Rest API testing through Rest Assured part 3

We have created response POJO in the last section now we will see the response which is coming from server through that response we will instantiate the POJO class which means we will inside the POJO class and then put the assertion so to initialize the response the POJO class we know the createPerson class is the main response POJO because this guy create person has a reference of response. It has reference of class so if you go to response class this guys it has reference of Address class. So one class is linking with other class internally. So in that case what we do it the CreatePerson is the Custom class which it has given the name because when we convert the response in to the POJO classes, this createPerson is a example in the google.

Now what we will do here is we are going to the service testing that whatever response is coming back. First we will write GSON class object and ones the object it has been created we will write gson.fromJson(String) and we can write what class you want to create that is CreatePerson class and data means response data.

***Gson gson = new Gson();***

***gson.fromJson(data.asString(), CreatePersonResponse.class);***

***And response data is in the form String so make it as data.asString.***

Now we will create object of requestPOJO of CreatePerson class and then we can get the value of this. Like get name, get surname, get city, etc. or whatever value it has we will get it.

CreatePersonResponse createPersonResponse = **new** CreatePersonResponse();

createPersonResponse.getResponse().get(0).getAddress();

here with CreatePersonresponse class object first we will get getResponse() method and using this method we will get get(index) method because it is in the form of list so we can pass index from zero because first value start with zero index then we will access get value of that class. For example getAddress(), GetName() etc.

Now getAddress also have some data so we want to access it. For that we have to do it

***createPersonResponse.getResponse().get(0).getAddress().getCity();***

similarly for other element of address also.

***Note: while accessing gson response you will get error it may be server error sometimes it won’t be start so make sure that it will be started. And after that you will get null pointer exception also to solve this error you need to debug the script and see the problem.***

***Basically null pointer exception will occur because just before the while creating object of this class. Here we have gson object will show null pointer. So to solve this we will return type of this class is CreatePersonResponse.***

***So we do not write the line of CreatePersonResponse class.***

Now we will put assertion on the response data.

***Assert.assertEquals(createPersonResponse.getResponse().get(0).getAddress().getCity(), "city");***

Note: here I have hard coded code of payload but what is unique?

When you get response data come from the data base instead of writing response code like city, name etc we can create a data base connection from data base we can read like what city is getting created. And whatever outcomes comes from data base we can supply as input in the assertion and it work fine.