Automation Testing

Get Request and Response Automation

A Day in the Life of an Automation Test Engineer

Thomas has decided to use REST APIs in his backend server to receive and send HTTP requests and responses respectively.

He would now like to automate this testing process.

To achieve this, he must make use of the REST Assured Java library.

Learning Objectives

By the end of this lesson, you will be able to:

- Automate a GET request using REST Assured
- Validate the response in REST Assured
- Verify the response header and body

Create and Prepare the Maven Project

Getting Started with REST Assured

REST Assured is a Java-based library used to test REST APIs.

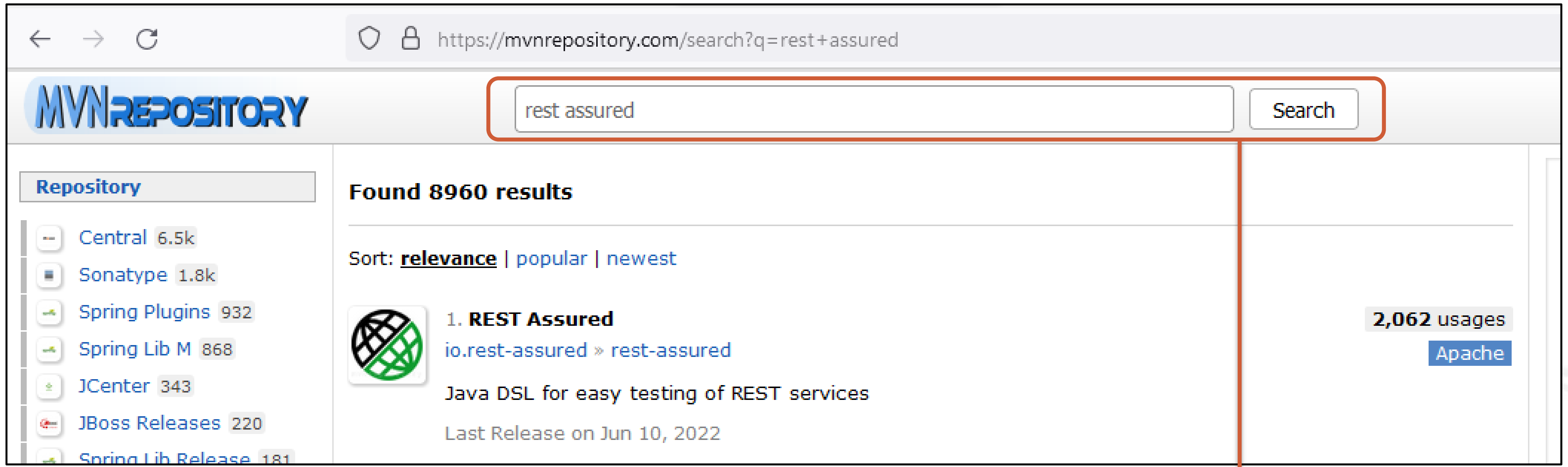
Before the users can start working with REST Assured, they will need to:

1. Install a JDK (Java Development Kit)
2. Install a Java IDE, such as Eclipse

Source: *https://www.eclipse.org/*

Maven Repository

REST Assured JAR files are available in **mvnrepository.com**.


https://mvnrepository.com/search?q=rest+assured

Found 8960 results

Sort: **relevance** | popular | newest

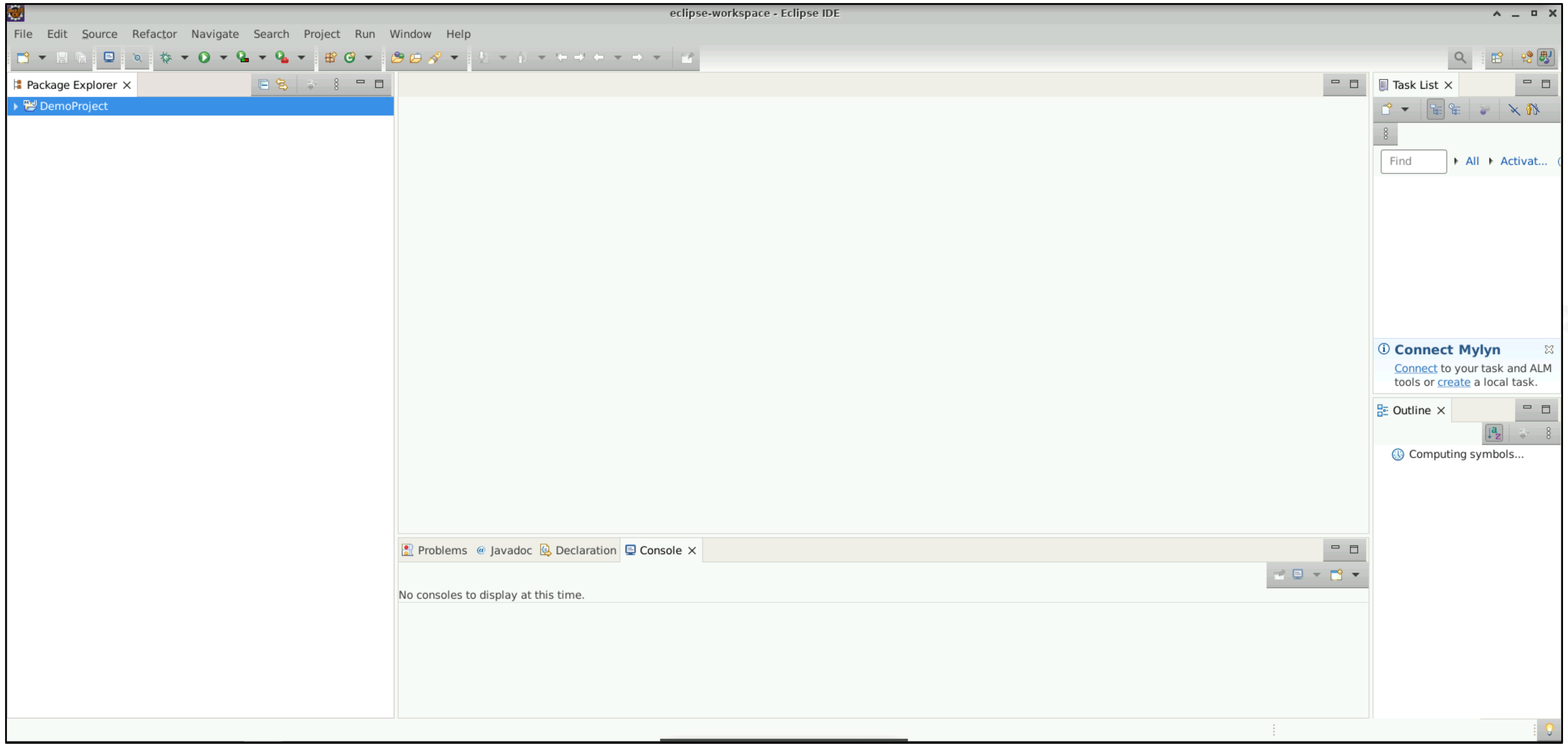
1. **REST Assured** 2,062 usages
io.rest-assured » rest-assured Apache
Java DSL for easy testing of REST services
Last Release on Jun 10, 2022

Maven Projects will automatically import JAR files specified in the **pom.xml** file.

A user must search for **rest assured** in the Maven Repository by using the search field.

Creating a Maven Project

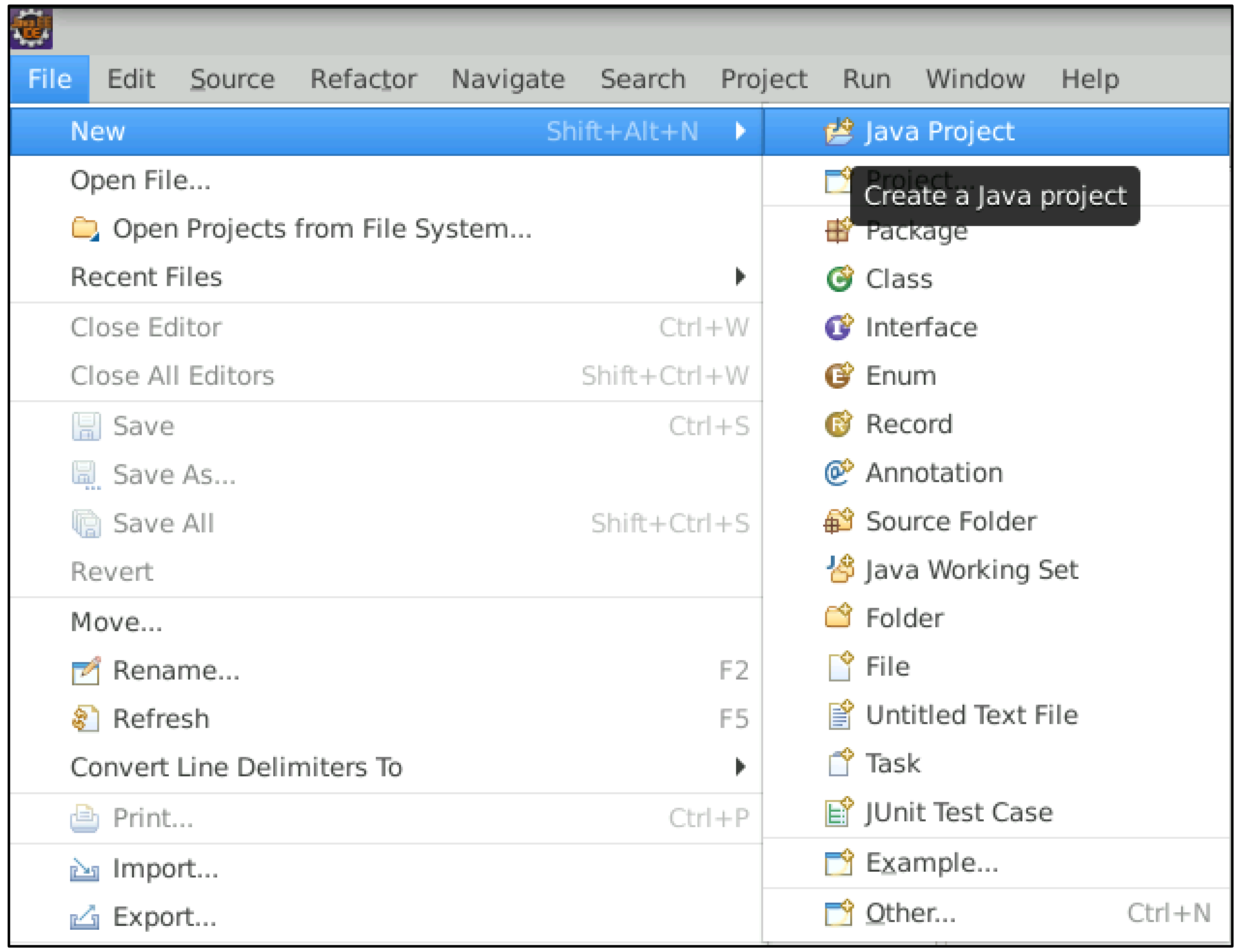
Step 1: Open Java IDE



No consoles to display at this time.

Creating a Maven Project

Step 2: Navigate to **File** > **New** > **Java Project**



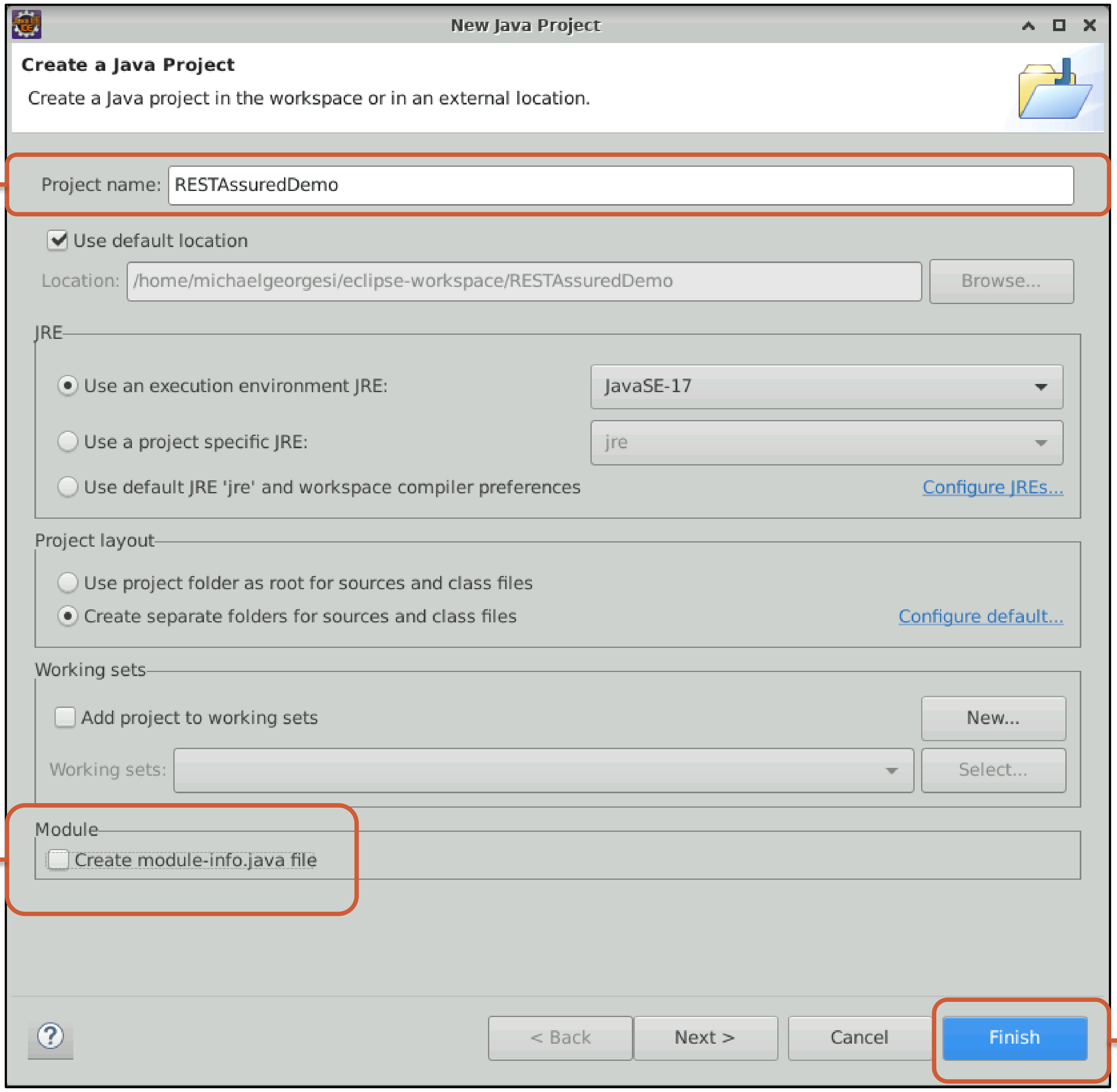
Creating a Maven Project

Step 3: Provide the necessary project details

Provide a project name

Uncheck **Create module-info.java file**

Click on **Finish** when done


New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: RESTAssuredDemo

[✓] Use default location

Location: /home/michaelgeorgesi/eclipse-workspace/RESTAssuredDemo

JRE

● Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE: jre

○ Use default JRE 'jre' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

☐ Add project to working sets

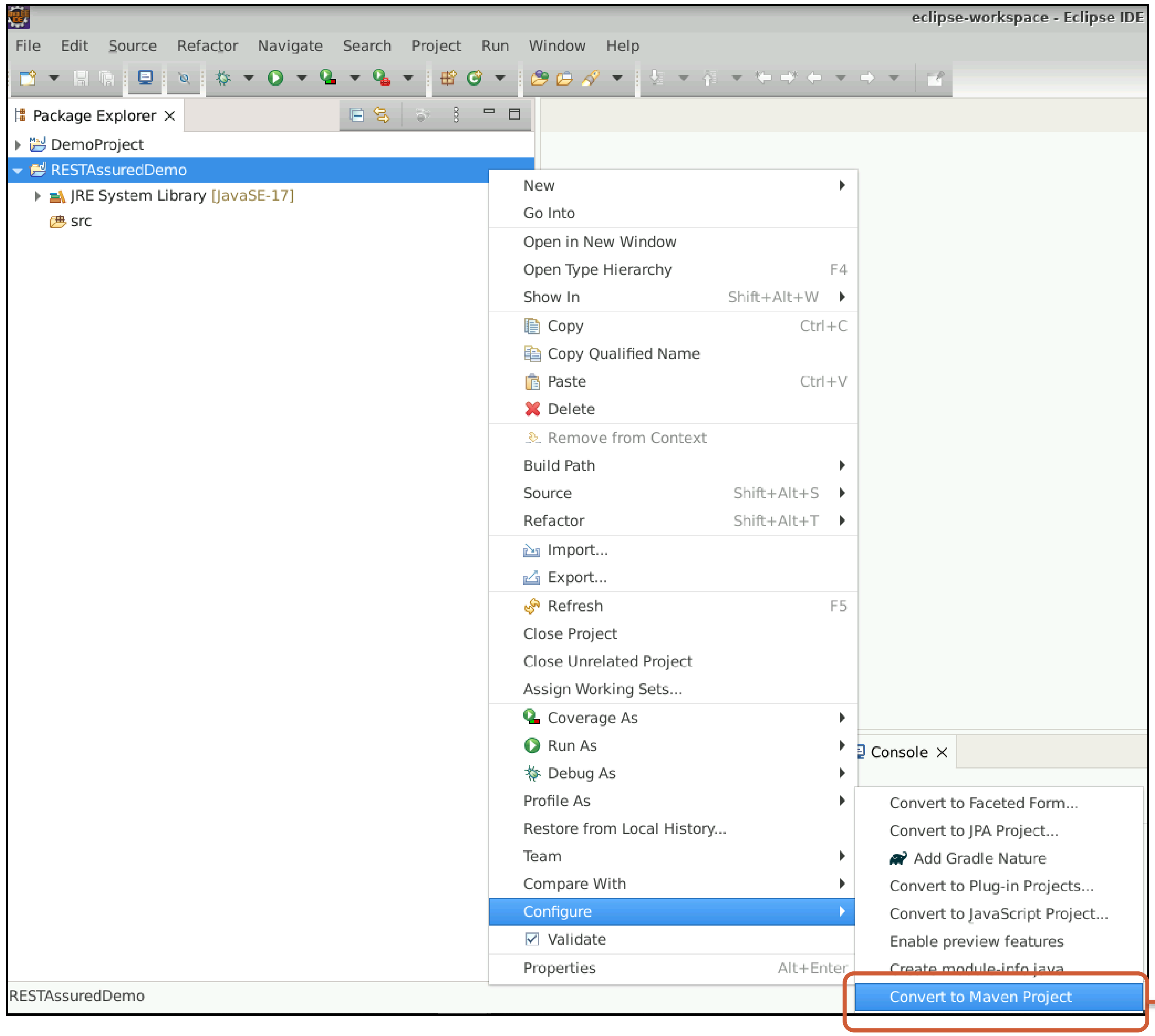
Working sets:

Module

☐ Create module-info.java file

Creating a Maven Project

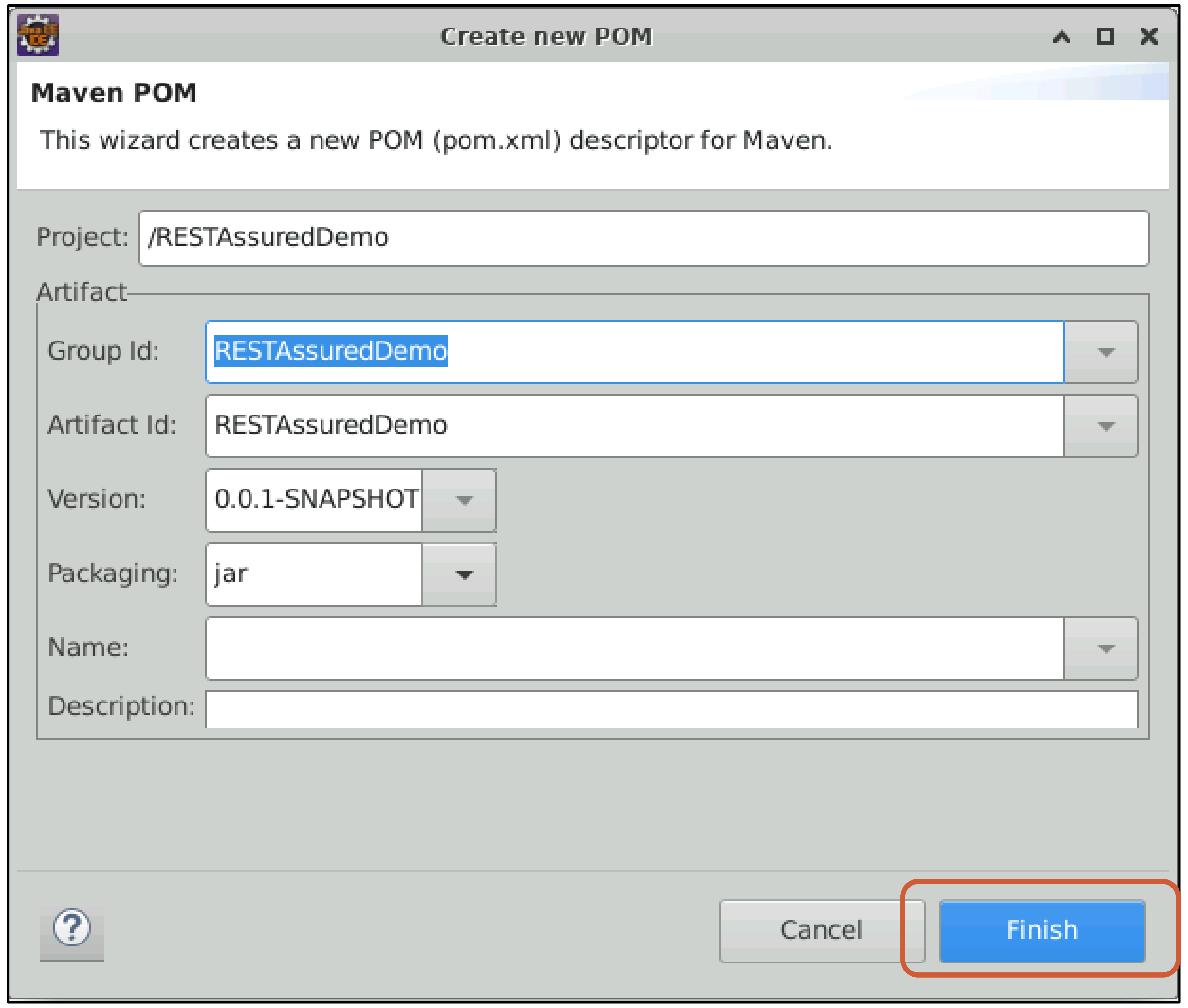
Step 4: Convert the Java Project to a Maven Project



Click on **Convert to Maven Project**

Creating a Maven Project

Step 5: A **Create new POM** dialog box shows up. Leave the entries as default and click on **Finish**



Maven POM

This wizard creates a new POM (pom.xml) descriptor for Maven.

Project: /RESTAssuredDemo

Artifact

Group Id: RESTAssuredDemo

Artifact Id: RESTAssuredDemo

Version: 0.0.1-SNAPSHOT

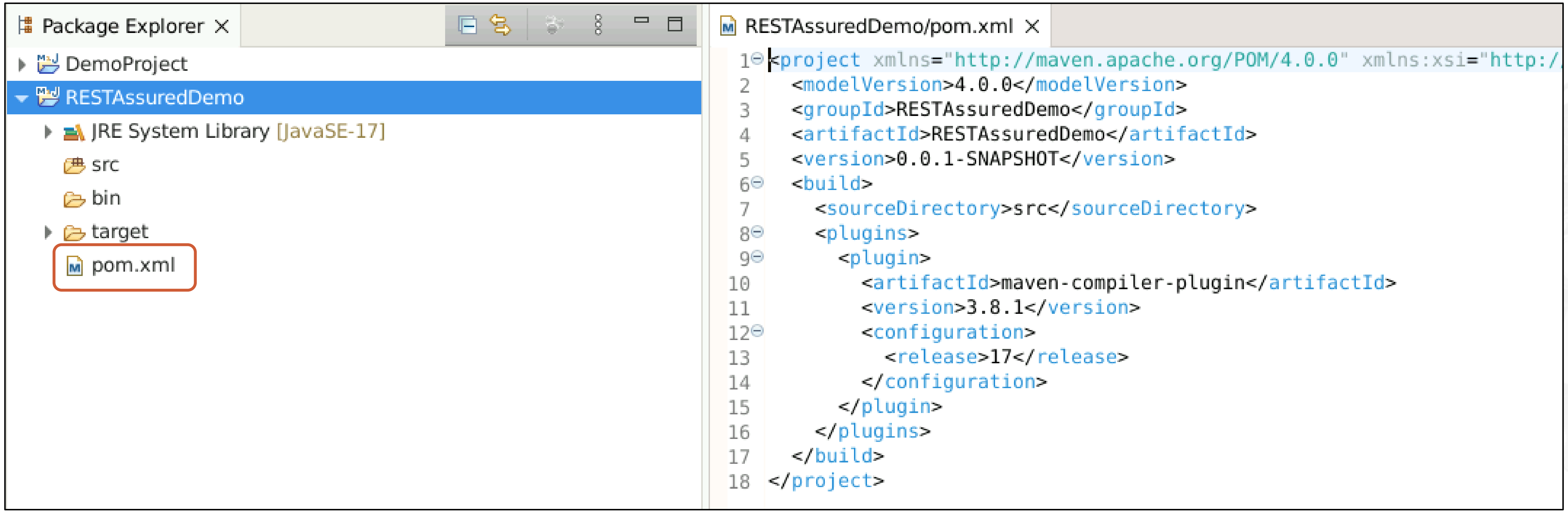
Packaging: jar

Name:

Description:

Creating a Maven Project

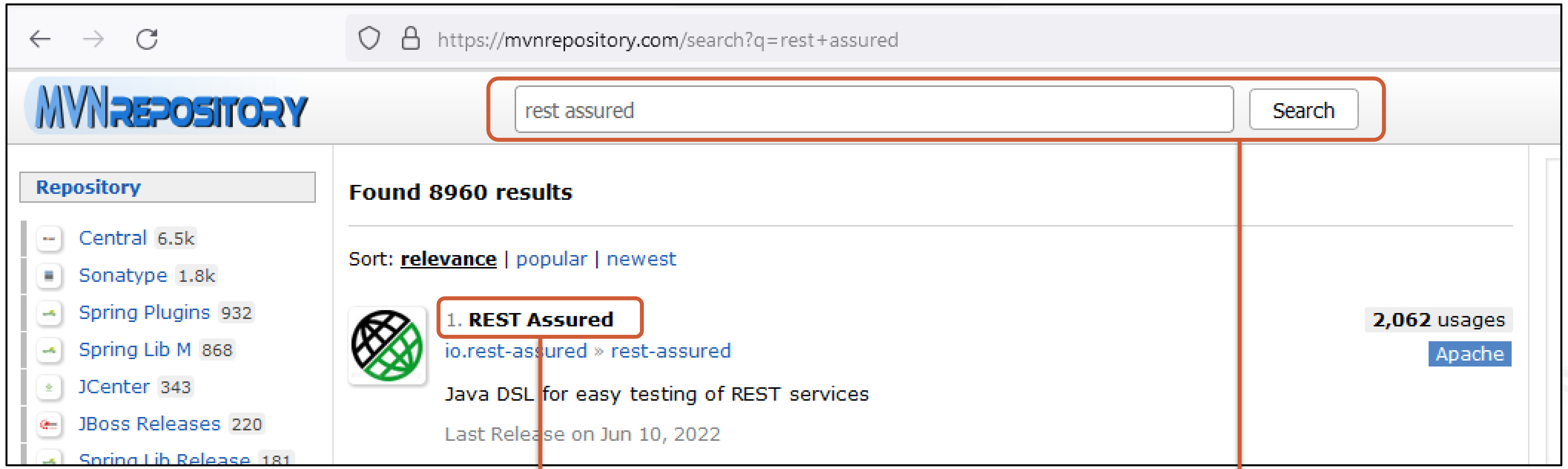
The **pom.xml** file is created for the project.



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http:/
  <modelVersion>4.0.0</modelVersion>
  <groupId>RESTAssuredDemo</groupId>
  <artifactId>RESTAssuredDemo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <release>17</release>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Importing REST Assured

Step 6: Navigate to **mvnrepository.com**, search for rest assured, and click on **REST Assured**



https://mvnrepository.com/search?q=rest+assured

Found 8960 results

Sort: relevance | popular | newest

1. REST Assured — 2,062 usages — Apache

io.rest-assured » rest-assured

Java DSL for easy testing of REST services

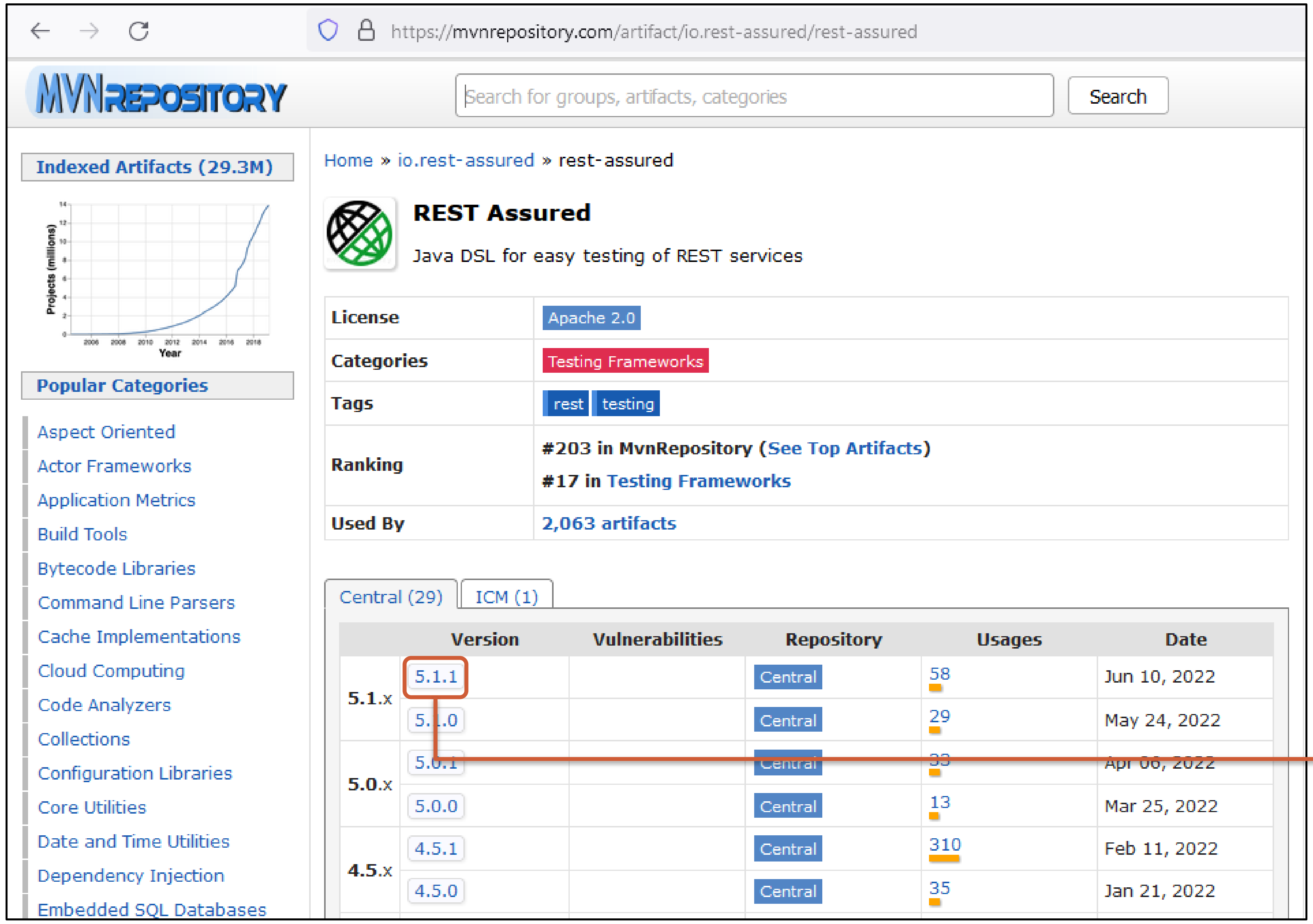
Last Release on Jun 10, 2022

Click on REST Assured

Search for **rest assured** in the Maven Repository

Importing REST Assured

Step 7: Click on the latest version of REST Assured



https://mvnrepository.com/artifact/io.rest-assured/rest-assured

MVNREPOSITORY

Search for groups, artifacts, categories | Search

Indexed Artifacts (29.3M)

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases

Home » io.rest-assured » rest-assured

REST Assured

Java DSL for easy testing of REST services

License	Apache 2.0
Categories	Testing Frameworks
Tags	rest testing
Ranking	#203 in MvnRepository (See Top Artifacts) #17 in Testing Frameworks
Used By	2,063 artifacts

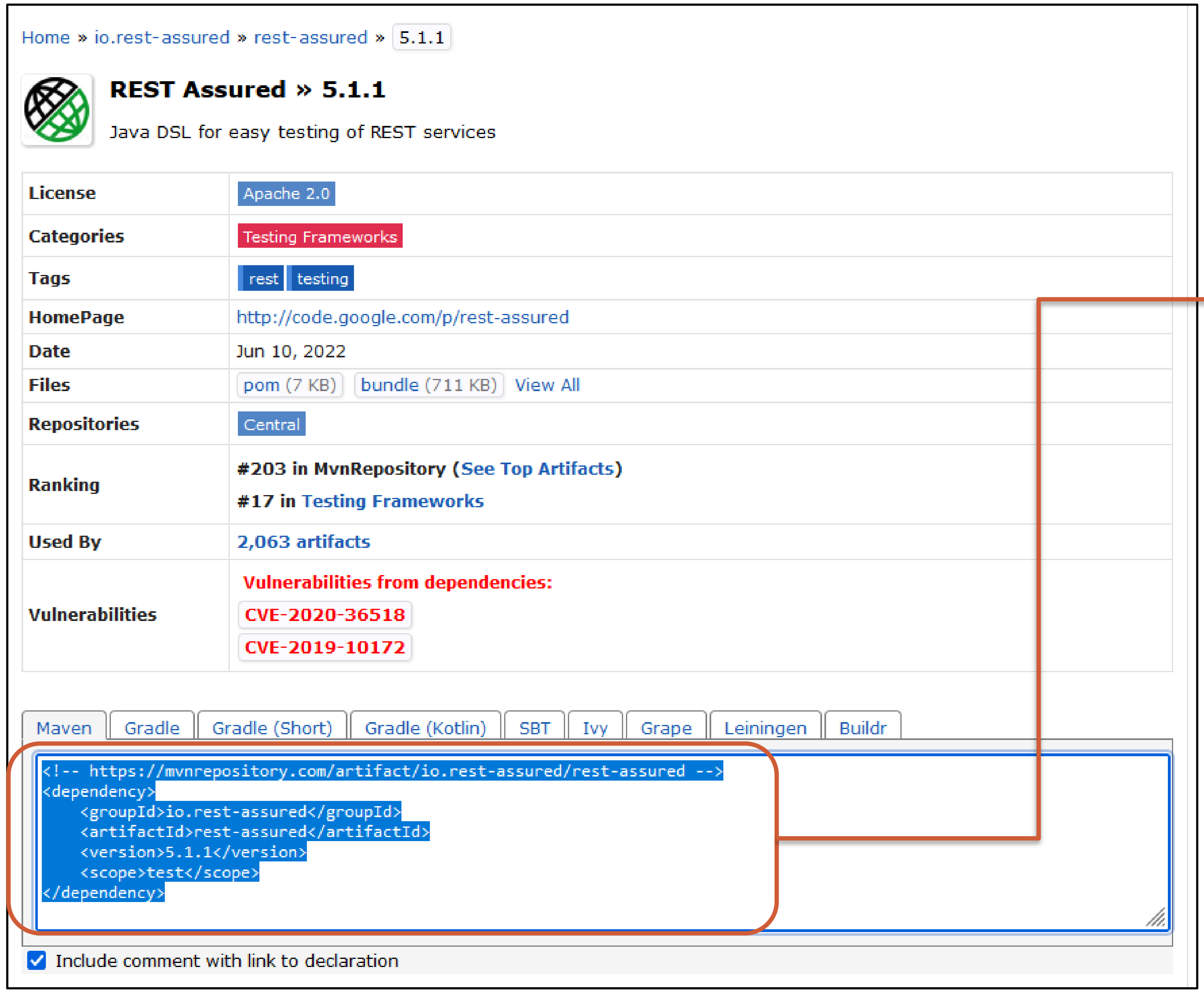
Central (29) ICM (1)

	Version	Vulnerabilities	Repository	Usages	Date
5.1.x	5.1.1		Central	58	Jun 10, 2022
	5.1.0		Central	29	May 24, 2022
5.0.x	5.0.1		Central	33	Apr 06, 2022
	5.0.0		Central	13	Mar 25, 2022
4.5.x	4.5.1		Central	310	Feb 11, 2022
	4.5.0		Central	35	Jan 21, 2022

The latest version of REST Assured

Importing REST Assured

Step 8: Copy dependency information, which will be added to the pom.xml file



Home » io.rest-assured » rest-assured » 5.1.1

REST Assured » 5.1.1

Java DSL for easy testing of REST services

License	Apache 2.0
Categories	Testing Frameworks
Tags	rest testing
HomePage	http://code.google.com/p/rest-assured
Date	Jun 10, 2022
Files	pom (7 KB) bundle (711 KB) View All
Repositories	Central
Ranking	#203 in MvnRepository (See Top Artifacts) #17 in Testing Frameworks
Used By	2,063 artifacts
Vulnerabilities	Vulnerabilities from dependencies: CVE-2020-36518 CVE-2019-10172

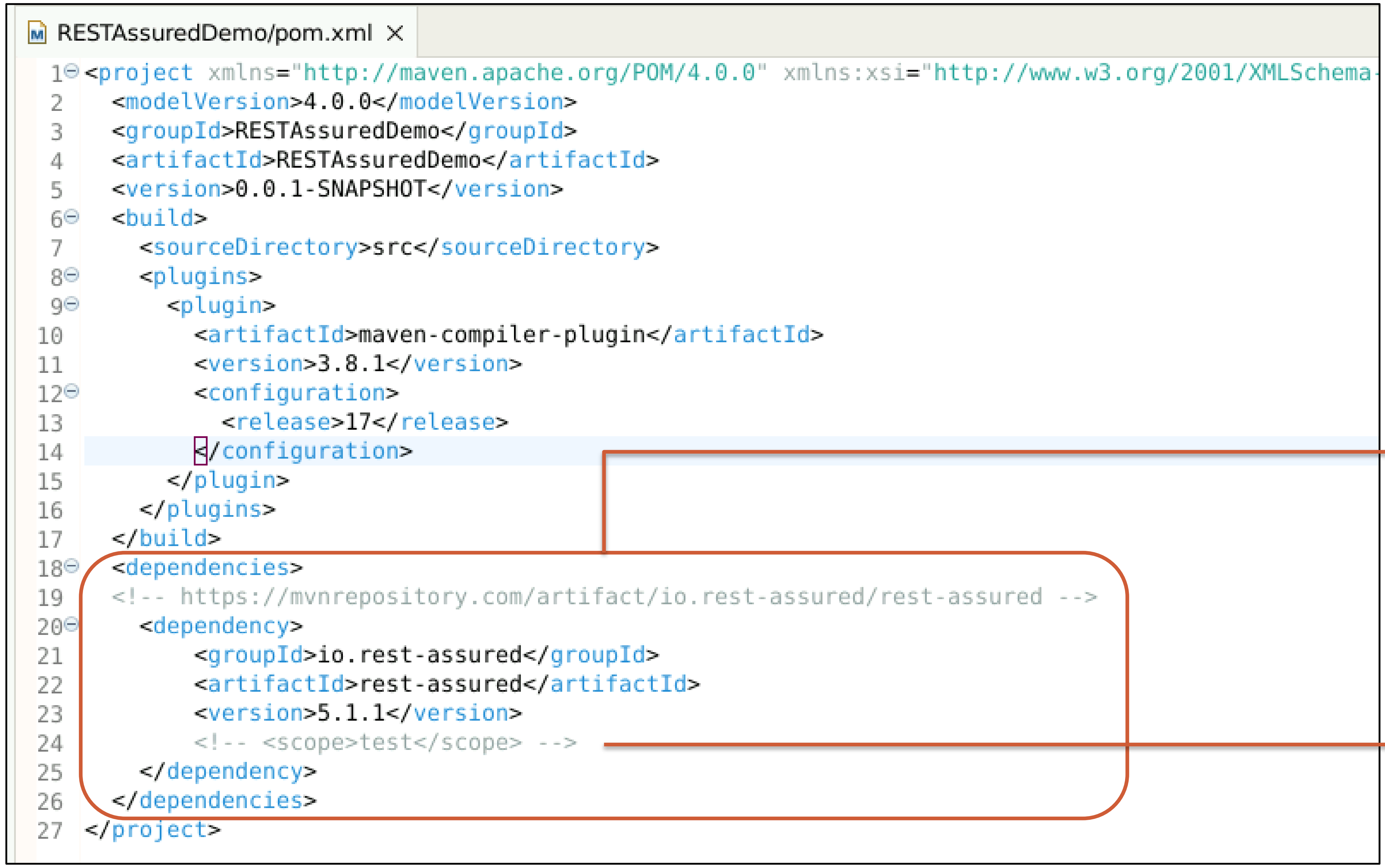
```
<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>5.1.1</version>
    <scope>test</scope>
</dependency>
```

☑ Include comment with link to declaration

Dependency information

Importing REST Assured

Step 9: Paste REST Assured dependency information within pom.xml file


RESTAssuredDemo/pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema
  <modelVersion>4.0.0</modelVersion>
  <groupId>RESTAssuredDemo</groupId>
  <artifactId>RESTAssuredDemo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <release>17</release>
        </configuration>
      </plugin>
    </plugins>
  </build>
  <dependencies>
  <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
    <dependency>
        <groupId>io.rest-assured</groupId>
        <artifactId>rest-assured</artifactId>
        <version>5.1.1</version>
        <!-- <scope>test</scope> -->
    </dependency>
  </dependencies>
</project>
```

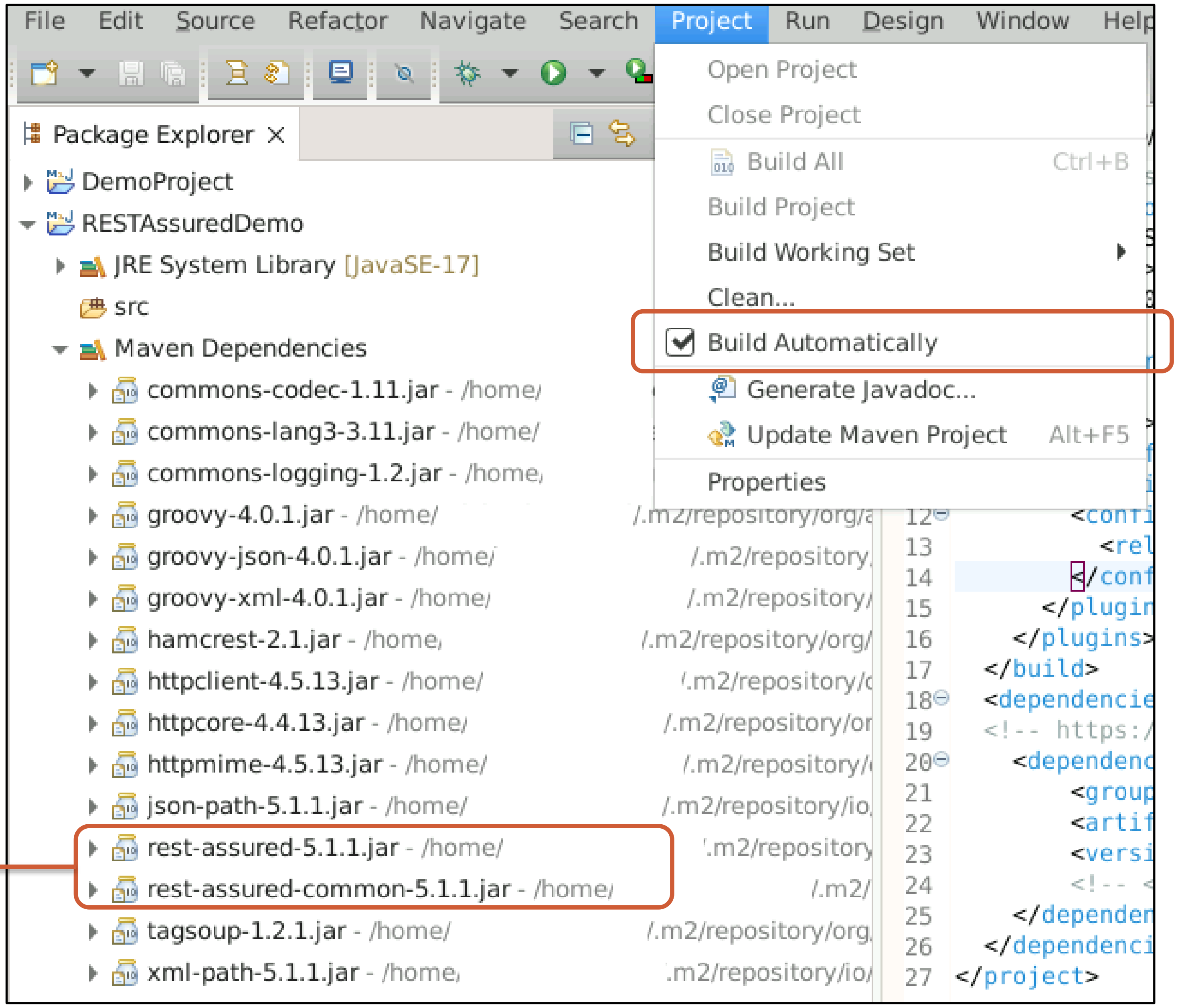
Add **<dependencies></dependencies>** tags

Remove the **<scope></scope>** tag

Importing REST Assured

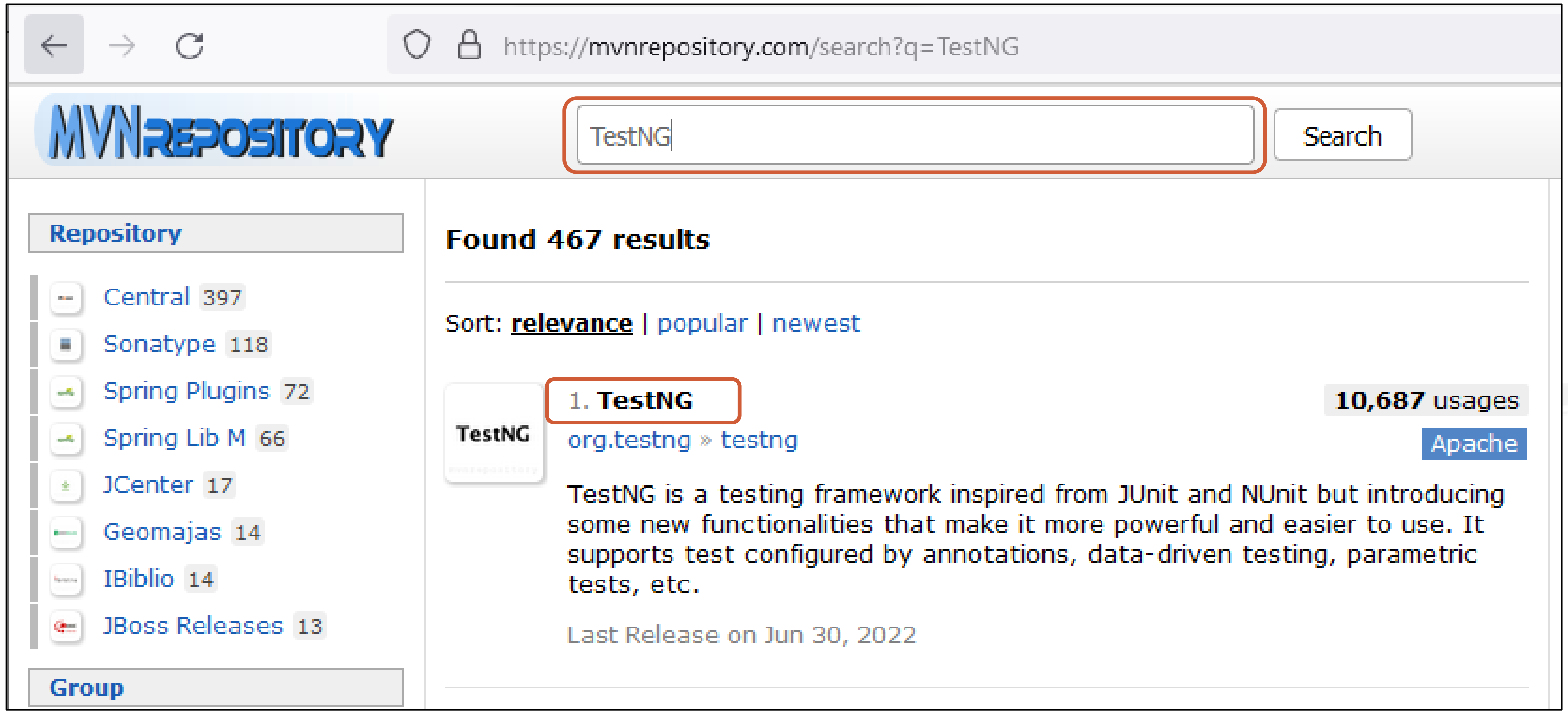
Step 10: Ensure **Build Automatically** is enabled and save the project

REST Assured libraries are automatically imported when the project is saved.


File
Edit
Source
Refactor
Navigate
Search
Project
Run
Design
Window
Help
Open Project
Close Project
Build All Ctrl+B
Build Project
Build Working Set
Clean...
Build Automatically
Generate Javadoc...
Update Maven Project Alt+F5
Properties
Package Explorer
DemoProject
RESTAssuredDemo
JRE System Library [JavaSE-17]
src
Maven Dependencies
commons-codec-1.11.jar - /home/
commons-lang3-3.11.jar - /home/
commons-logging-1.2.jar - /home/
groovy-4.0.1.jar - /home/
groovy-json-4.0.1.jar - /home/
groovy-xml-4.0.1.jar - /home/
hamcrest-2.1.jar - /home/
httpclient-4.5.13.jar - /home/
httpcore-4.4.13.jar - /home/
httpmime-4.5.13.jar - /home/
json-path-5.1.1.jar - /home/
rest-assured-5.1.1.jar - /home/
rest-assured-common-5.1.1.jar - /home/
tagsoup-1.2.1.jar - /home/
xml-path-5.1.1.jar - /home/

Importing TestNG

Step 11: Navigate to **mvnrepository.com**, search for TestNG, and click on **TestNG**


https://mvnrepository.com/search?q=TestNG

MVNREPOSITORY

TestNG

Search

Repository

Central	397
Sonatype	118
Spring Plugins	72
Spring Lib M	66
JCenter	17
Geomajas	14
IBiblio	14
JBoss Releases	13

Group

Found 467 results

Sort: relevance | popular | newest

1. TestNG

10,687 usages

org.testng » testng

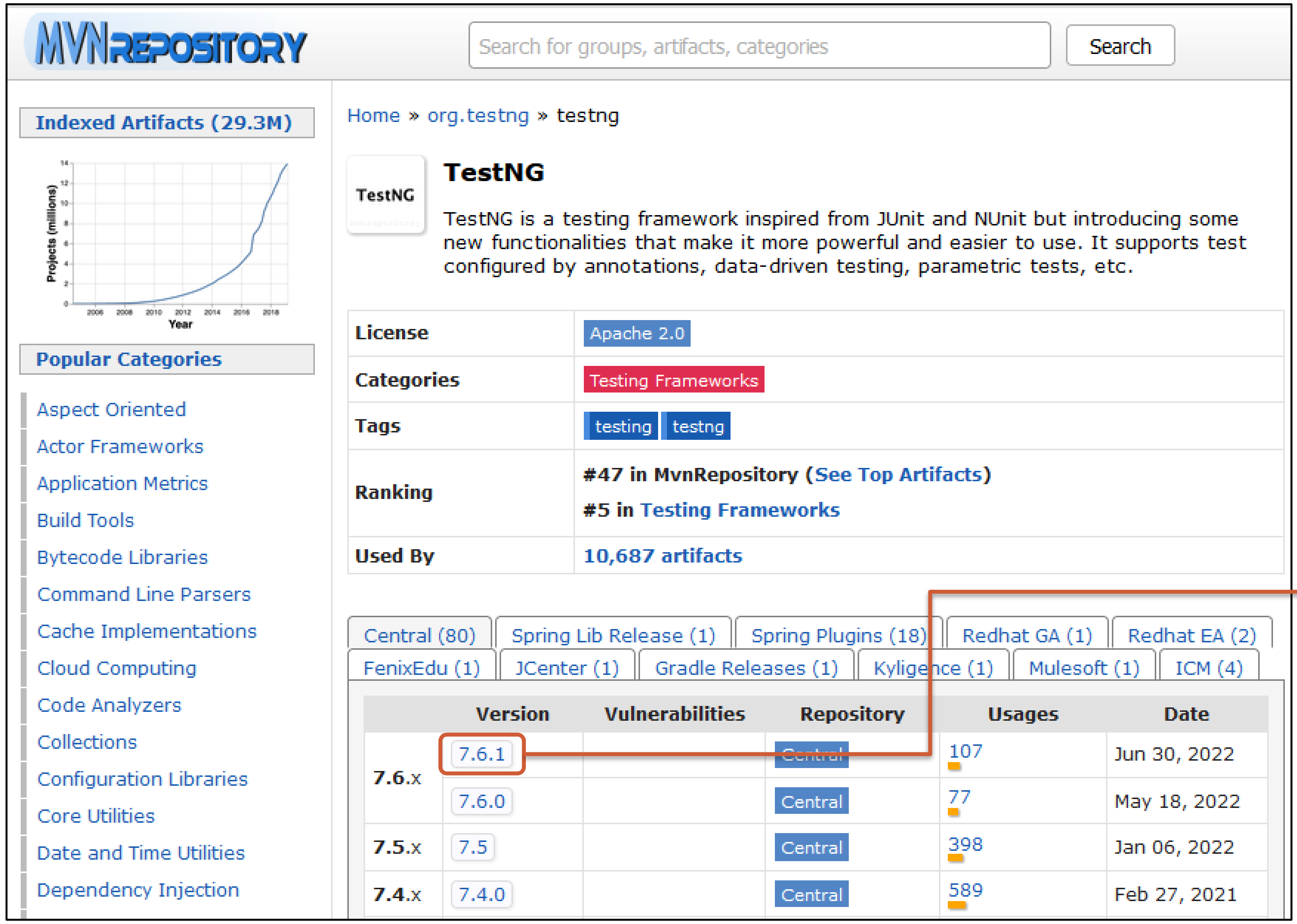
Apache

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use. It supports test configured by annotations, data-driven testing, parametric tests, etc.

Last Release on Jun 30, 2022

Importing TestNG

Step 12: Click on the latest version of TestNG



MVNREPOSITORY

Search for groups, artifacts, categories | Search

Indexed Artifacts (29.3M)

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection

Home » org.testng » testng

TestNG

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use. It supports test configured by annotations, data-driven testing, parametric tests, etc.

License	Apache 2.0
Categories	Testing Frameworks
Tags	testing testng
Ranking	**#47 in MvnRepository (See Top Artifacts)** **#5 in Testing Frameworks**
Used By	**10,687 artifacts**

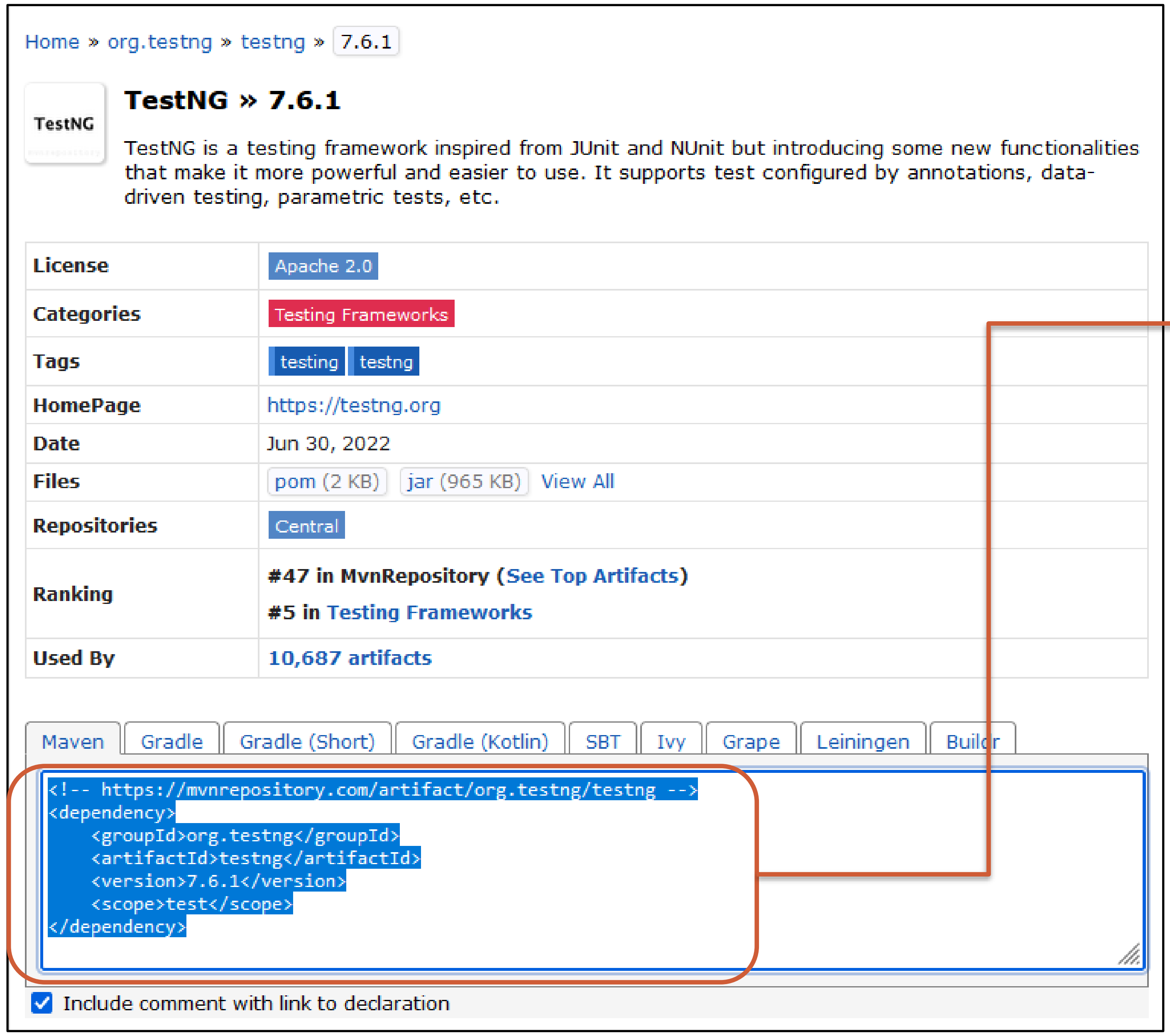
Central (80) | Spring Lib Release (1) | Spring Plugins (18) | Redhat GA (1) | Redhat EA (2) | FenixEdu (1) | JCenter (1) | Gradle Releases (1) | Kyligence (1) | Mulesoft (1) | ICM (4)

	Version	Vulnerabilities	Repository	Usages	Date
7.6.x	7.6.1		Central	107	Jun 30, 2022
	7.6.0		Central	77	May 18, 2022
7.5.x	7.5		Central	398	Jan 06, 2022
7.4.x	7.4.0		Central	589	Feb 27, 2021

Latest version of TestNG: 7.6.1

Importing TestNG

Step 12: Copy TestNG dependency information



Home » org.testng » testng » 7.6.1

TestNG » 7.6.1

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use. It supports test configured by annotations, data-driven testing, parametric tests, etc.

License	Apache 2.0
Categories	Testing Frameworks
Tags	testing testng
HomePage	https://testng.org
Date	Jun 30, 2022
Files	pom (2 KB) jar (965 KB) View All
Repositories	Central
Ranking	#47 in MvnRepository (See Top Artifacts) #5 in Testing Frameworks
Used By	10,687 artifacts

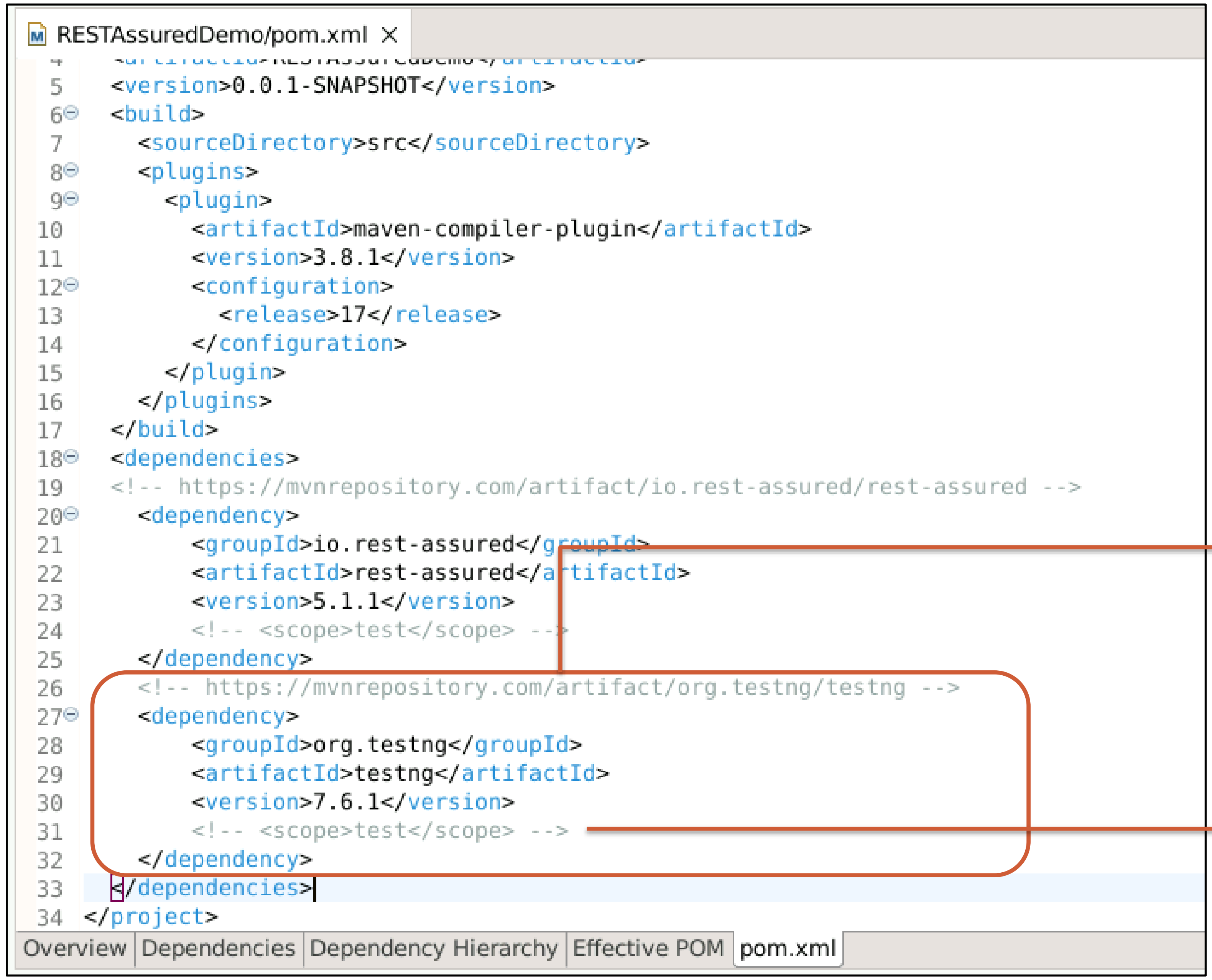
```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.6.1</version>
    <scope>test</scope>
</dependency>
```

☑ Include comment with link to declaration

Dependency information

Importing TestNG

Step 13: Paste TestNG dependency information within the **pom.xml** file and save the project



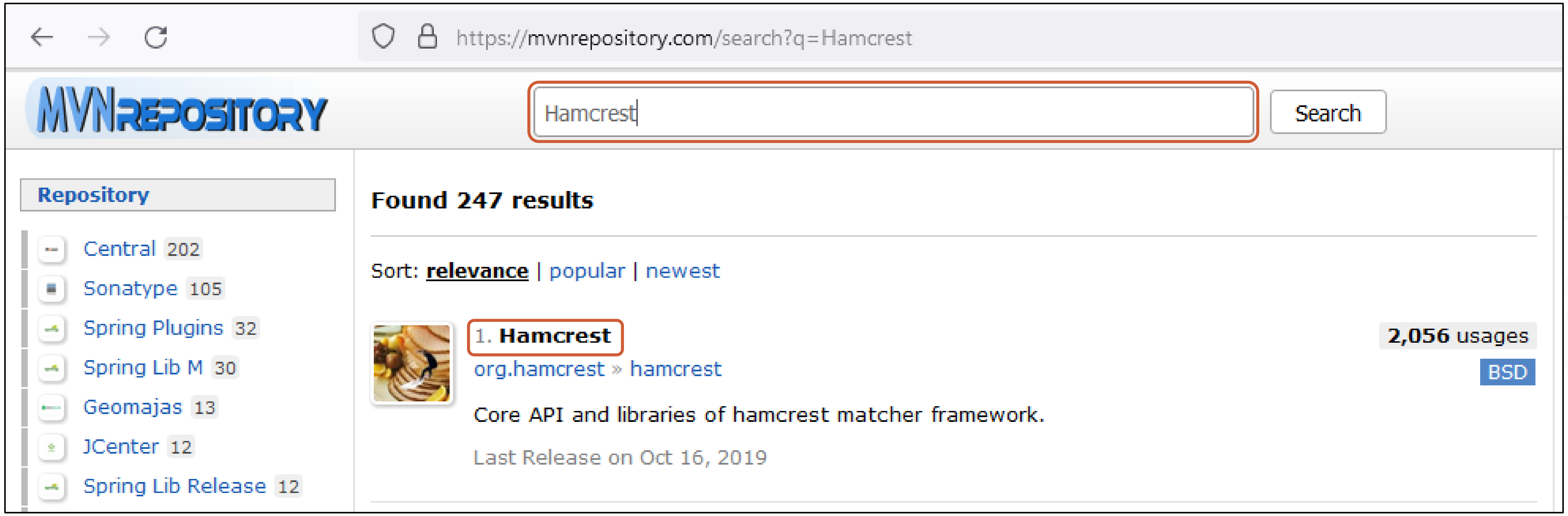
```
<version>0.0.1-SNAPSHOT</version>
<build>
  <sourceDirectory>src</sourceDirectory>
  <plugins>
    <plugin>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <configuration>
        <release>17</release>
      </configuration>
    </plugin>
  </plugins>
</build>
<dependencies>
<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
  <dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>rest-assured</artifactId>
      <version>5.1.1</version>
      <!-- <scope>test</scope> -->
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.testng/testng -->
  <dependency>
      <groupId>org.testng</groupId>
      <artifactId>testng</artifactId>
      <version>7.6.1</version>
      <!-- <scope>test</scope> -->
  </dependency>
</dependencies>
</project>
```

Paste TestNG dependency information in the **pom.xml** file

Remove the **<scope></scope>** tag

Importing Hamcrest

Step 14: Navigate to **mvnrepository.com**, search for Hamcrest, and click on **Hamcrest**



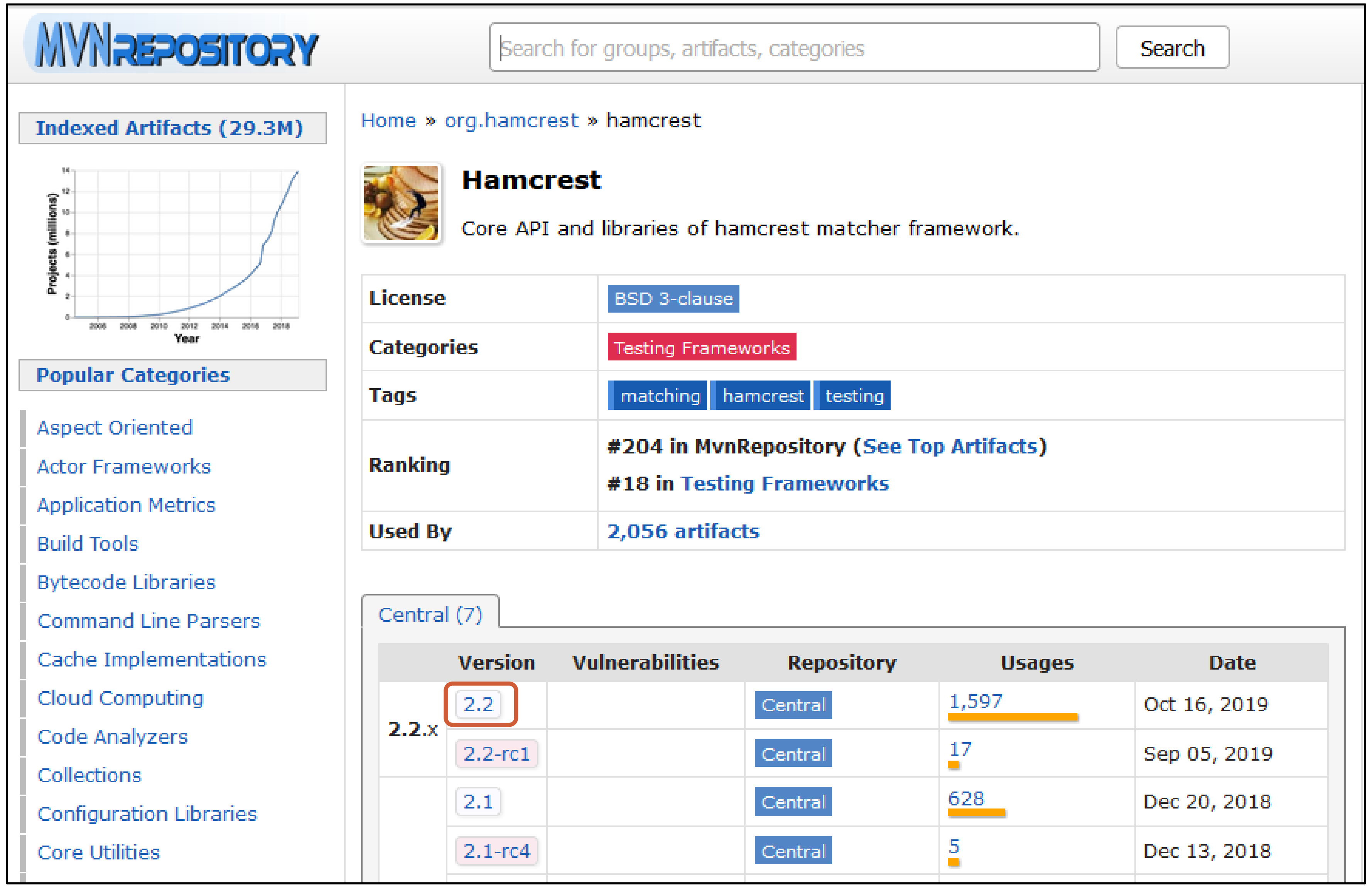
Found 247 results

Sort: relevance | popular | newest

1. Hamcrest
org.hamcrest » hamcrest
2,056 usages
BSD
Core API and libraries of hamcrest matcher framework.
Last Release on Oct 16, 2019

Importing Hamcrest

Step 15: Click on the latest version of Hamcrest


MVNREPOSITORY
Search for groups, artifacts, categories
Search
Indexed Artifacts (29.3M)
Projects (millions)
Year
Popular Categories

Aspect Oriented
Actor Frameworks
Application Metrics
Build Tools
Bytecode Libraries
Command Line Parsers
Cache Implementations
Cloud Computing
Code Analyzers
Collections
Configuration Libraries
Core Utilities

Home » org.hamcrest » hamcrest

Hamcrest

Core API and libraries of hamcrest matcher framework.

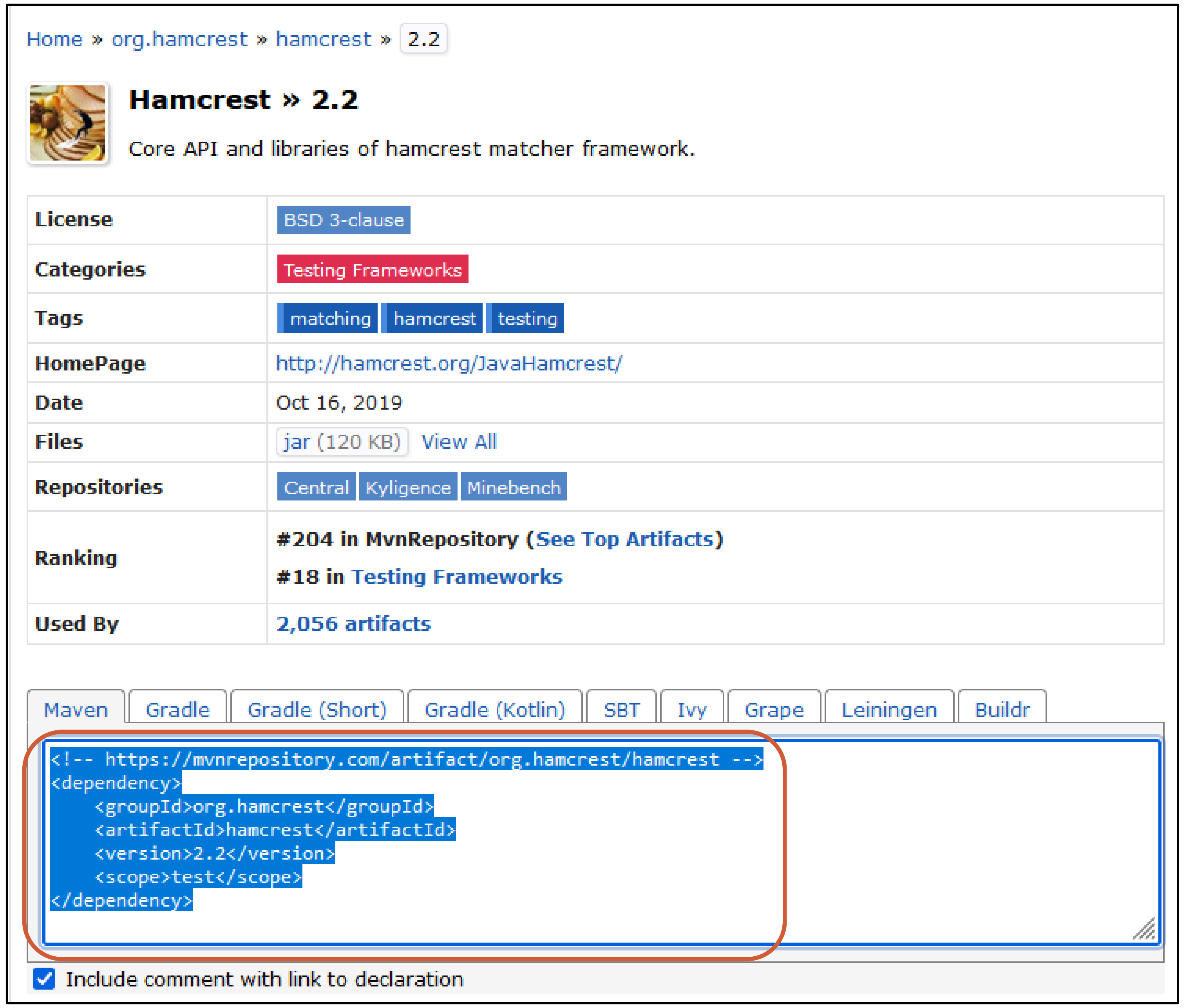
License	BSD 3-clause
Categories	Testing Frameworks
Tags	matching hamcrest testing
Ranking	#204 in MvnRepository (See Top Artifacts) #18 in Testing Frameworks
Used By	2,056 artifacts

Central (7)

	Version	Vulnerabilities	Repository	Usages	Date
2.2.x	2.2		Central	1,597	Oct 16, 2019
	2.2-rc1		Central	17	Sep 05, 2019
	2.1		Central	628	Dec 20, 2018
	2.1-rc4		Central	5	Dec 13, 2018

Importing Hamcrest

Step 16: Copy Hamcrest dependency information



Home » org.hamcrest » hamcrest » 2.2

Hamcrest » 2.2

Core API and libraries of hamcrest matcher framework.

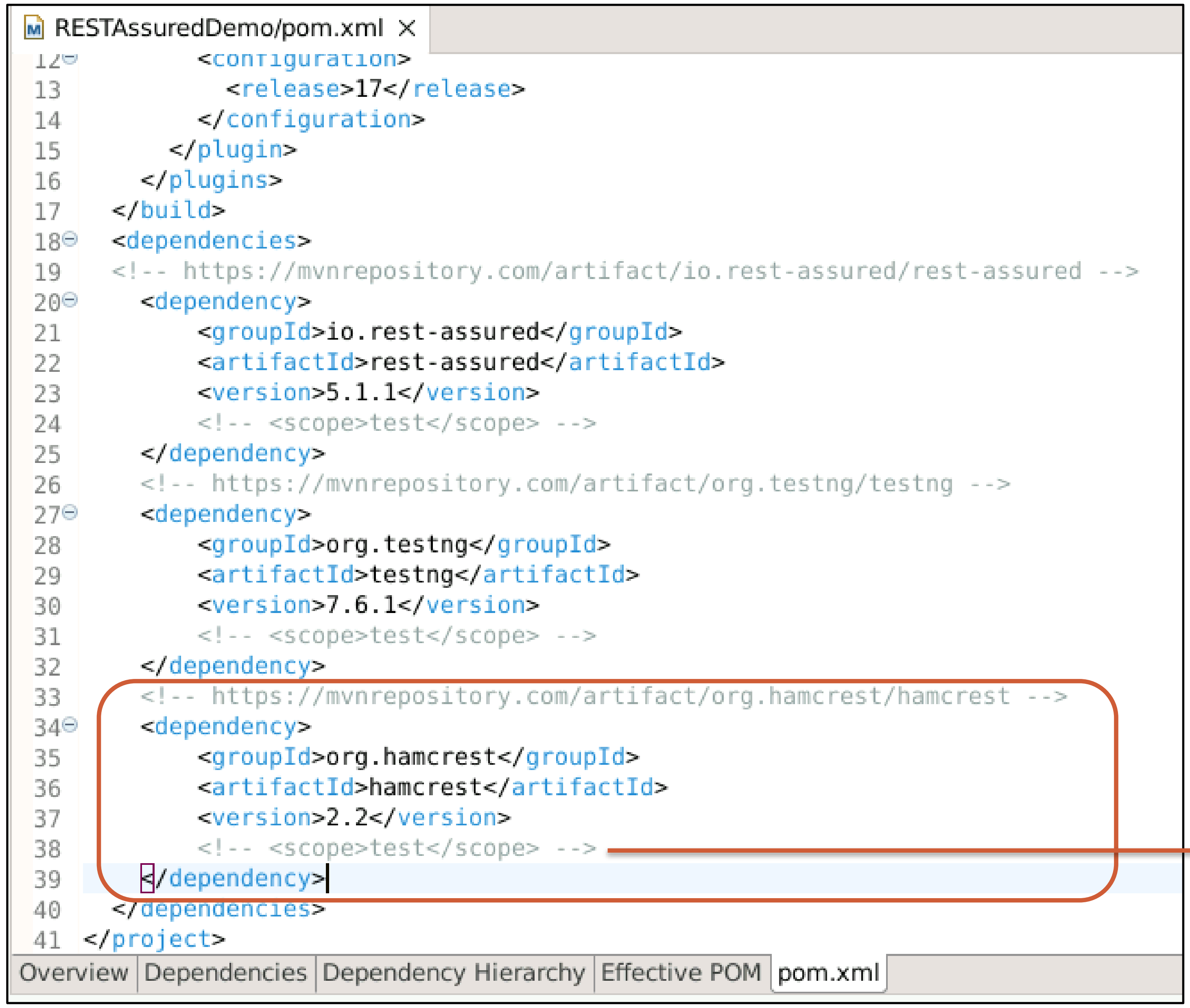
License	BSD 3-clause
Categories	Testing Frameworks
Tags	matching hamcrest testing
HomePage	http://hamcrest.org/JavaHamcrest/
Date	Oct 16, 2019
Files	jar (120 KB) View All
Repositories	Central Kyligence Minebench
Ranking	**#204 in MvnRepository (See Top Artifacts)** **#18 in Testing Frameworks**
Used By	**2,056 artifacts**

```
<!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest -->
<dependency>
    <groupId>org.hamcrest</groupId>
    <artifactId>hamcrest</artifactId>
    <version>2.2</version>
    <scope>test</scope>
</dependency>
```

☑ Include comment with link to declaration

Importing Hamcrest

Step 17: Paste Hamcrest dependency information within the **pom.xml** file and save the project



```
          <configuration>
            <release>17</release>
          </configuration>
        </plugin>
      </plugins>
    </build>
    <dependencies>
    <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
      <dependency>
          <groupId>io.rest-assured</groupId>
          <artifactId>rest-assured</artifactId>
          <version>5.1.1</version>
          <!-- <scope>test</scope> -->
      </dependency>
      <!-- https://mvnrepository.com/artifact/org.testng/testng -->
      <dependency>
          <groupId>org.testng</groupId>
          <artifactId>testng</artifactId>
          <version>7.6.1</version>
          <!-- <scope>test</scope> -->
      </dependency>
      <!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest -->
      <dependency>
          <groupId>org.hamcrest</groupId>
          <artifactId>hamcrest</artifactId>
          <version>2.2</version>
          <!-- <scope>test</scope> -->
      </dependency>
    </dependencies>
  </project>
```

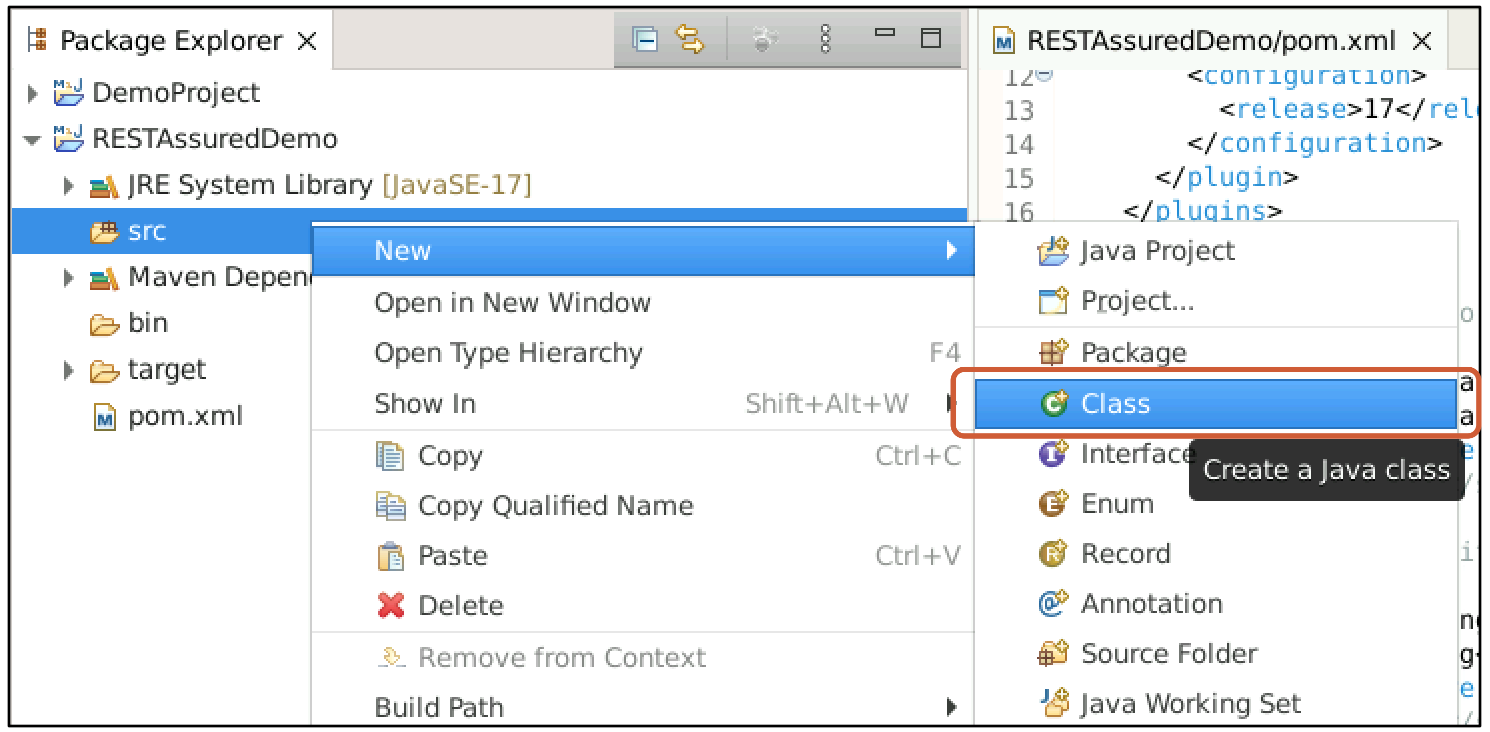
Remove the **<scope></scope>** tag

FULL STACK

Automate GET Request Using REST Assured

Create a Java Class

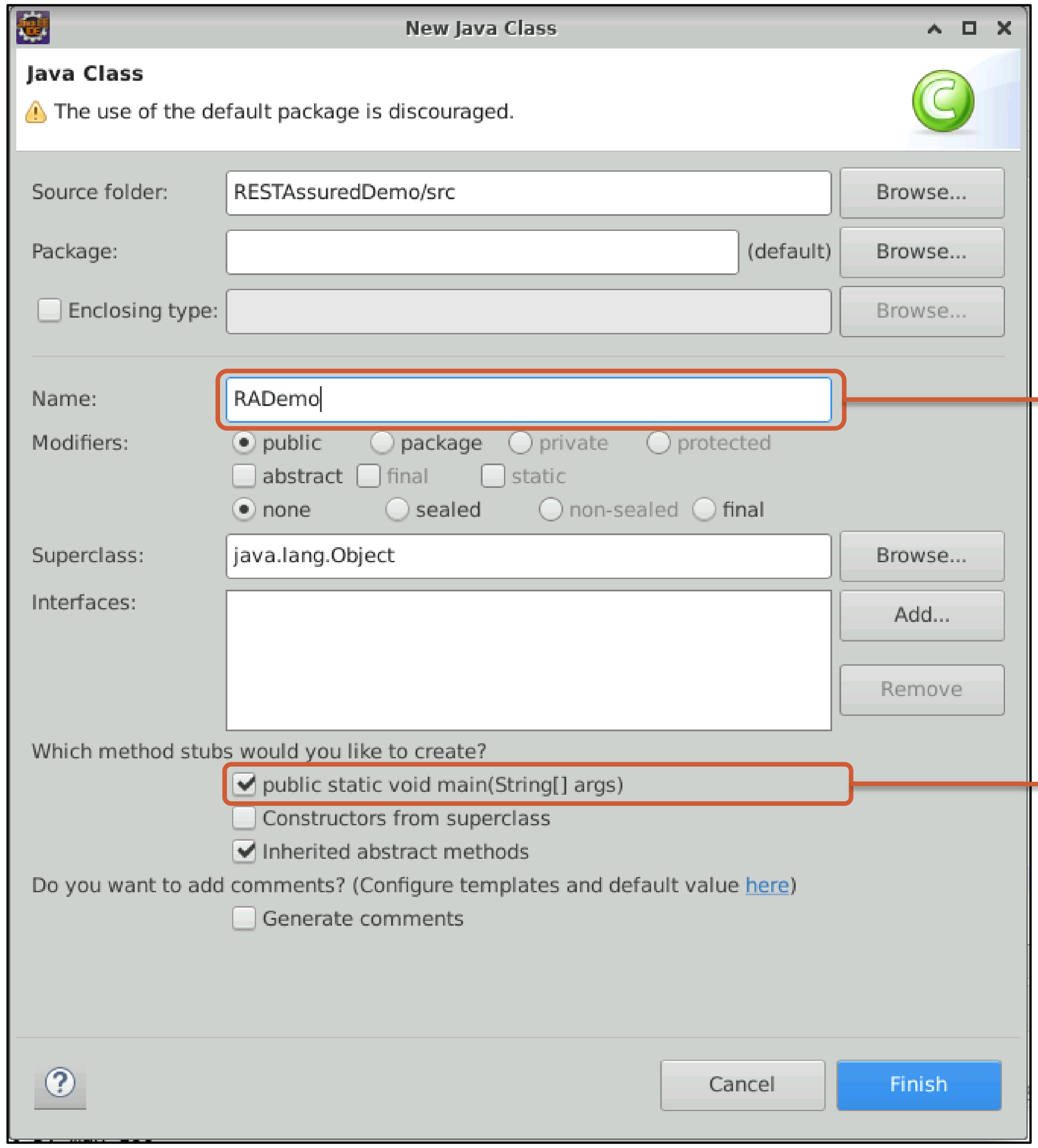
Step 1: Right-click on the **src** folder, navigate to **New**, and click on **Class**



```
        <configuration>
          <release>17</rel
        </configuration>
      </plugin>
    </plugins>
```

Create a Java Class

Step 2: Provide a name to the class



New Java Class

Java Class

⚠ The use of the default package is discouraged.

Source folder: RESTAssuredDemo/src

Package: (default)

☐ Enclosing type:

Name: RADemo

Modifiers: ● public ○ package ○ private ○ protected

☐ abstract ☐ final ☐ static

● none ○ sealed ○ non-sealed ○ final

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)

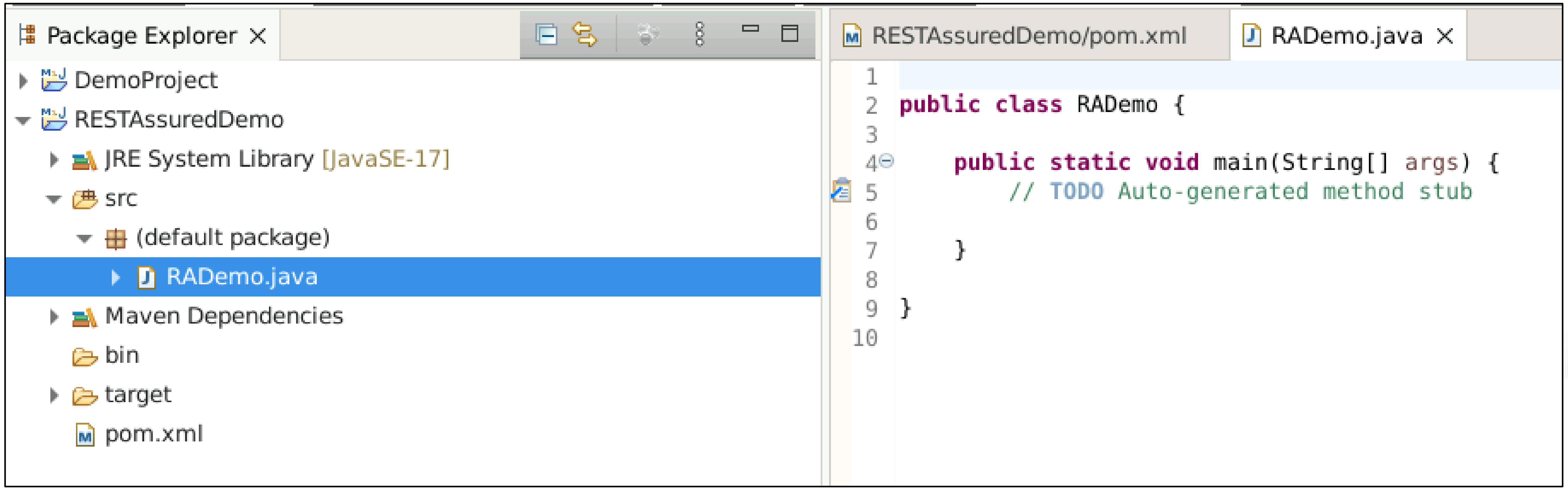
☐ Generate comments

Provide a name for the class

Select this option

Create a Java Class

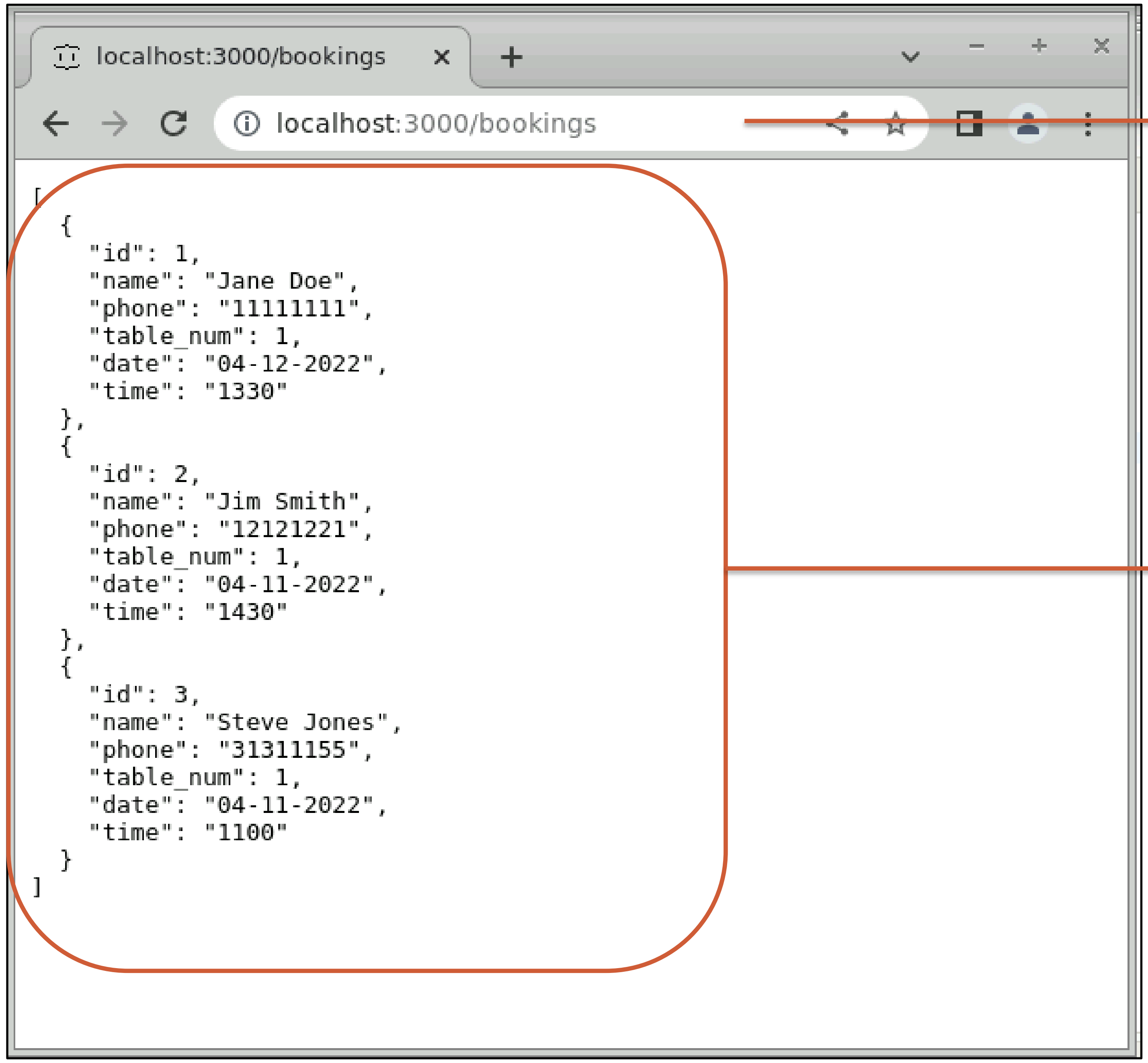
The following Java file will be created:



```java

public class RADemo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

}

```

Simulated REST Service

The following REST API with JSON Server will be used for the example:



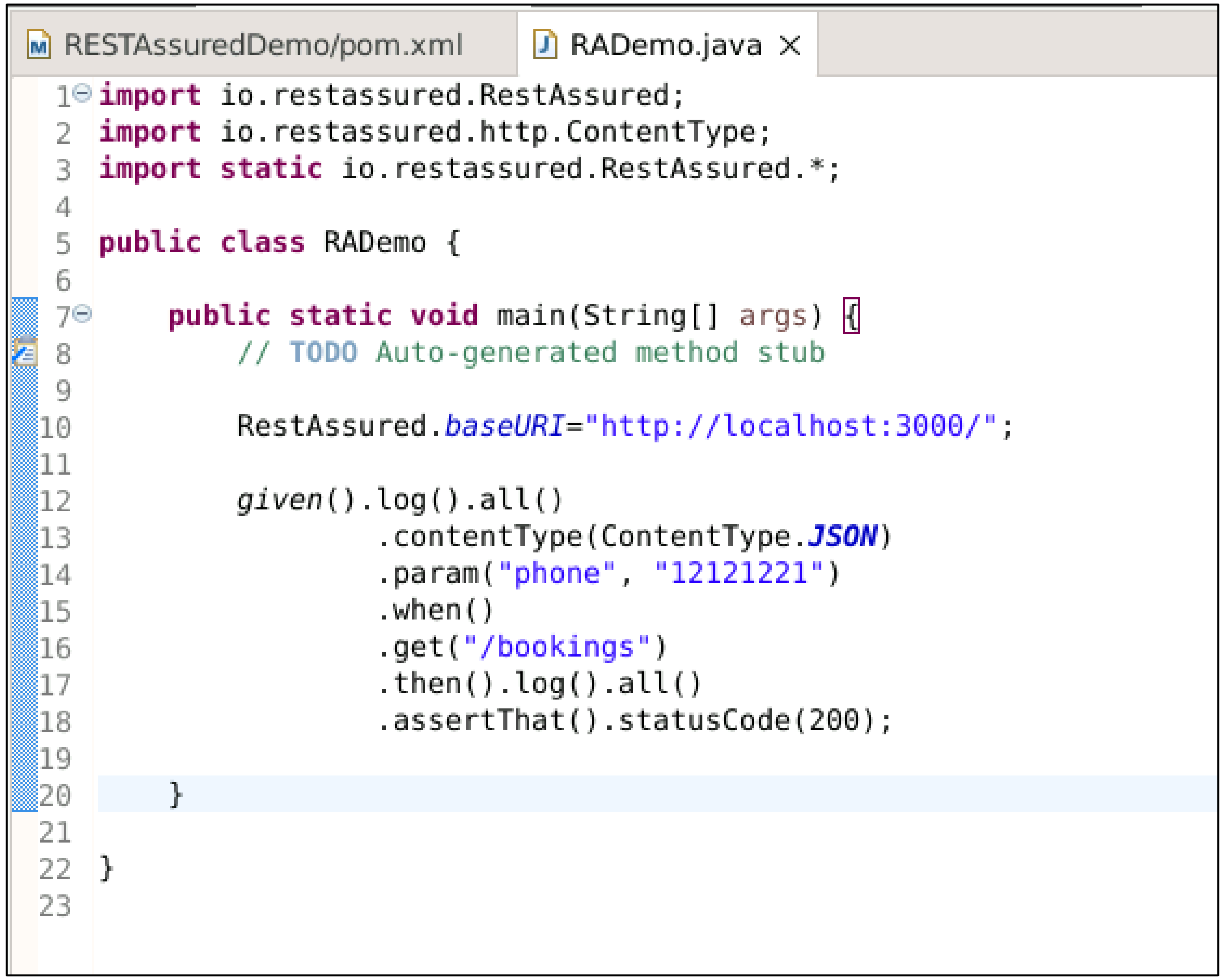
```
[
  {
    "id": 1,
    "name": "Jane Doe",
    "phone": "11111111",
    "table_num": 1,
    "date": "04-12-2022",
    "time": "1330"
  },
  {
    "id": 2,
    "name": "Jim Smith",
    "phone": "12121221",
    "table_num": 1,
    "date": "04-11-2022",
    "time": "1430"
  },
  {
    "id": 3,
    "name": "Steve Jones",
    "phone": "31311155",
    "table_num": 1,
    "date": "04-11-2022",
    "time": "1100"
  }
]
```

URI of REST Service

Three database records in JSON format

GET Request

Step 3: Enter the following lines of code within the Java file:



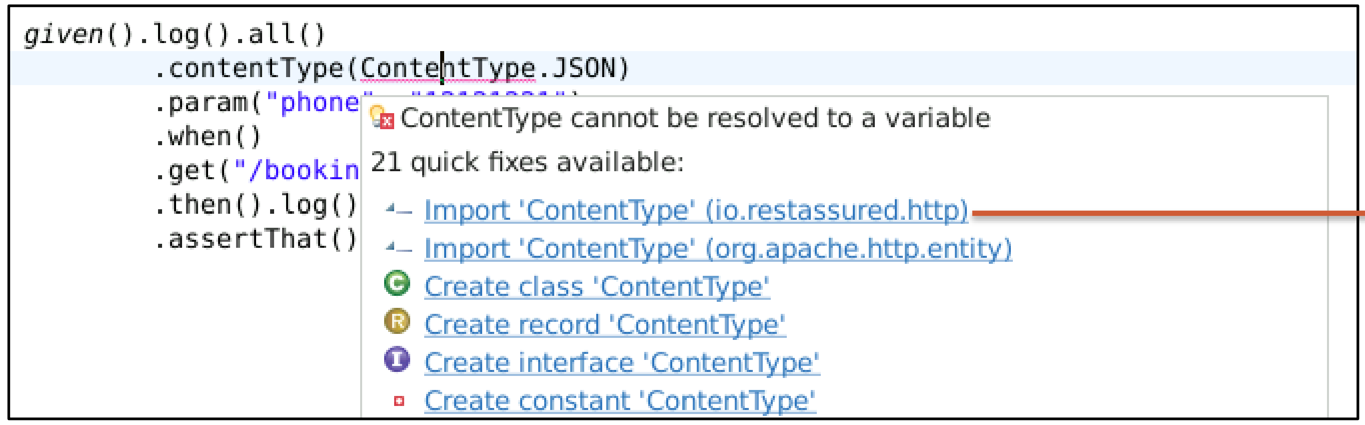
```java
import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import static io.restassured.RestAssured.*;

public class RADemo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        RestAssured.baseURI="http://localhost:3000/";

        given().log().all()
                .contentType(ContentType.JSON)
                .param("phone", "12121221")
                .when()
                .get("/bookings")
                .then().log().all()
                .assertThat().statusCode(200);

    }

}
```

GET Request

Users will need to **import** the following classes:

```
import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import static io.restassured.RestAssured.*;
```

The Eclipse IDE will highlight when a particular function usage requires an **import** statement:



```
given().log().all()
        .contentType(ContentType.JSON)
        .param("phone"
        .when()
        .get("/bookin
        .then().log()
        .assertThat()
```

ContentType cannot be resolved to a variable

21 quick fixes available:

- Import 'ContentType' (io.restassured.http)
- Import 'ContentType' (org.apache.http.entity)
- Create class 'ContentType'
- Create record 'ContentType'
- Create interface 'ContentType'
- Create constant 'ContentType'

Click here to automatically add the necessary **import** statement

Statically import classes to access static methods:

`import` **`static`** `io.restassured.RestAssured.*;`

GET Request

Users can change the default values for the following:

- RestAssured.baseURI
- RestAssured.port
- RestAssured.basePath
- RestAssured.authentication
- RestAssured.rootPath

```
RestAssured.baseURI = "http://localhost:3000/";
```

The default Base URI has been changed for all future requests.

GET Request

The given(), when(), and then() functions are used to build the GET request.

given()

Specifies the input details (like request parameters) needed to build the request

when()

Sends the request to the API by specifying method and path parameters

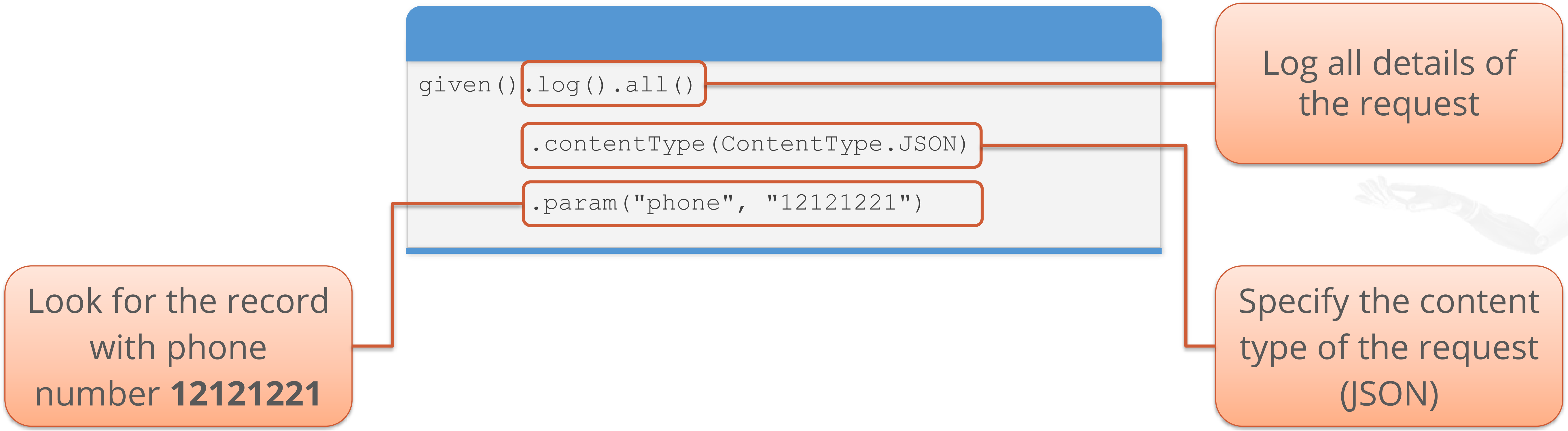
then()

Returns a response that can be validated

GET Request: given() Method

given()

Specifies the input details (like request parameters) needed to build the request



```
given().log().all()
        .contentType(ContentType.JSON)
        .param("phone", "12121221")
```

Log all details of the request
Specify the content type of the request (JSON)
Look for the record with phone number **12121221**

GET Request: when() Method

when()

Sends the request to the API by specifying method and path parameters

```
.when()
.get("/bookings")
```

Perform a GET HTTP request to the specified path

Validate GET Response in REST Assured

GET Request – then() Method

then()

Returns a response that can be validated

```
.then().log().all()
.assertThat().statusCode(200);
```

Log all details of the response

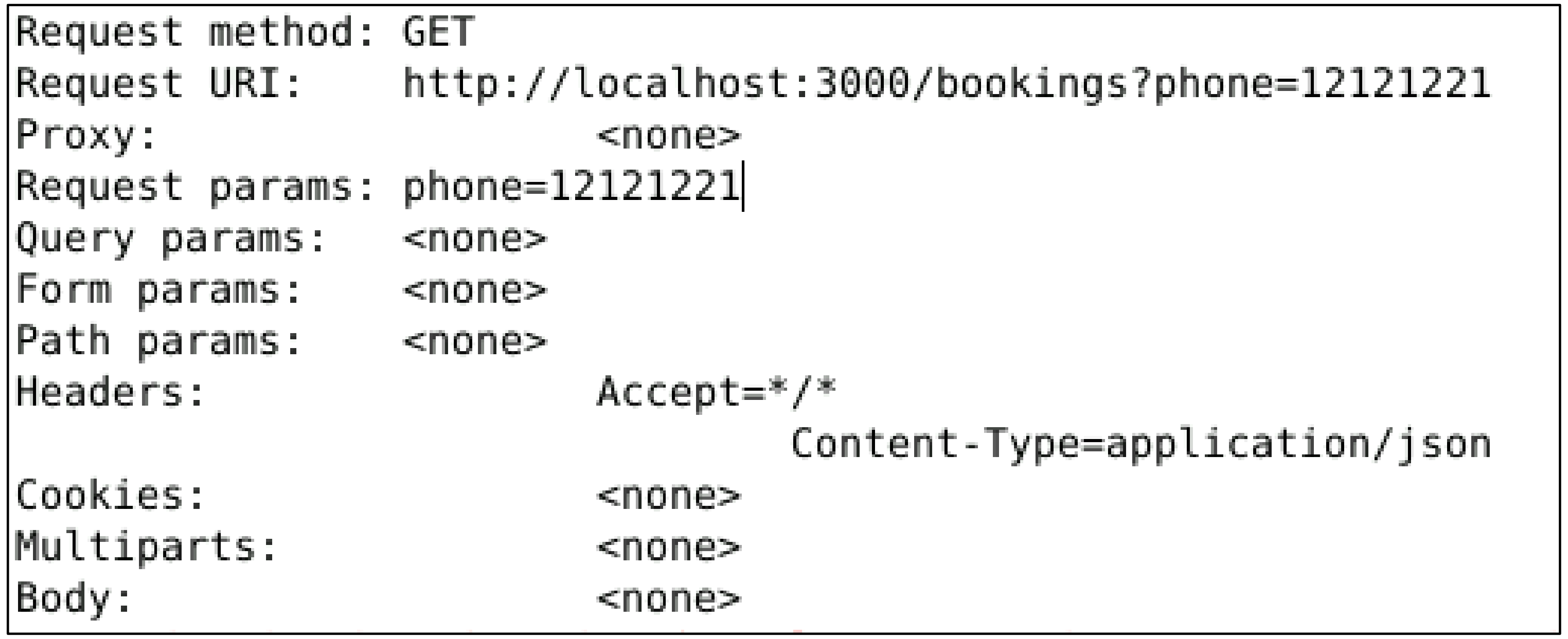
Verify whether the status code of the response is 200

If the returned status code is different from the one specified, the following error is displayed:

```
Exception in thread "main" java.lang.AssertionError: 1 expectation failed.
```

Output of Request

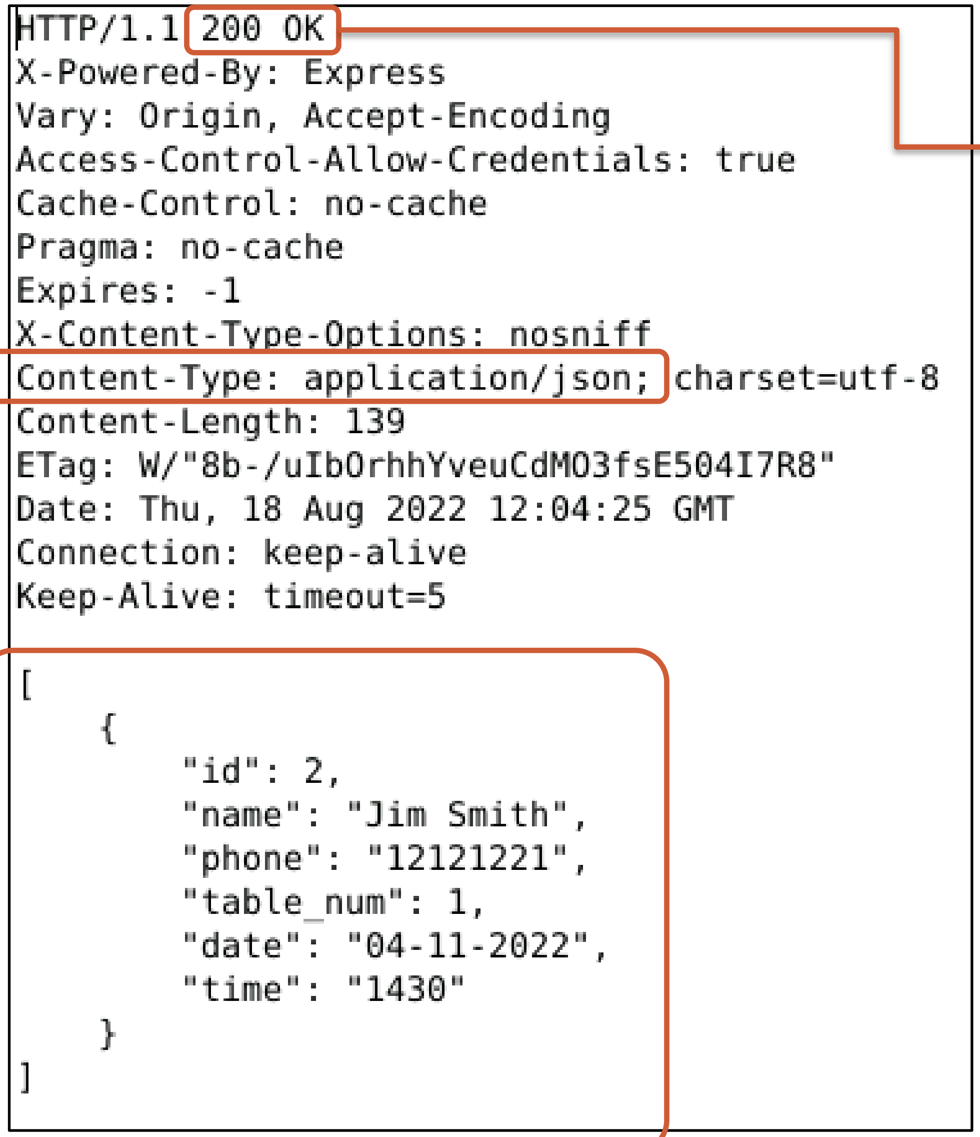
The following is the GET request sent to the server:



```
Request method: GET
Request URI:    http://localhost:3000/bookings?phone=12121221
Proxy:                  <none>
Request params: phone=12121221
Query params:   <none>
Form params:    <none>
Path params:    <none>
Headers:                Accept=*/*
                                Content-Type=application/json
Cookies:                <none>
Multiparts:             <none>
Body:                   <none>
```

Output of Response

The following is the response received from the server:



```
HTTP/1.1 200 OK
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Content-Type-Options: nosniff
Content-Type: application/json; charset=utf-8
Content-Length: 139
ETag: W/"8b-/uIbOrhhYveuCdMO3fsE504I7R8"
Date: Thu, 18 Aug 2022 12:04:25 GMT
Connection: keep-alive
Keep-Alive: timeout=5

[
    {
        "id": 2,
        "name": "Jim Smith",
        "phone": "12121221",
        "table_num": 1,
        "date": "04-11-2022",
        "time": "1430"
    }
]
```

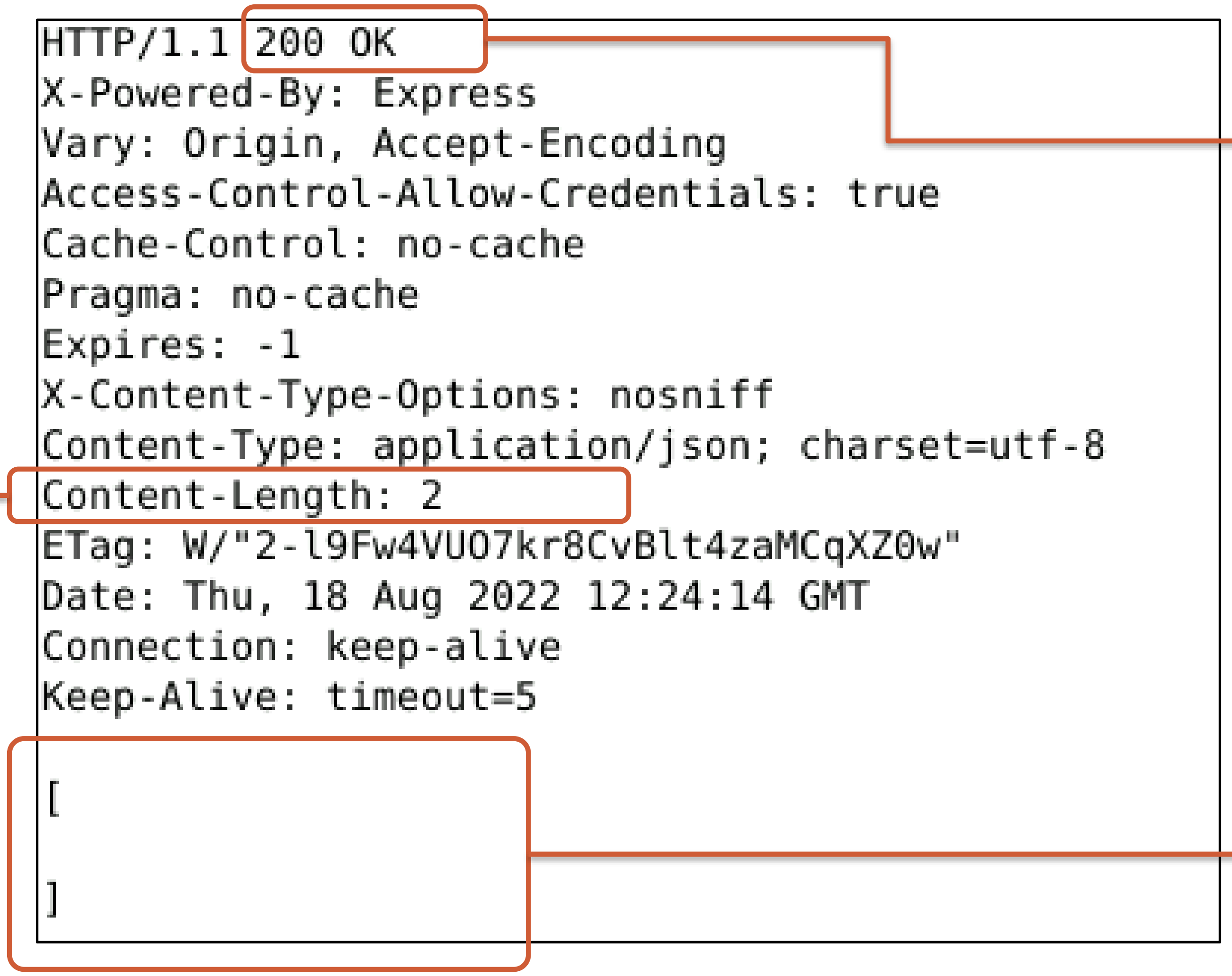
Response body format

200 indicates a successful request.

Response body with record corresponding to request parameter

Output of Response

The record not found is still a successful GET request:



```
HTTP/1.1 200 OK
X-Powered-By: Express
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Content-Type-Options: nosniff
Content-Type: application/json; charset=utf-8
Content-Length: 2
ETag: W/"2-l9Fw4VUO7kr8CvBlt4zaMCqXZ0w"
Date: Thu, 18 Aug 2022 12:24:14 GMT
Connection: keep-alive
Keep-Alive: timeout=5

[

]
```

Content length

200 indicates a successful request.

An empty response body indicates that the record was not found.

Verify Response Header and Body

Verifying Response Body

Here is the code to verify if the booking date in the response is equal to today's date:

Specify the base URI

```
RestAssured.baseURI="http://localhost:3000/";
```

Extract the current date in the required format

```
DateTimeFormatter dtf = DateTimeFormatter.ofPattern("MM-dd-yyyy");
LocalDateTime now = LocalDateTime.now();
```

Extract the response body

```
Response response = given()
        .param("phone","12121221")
        .get("/bookings");

ResponseBody body = response.getBody();

System.out.println("Response body is: "+ body.asString());
```

Extract the date field from the response body

```
JsonPath jsonPathEvaluator = response.jsonPath();

String bookingDate = jsonPathEvaluator.get("date").toString();
bookingDate = bookingDate.substring(1, bookingDate.length() - 1);
```

Compare the date with the current date using an assertion

```
try {
    Assert.assertEquals(bookingDate, dtf.format(now));
    System.out.println("The booking is for today. Bon Apetit!");
}
catch(AssertionError e) {
    System.out.println("The booking is not for today " + e);
}
```

Extracting the Response Body

Here is the code to extract the response body from the response:

```
Response response = given()
        .param("phone","12121221")
        .get("/bookings");
```

Extract the response of the GET HTTP request

```
ResponseBody body = response.getBody();

System.out.println("Response body is: "+ body.asString());
```

Extract body from the response

Output:

```
Response body is: [
  {
    "id": 2,
    "name": "Jim Smith",
    "phone": "12121221",
    "table_num": 1,
    "date": "09-19-2022",
    "time": "1430"
  }
]
```

Extracting and Comparing a Single Field

Here is the code to extract and compare the date field from the response:

```java
JsonPath jsonPathEvaluator = response.jsonPath();

String bookingDate = jsonPathEvaluator.get("date").toString();
bookingDate = bookingDate.substring(1, bookingDate.length() - 1);
```

```java
try {
    Assert.assertEquals(bookingDate, dtf.format(now));
    System.out.println("The booking is for today. Bon Apetit!");
}
catch(AssertionError e) {
    System.out.println("The booking is not for today " + e);
}
```

Edit the string to remove the leading and trailing square brackets

Extract the date field from the JSON response

Use assertions within the try-catch block to compare dates

Output of a successful assertion:

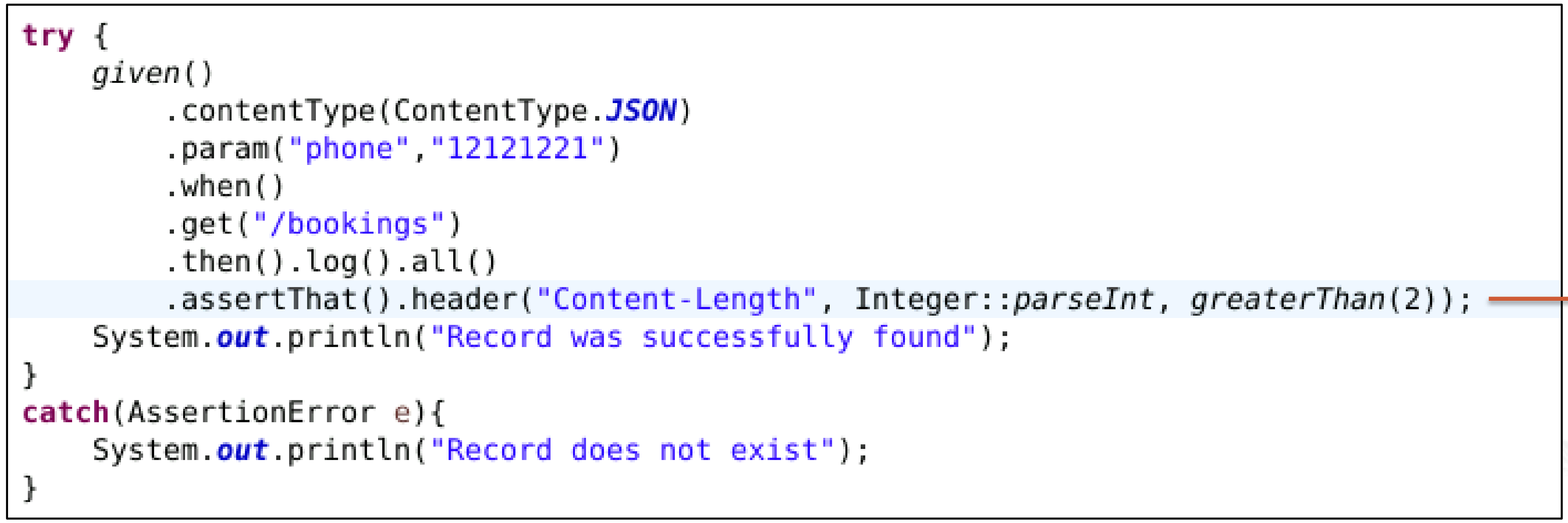
```
The booking is for today. Bon Apetit!
```

Output of a failed assertion:

```
The booking is not for today java.lang.AssertionError: expected [08-19-2022] but found [09-19-2022]
```

Verifying Response Header

Use the **header()** function to validate a header field and determine whether record was found:



```
try {
    given()
        .contentType(ContentType.JSON)
        .param("phone","12121221")
        .when()
        .get("/bookings")
        .then().log().all()
        .assertThat().header("Content-Length", Integer::parseInt, greaterThan(2));
    System.out.println("Record was successfully found");
}
catch(AssertionError e){
    System.out.println("Record does not exist");
}
```

Use the Content-Length header attribute to determine whether the response body is empty

Key Takeaways

- REST Assured and other necessary Java libraries are available in the Maven repository and can be imported to the Maven project.
- A GET HTTP request can be built using the given() and when() methods.
- The then() method combined with assertions can be used to validate the GET response.
- The response body and values can be extracted with the Response interface and JsonPath class.