Prepared by -Pasindya Chavindee -2558

What is REST Assured?

REST Assured is an open-source **Java-based library** used to test RESTful web services. It provides a **domain-specific language (DSL)** for writing powerful, readable, and maintainable tests for REST APIs.

It's widely used in **QA automation** for validating **HTTP requests and responses** (like GET, POST, PUT, DELETE).

Why Use REST Assured?

Feature	**Benefit**
Simple Syntax	You write tests in natural, readable Java
No Need for Manual JSON Parsing	Automatically handles response parsing
TestNG/JUnit Integration	Works with standard test frameworks
Supports Authentication	Basic, Digest, OAuth, Bearer, etc
Built-in Assertion	Hamcrest-style expressive assertions
Fast Execution	Executes quickly with minimal setup

Common HTTP Methods

Method	**Use Case**
GET	Read/fetch resource
POST	Create resource
PUT	Update entire resource
PATCH	Partially update resource
DELETE	Delete resource

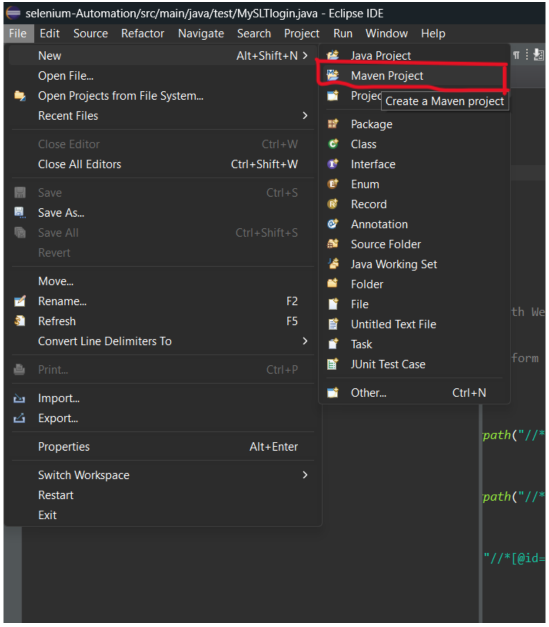
Prerequisites & Tools

Tool	Description
Java JDK	Install JDK 11 or later
IDE	IntelliJ IDEA / VS Code
Maven/Gradle	For managing dependencies
Postman	(Optional) For manual API testing

Project Setup (Maven)

➤ Step 1: Create Maven Project

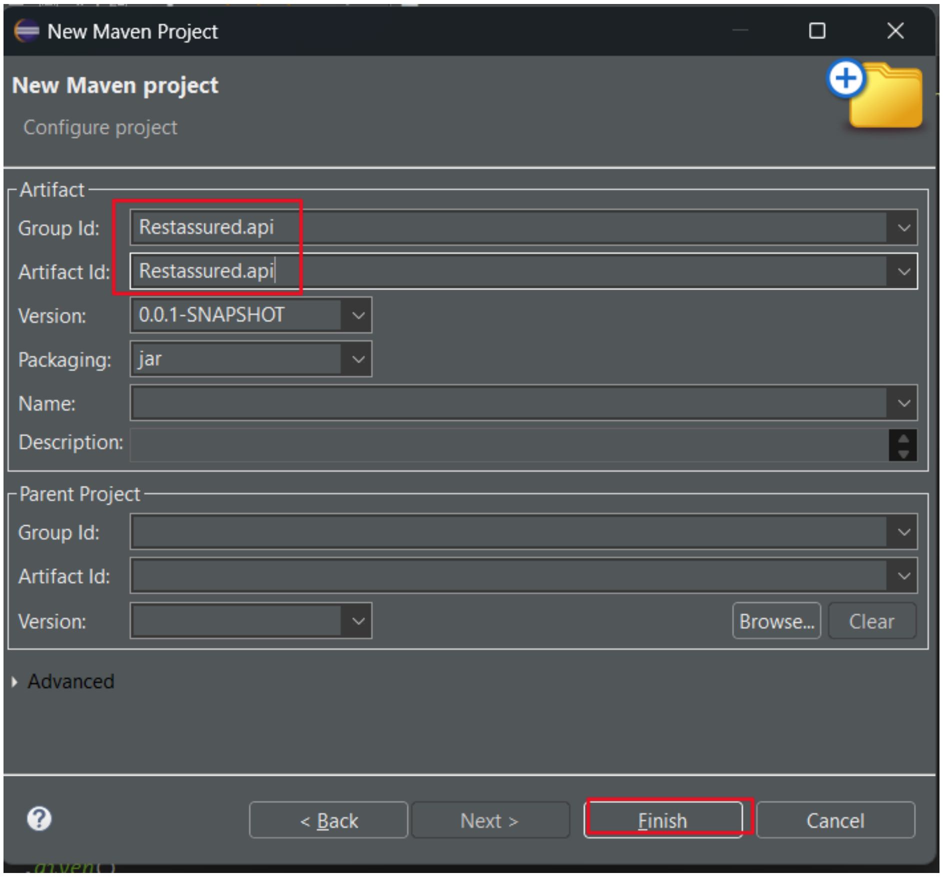
1. Open Eclipse
2. Go to: File → New → Project → Select **Maven Project**



3.Click Next → Check Create a simple project → Click Next

4. Fill in:

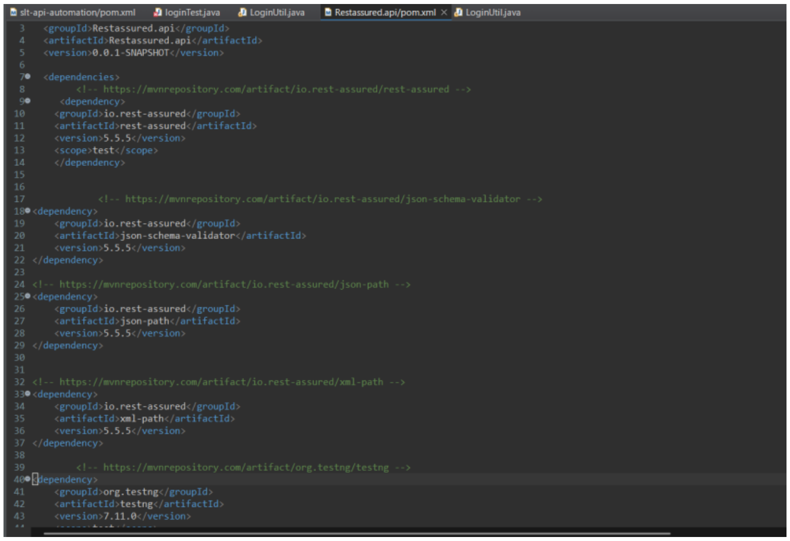
- **Group ID**: com.myslt
- **Artifact ID**: slt-api-automation
- Click Finish



Group Id: Restassured.api
Artifact Id: Restassured.api
Version: 0.0.1-SNAPSHOT
Packaging: jar
Name:
Description:

Parent Project
Group Id:
Artifact Id:
Version:

Step 2: Add Dependencies in pom.xml

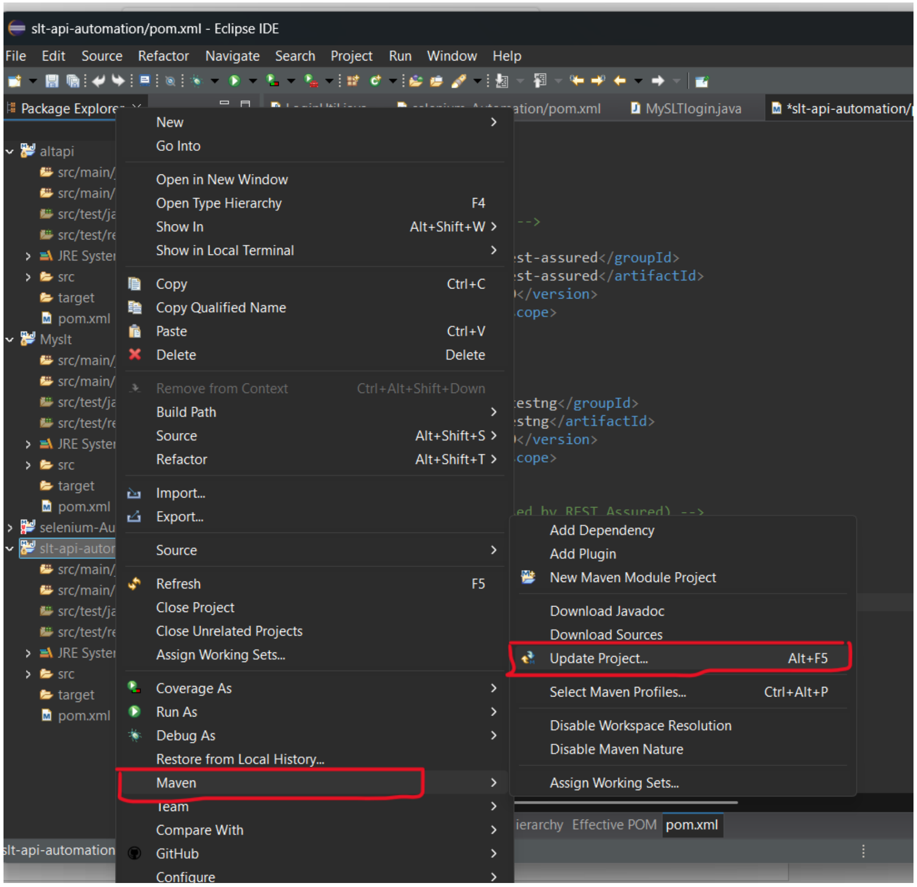


```
  <groupId>Restassured.api</groupId>
  <artifactId>Restassured.api</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencies>
        <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
     <dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>5.5.5</version>
    <scope>test</scope>
    </dependency>


            <!-- https://mvnrepository.com/artifact/io.rest-assured/json-schema-validator -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>json-schema-validator</artifactId>
    <version>5.5.5</version>
</dependency>

<!-- https://mvnrepository.com/artifact/io.rest-assured/json-path -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>json-path</artifactId>
    <version>5.5.5</version>
</dependency>


<!-- https://mvnrepository.com/artifact/io.rest-assured/xml-path -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>xml-path</artifactId>
    <version>5.5.5</version>
</dependency>

        <!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.11.0</version>
```

Step 3: Update Maven Project

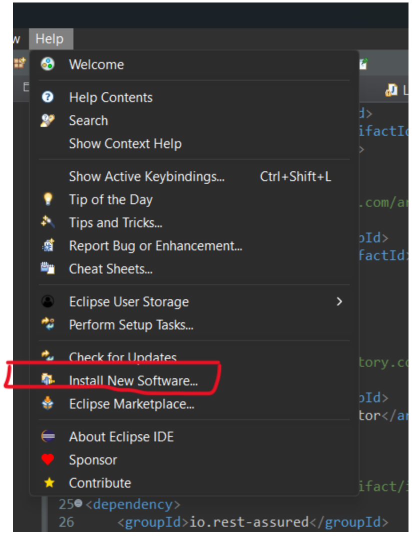
1.Right-click the project → Maven → Update Project

2. Wait for dependencies to download (check in Maven Dependencies folder)



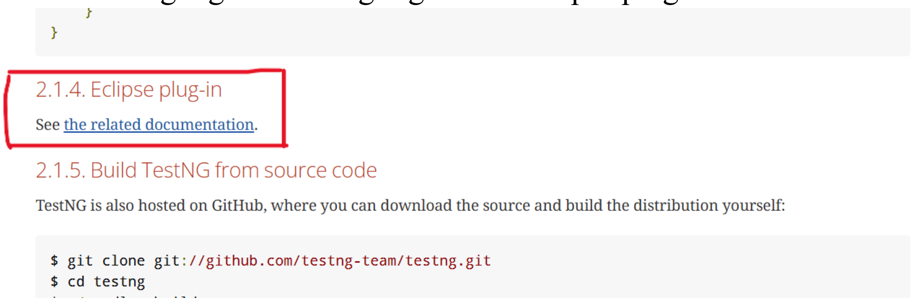
Step 4: Add Test plugin for eclipse

1.click on help & install new software



```
<dependency>
    <groupId>io.rest-assured</groupId>
```

2.Then search on google as Testng.org & find eclipse plugin document



```
    }
}
```

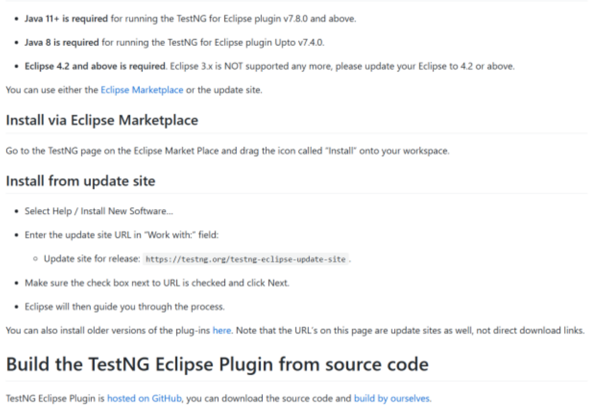
2.1.4. Eclipse plug-in

See the related documentation.

2.1.5. Build TestNG from source code

TestNG is also hosted on GitHub, where you can download the source and build the distribution yourself:

```
$ git clone git://github.com/testng-team/testng.git
$ cd testng
```



- **Java 11+ is required** for running the TestNG for Eclipse plugin v7.8.0 and above.
- **Java 8 is required** for running the TestNG for Eclipse plugin Upto v7.4.0.
- **Eclipse 4.2 and above is required**. Eclipse 3.x is NOT supported any more, please update your Eclipse to 4.2 or above.

You can use either the Eclipse Marketplace or the update site.

Install via Eclipse Marketplace

Go to the TestNG page on the Eclipse Market Place and drag the icon called "Install" onto your workspace.

Install from update site

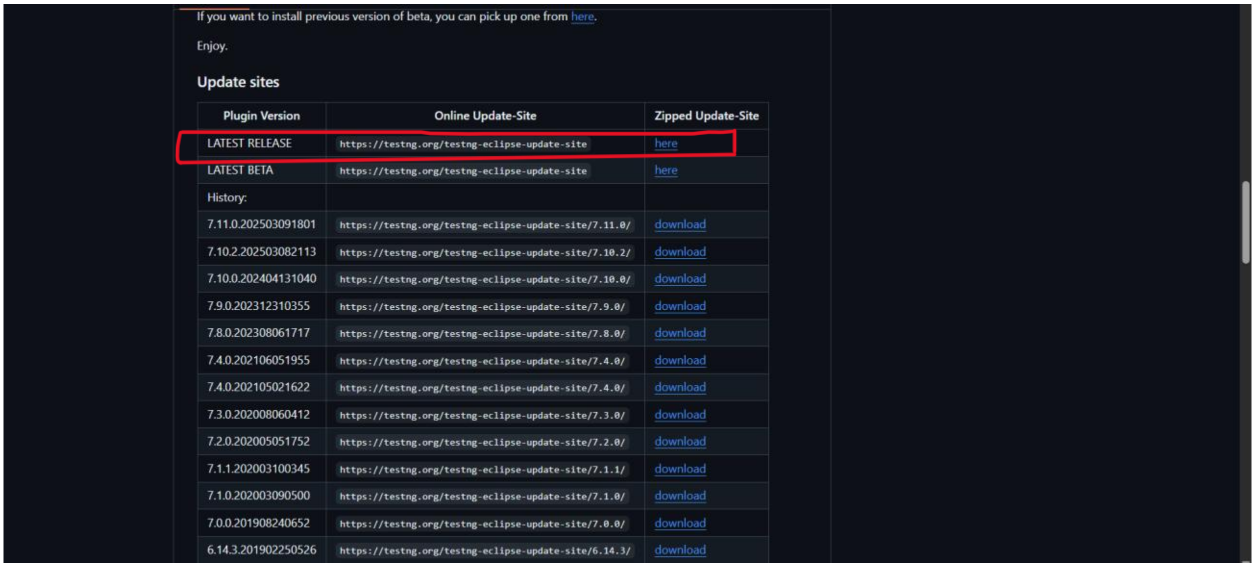
- Select Help / Install New Software...
- Enter the update site URL in "Work with:" field:
  - Update site for release: `https://testng.org/testng-eclipse-update-site` .
- Make sure the check box next to URL is checked and click Next.
- Eclipse will then guide you through the process.

You can also install older versions of the plug-ins here. Note that the URL's on this page are update sites as well, not direct download links.

Build the TestNG Eclipse Plugin from source code

TestNG Eclipse Plugin is hosted on GitHub, you can download the source code and build by ourselves.

3.Then you will get git hub page like this .Copy the url of latest release from here



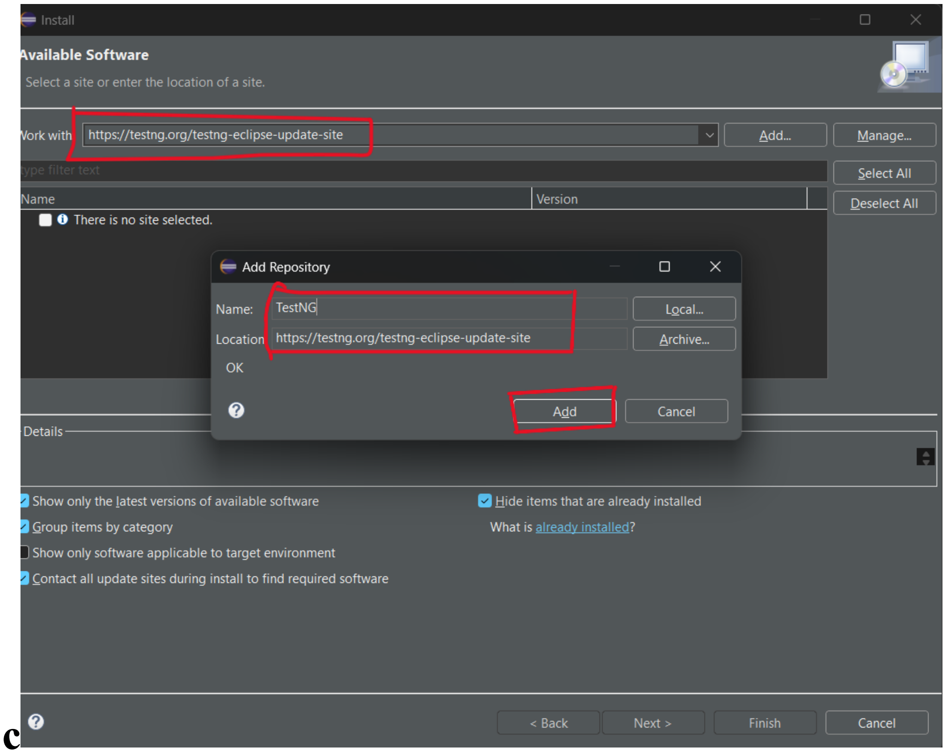
If you want to install previous version of beta, you can pick up one from here.

Enjoy.

Update sites

Plugin Version	Online Update-Site	Zipped Update-Site
LATEST RELEASE	`https://testng.org/testng-eclipse-update-site`	here
LATEST BETA	`https://testng.org/testng-eclipse-update-site`	here
History:		
7.11.0.202503091801	`https://testng.org/testng-eclipse-update-site/7.11.0/`	download
7.10.2.202503082113	`https://testng.org/testng-eclipse-update-site/7.10.2/`	download
7.10.0.202404131040	`https://testng.org/testng-eclipse-update-site/7.10.0/`	download
7.9.0.202312310355	`https://testng.org/testng-eclipse-update-site/7.9.0/`	download
7.8.0.202308061717	`https://testng.org/testng-eclipse-update-site/7.8.0/`	download
7.4.0.202106051955	`https://testng.org/testng-eclipse-update-site/7.4.0/`	download
7.4.0.202105021622	`https://testng.org/testng-eclipse-update-site/7.4.0/`	download
7.3.0.202008060412	`https://testng.org/testng-eclipse-update-site/7.3.0/`	download
7.2.0.202005051752	`https://testng.org/testng-eclipse-update-site/7.2.0/`	download
7.1.1.202003100345	`https://testng.org/testng-eclipse-update-site/7.1.1/`	download
7.1.0.202003090500	`https://testng.org/testng-eclipse-update-site/7.1.0/`	download
7.0.0.201908240652	`https://testng.org/testng-eclipse-update-site/7.0.0/`	download
6.14.3.201902250526	`https://testng.org/testng-eclipse-update-site/6.14.3/`	download

4.Then add url in here & click on add .After click on finish button



Install

Available Software

Select a site or enter the location of a site.

ork with: https://testng.org/testng-eclipse-update-site

☐ There is no site selected.

Add Repository

Name: TestNG

Location: https://testng.org/testng-eclipse-update-site

OK

☑ Show only the latest versions of available software

☑ Group items by category

☐ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

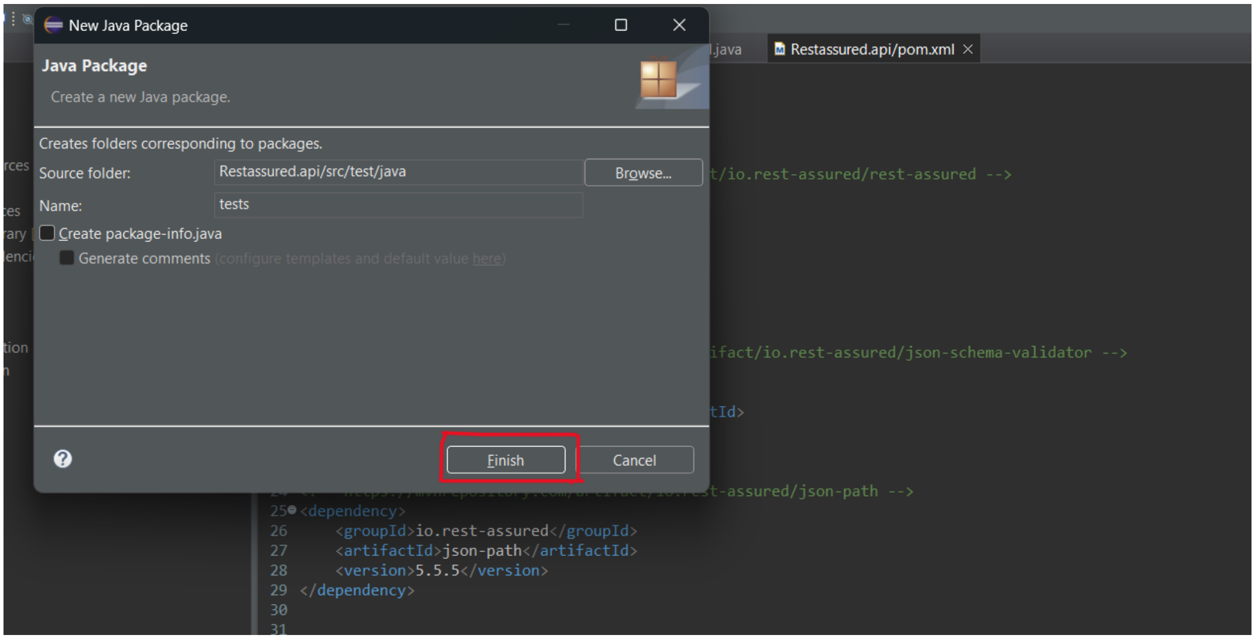
☑ Hide items that are already installed

What is already installed?

c

Step 5: Create the Package and Test Class

1. Right-click src/test/java → New → Package → Name it: tests



Creates folders corresponding to packages.

Source folder: Restassured.api/src/test/java

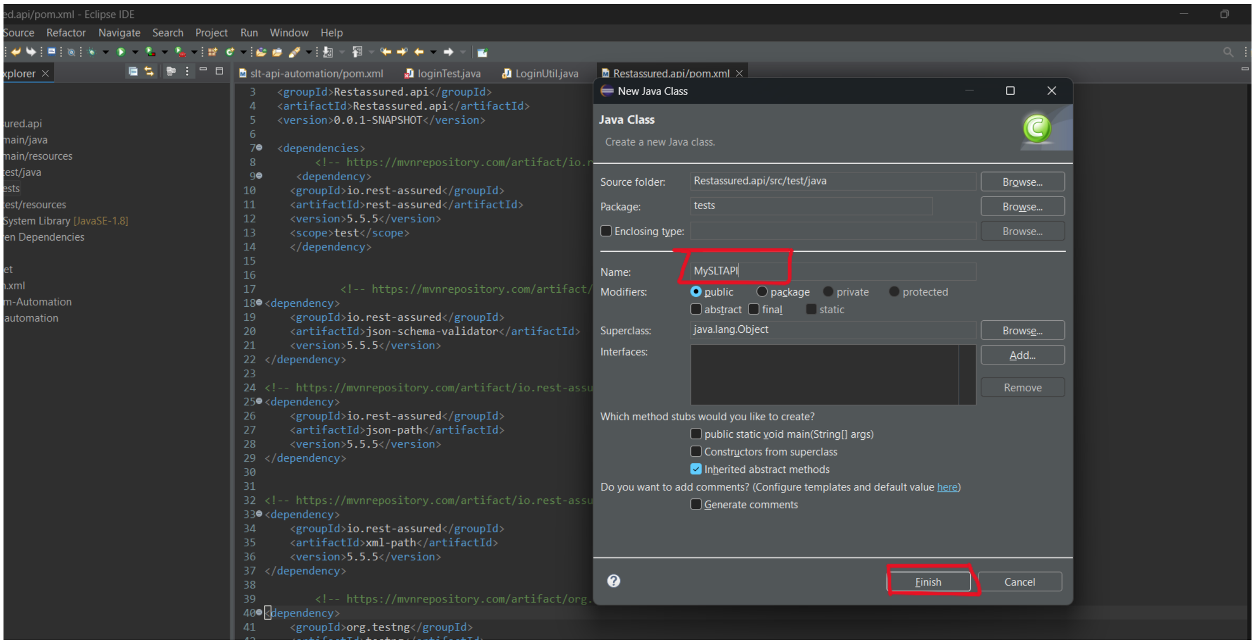
Name: tests

□ Create package-info.java

□ Generate comments (configure templates and default value here)

```
t/io.rest-assured/rest-assured -->

ifact/io.rest-assured/json-schema-validator -->

tId>

t-assured/json-path -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>json-path</artifactId>
    <version>5.5.5</version>
</dependency>
```

2. Right-click on the package → New → Class → Name it: MySLTAPI



```
<groupId>Restassured.api</groupId>
<artifactId>Restassured.api</artifactId>
<version>0.0.1-SNAPSHOT</version>

<dependencies>
      <!-- https://mvnrepository.com/artifact/io.r
   <dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>5.5.5</version>
    <scope>test</scope>
    </dependency>


          <!-- https://mvnrepository.com/artifact/
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>json-schema-validator</artifactId>
    <version>5.5.5</version>
</dependency>

<!-- https://mvnrepository.com/artifact/io.rest-assu
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>json-path</artifactId>
    <version>5.5.5</version>
</dependency>


<!-- https://mvnrepository.com/artifact/io.rest-assu
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>xml-path</artifactId>
    <version>5.5.5</version>
</dependency>

      <!-- https://mvnrepository.com/artifact/org.
<dependency>
    <groupId>org.testng</groupId>
```

Source folder: Restassured.api/src/test/java

Package: tests

□ Enclosing type:

Name: MySLTAPI

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

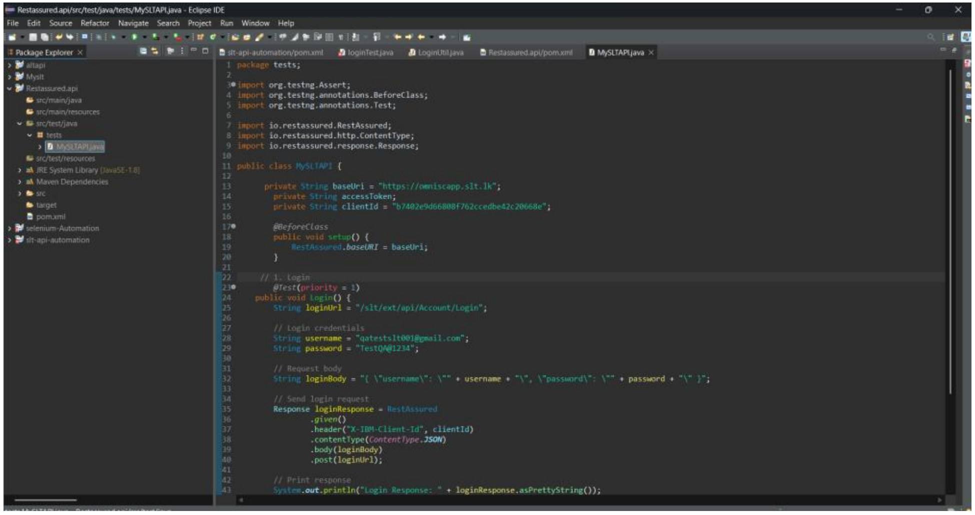
□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

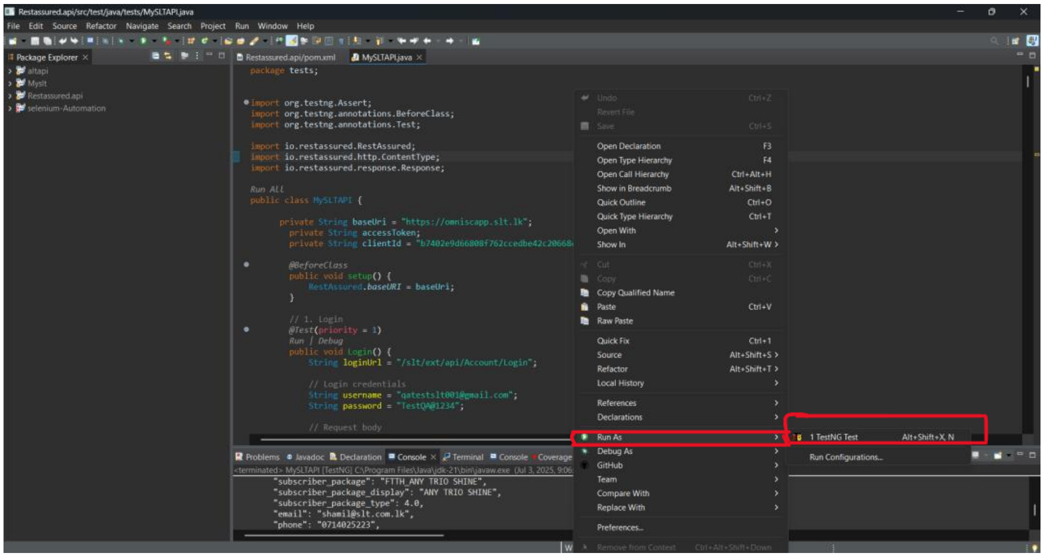
□ Generate comments

3.then add your script like this



```java
package tests;

import org.testng.Assert;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import io.restassured.response.Response;

public class MySLTAPI {

      private String baseUri = "https://omniscapp.slt.lk";
        private String accessToken;
        private String clientId = "b7402e9d66808f762ccedbe42c20668e";

        @BeforeClass
        public void setup() {
            RestAssured.baseURI = baseUri;
        }

     // 1. Login
        @Test(priority = 1)
    public void Login() {
        String loginUrl = "/slt/ext/api/Account/Login";

        // Login credentials
        String username = "qatestslt001@gmail.com";
        String password = "TestQA@1234";

        // Request body
        String loginBody = "{ \"username\": \"" + username + "\", \"password\": \"" + password + "\" }";

        // Send login request
        Response loginResponse = RestAssured
                .given()
                .header("X-IBM-Client-Id", clientId)
                .contentType(ContentType.JSON)
                .body(loginBody)
                .post(loginUrl);

        // Print response
        System.out.println("Login Response: " + loginResponse.asPrettyString());
```

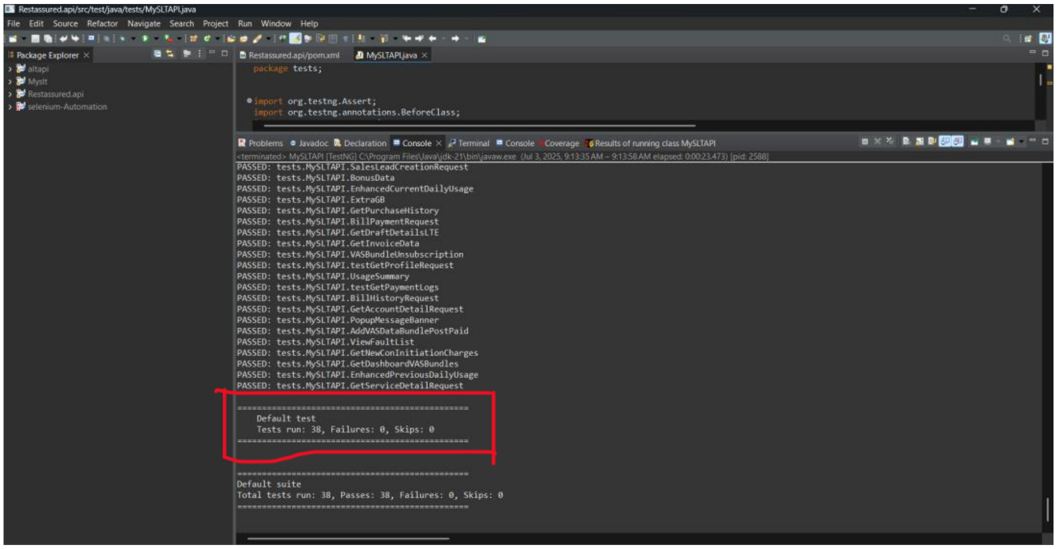
4.After adding script right click on it & click on **Run as** button .then you will receive TestNG test .click on it



```java
package tests;

import org.testng.Assert;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import io.restassured.response.Response;

Run All
public class MySLTAPI {

      private String baseUri = "https://omniscapp.slt.lk";
        private String accessToken;
        private String clientId = "b7402e9d66808f762ccedbe42c20668

        @BeforeClass
        public void setup() {
            RestAssured.baseURI = baseUri;
        }

        // 1. Login
        @Test(priority = 1)
        Run | Debug
        public void Login() {
            String loginUrl = "/slt/ext/api/Account/Login";

            // Login credentials
            String username = "qatestslt001@gmail.com";
            String password = "TestQA@1234";

            // Request body
```

```
<terminated> MySLTAPI [TestNG] C:\Program Files\Java\jdk-21\bin\javaw.exe (Jul 3, 2025, 9:06
        "subscriber_package": "FTTH_ANY TRIO SHINE",
        "subscriber_package_display": "ANY TRIO SHINE",
        "subscriber_package_type": 4.0,
        "email": "shamil@slt.com.lk",
        "phone": "0714025223",
```

5.Then script will run this & test result will display buttom of the script in console area like this .


Restassured.api/src/test/java/tests/MySLTAPI.java
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
altapi
Myslt
Restassured.api
selenium-Automation
Restassured.api/pom.xml
MySLTAPI.java
Problems Javadoc Declaration Console Terminal Console Coverage Results of running class MySLTAPI

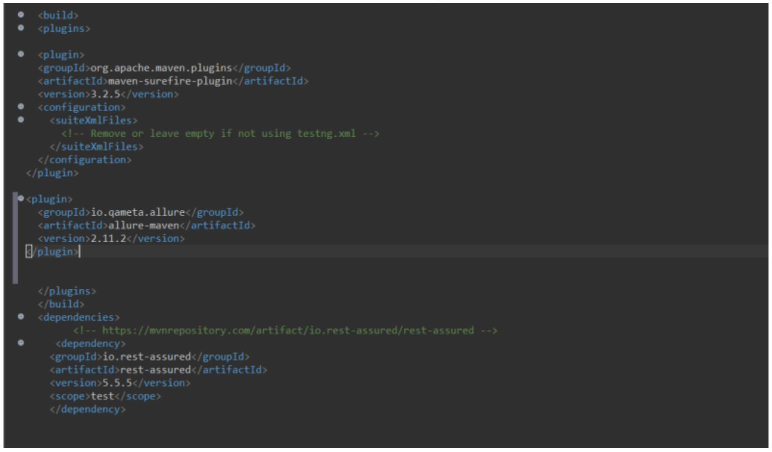
```java
package tests;


import org.testng.Assert;
import org.testng.annotations.BeforeClass;
```

```
<terminated> MySLTAPI [TestNG] C:\Program Files\Java\jdk-21\bin\javaw.exe (Jul 3, 2025, 9:13:35 AM – 9:13:58 AM elapsed: 0:00:23.473) [pid: 2588]
PASSED: tests.MySLTAPI.SalesLeadCreationRequest
PASSED: tests.MySLTAPI.BonusData
PASSED: tests.MySLTAPI.EnhancedCurrentDailyUsage
PASSED: tests.MySLTAPI.ExtraGB
PASSED: tests.MySLTAPI.GetPurchaseHistory
PASSED: tests.MySLTAPI.BillPaymentRequest
PASSED: tests.MySLTAPI.GetDraftDetailsLTE
PASSED: tests.MySLTAPI.GetInvoiceData
PASSED: tests.MySLTAPI.VASBundleUnsubscription
PASSED: tests.MySLTAPI.testGetProfileRequest
PASSED: tests.MySLTAPI.UsageSummary
PASSED: tests.MySLTAPI.testGetPaymentLogs
PASSED: tests.MySLTAPI.BillHistoryRequest
PASSED: tests.MySLTAPI.GetAccountDetailRequest
PASSED: tests.MySLTAPI.PopupMessageBanner
PASSED: tests.MySLTAPI.AddVASDataBundlePostPaid
PASSED: tests.MySLTAPI.ViewFaultList
PASSED: tests.MySLTAPI.GetNewConInitiationCharges
PASSED: tests.MySLTAPI.GetDashboardVASBundles
PASSED: tests.MySLTAPI.EnhancedPreviousDailyUsage
PASSED: tests.MySLTAPI.GetServiceDetailRequest

===============================================
    Default test
    Tests run: 38, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 38, Passes: 38, Failures: 0, Skips: 0
===============================================
```

How to get Allure report for rest assure

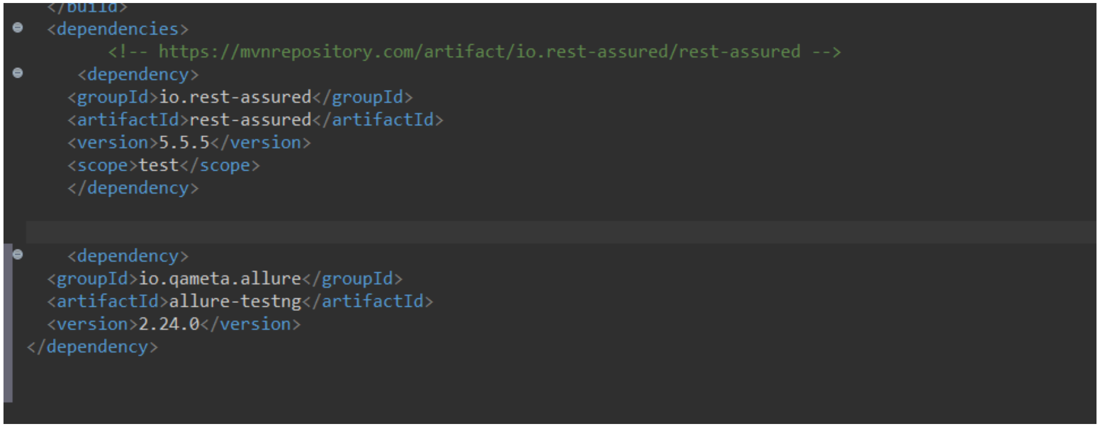
1. Add Allure Maven Plugin to pom.xml

Add this inside your <build><plugins> section:



```xml
<build>
<plugins>

<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>3.2.5</version>
<configuration>
  <suiteXmlFiles>
    <!-- Remove or leave empty if not using testng.xml -->
  </suiteXmlFiles>
</configuration>
</plugin>

<plugin>
<groupId>io.qameta.allure</groupId>
<artifactId>allure-maven</artifactId>
<version>2.11.2</version>
</plugin>


</plugins>
</build>
<dependencies>
      <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
   <dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>rest-assured</artifactId>
  <version>5.5.5</version>
  <scope>test</scope>
  </dependency>
```

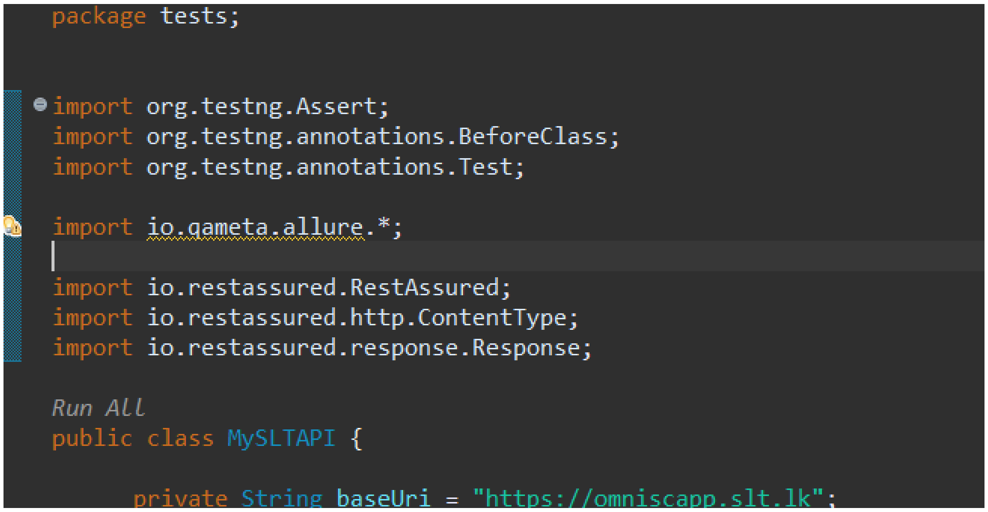
if you're using **TestNG**:
add this under dependencies



```
</build>
<dependencies>
      <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
   <dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>rest-assured</artifactId>
  <version>5.5.5</version>
  <scope>test</scope>
  </dependency>


  <dependency>
<groupId>io.qameta.allure</groupId>
<artifactId>allure-testng</artifactId>
<version>2.24.0</version>
</dependency>
```

2.Annotate Your Test Classes

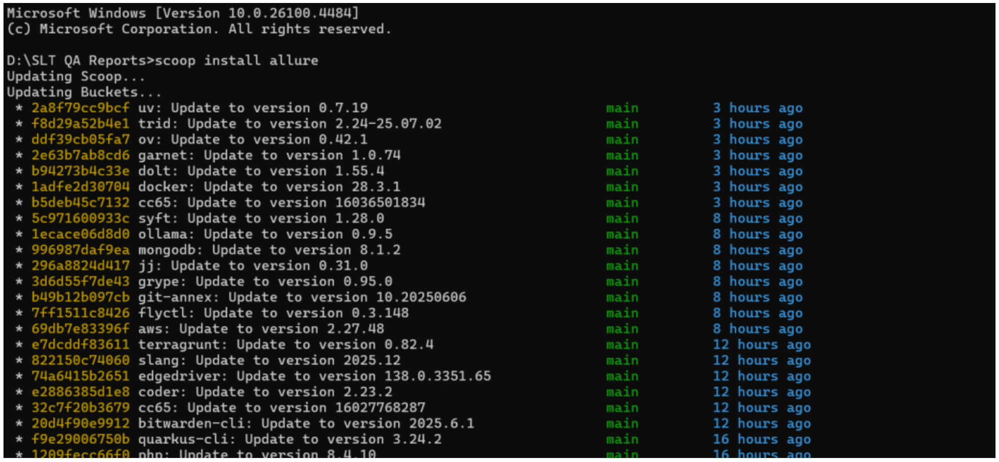
Use Allure annotations like:



```java
package tests;


import org.testng.Assert;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;

import io.qameta.allure.*;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import io.restassured.response.Response;

Run All
public class MySLTAPI {

      private String baseUri = "https://omniscapp.slt.lk";
```

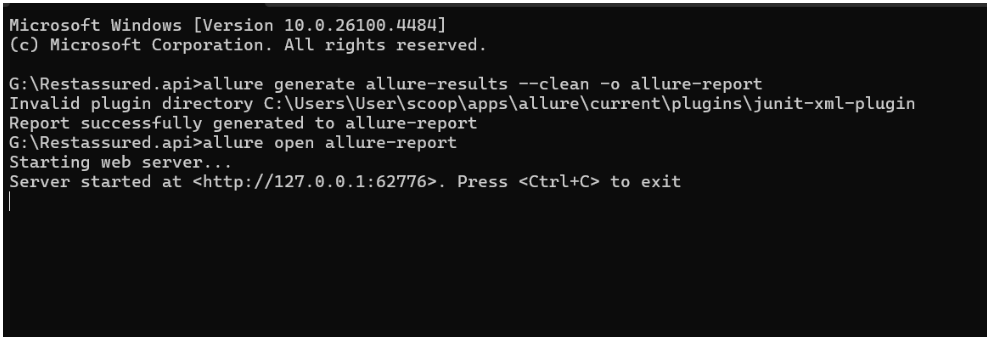
3. Run Your Tests

4. Generate Allure Report
   Option A: Install Allure CLI (Recommended)
   Install Allure CLI globally:
   Windows (via Scoop):



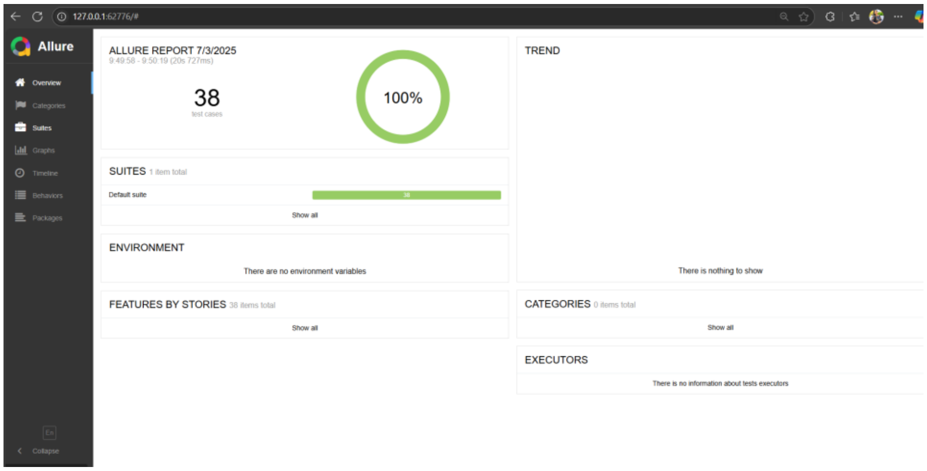
```
Microsoft Windows [Version 10.0.26100.4484]
(c) Microsoft Corporation. All rights reserved.

D:\SLT QA Reports>scoop install allure
Updating Scoop...
Updating Buckets...
 * 2a8f79cc9bcf uv: Update to version 0.7.19                              main         3 hours ago
 * f8d29a52b4e1 trid: Update to version 2.24-25.07.02                     main         3 hours ago
 * ddf39cb05fa7 ov: Update to version 0.42.1                              main         3 hours ago
 * 2e63b7ab8cd6 garnet: Update to version 1.0.74                          main         3 hours ago
 * b94273b4c33e dolt: Update to version 1.55.4                            main         3 hours ago
 * 1adfe2d30704 docker: Update to version 28.3.1                          main         3 hours ago
 * b5deb45c7132 cc65: Update to version 16036501834                       main         3 hours ago
 * 5c971600933c syft: Update to version 1.28.0                            main         8 hours ago
 * 1ecace06d8d0 ollama: Update to version 0.9.5                           main         8 hours ago
 * 996987daf9ea mongodb: Update to version 8.1.2                          main         8 hours ago
 * 296a8824d417 jj: Update to version 0.31.0                              main         8 hours ago
 * 3d6d55f7de43 grype: Update to version 0.95.0                           main         8 hours ago
 * b49b12b097cb git-annex: Update to version 10.20250606                  main         8 hours ago
 * 7ff1511c8426 flyctl: Update to version 0.3.148                         main         8 hours ago
 * 69db7e83396f aws: Update to version 2.27.48                            main         8 hours ago
 * e7dcddf83611 terragrunt: Update to version 0.82.4                      main         12 hours ago
 * 822150c74060 slang: Update to version 2025.12                          main         12 hours ago
 * 74a6415b2651 edgedriver: Update to version 138.0.3351.65               main         12 hours ago
 * e2886385d1e8 coder: Update to version 2.23.2                           main         12 hours ago
 * 32c7f20b3679 cc65: Update to version 16027768287                       main         12 hours ago
 * 20d4f90e9912 bitwarden-cli: Update to version 2025.6.1                 main         12 hours ago
 * f9e29006750b quarkus-cli: Update to version 3.24.2                     main         16 hours ago
 * 1209fecc66f0 php: Update to version 8.4.10                             main         16 hours ago
```

Then run:
allure generate allure-results --clean -o allure-report
allure open allure-report



```
Microsoft Windows [Version 10.0.26100.4484]
(c) Microsoft Corporation. All rights reserved.

G:\Restassured.api>allure generate allure-results --clean -o allure-report
Invalid plugin directory C:\Users\User\scoop\apps\allure\current\plugins\junit-xml-plugin
Report successfully generated to allure-report
G:\Restassured.api>allure open allure-report
Starting web server...
Server started at <http://127.0.0.1:62776>. Press <Ctrl+C> to exit
```

Then you will receive allure report like this :



127.0.0.1:62776/#

ALLURE REPORT 7/3/2025
9:49:58 - 9:50:19 (20s 727ms)

38
test cases

100%

SUITES 1 item total

Default suite 38

Show all

ENVIRONMENT

There are no environment variables

FEATURES BY STORIES 38 items total

Show all

TREND

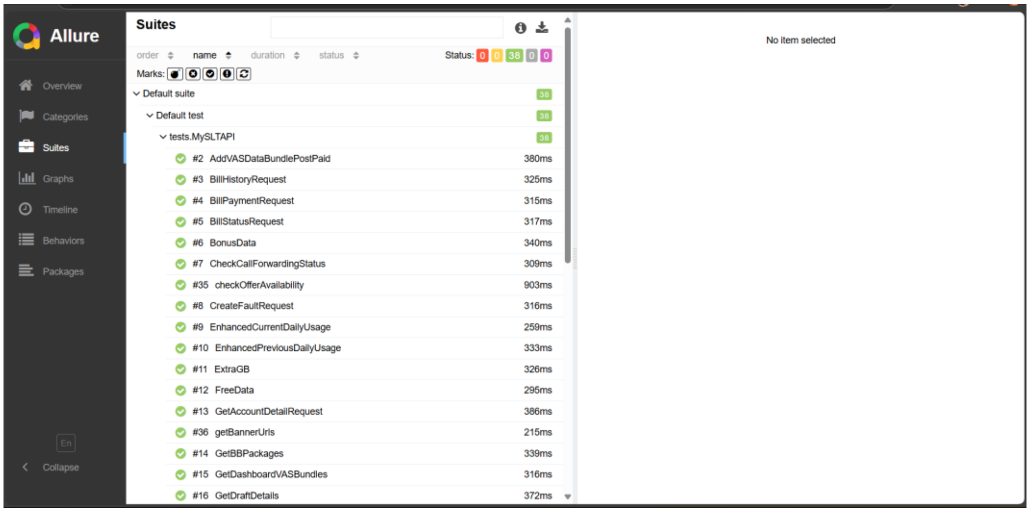
There is nothing to show

CATEGORIES 0 items total

Show all

EXECUTORS

There is no information about tests executors



Suites

order name duration status Status: 0 0 38 0 0

Marks:

Default suite 38

Default test 38

tests.MySLTAPI 38

#2 AddVASDataBundlePostPaid	380ms
#3 BillHistoryRequest	325ms
#4 BillPaymentRequest	315ms
#5 BillStatusRequest	317ms
#6 BonusData	340ms
#7 CheckCallForwardingStatus	309ms
#35 checkOfferAvailability	903ms
#8 CreateFaultRequest	316ms
#9 EnhancedCurrentDailyUsage	259ms
#10 EnhancedPreviousDailyUsage	333ms
#11 ExtraGB	326ms
#12 FreeData	295ms
#13 GetAccountDetailRequest	386ms
#36 getBannerUrls	215ms
#14 GetBBPackages	339ms
#15 GetDashboardVASBundles	316ms
#16 GetDraftDetails	372ms

No item selected