**A session filter** can be used to record the session id returned from the server and apply this session **ID** in subsequent requests. So that the same logged in session will be continue and it can be used in multiple tests. So login once and use the same session id everywhere.

Example :

**public** **void** add()

{

RestAssured.*baseURI*="http://localhost:8080";

SessionFilter ss = **new** SessionFilter(); // create object of session filter

//Login

// String StrRes =

*given*()

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

.body("{ \"username\": \"Pratik.toke07\", \"password\": \"Admin@123\" }")

.log().all()

.filter(ss)// use it after Body so that the same logged in session will be continue and it can be used in multiple tests. So login once and use the same session id everywhere

.when()

.post("/rest/auth/1/session")

.then()

.assertThat().statusCode(200);

// .extract().response().asString();

// JsonPath js = new JsonPath(StrRes);

// String sessionId = js.get("value");

OR > using auth >

Given().auth().basic(“username”,”password”).filter(ss).when().get(“URL/end point”).then () .. .. .

//Add comment in Bug

*given*()

.pathParam("key","10001")// key of issue for which comment needs to be added and use this variable below

.header("Content-Type","application/json")//.header("Cookie",sessionId) no need of this now since we are maintaining session by session object

.body("{\r\n"

+ " \"body\": \"My comment 132.\",\r\n"

+ " \"visibility\": {\r\n"

+ " \"type\": \"role\",\r\n"

+ " \"value\": \"Administrators\"\r\n"

+ " }\r\n"

+ "}")

.filter(ss)//use sessionfilter object to continue the earlier logged-in session. This is another request, but here we are using session id from previous request to continue the session.

.when()

.post("/rest/api/2/issue/{key}/comment")// this is a path parameter we are using, on line#18 we have mentioned key - 10001, we use same key parameter here

.then()

.assertThat()

.statusCode(201);