetch the username

def username = context.expand('${Login#Request#//\*:username}')

if(username!="Aishwarya")

{

log.info "Not a Previlaged user -No purchase"

//if i want to skip any step then i do this

testRunner.gotoStepByName("Logout")

}

else

{

log.info "processd to purchase"

}

**4) Execution the property expansion login and logout**

def tc = testRunner.testCase.testSuite.project.testSuites["Shopping Service Functional Test"].testCases["Property Expansion Login and Logout"]

tc.testSteps["username and password properties"].setPropertyValue("username","Aishwarya")

tc.testSteps["username and password properties"].setPropertyValue("password","Joush12")

//this line will execute the entire testcase

def runner = tc.run(null,false)

def teststeplist = tc.getTestStepList()

//log.info TSL

//i want see the names of the my test steps

teststeplist.each

{

def tsname = it.name

log.info tsname

//fetch the currect step which i am executing

//log.info tc.getCurrentStep().name

}

//i want to know the status of my execution

//this will provide me the execution status

log.info "Status: $runner.status "

//This will provide me the time taken forexecution

log.info "TimeTaken: $runner.timeTaken"

//i want to know the number of steps executed

log.info "Number of Steps executed is : "+runner.results.size()

//Assert

assert runner.results.size()>=5

**REST PROJECT**

**SBE BY CREATING GROOVY SCRIPT( JSON PATH)**

import groovy.json.JsonSlurper

def response = context.testCase.getTestStepByName('Add').getPropertyValue("Response")

//holder

//i want to store this entire response in a holder

//so that i can get a specify data and storeit again

def holder = new JsonSlurper().parseText(response)

//when you are using json we have to provide object name after holder along with.

def placeid = holder.place\_id

//storing it in test suite

context.testCase.testSuite.setPropertyValue("placeid",placeid)

**SBE BY CREATING GROOVY SCRIPT(XML PATH)**

import com.eviware.soapui.support.XmlHolder

def response = context.testCase.getTestStepByName("Add").getPropertyValue("Response")

//holder

//i want to store this entire response in a holder

//so that i can get a specify data and storeit again

def holder = new XmlHolder(response)

//get node value is used when i want to fetch a specific response data

def sess = holder.getNodeValue("//place\_id")

context.testCase.testSuite.setPropertyValue("placeid",sess)

**SBE FOR REST PROJECT 2 (USER)**

import groovy.json.JsonSlurper

def response = context.testCase.getTestStepByName('Login').getPropertyValue("Response")

//holder

//i want to store this entire response in a holder

//so that i can get a specify data and storeit again

def holder = new JsonSlurper().parseText(response)

//when you are using json we have to provide object name after holder along with.

def token = holder.accessToken

//storing it in test suite

context.testCase.testSuite.setPropertyValue("accesstoken","bearer " +token)

**[Note: In code give space between “bearer space” space +token coz here we are getting the authorization(header value) as access token in create user step.**

**In header🡪 give authorization 🡪 in value give ${#TestSuite#accesstoken}**

**Check your custom properties and wat ever name u have given der give after test suite… here I have given accesstoken]**

**SCRIPT ASSERTION AFTER DOIN JSON MATCH CONFIGURATIONS AND ALL IN GET ALL VALUES UNDER SBE**

1. **TO KNOW THE COUNT OF USERS**

import groovy.json.JsonSlurper  
def response = context.testCase.getTestStepByName("getalluser").getPropertyValue("response")def holder = new JsonSlurper().parseText(response)def count = holder.count.toInteger()

log.info count

1. **TO ITERATE THE ID FROM THE GIVE EMAIL ID(EXPLAINATION IN NOTES)**

import groovy.json.JsonSlurper

def response = context.testCase.getTestStepByName("Get all users by id").getPropertyValue("response")

def holder = new JsonSlurper().parseText(response)

def count = holder.count.toInteger()

for(def i=0;i<count;i++)

{

if(holder.users[i].email=="aditi@gmail.com")

{

def id = holder.users[i].\_id

log.info id

def id1 = id.toString()

context.testCase.testSuite.setPropertyValue("id",id1)

}

}

**TO DELECT THAT ID (DONE IN SCRIPT ASSERTION IN DELECT REST FROM SBE)**

// We need to input the library to read json files  
import groovy.json.JsonSlurper  
//we need to store the response  
def response = context.testCase.getTestStepByName("Get All Users").getPropertyValue("Response")  
//Stripping the response into a json script  
def holder = new JsonSlurper().parseText(response)  
// getting only the count  
// def count = holder.count  
def id = holder.users[200].\_id  
context.testCase.testSuite.setPropertyValue("id", id)  
// holding the count  
/\*for(def i=0; i< count; i++){  
if(holder.users[i].email == '[sachi@gmail.com](mailto:sachi@gmail.com)'){  
def id = holder.users[i].\_id  
[log.info](http://log.info/) id  
def idle = id.toString()  
context.testCase.testSuite.setPropertyValue("id", idle)  
}  
}  
\*/

**REST 3 PROJECT (PRODUCT) ) [SCRIPT ASSERTION IS WRITTEN IN CREATE PRODUCT UNDER SBE AS WE ARE FETECHING DATA FROM CREATE PRODUCTS AND STORING IT IN RESPOSE AND HOLDER AND TEST SUITE LEVEL]**

import groovy.json.JsonSlurper

def response = context.testCase.getTestStepByName('create product').getPropertyValue("Response")

//holder

//i want to store this entire response in a holder

//so that i can get a specify data and storeit again

def holder = new JsonSlurper().parseText(response)

//when you are using json we have to provide object name after holder along with.

def id1 = holder.product.\_id

//storing it in test suite

context.testCase.testSuite.setPropertyValue("id",id1)

log.info id1

**NOTE: HERE NO NEED TO PUT “ bearer ” +token in the 5th line coz we are giving the authorization(header value) from the previous rest project (rest 2(user) project) to all our services and test steps.**

**REST PROJECT 4 (ORDERS) [SCRIPT ASSERTION IS WRITTEN IN CREATE ORDERS UNDER SBE, AS WE ARE FETECHING DATA FROM CREATE ORDERS AND STORING IT IN RESPOSE AND HOLDER AND TEST SUITE LEVEL]**

import groovy.json.JsonSlurper

def response = context.testCase.getTestStepByName('create order').getPropertyValue("Response")

//holder

//i want to store this entire response in a holder

//so that i can get a specify data and storeit again

def holder = new JsonSlurper().parseText(response)

//when you are using json we have to provide object name after holder along with.

def id1 = holder.order.\_id

//storing it in test suite

context.testCase.testSuite.setPropertyValue("id",id1)

log.info id1

**NOTE: HERE NO NEED TO PUT “ bearer ” +token in the 5th line coz we are giving the authorization(header value) from the previous rest project (rest 2(user) project) to all our services and test steps.**

**NOTE FOR REST PROJECTS: ONCE U CREATE SBE, THEN CREATE AN ASSERTION 🡪 COMPLAINES🡪 VALID HTTP🡪 ADD🡪 GIVEN 200 OR 201(COZ THE STATUES IS OK FOR 200) FOR EACH OF UR TEST STEPS UNDER SBE**

**REST ASSURED PROJECTS IN ECLIPSE IDE**

1. **WE ARE TESTING THE TRELLO BOARD BY CREATING A BOARD IN TRELLO FROM REST ASSURED PROJECT IN ECLIPSE.**

**(HERE WE ARE TAKING THE QUERY PARAMS AND BASEURL FROM POSTMAN TRELLO PROJECT)**

**package** RestTesting;

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

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

**import** io.restassured.RestAssured;

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

**public** **class** Trello

{

**public** **static** String *baseurl* ="https://api.trello.com";

@Test

**public** **void** createaboard()

{

RestAssured.*baseURI* = *baseurl*;

*given*().queryParam("name", "Divyaboard")

.queryParam("key", "b39219fe7dbd1d6144c6700033e3b9a5")

.queryParam("token", "475d47aa08cebeb6bdef7da62aecf29ba71113f671b5752437f0ab28744f2c62")

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

.when()

.post("/1/boards/")

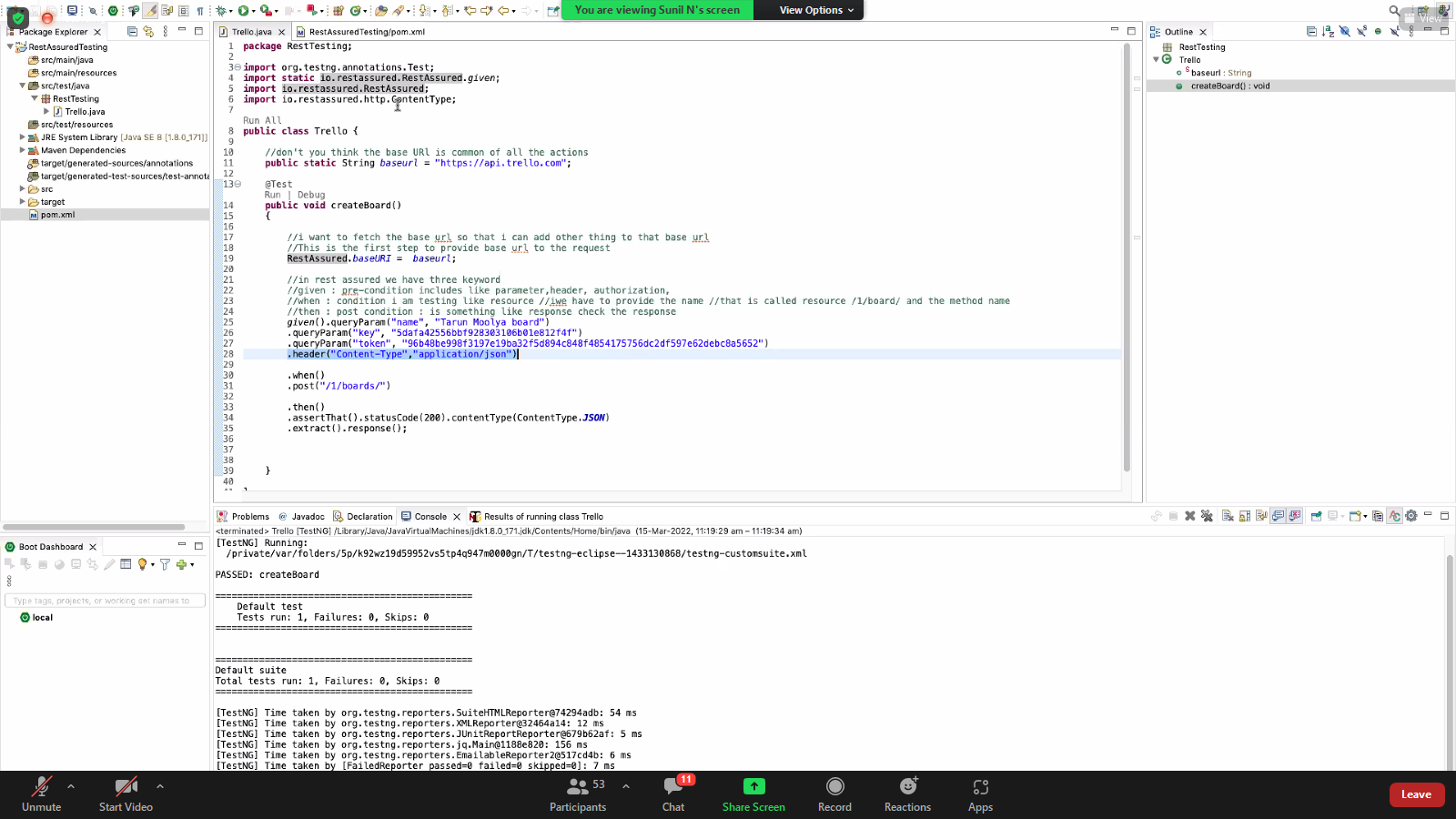
.then()

.assertThat().statusCode(200).contentType(ContentType.***JSON***)

.extract().response();

}}

**PICTURE FOR THE 1ST CODE (SUNIL’S SCREENSHOT)**

****

1. **WE ARE TESTING THE TRELLO BOARD WHICH WE INTIALLY CREATING USING POSTMAN AS A REST ASSURED PROJECT IN ECLIPSE.**

**Here we are creating a board, getting the board and deleting the board, HERE ALSO WE ARE TAKING THE QUERY PARAMS AND BASEURL FROM POSTMAN TRELLO PROJECT**

**package** RestTesting;

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

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

**import** io.restassured.RestAssured;

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

**import** io.restassured.path.json.JsonPath;

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

**public** **class** Trello

{

**public** **static** String *baseurl* = "https://api.trello.com";

**public** **static** String *id*;

@Test(priority = 0)

**public** **void** createBoard()

{

//i want to fetch the base url so that i can add other thing to that base url

//This is the first step to provide base url to the request

RestAssured.*baseURI* = *baseurl*;

//in rest assured we have three keyword

//given : pre-condition includes like parameter,header, authorization,

//when : condition i am testing like resource //iwe have to provide the name //that is called resource /1/board/ and the method name

//then : post condition : is something like response check the response

Response response = *given*().queryParam("name", "Divyaboard")

.queryParam("key", "b39219fe7dbd1d6144c6700033e3b9a5")

.queryParam("token", "475d47aa08cebeb6bdef7da62aecf29ba71113f671b5752437f0ab28744f2c62")

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

.when()

.post("/1/boards/")

.then()

.assertThat().statusCode(200).contentType(ContentType.***JSON***)

.extract().response();

//This is to fetch the response as a string

String jsonresponse = response.asString();

//i want to convert the response in to json format

//why do i use jsonpath to convert the string in to a json format

JsonPath js = **new** JsonPath(jsonresponse);

//nw i have to fetch the id

*id* = js.get("id");

}

//if i make any @Test method enabled =false/ that method will not execute

@Test(priority = 1)

**public** **void** getBoard()

{

RestAssured.*baseURI* = *baseurl*;

Response response = *given*().queryParam("name", "Divyaboard")

.queryParam("key", "b39219fe7dbd1d6144c6700033e3b9a5")

.queryParam("token", "475d47aa08cebeb6bdef7da62aecf29ba71113f671b5752437f0ab28744f2c62")

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

.when()

.get("/1/boards/3YtTVHAS")

.then()

.assertThat().statusCode(200).contentType(ContentType.***JSON***)

.extract().response();

String jsonresponse = response.asString();

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

}

@Test(priority = 2)

**public** **void** DeleteBoard()

{

RestAssured.*baseURI* = *baseurl*;

Response response = *given*().queryParam("name", "Divyaboard")

.queryParam("key", "b39219fe7dbd1d6144c6700033e3b9a5")

.queryParam("token", "475d47aa08cebeb6bdef7da62aecf29ba71113f671b5752437f0ab28744f2c62")

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

.when()

.delete("/1/boards/3YtTVHAS")

.then()

.assertThat().statusCode(200).contentType(ContentType.***JSON***)

.extract().response();

String jsonresponse = response.asString();

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

}

}

1. **WE ARE TESTING THE ECOMMERCE(USERS)(FOLDER GIVEN BY SUNIL) WHICH WE INTIALLY CREATING USING POSTMAN AS A REST ASSURED PROJECT IN ECLIPSE.**

**package** RestTesting;

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

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

**import** io.restassured.RestAssured;

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

**import** io.restassured.path.json.JsonPath;

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

**public** **class** EcommerceUser

{

**public** **static** String *baseurl* = "https://ecommerceservice.herokuapp.com/";

**public** **static** String *message*;

**public** **static** String *accessToken*;

@Test(priority = 0)

**public** **void** signup()

{

RestAssured.*baseURI* = *baseurl*;

String requestbody = "{\r\n"

+ " \"email\": \"krishnakumar176@gmail.com\",\r\n"

+ " \"password\": \"krishna@123\"\r\n"

+ "}\r\n"

+ "";

Response response = *given*()

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

.body(requestbody)

.when()

.post("/user/signup")

.then()

.assertThat().statusCode(201).contentType(ContentType.***JSON***)

.extract().response();

String jsonresponse = response.asString();

//i want to convert the response in to json format

//why do i use jsonpath to convert the string in to a json format

JsonPath js = **new** JsonPath(jsonresponse);

//nw i have to fetch the id

*message* = js.get("message");

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

}

@Test(priority = 1)

**public** **void** Login()

{

RestAssured.*baseURI* =*baseurl*;

String requestbody = "{\r\n"

+ " \"email\": \"krishnakumar176@gmail.com\",\r\n"

+ " \"password\": \"krishna@123\"\r\n"

+ "}\r\n"

+ "";

Response resposne = *given*()

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

.body(requestbody)

.when()

.post("/user/login")

.then()

.assertThat().statusCode(200).contentType(ContentType.***JSON***)

.extract().response();

String jsonresponse = resposne.asString();

//i want to convert the response in to json format

//why do i use jsonpath to convert the string in to a json format

JsonPath js = **new** JsonPath(jsonresponse);

//nw i have to fetch the id

*accessToken* = js.get("accessToken");

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

}}

**DIFFERENCE BETWEEN JUNIT AND TESNG TESTING FRAMEWORKS**

**1)TESTNG**

@BeforeTest  
**public** **void** BT()  
{  
String Name = "Sunil";  
String SName = "Nishchith";  
Assert.assertEquals(SName, Name);  
}  
  
@Test  
**public** **void** testcase1()  
{  
System.***out***.println("This is my first teestcase1");  
}  
  
@AfterTest  
**public** **void** AT()  
{  
System.***out***.println("This is After test");  
}

**EXPLAINATION: HERE AS NAME IS SUNIL AND SNAME IS NISHCHITH, THIS LINE THAT IS Assert.assertEquals(SName, Name); IS FAILED🡪 THERFORE IN JUNIT AS @BEFORE IS FAILED, THE @TEST METHOD WILL FAIL AND IT WILL ONLY EXECUTE @AFTER METHOD.**

1. **JUNIT**

@Before  
**public** **void** BT()  
{  
String Name = "Sunil";  
String SName ="Nishchith";  
  
Assert.*assertEquals*(SName,Name);  
System.***out***.println("Both the names are same");  
}  
  
@Test  
**public** **void** testcase1()  
{  
System.***out***.println("This is the first method");  
}  
  
@After  
**public** **void** AT()  
{  
System.***out***.println("This is after method");  
}

**EXPLAINATION: HERE AS NAME IS SUNIL AND SNAME IS NISHCHITH, THIS LINE THAT IS Assert.assertEquals(SName, Name); IS FAILED🡪 AS THE ASSERT IS FAILED IT WILL TOTALLY NOT GET EXECUTED(@TEST AND @AFTER METHOD WILL NOT BE EXECUTED)**