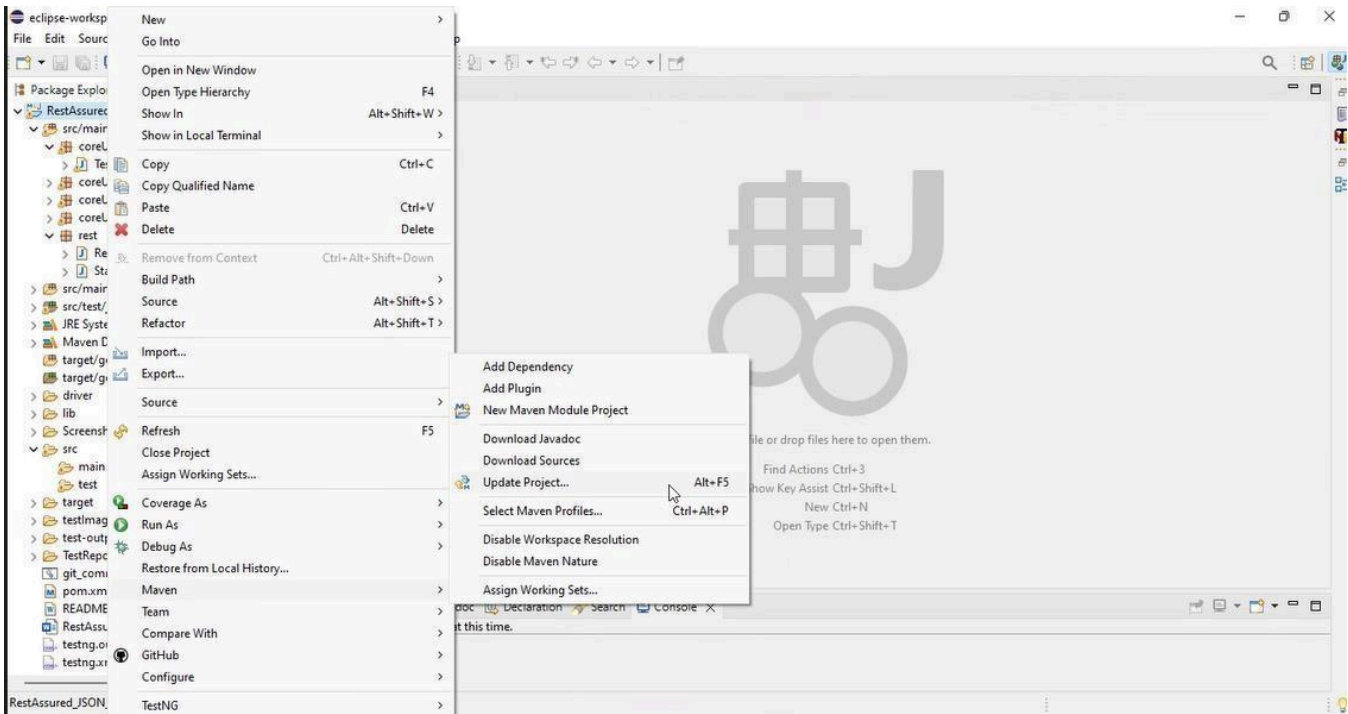# REST-ASSURED
(API - AUTOMATION PROJECT – PL1)

Login A
RestAssured Automation

Login A
RestAssured Automation

Pre-requisite:

As soon as you import project in eclipse, update the project using maven update option as below. This is to resolve issue if any maven dependency not downloaded properly:

1. Right click on project : Go to "Maven" : Select "Update Project"

2. In Update Maven Project Box Select "Force Update of Snapshots/Releases" and click OK

Login A
RestAssured Automation
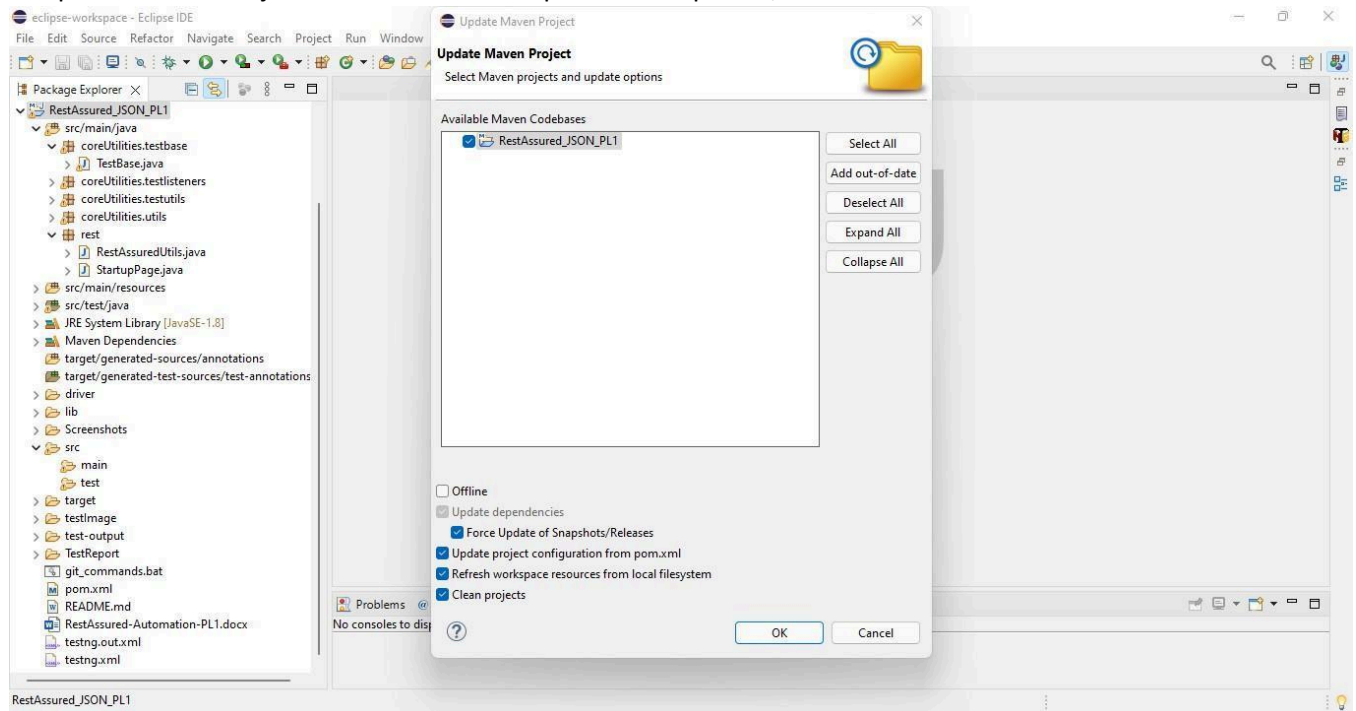
Template Code Structure:

a. Below are the packages and files you will be required to work upon.

b. Other Files and packages you can ignore.

c. In other Files and packages do not do any changes. It would affect your evaluation.

d. You are not required to work in "Test" Folder. Files there are noneditable. Editing those files and trying to save them will throw error and would affect your evaluation.

Package	Class/File	Description
src/main/java/coreUtilities/utils/	FileOperations.java	1. Contains methods to read from excel file. 2. Method is in templated form. 3. **You will be required to implement these methods as very first activity, because for creating post the data should be read from excel.**
/src/main/java/rest	ApiUtil.java	1. All core activities to be performed here. 2. The comments associated with each templated method here describe the expectation. 3. Declare any variable/object you need to share data/status between different methods. 4. Do not modify the signature of methods declared here. 5. You can create additional supportive common methods in CommonEvents class.
/src/main/resources/	TestData.xlsx	Data is present for creating post and put requests

Login A
RestAssured Automation

/src/main/java/coreUtilities/utils	CommonEvents.java	1. Contains all common activities. 2. Certain templated common method declared here. 3. You implement them as per your need. 4. You can add any additional method for common activity here
	Testng.xml	Execution needs to kick started from TestNG xml

PROBLEM STATEMENT

Need to automate the following activities using RestAssured.

Key Activities to implement:

Sl No.	Summary	Action	Expected Result

1	Retrieve the Employee Status:
	
	 GetEmpStatus(String endpoint, String cookieValue, Map<String, String> body)
		1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/admin/employment-statuses?limit=0 ")
		 URL becomes: "opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/employment-statuses?limit=0"
		 2. Fetch the Cookie for the session and set the Header Content Type to "application/json".
		 3. Trigger a GET call to the specified endpoint.
		 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object:
		     ● Response
		
		     ● statusCode (extracted from Response object)
		
		 5. Return the CustomResponse object from the method.
			-The method should Returns an object of type CustomResponse containing:
			   ● statusCode=200
			   ● status=OK.
			
			 -Implementation should be corrects
			 -Non null and non empty List for:
			   ● ItemId
			   ● ItemName .
			
			 -The complete raw response.

2	Retrieve the job-titles Data in the Method:
	
	 GetJobTitle(String endpoint, String cookieValue, Map<String, String> body).
		1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/admin/job-titles?limit=0")
		 URL becomes: "opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/job-titles?limit=0"
		 2. Fetch the Cookie for the session and set the Header Content Type to "application/json".
		 3.. Trigger a GET call to the specified endpoint.
		 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object:
		     ● Response
		     ● statusCode (extracted from Response object)
			-The method should Returns an object of type CustomResponse containing:
			   ● statusCode=200
			   ● status=OK.
			
			 -Implementation should be corrects
			 -Non null and non empty List for:
			   ● Ids
			   ● Name .

Login A
RestAssured Automation

		● Status (extracted from Response object) ● Id list (extracted from Response object) ● Name 5. Return the CustomResponse object from the method.	-The complete raw response.
3	Retrieve the Admin subbunit in the Method: GetAdminSubunit(String endpoint, String cookieValue, Map<String, String> body)	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/admin/subunits") URL becomes: "opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/subunits" 2. Fetch the Cookie for the session and set the Header Content Type to "application/json". 3.. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response ● statusCode (extracted from Response object) ● Status (extracted from Response object) ● empCount. 5. Return the CustomResponse object from the method.	-The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● status=OK. -Implementation should be corrects. -Non null and non empty List for: ● Ids ● Names ● UnitIds ● Description ● Level ● Left ● Right. -The complete raw response.
4	Retrieve PIM Employee Using the Method: GetPimEmp(String endpoint, String cookieValue, Map<String, String> body).	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/employees") URL becomes: "opensource-demo.orangehrmlive.com/web/index.php/api/v2/pim/employees" 2. Fetch the Cookie for the session and set the Header Content Type to "application/json". 3.. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● response, ● statusCode, ● status, ● employeeNumbers, ● lastNames, ● firstNames, ● employeeId. 5. Return the CustomResponse object from the method	-The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● status=OK. -Implementation should be corrects. -Non null and non empty List for: ● employeeNumbers ● lastNames ● firstNames ● employeeIds. -The complete raw response.

5	Retrieve the Reports in ascending order: GetReportASC(String endpoint, String cookieValue, Map<String, String> body).	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/reports/defined?limit=50&offset=0&sortField=report.name&sortOrder=ASC") URL becomes: "opensource-demo.orangehrmlive.com/web/index.php/api/v2/pim/reports/defined?limit=50&offset=0&sortField=report.name&sortOrder=ASC" 2. Fetch the Cookie for the session and set the Header Content Type to "application/json". 3.. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response, ● statusCode, ● status, ● ids, ● names. 5. Return the CustomResponse object from the method	- The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● status=OK. -Implementation should be corrects. -Non null and non empty List for: ● names ● id. -The complete raw response.
6	Retrive the Leave Eligibility list using the method: GetLeaveEligibility(String endpoint, String cookieValue, Map<String, String> body).	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/leave/leave-types/eligible?includeAllocated=true") URL becomes: "opensource-demo.orangehrmlive.com/web/index.php/api/v2/leave/leave-types/eligible?includeAllocated=true" 2. Fetch the Cookie for the session and set the Header Content Type to "application/json". 3.. Trigger a GET call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: ● Response, ● StatusCode, status, ● ids, ● names, ● isDeleted, ● isSituational. 5. Return the CustomResponse object from the method.	- The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● Status=OK. -Implementation should be corrects. -Non null and non empty List for: ● employeeNumbers ● lastNames ● firstNames ● employeeId. -The complete raw response.

| 7 | Update the admin ldap config using Method:

PutAdminConfig(String endpoint, String cookieValue, String body). | 1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/admin/ldap-config")
URL becomes:
"opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/ldap-config"
2. Fetch the Cookie for the session and set the Header Content Type to "application/json".
3. Trigger a PUT call to the specified endpoint.
4. Create an object of type CustomResponse and initialize it with values extracted from the Response object:
    ● response
    ● statusCode,
    ● status,
    ● enable,
    ● hostname,
    ● port,
    ● encryption,
    ● ldapImplementation,
    ● bindAnonymously,
    ● bindUserDN,
    ● hasBindUserPassword,
    ● userLookupSettings,
    ● dataMapping,
    ● mergeLDAPUsersWithExistingSystemUsers,
    ● syncInterval.
5. Return the CustomResponse object from the met
. | - The method should Returns an object of type CustomResponse containing:
    ● statusCode=200
    ● Status=HTTP/1.1 200 OK.

-Implementation should be corrects.

-Non null and non empty List for:
    ● Enable
    ● hostname
    ● port
    ● encryption
    ● ldapImplementation
    ● bindAnonymously
    ● bindUserDN
    ● hasBindUserPassword
    ● userLookupSettings
    ● dataMapping
    ● mergeLDAPUsersWithExistingSystemUsers
    ● syncInterval.

-The complete raw response. |

8	Update the Optional Fields using Method: PutOptionalField(String endpoint, String cookieValue, Object body).	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/optional-field") URL becomes: "opensource-demo.orangehrmlive.com /web/index.php/api/v2/pim/optional-field" 2. Fetch the Cookie for the session and set the Header Content Type to "application/json". 3. Trigger a PUT call to the specified endpoint. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: • Response • statusCode (extracted from Response object) • Status (extracted from Response object) • pimShowDeprecatedFields, • showSIN, • showSSN, • showTaxExemptions. 5. Return the CustomResponse object from the met.	-The method should Returns an object of type CustomResponse containing: • statusCode=200 • status=OK. -Implementation should be corrects. -Request Body should have: • pimShowDeprecatedFields • showSIN • showSSN • showTaxExemptions -The complete raw response.
9	Create a custom field using the method: PostCustomField(String endpoint, String cookieValue, String body).	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/custom-fields") URL becomes: "opensource-demo.orangehrmlive.com /web/index.php/api/v2/pim/custom-fields " 2. Fetch the Cookie for the session and set the Header Content Type to "application/json". 3. Trigger a POST call to the specified endpoint and also pass the body. 4. Create an object of type CustomResponse and initialize it with values extracted from the Response object: • response, • statusCode, • status, • id, • fieldName, • fieldType, • extraData, • screen. 5. Return the CustomResponse object from the met.	-The method should Returns an object of type CustomResponse containing: • statusCode=200 • status=OK. -Implementation should be corrects. -Response should have not NULL: • ID • fieldName • fieldType • extraData • screen. -The complete raw response.

10	Update the pim employee custom field using the method: PutCustomField(String endpoint, String cookieValue, String body).	1. Create an URL by combining the BASE_URL (already declared) and endpoint ("/web/index.php/api/v2/pim/custom-fields/1 ") URL becomes: "opensource-demo.orangehrmlive.com /web/index.php/api/v2/pim/custom-fields/1 " 2. Fetch the Cookie for the session and set the Header Content Type to "application/json". 3. Fetch the id of the employee to delete. 4. Send a PUT req to the url and also pass the body. 5. CustomResponse should contain: ● response, ● statusCode, ● status, ● id, ● fieldName, ● fieldType, ● extraData, ● screen	-The method should Returns an object of type CustomResponse containing: ● statusCode=200 ● status=OK. -Implementation should be corrects. -Response should have not NULL: ● id ● fieldname ● fieldType ● extraData ● screen. -The complete raw response.

NOTE: "Please do not delete any file in the src folder. But you are free to add any other file".

Expectations:

1) **Learners should write automation scripts using Java and REST Assured to automate the API testing for all the provided methods (e.g., GET, POST, PUT, DELETE).** In other words, the automation script should perform all mentioned API interactions, including validation of responses.

2) **Learners should not use any pre-built libraries or tools to validate API responses (e.g., JSON schema validation tools).** They should manually validate the response content (e.g., status codes, response body, etc.) by writing their own logic for assertion.


IMPLEMENTATION/FUNCTIONAL REQUIREMENT

Login A
RestAssured Automation

1.1   CODE QUALITY/OPTIMIZATIONS

1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

EXECUTION STEPS TO FOLLOW

1. **You are mandatory required to run test cases for applications before final submission. Without this project evaluation will not happen.**

2. **You can launch test cases any time as follows: Right-click on testng.xml and run TestNGSuite.**
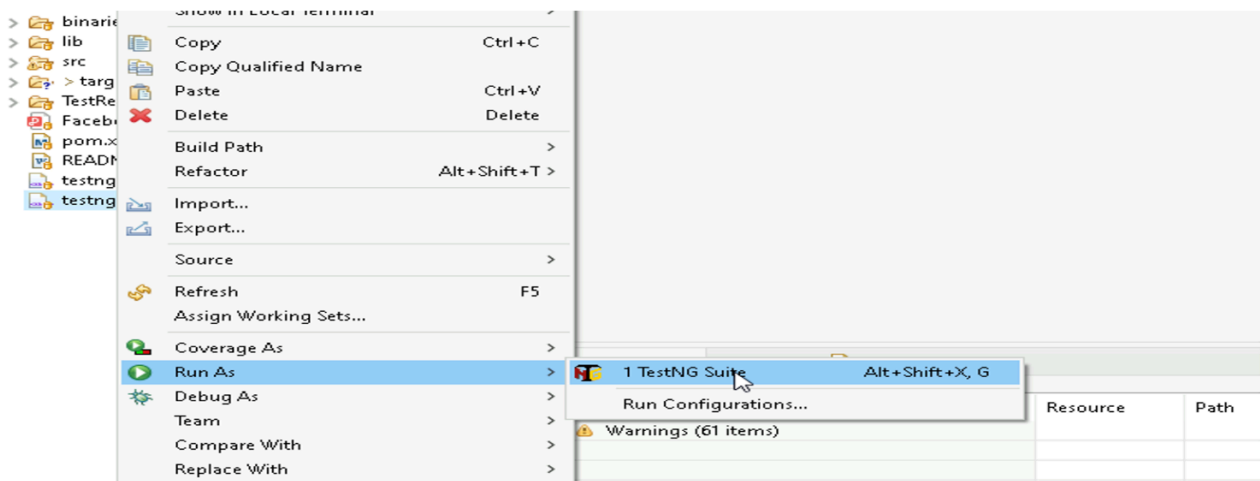
NOTE:
- **When executing test cases via TestNG.xml, it launches the application UI using Selenium WebDriver. This is intentional behavior to facilitate cookie extraction for authentication.**

3. To do the final submission of the assessment :
   a. Press escape to come out of Fullscreen mode.
   b. Submit the assessment.
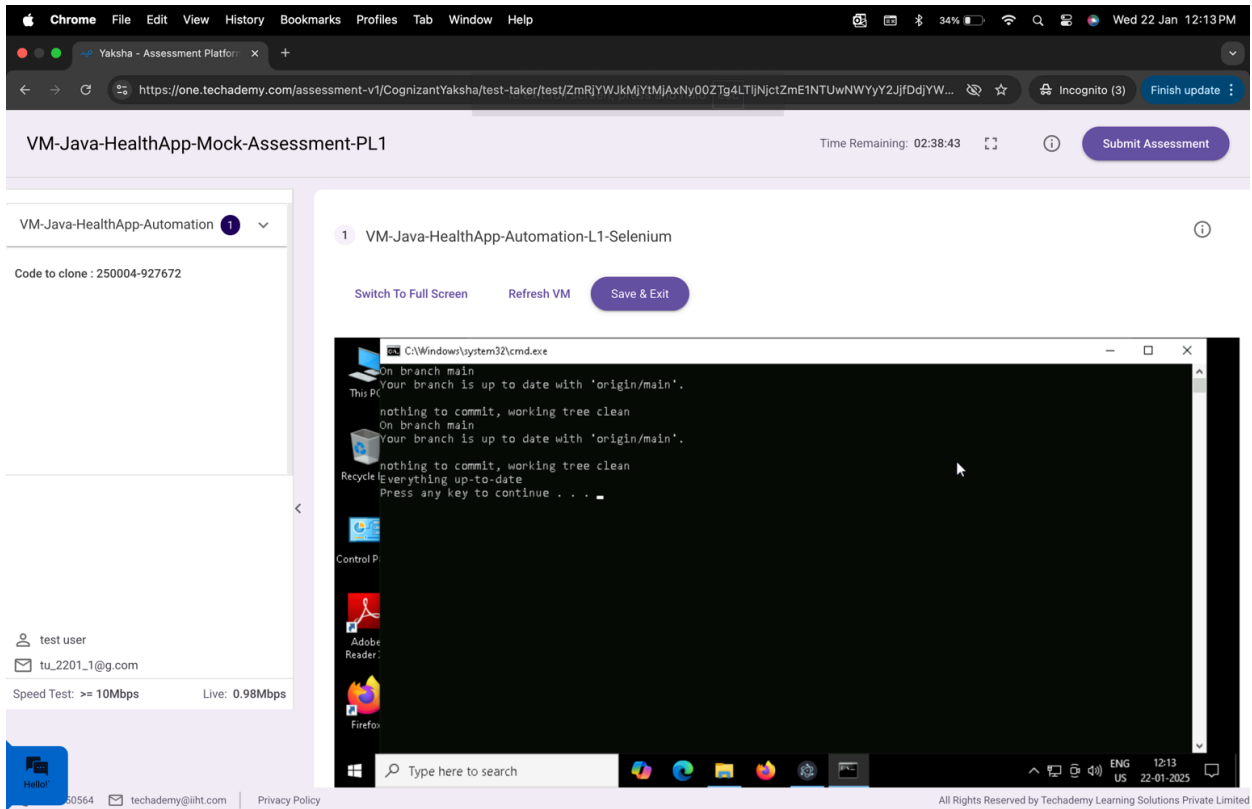
After the successful submission of the assessment, you will get a confirmation message displayed on your screen.

===

All the Best

Login A
RestAssured Automation