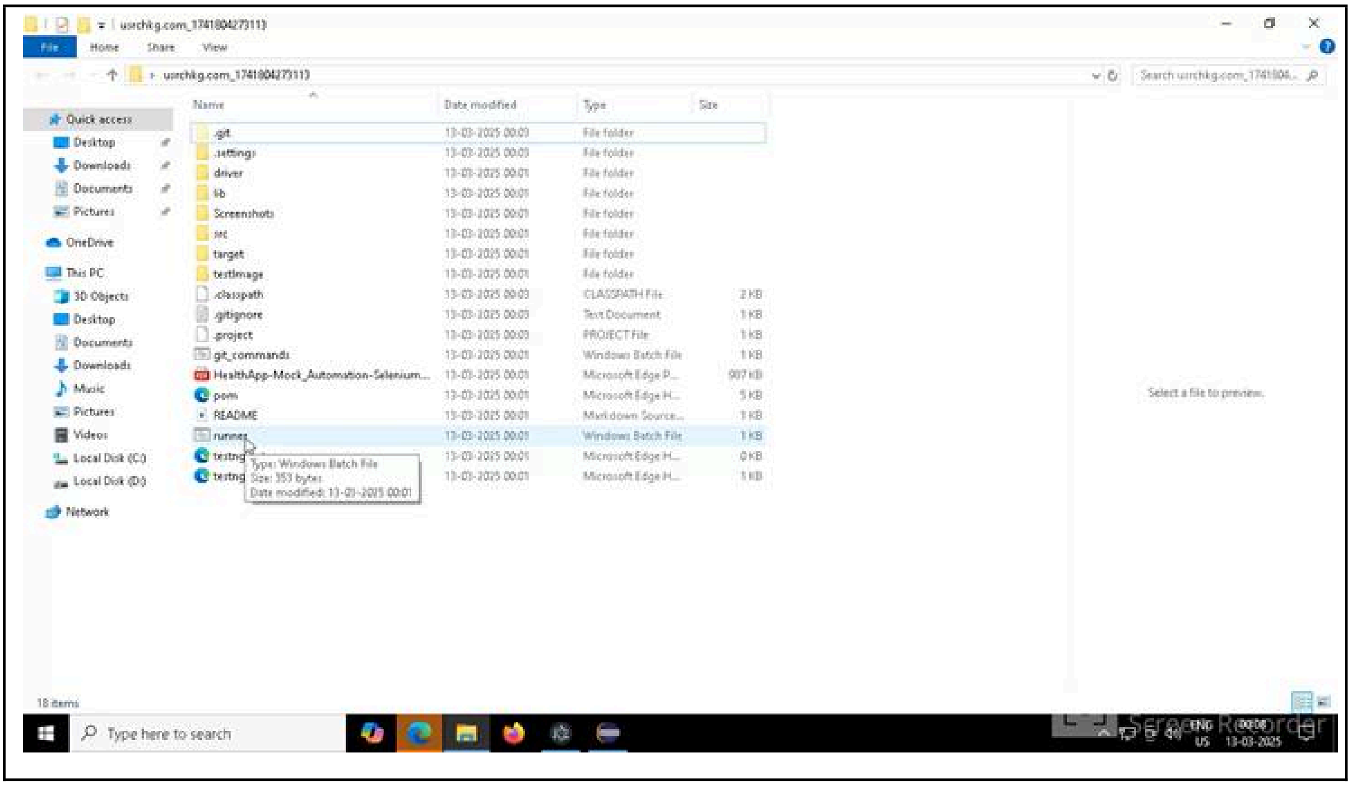
REST-ASSURED
(API - AUTOMATION PROJECT – PL1)

Pre-requisite:

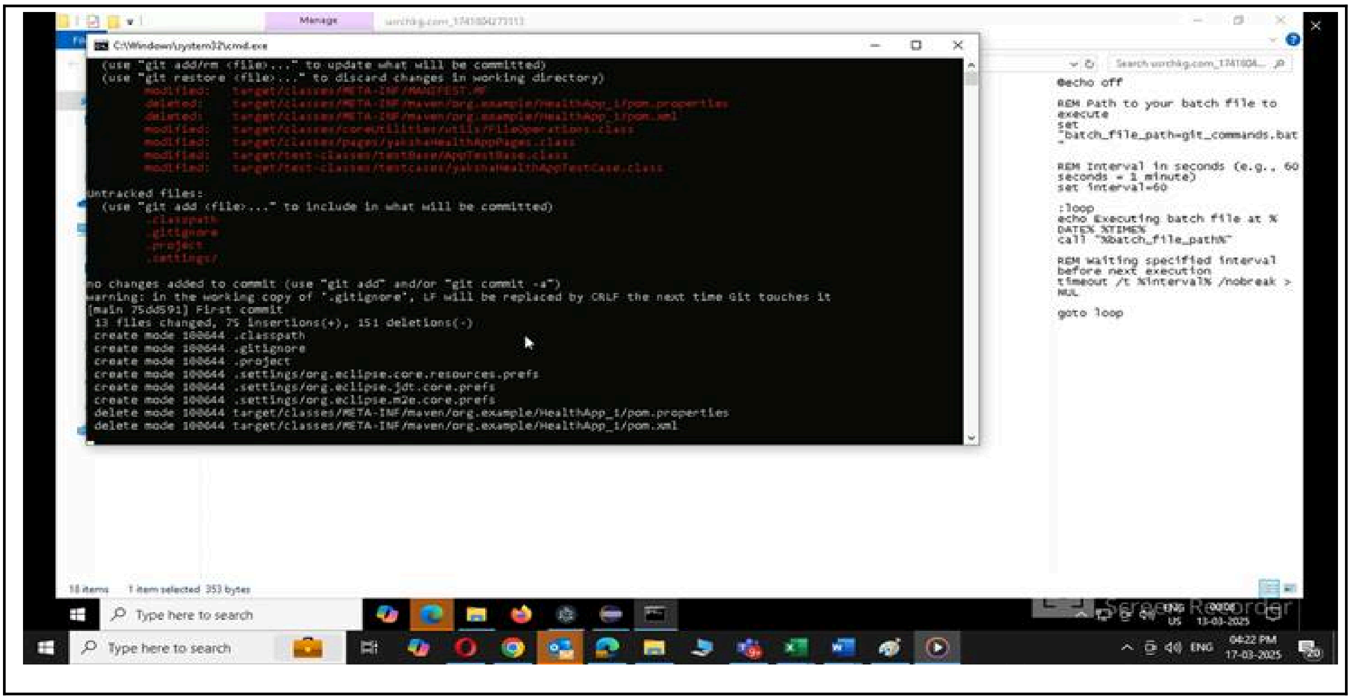
Before you start working on your project, execute the runner file present in your project folder (Simply by double click). This is mandatory.



Name	Date modified	Type	Size
.git	13-03-2025 00:03	File folder	
.settings	13-03-2025 00:03	File folder	
driver	13-03-2025 00:01	File folder	
lib	13-03-2025 00:01	File folder	
Screenshots	13-03-2025 00:01	File folder	
src	13-03-2025 00:01	File folder	
target	13-03-2025 00:01	File folder	
testImage	13-03-2025 00:01	File folder	
.classpath	13-03-2025 00:03	CLASSPATH File	2 KB
.gitignore	13-03-2025 00:03	Text Document	1 KB
.project	13-03-2025 00:03	PROJECT File	1 KB
git_commands	13-03-2025 00:01	Windows Batch File	1 KB
HealthApp-Mock_Automation-Selenium...	13-03-2025 00:01	Microsoft Edge P...	907 KB
pom	13-03-2025 00:01	Microsoft Edge H...	5 KB
README	13-03-2025 00:01	Markdown Source...	1 KB
runner	13-03-2025 00:01	Windows Batch File	1 KB
testng	13-03-2025 00:01	Microsoft Edge H...	0 KB
testng	13-03-2025 00:01	Microsoft Edge H...	1 KB

Type: Windows Batch File
Size: 353 bytes
Date modified: 13-03-2025 00:01

This will launch a command terminal for you where it will keep on pushing your updated code to GIT on regular intervals. Keep that command terminal open at backend and you can continue working on your project.



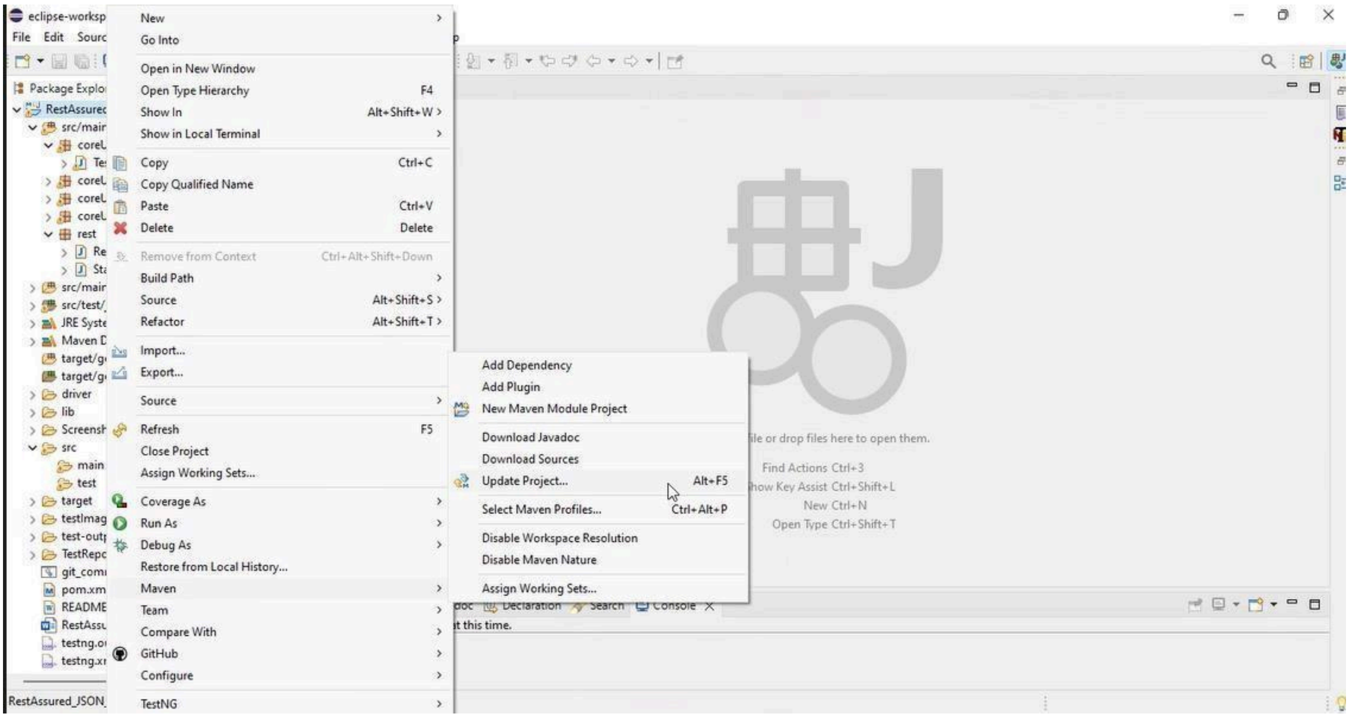
Manage
usrchkg.com_1741804273113
C:\Windows\system32\cmd.exe
Search usrchkg.com_174180...
18 items
1 item selected 353 bytes
Type here to search
Type here to search

```
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   target/classes/META-INF/MANIFEST.MF
        deleted:    target/classes/META-INF/maven/org.example/HealthApp_1/pom.properties
        deleted:    target/classes/META-INF/maven/org.example/HealthApp_1/pom.xml
        modified:   target/classes/coreUtilities/utils/FileOperations.class
        modified:   target/classes/pages/yakshaHealthAppPages.class
        modified:   target/test-classes/testBase/AppTestBase.class
        modified:   target/test-classes/testcases/yakshaHealthAppTestCase.class

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .classpath
        .gitignore
        .project
        .settings/

no changes added to commit (use "git add" and/or "git commit -a")
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
[main 75dd591] First commit
 13 files changed, 75 insertions(+), 151 deletions(-)
 create mode 100644 .classpath
 create mode 100644 .gitignore
 create mode 100644 .project
 create mode 100644 .settings/org.eclipse.core.resources.prefs
 create mode 100644 .settings/org.eclipse.jdt.core.prefs
 create mode 100644 .settings/org.eclipse.m2e.core.prefs
 delete mode 100644 target/classes/META-INF/maven/org.example/HealthApp_1/pom.properties
 delete mode 100644 target/classes/META-INF/maven/org.example/HealthApp_1/pom.xml
```

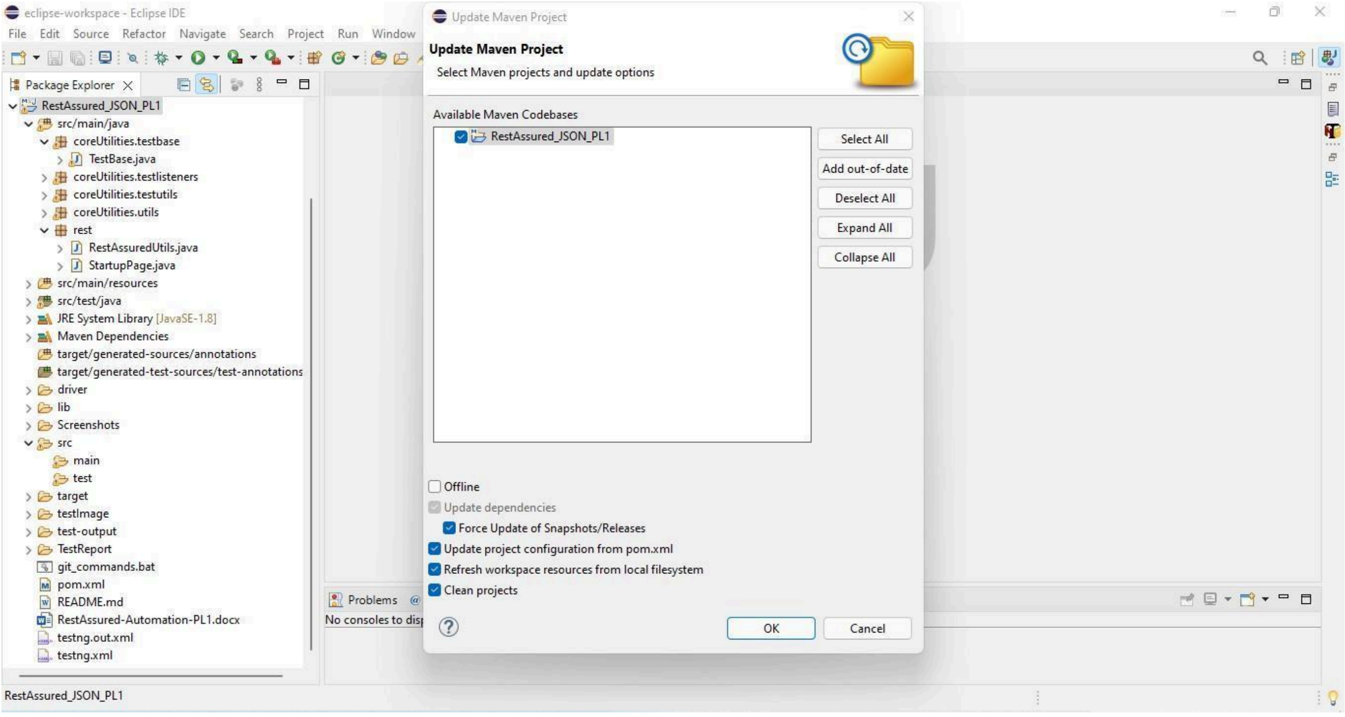
```
@echo off

REM Path to your batch file to
execute
set
"batch_file_path=git_commands.bat
"

REM Interval in seconds (e.g., 60
seconds = 1 minute)
set interval=60

:loop
echo Executing batch file at %
DATE% %TIME%
call "%batch_file_path%"

REM Waiting specified interval
before next execution
timeout /t %interval% /nobreak >
NUL

goto loop
```

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



Update Maven Project

Select Maven projects and update options

Available Maven Codebases

☑ RestAssured_JSON_PL1

□ Offline

☑ Update dependencies

☑ Force Update of Snapshots/Releases

☑ Update project configuration from pom.xml

☑ Refresh workspace resources from local filesystem

☑ Clean projects

Template Code Structure:

- Below are the packages and files you will be required to work upon.
- Other Files and packages you can ignore.
- In other Files and packages do not do any changes. It would affect your evaluation.
- 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. It contains methods to read data from Excel files. 2. The method is in templated form. 3. You will be required to implement these methods as the first activity, because even the URL to navigate to is read using these methods.
/src/main/java/pages	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 4. variable/object you need to share data/status between different methods. 5. Do not modify the signature of methods declared here. 6. You can create additional supportive common methods in 7. CommonEvents class.
/src/main/resources/	Config.xlsx	1. Data present to be used in creating a new Post.
	testData.xlsx	1. Data is present for creating post and put requests
/src/main/java/coreUtilities/utils	CommonEvents.java	1. Contains all common activities. 2. Certain templated common methods are declared here. 3. You implement them as per your needs. 4. You can add any additional method for common activity here
	Testng.xml	Execution needs to be kick-started from TestNG.xml

PROBLEM STATEMENT

Need to automate the following activities using RestAssured.

Key Activities to implement:

#	Summary	Action	Expected Result
1	Retrieve sub units tree with Valid Cookie Authentication using method: **getSubunitsTree(String endpoint, String cookieValue, Map<String, String> body)**	1. Construct the final URL by combining the BASE_URL and the provided endpoint parameter: a. https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/subunits?mode=tree 2. Initialize a RequestSpecification using RestAssured: a. Add a cookie with name "orangehrm" and the value passed as cookieValue. b. Set the "Content-Type" header to "application/json". 3. Trigger a GET request to the above URL. 4. Create a CustomResponse object with: a. Full Response b. Extracted statusCode c. Extracted statusLine d. Extracted list id e. Extracted list unitId f. Extracted list name g. Extracted list level h. Extracted list children 5. Return the CustomResponse object.	1. Method should return a fully initialized CustomResponse object with: a. statusCode = 200 b. status = OK (or valid status line like "HTTP/1.1 200 OK") c. A non-null and non-empty languages list d. A non-null and non-empty id list e. A non-null and non-empty unitId list f. A non-null and non-empty unitId name g. A non-null and non-empty unitId level h. A non-null and non-empty unitId children i. The complete raw Response object
2	Retrieve admin skills using session cookie authentication through method: **getAdminSkills(String endpoint, String cookieValue, Map<String, String> body)**	1. Construct the complete URL by concatenating the BASE_URL and the provided endpoint: a. https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/skills?limit=50&offset=0 2. Create a RequestSpecification using RestAssured:	1. The method should return a CustomResponse object with: a. statusCode = 200 b. status equal to a valid HTTP status line like "HTTP/1.1 200 OK" c. Non-null and non-empty lists for: i. id ii. name iii. description

		a. Add a session cookie named "orangehrm" with the provided cookieValue. b. Add a header "Content-Type" with the value "application/json". 3. Send a GET request to the constructed URL. 4. Wrap the extracted fields into lists<Object>: a. id, name, description 5. Create and return a CustomResponse object initialized with: a. The complete Response b. Extracted statusCode c. Extracted statusLine d. id, name, description	d. The complete raw Response object
3	Retrieve admin educations using session-based authentication through the method: **getAdminEdu(String endpoint, String cookieValue, Map<String, String> body)**	1. Construct the final URL by combining the BASE_URL and the provided endpoint: ● https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/educations?limit=50&offset=0 2. Create a RequestSpecification using RestAssured: ● Add a session cookie named "orangehrm" using the provided cookieValue. ● Set the request header "Content-Type" to "application/json". 3. Wrap each of these fields into separate List<Object> collections: ● id ● name 4. Create a CustomResponse object using: ● The complete Response object ● Extracted statusCode and statusLine ● id and name field lists 5. Return the populated CustomResponse object.	1. The method should return a CustomResponse object containing: ● statusCode = 200 ● A valid status line such as "HTTP/1.1 200 OK" ● Non-null and non-empty lists for the following keys: ➔ id and name ● The complete Response object from the API call

4	Retrieve admin licenses using session-based authentication through method: **getAdminLicenses(String endpoint, String cookieValue, Map<String, String> body)**	1. Construct the final URL by concatenating BASE_URL and the given endpoint: https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/licenses?limit=50&offset=0 2. Create a **RequestSpecification** using **RestAssured**: ● Include the session cookie "orangehrm" using the provided cookieValue ● Add the header "Content-Type" with value "application/json" 3. Trigger a **GET** request to the above URL. 4. Store each field as a List<Object>: ● id, name 5. Return a CustomResponse object populated with: ● Full Response object ● statusCode and statusLine ● Extracted field lists	● Method must return a valid CustomResponse object containing: ● statusCode = 200 ● status like "HTTP/1.1 200 OK" ● Non-null and non-empty lists for: ▪ id ▪ name o The full raw Response object
5	Retrieve admin languages using session-based cookie authentication via method: **getAdminLanguages(String endpoint, String cookieValue, Map<String, String> body)**	1. Construct the final URL by appending the endpoint to the base URL: `https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/licenses?limit=50&offset=0` 2. Use **RestAssured** to initialize a `RequestSpecification`: ● Add the session cookie `"orangehrm"` using the provided `cookieValue` ● Set the request header `"Content-Type"` to `"application/json"`	● The method must return a `CustomResponse` containing: ● `statusCode = 200` ● A valid `status` line like `"HTTP/1.1 200 OK"` ● An `CustomResponse` object with non-null and non-empty lists for: ▪ `id` ▪ `name` ● The full raw `Response` object

		3. If a request body (`body`) is not null, include it in the request. 4. Store all extracted fields in respective `List<Object>` collections. o id o name 5. Return a `CustomResponse` object initialized with: ● The full `Response` ● `statusCode` and `statusLine`	
6	Delete valid user using session-based authentication via method: **deleteValidUsers(String endpoint, String cookieValue, String requestBody)**	1. Construct the complete URL by appending the endpoint to the base URL: `https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/users` 2. Initialize a **RequestSpecification** using **RestAssured**: ● Add the session cookie `"orangehrm"` with the provided `cookieValue`. ● Set the header `"Content-Type"` to `"application/json"`. 3. If a request body (`body`) is provided (not null), include it in the request using `.body()`. 4. Trigger a **DELETE** request to the constructed URL and extract the **Response**. 5. Create and return a **CustomResponse** object containing: ● The full `Response` ● The extracted `statusCode` and `statusLine`	● The method should return a `CustomResponse` object with: ● `statusCode = 200` ● A valid `status` string (e.g., `"HTTP/1.1 200 OK"`) ● The complete raw `Response` object

7	Create new job titles using session-based authentication through method: **postNewJobTitles(String endpoint, String cookieValue, String requestBody)**	1. Construct the final URL using the base URL and the language package ID as a path variable: `https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/job-titles` 2. Initialize a **RequestSpecification** using **RestAssured**: ● Add the `"orangehrm"` cookie with the given `cookieValue`. ● Set the request header `"Content-Type"` to `"application/json"`. 3. If the `body` is not null, include it using `.body()`. 4. Trigger a **POST** request to the endpoint and extract the **Response**. 5. Store these values in individual string variables: ● `title` 6. Return a **CustomResponse** object initialized with: ● The full `Response` ● `statusCode` and `statusLine` ● Extracted string values: title	● The method should return a valid `CustomResponse` containing: ● `statusCode = 200` ● A valid HTTP `status` (e.g., `"HTTP/1.1 200 OK"`) ● Non-empty string values for: ▪ `title` ● The complete raw `Response` object
8	Update job title by ID using session-based authentication via method: **putJobTitleByID(String endpoint, String cookieValue, String requestBody)**	1. Construct the complete URL by appending the endpoint to the base URL: `https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/job-titles/{id}` 2. Initialize a **RequestSpecification** using **RestAssured**: ● Add a session cookie named `"orangehrm"` with	● The method should return a `CustomResponse` containing: ● `statusCode = 200` ● A valid HTTP `status` line such as `"HTTP/1.1 200 OK"` ● A non-null and non-empty `List<Object>` ▪ id

		the provided `cookieValue` ● Set the `"Content-Type"` header to `"application/json"` 3. If the `body` is not null, include it using `.body()`. 4. Send a **PUT** request to the full URL and extract the **Response** object. 5. Return a **CustomResponse** object with: ● The full `Response` object ● Extracted `statusCode` and `statusLine` ● The List<Object> of `id` ● The List<Object> of `title` ● The Object type `description` ● The Object type `note` ● The `Map<String, Object> jobSpecification`	▪ title ● The full raw `Response` object
9	Create admin pay-grades with session-based cookie authentication using method: **postUniquePaygrade(String endpoint, String cookieValue, String requestBody)**	1. Construct the complete URL by appending the endpoint to the base URL: `https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/pay-grades` 2. Initialize a **RequestSpecification** using **RestAssured**: ● Add the cookie `"orangehrm"` with the value provided by `cookieValue`. ● Set the header `"Content-Type"` to `"application/json"`.	● The method must return a `CustomResponse` object that includes: ● `statusCode = 200` ● A valid `status` line (e.g., `"HTTP/1.1 200 OK"`) ● A non-null and non-empty Object ▪ id ▪ name ● The complete raw `Response` object

		● If the `body` is not null, include it using `.body()`. 3. Trigger a **POST** request to the above endpoint and extract the **Response**. 4. Parse the response using **JsonPath** and retrieve the `"data"` object as a `Map<String, Object>` which holds the current module status. 5. Return a **CustomResponse** object initialized with: ● The full `Response` object ● `statusCode` and `statusLine` ● `Extracted id, name of type Object.`	
10	Delete paygrade by ID using a DELETE request with session-based authentication via method: **deletePaygradeById(String endpoint, String cookieValue, String requestBody)**	1. Construct the complete URL by appending the endpoint to the base URL: https://opensource-demo.orangehrmlive.com/web/index.php/api/v2/admin/pay-grades 2. Initialize a **RequestSpecification** using **RestAssured**: ● Add the session cookie `"orangehrm"` with the provided `cookieValue`. ● Set the header `"Content-Type"` to `"application/json"`. 3. If the `body` is not null, include it using `.body()`. 4. Trigger a **DELETE** request to the constructed URL and extract the **Response**. 5. Create and return a **CustomResponse** object containing: ● The full `Response`	● The method should return a valid `CustomResponse` object containing: ● `statusCode = 200` ● A valid `status` string like `"HTTP/1.1 200 OK"` ● The full `Response` object

		● The extracted `statusCode` and `statusLine`	

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

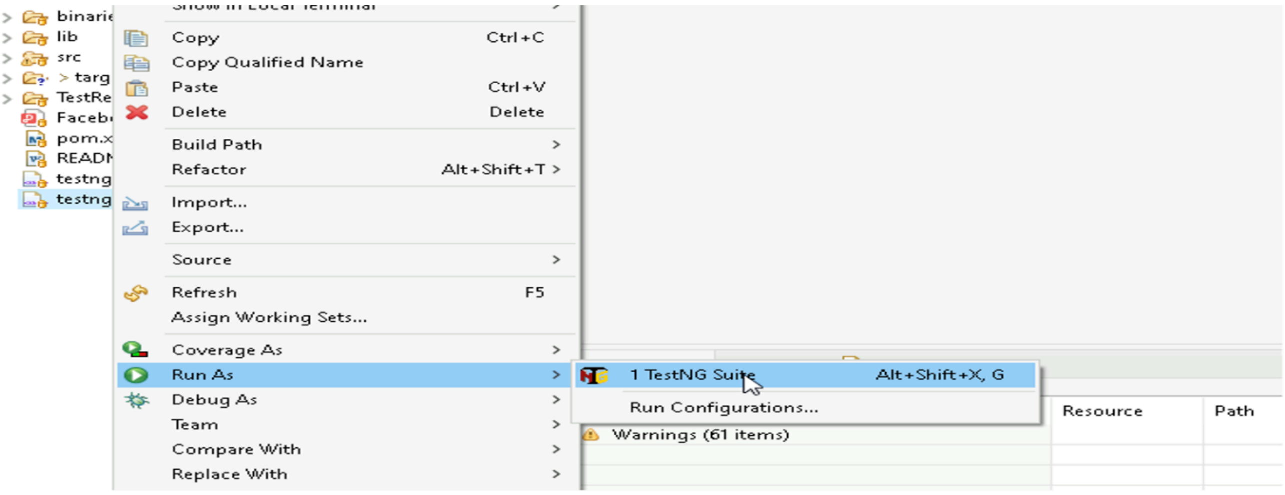
1.1 CODE QUALITY/OPTIMIZATIONS

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

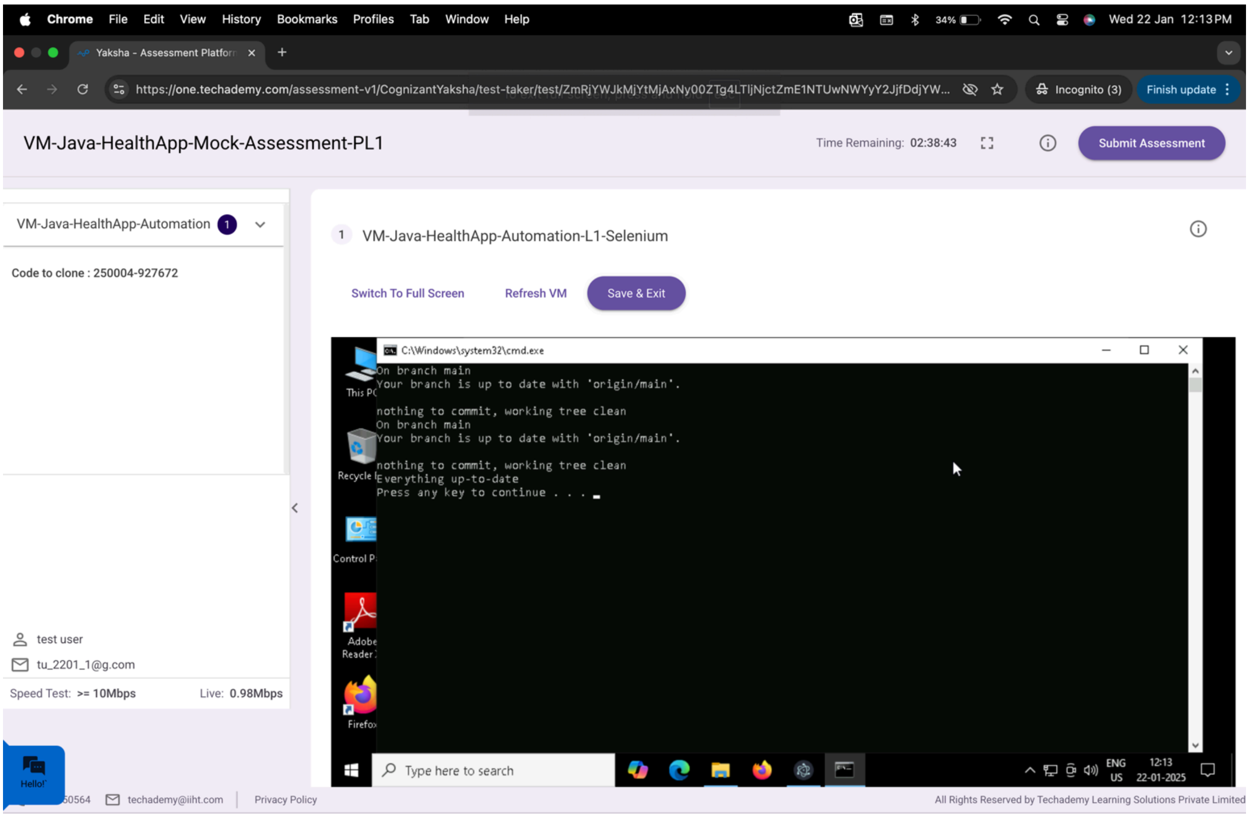
2. **You can launch test cases any time as follows: Right-click on testng.xml and run TestNGSuite.**



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.



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 22 Jan 12:13 PM
Yaksha - Assessment Platform
https://one.techademy.com/assessment-v1/CognizantYaksha/test-taker/test/ZmRjYWJkMjYtMjAxNy00ZTg4LTljNjctZmE1NTUwNWYyY2JjfDdjYW...
Incognito (3)
Finish update
Submit Assessment
VM-Java-HealthApp-Automation
Switch To Full Screen
Refresh VM
Save & Exit
C:\Windows\system32\cmd.exe
This PC
Recycle Bin
Control Panel
Adobe Reader
Firefox
Type here to search
ENG US
12:13 22-01-2025
Hello!
techademy@iiht.com
Privacy Policy
All Rights Reserved by Techademy Learning Solutions Private Limited

VM-Java-HealthApp-Mock-Assessment-PL1

Time Remaining: 02:38:43

1 VM-Java-HealthApp-Automation-L1-Selenium

Code to clone : 250004-927672

test user
tu_2201_1@g.com
Speed Test: >= 10Mbps Live: 0.98Mbps

```
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
Press any key to continue . . .
```

After the successful submission of the assessment, you will get a confirmation message displayed on your screen.

==

All the Best