

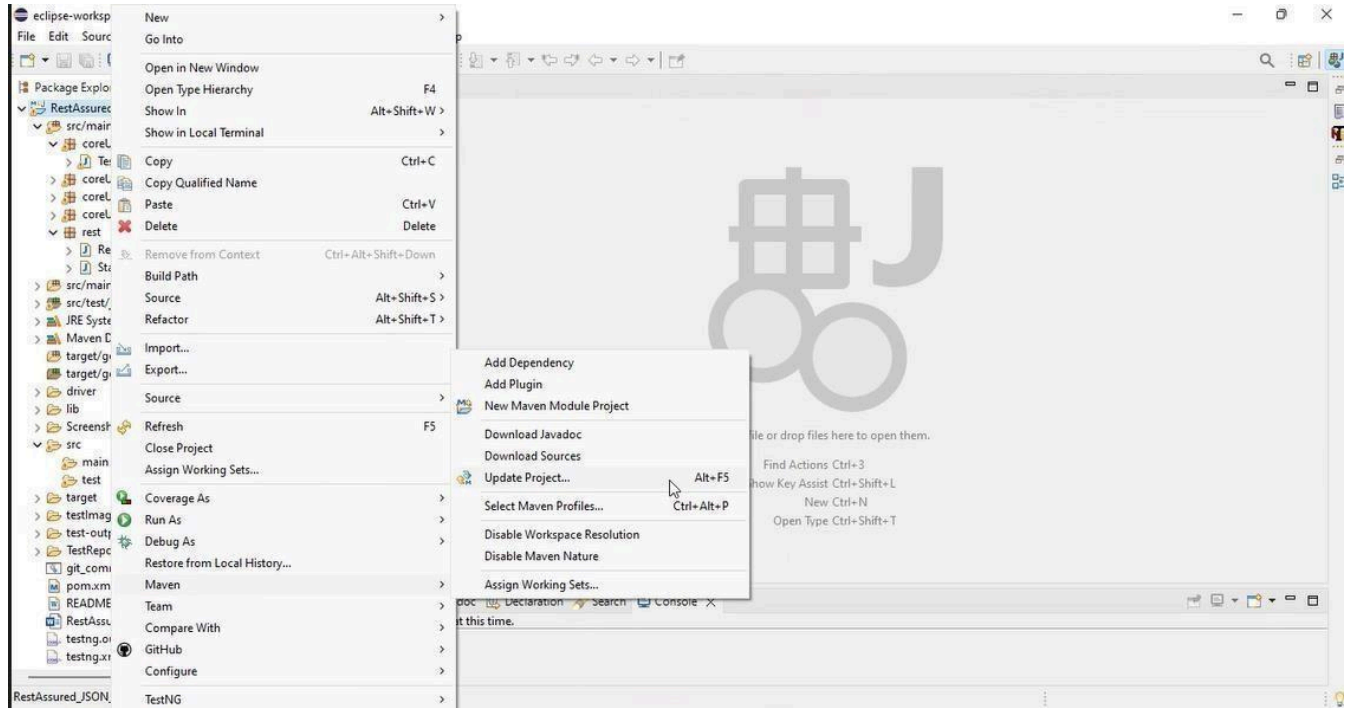
# REST ASSURED API AUTOMATION PROJECT

---

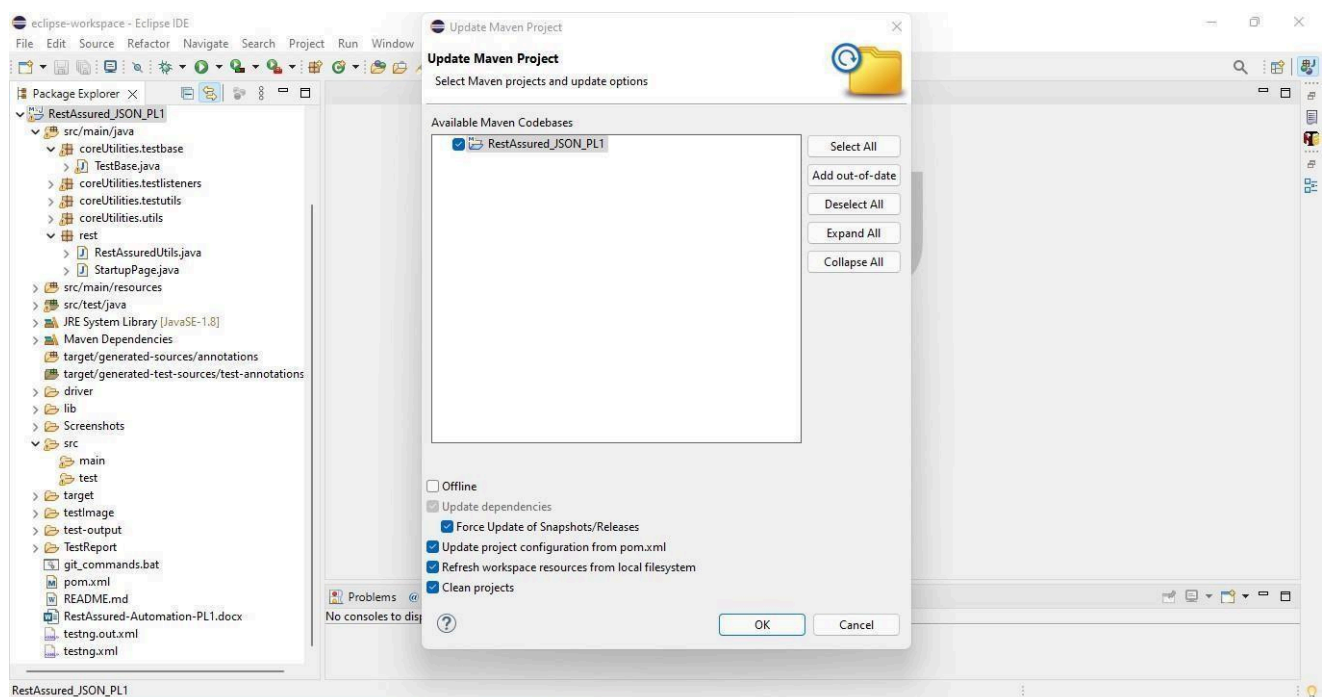
## Pre-requisite:

As soon as you import a project in eclipse, update the project using the 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



## 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 non-editable. Editing those files and trying to save them will throw error and would affect your evaluation.

Package	Class/File	Description
src/main/java/coreUtilities/utills/	FileOperations.java	<ol style="list-style-type: none"><li>1. Contains methods to read from excel file.</li><li>2. Method is in templated form.</li><li>3. <b>You will be required to implement these methods as very first activity, because for creating post the data should be read from excel.</b></li></ol>
/src/main/java/rest	ApiUtil.java	<ol style="list-style-type: none"><li>1. All core activities to be performed here.</li><li>2. The comments associated with each templated method here describe the expectation.</li><li>3. Declare any variable/object you need to share data/status between different methods.</li><li>4. Do not modify the signature of methods declared here.</li><li>5. You can create additional supportive common methods in CommonEvents class.</li></ol>
/src/main/java/rest	AuthUtil.java	<ol style="list-style-type: none"><li>1. Class already defined to read and return bearer token from</li></ol>

		config.properties file.
/src/main/resources/	Config.xlsx	Data present to be used in Implementing functions.
/src/main/java/coreUtilities/utis	CommonEvents.java	<ol style="list-style-type: none"> <li>1. Contains all common activities.</li> <li>2. Certain templated common methods declared here.</li> <li>3. You implement them as per your need.</li> <li>4. You can add any additional method for common activity here</li> </ol>
	Testng.xml	Execution needs to kick started from TestNG xml

## PROBLEM STATEMENT

Need to automate the following activities using RestAssured.

## Key Activities to implement:

Below activities need to be implemented in the ApiUtil.java file present in src/main/java/rest package.

SI No.	Summary	Action	Expected Result
1	Retrieve list of stocks in Method:  <code>getAllStocks(String endpoint, Object body)</code>	<ol style="list-style-type: none"><li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/PharmacyStock/AllStockDetails</code></b></li><li>2. Include the necessary query parameters if required, otherwise, leave it null.</li><li>3. Include an authorization header for the request.</li><li>4. Validate the response contains:<ul style="list-style-type: none"><li>- ItemId</li><li>- ItemName</li><li>- GenericName</li></ul></li><li>5. Ensure all these fields are not null.</li><li>6. Validate the request indicates successful retrieval of stocks.</li><li>7. Validate that:<ol style="list-style-type: none"><li>a. The response status code is 200 (OK).</li><li>b. The Status field in the response is "OK".</li></ol></li></ol>	Returns a 200 OK status with the list of stocks in the "Results" field, containing ItemId, ItemName, and GenericName.
2	Retrieve main store details in Method:  <code>getMainStore(String endpoint, Object body)</code>	<ol style="list-style-type: none"><li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/PharmacySettings/MainStore</code></b></li><li>2. Include the necessary query parameters if required, otherwise, leave it null.</li><li>3. Include an authorization header for the request.</li><li>4. Validate the response contains:<ul style="list-style-type: none"><li>- StoreId.</li></ul></li><li>5. Ensure <code>StoreId</code> is not null.</li><li>6. Validate the request indicates successful retrieval of the main store details.</li><li>7. Validate that:<ol style="list-style-type: none"><li>a. The response status code is 200 (OK).</li></ol></li></ol>	Returns a 200 OK status with the main store details in the "Results" field, including a non-null StoreId.

		b. The Status field in the response is "OK".	
3	<p>Retrieve dispensary requisitions in Method:</p> <p><code>getRequisitionByDateRange</code> (String endpoint, Object body)</p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/DispensaryRequisition/Dispensary/1?FromDate=" + fromDate + "&amp;ToDate=" + toDate</code></b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Add an authorization header to the request.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- RequisitionId</li> <li>- RequisitionNo</li> <li>- RequisitionStatus</li> </ul> </li> <li>5. Ensure <code>RequisitionNo</code> and <code>RequisitionStatus</code> are not null.</li> <li>6. Ensure <code>RequisitionId</code> values are unique.</li> <li>7. Validate the request indicates successful retrieval of requisitions.</li> <li>8. Validate that: <ol style="list-style-type: none"> <li>a. The response status code is 200 (OK).</li> <li>b. The Status field in the response is "OK".</li> </ol> </li> </ol>	Returns a 200 OK status with requisition details in the "Results" field, including RequisitionId, RequisitionNo, and RequisitionStatus.
4	<p>Retrieve patient consumptions in Method:</p> <p><code>getPatientConsumptions</code>(String endpoint, Object body)</p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/PatientConsumption/PatientConsumptions</code></b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Add the required authorization header.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- PatientId</li> <li>- HospitalNo.</li> </ul> </li> <li>5. Ensure these fields are not null.</li> <li>6. Validate the request indicates successful retrieval of patient consumptions.</li> <li>7. Validate that: <ol style="list-style-type: none"> <li>a. The response status code is 200 (OK).</li> <li>b. The Status field in the response is "OK".</li> </ol> </li> </ol>	Returns a 200 OK status with patient consumption details in the Results field, including non-null PatientId and HospitalNo.

5	<p>Retrieve patient consumption info in Method:</p> <p>getPatientConsumptionInfoByPatientIdAndVisitId(String endpoint, Object body)</p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b>https://healthapp.yaksha.com/api/PatientConsumption/PatientConsumptionInfo?patientId=" + patientId + "&amp;patientVisitId=" + patientVisitId</b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Add the required authorization header.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- PatientConsumption.PatientId</li> <li>- PatientConsumption.PatientVisitId</li> </ul> </li> <li>5. Ensure these fields match the request parameters.</li> <li>6. Validate the request indicates successful retrieval of patient consumption info.</li> <li>7. Validate that: <ol style="list-style-type: none"> <li>a. The response status code is 200 (OK).</li> <li>b. The Status field in the response is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with patient consumption information in the "Results" field, including matching PatientId and PatientVisitId.</p>
6	<p>Retrieve billing scheme details in Method:</p> <p>getBillingSchemeBySchemeId(String endpoint, Object body)</p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b>https://healthapp.yaksha.com/api/PatientConsumption/PharmacyIpBillingScheme?schemeId=" + schemeId + ""</b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Add the necessary authorization header.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- SchemeId</li> <li>- SchemeCode</li> </ul> </li> <li>5. Ensure <b>SchemeCode</b> is not null and <b>SchemeId</b> matches the request parameter.</li> <li>6. Validate the request indicates successful retrieval of billing scheme details.</li> <li>7. Validate that: <ol style="list-style-type: none"> <li>a. The response status code is 200 (OK).</li> <li>b. The Status field in the response is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with billing scheme details in the "Results" field, including matching SchemeId and a non-null SchemeCode.</p>

7	<p>Retrieve billing summary in Method:</p> <p><code>getBillingSummaryByPatientId(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/PharmacySales/PatientBillingSummary?patientId=" + patientId</code></b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Add the required authorization header.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- PatientId</li> <li>- TotalDue</li> </ul> </li> <li>5. Ensure <b>TotalDue</b> is not null and <b>PatientId</b> matches the request parameter.</li> <li>6. Validate the request indicates successful retrieval of billing summary.</li> <li>7. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with billing summary details in the "Results" field, including matching PatientId and a non-null TotalDue.</p>
8	<p>Retrieve patient consumption list in Method:</p> <p><code>getConsumptionsListOfAPatientById(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/PatientConsumption/ConsumptionsOfPatient?patientId=" + patientId + "&amp;patientVisitId=" + patientVisitId</code></b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Validate the response contains: <ul style="list-style-type: none"> <li>- PatientConsumptionId</li> <li>- TotalAmount.</li> </ul> </li> <li>4. Ensure <b>PatientConsumptionId</b> values are unique and <b>TotalAmount</b> is not null.</li> <li>5. Validate the request indicates successful retrieval of the consumption list.</li> <li>6. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The Results field contains a StatusCode of "200".</li> <li>b. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with consumption details in the "Results" field, including unique PatientConsumptionId values and non-null TotalAmount.</p>



9	<p>Retrieve return consumption list in Method:</p> <p><i>getReturnConsumptionsList(String endpoint, Object body)</i></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><a href="https://healthapp.yaksha.com/api/PatientConsumption/Returns">https://healthapp.yaksha.com/api/PatientConsumption/Returns</a></b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Add the required authorization header.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- ConsumptionReturnReceiptNo</li> <li>- PatientId</li> </ul> </li> <li>5. Ensure <b>ConsumptionReturnReceiptNo</b> values are unique and <b>PatientId</b> is not null.</li> <li>6. Validate the request indicates successful retrieval of the return consumption list.</li> <li>7. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with return consumption details in the "Results" field, including unique ConsumptionReturnReceiptNo values and non-null PatientId.</p>
10	<p>Retrieve discharged patients in Method:</p> <p><i>getDischargedPatients(String endpoint, Object body)</i></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><a "&amp;todate=" + toDate" +="" fromdate="" href="https://healthapp.yaksha.com/api/Admission/DischargedPatients?admissionStatus=discharged&amp;FromDate=">https://healthapp.yaksha.com/api/Admission/DischargedPatients?admissionStatus=discharged&amp;FromDate=" + fromDate + "&amp;ToDate=" + toDate</a></b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Add the necessary authorization header.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- PatientVisitId</li> <li>- PatientAdmissionId</li> <li>- PatientId</li> </ul> </li> <li>5. Ensure <b>PatientVisitId</b> and <b>PatientAdmissionId</b> values are unique.</li> <li>6. Ensure <b>PatientId</b> is not null.</li> <li>7. Validate the request indicates successful retrieval of discharged patients.</li> <li>8. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the list of discharged patients in the "Results" field, containing unique PatientVisitId and PatientAdmissionId and non-null PatientId.</p>

11	<p>Retrieve admitted patients in Method:</p> <p><code>getAdmittedPatients(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/IpBilling/AdmittedPatients</code></b></li> <li>2. If a request body is needed, include it, similarly if parameters are required, you should add.</li> <li>3. Ensure an authorization header is included.</li> <li>4. Validate the response contains the profile details: <ul style="list-style-type: none"> <li>- PatientId</li> <li>- VisitId</li> <li>- DischargeDate</li> </ul> </li> <li>5. Ensure <code>PatientId</code> and <code>VisitId</code> values are unique.</li> <li>6. Ensure <code>DischargeDate</code> is null for all patients.</li> <li>7. Validate the request indicates successful retrieval of admitted patients.</li> <li>8. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the list of admitted patients in the "Results" field, containing unique PatientId and VisitId values and null DischargeDate for all patients.</p>
12	<p>Search for IPD patients by patient ID in Method:</p> <p><code>searchIpdPatientByPatientId(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> <li>1. Call the GET on the endpoint i.e., <b><code>https://healthapp.yaksha.com/api/Patient/IPDPatientSearch?search="+ patientName</code></b></li> <li>2. Include the necessary query parameters if required, otherwise, leave it null.</li> <li>3. Include the authorization header.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- PatientId</li> <li>- PatientCode</li> </ul> </li> <li>5. Ensure <code>PatientId</code> and <code>PatientCode</code> are not null.</li> <li>6. Validate the request indicates successful retrieval of IPD patients.</li> <li>7. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the list of IPD patients in the "Results" field, containing non-null PatientId and PatientCode.</p>

13	<p>Retrieve patient provisional information in Method:</p> <p><code>getPatientProvisionalInfo(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/Billing/PatientsProvisionalInfo</code></b></li> <li>2. If the body is needed, include it.</li> <li>3. Ensure an authorization header is included.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- PatientId</li> <li>- PatientCode.</li> </ul> </li> <li>5. Ensure <code>PatientId</code> and <code>PatientCode</code> are not null.</li> <li>6. Validate the request indicates successful retrieval of patient provisional information.</li> <li>7. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the provisional information in the "Results" field, containing non-null PatientId and PatientCode.</p>
14	<p>Retrieve provisional items for a specific patient and scheme in Method:</p> <p><code>getProvisionalItemsListByPatientIdAndSchemeId(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/Billing/ProvisionalItemsByPatientId?patientId=" + patientId + "&amp;schemeId=" + schemeId</code></b></li> <li>2. Include necessary request parameters.</li> <li>3. Add the required authorization header.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- Patient.PatientId</li> <li>- Patient.ProvisionalItems.</li> </ul> </li> <li>5. Ensure <code>ProvisionalItems</code> is not empty and matches the patient ID.</li> <li>6. Validate the request indicates successful retrieval of provisional items.</li> <li>7. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the list of provisional items in the "Results" field, containing ProvisionalItems matching the patient ID and non-empty list.</p>

15	<p>Retrieve invoices by date range in Method:</p> <p><i>getInvoicesByDateRange(String endpoint, Object body)</i></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/Billing/Invoices?FromDate=" + fromDate + "&amp;ToDate=" + toDate</code></b></li> <li>2. Include required request parameters.</li> <li>3. Ensure an authorization header is included.</li> <li>4. Validate the response contains: <ul style="list-style-type: none"> <li>- InvoiceNumber</li> <li>- InvoiceCode</li> </ul> </li> <li>5. Ensure these fields are not null for each invoice.</li> <li>6. Validate the request indicates successful retrieval of invoices.</li> <li>7. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. Results array is not null and contains at least one invoice.</li> <li>c. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the list of invoices in the "Results" field, containing non-null InvoiceNumber and InvoiceCode for each invoice.</p>
16	<p>Retrieve provider list in Method:</p> <p><i>getProviderList(String endpoint, Object body)</i></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/Billing/GetProviderList</code></b></li> <li>2. Ensure an authorization header is included.</li> <li>3. Validate the response contains: <ul style="list-style-type: none"> <li>- EmployeeId</li> <li>- EmployeeName</li> </ul> </li> <li>4. Ensure these fields are not null for each provider.</li> <li>5. Validate the request indicates successful retrieval of provider list.</li> <li>6. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. Results array is not null and contains at least one provider.</li> <li>c. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the list of providers in the "Results" field, containing non-null EmployeeId and EmployeeName for each provider.</p>

17	<p>Retrieve users list in Method:</p> <p><code>getUserslist(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/Billing/ListUsers</code></b></li> <li>2. Ensure an authorization header is included.</li> <li>3. Validate the response contains: <ul style="list-style-type: none"> <li>- UserId</li> <li>- ShortName</li> <li>- DepartmentName</li> </ul> </li> <li>4. Ensure <code>UserId</code> is unique and the other fields are not null.</li> <li>5. Validate the request indicates successful retrieval of users list.</li> <li>6. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. Results array is not null and contains at least one user.</li> <li>c. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the list of users in the "Results" field, containing unique UserId and non-null ShortName and DepartmentName.</p>
18	<p>Retrieve current fiscal year details in Method:</p> <p><code>getCurrentFiscalYearDetails(String endpoint, Object body)</code></p>	<ol style="list-style-type: none"> <li>1. Call the GET on endpoint i.e <b><code>https://healthapp.yaksha.com/api/Billing/CurrentFiscalYear</code></b></li> <li>2. Ensure an authorization header is included.</li> <li>3. Validate the response contains: <ul style="list-style-type: none"> <li>- FiscalYearId</li> <li>- FiscalYearName</li> </ul> </li> <li>4. Ensure these fields are not null.</li> <li>5. Validate the request indicates successful retrieval of fiscal year details.</li> <li>6. Validate the response to ensure: <ol style="list-style-type: none"> <li>a. The status code is 200 (OK).</li> <li>b. The Results object is not null.</li> <li>c. The Status field is "OK".</li> </ol> </li> </ol>	<p>Returns a 200 OK status with the fiscal year details in the "Results" field, containing non-null FiscalYearId and FiscalYearName.</p>

**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.
- 

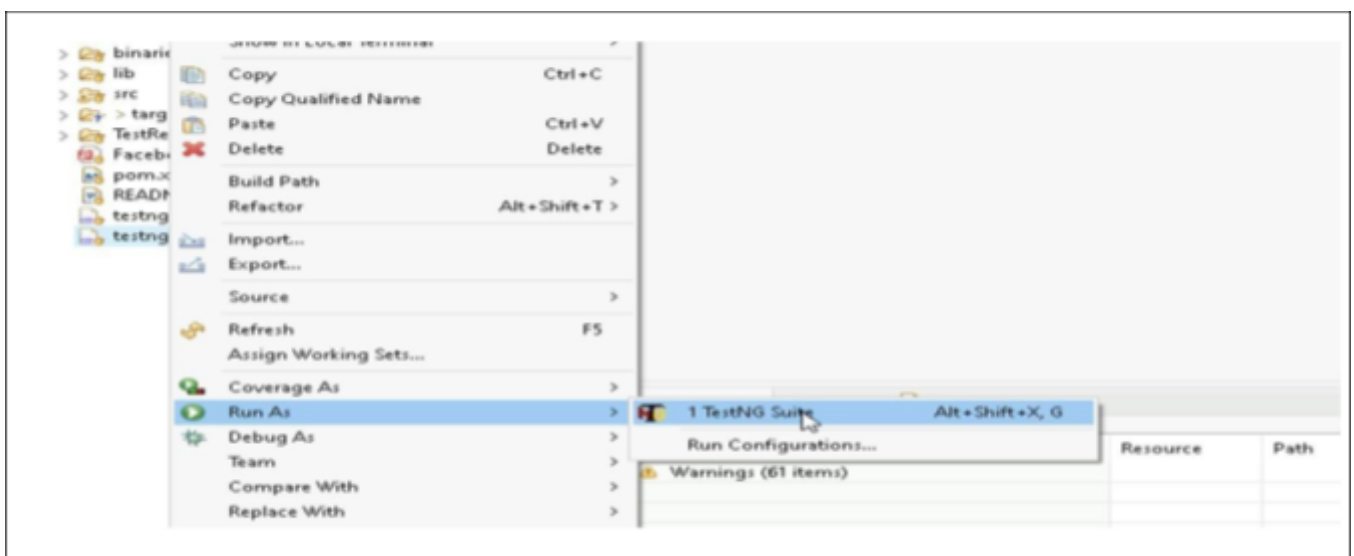
## IMPLEMENTATION/FUNCTIONAL REQUIREMENT

### 1.1 CODE QUALITY/OPTIMIZATIONS

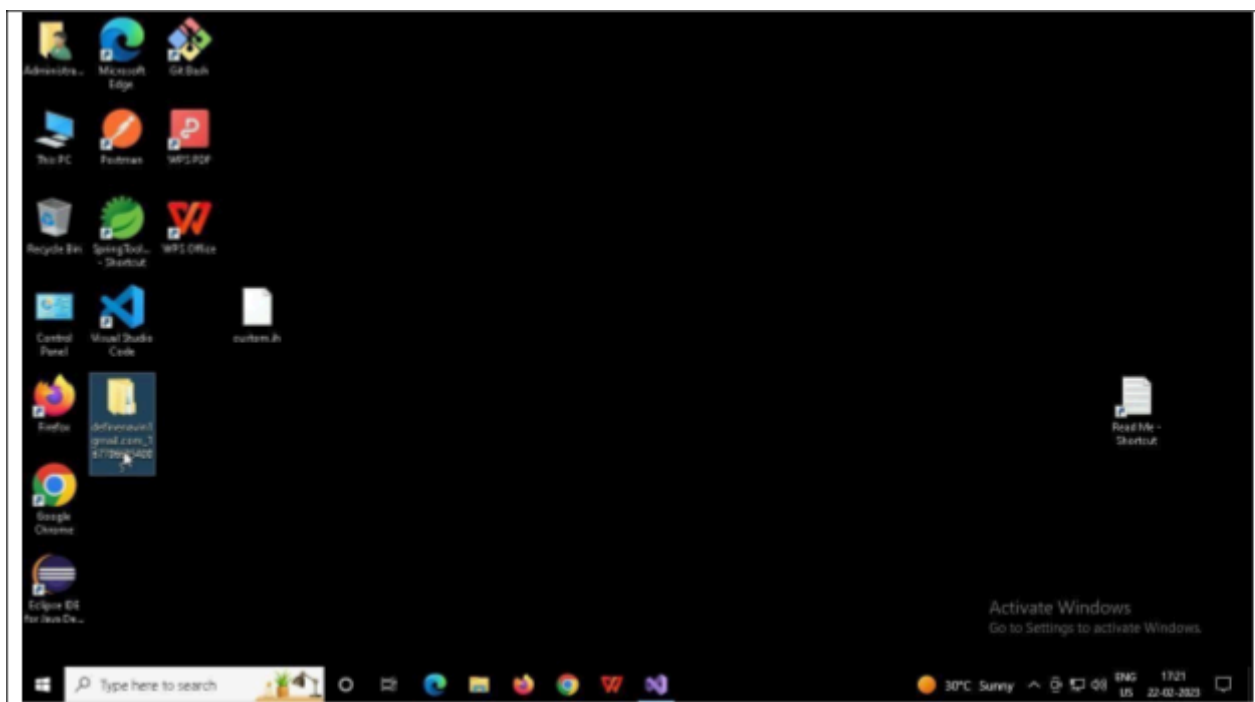
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

## EXECUTION STEPS TO FOLLOW

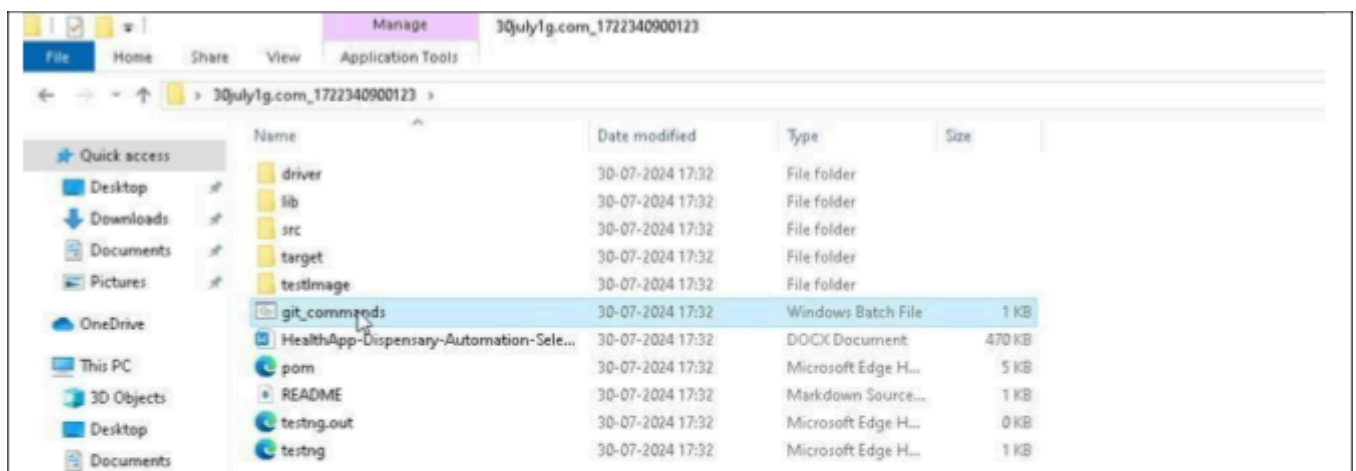
1. **You are mandatorily required to run test cases for applications before final submission. Without which project evaluation will not happen.**
2. **You can launch test cases any time as follows: Right click on testng.xml and run TestNGSuite**



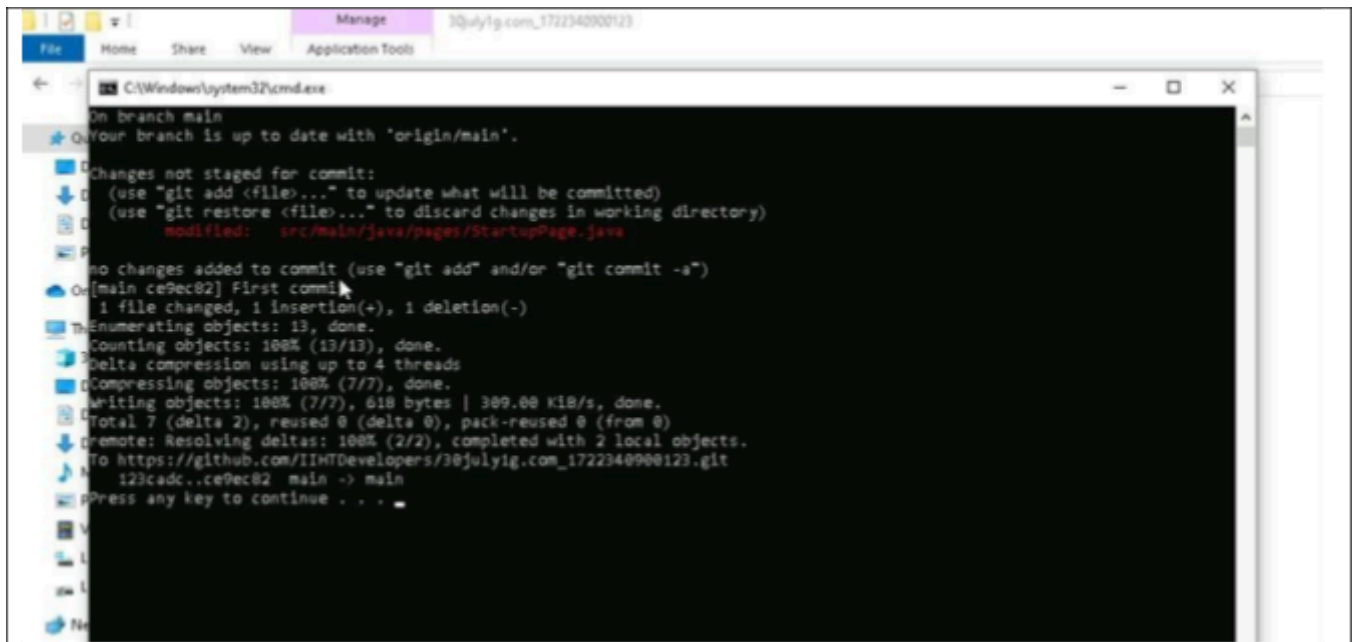
3. Before final submission, you are also required to push your code to GIT. Following are the steps to follow:



In your project folder, you will find a batch file named git\_commands



Double-click the batch file to run it. It will run the commands to push your code to GIT.



```
C:\Windows\system32\cmd.exe
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/main/java/pages/StartupPage.java

no changes added to commit (use "git add" and/or "git commit -s")
[main ce9ec82] First commit
1 file changed, 1 insertion(+), 1 deletion(-)
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 618 bytes | 309.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/IIHTDevelopers/30julyig.com_1722340900123.git
   123cadc..ce9ec82  main -> main
Press any key to continue . . .
```

=====

All the Best