Lab Week 04 | BITP 3123 Distributed Application Development

Preparation of Development Environment for a Web Service Provider Application

By

Emaliana Kasmuri

FTMK, UTeM

Table of Contents

1 Learning Outcomes

At the end of this lab exercise, the student should be able to: -

1. Implement the data layer for the REST application.
2. Prepare the development environment for the REST application.
3. Prepare the testing environment for the REST application.

2 Software Tools for the Lab Exercise

This lab exercise requires the following software tools to implement the case study.

1. Eclipse
2. MySQL and MySQL Workbench
3. Spring Initializr
4. Postman
5. Web browser
6. Internet connection

3 An Overview of the Case Study

tba

4 Implementation of the Lab Exercise

There are three implementation processes in this lab exercise. The processes are:-

1. Data layer implementation
2. Development environment preparation
3. Testing environment preparation

The relationship between these processes is shown in Figure 4.1.

Figure 4.1: Relation of processes for the lab exercises

5 Data Layer Implementation

There are three activities in the data layer implementation. The activities are:-

1. Create a database.
2. Create tables.
3. View data in tables.

The relationship between these activities is shown in Figure 5.1.

Figure 5.1: Relation of activities in the data layer implementation

5.1 Create a Database

1. Open **MySQL Workbench**.

2. Click the **Schemas** tab.
3. Click the button as shown in the dotted red box of Figure 5.2.

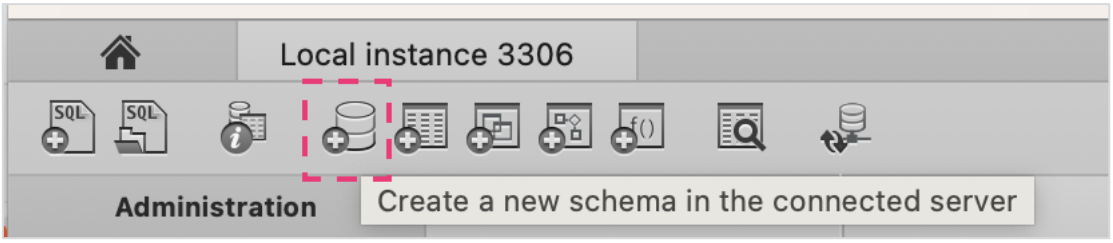

Figure 5.2: Button to create a new schema

A new tab named **Schema Editor**, as shown in Figure 5.3 will appear.

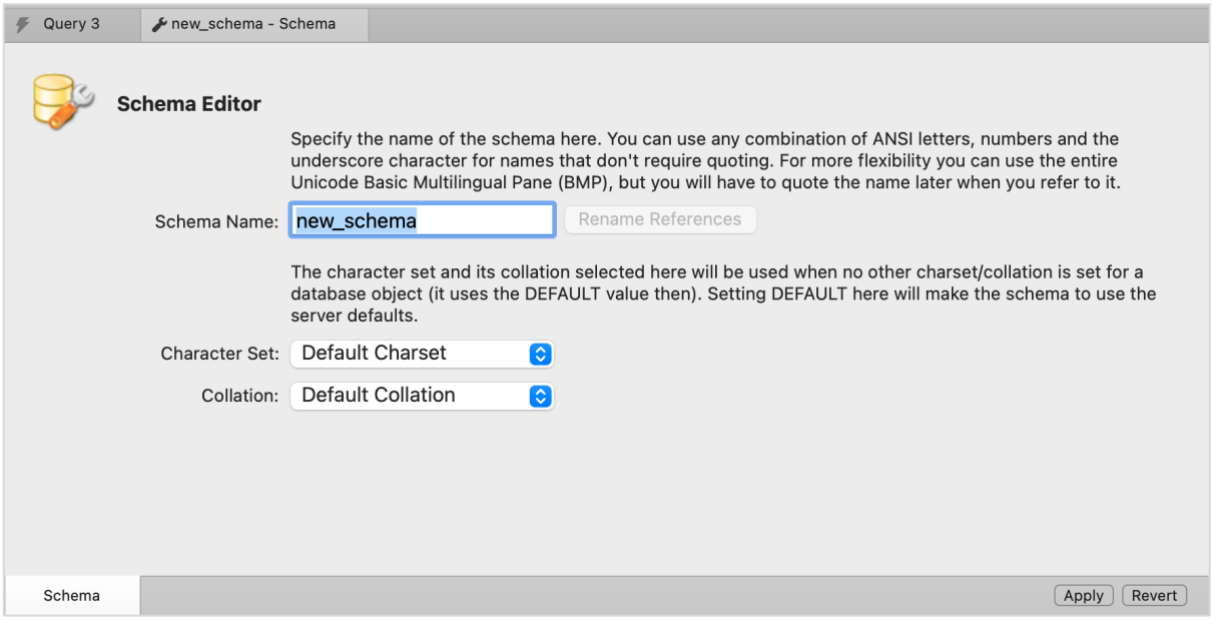

Figure 5.3: Tab to create a new schema

4. Enter orderdb_dev in the **Schema Name** text box.
5. Then, click the **Apply** button. The **orderdb_dev** will be created and appear in the **Schemas** list as shown in Figure 5.4.

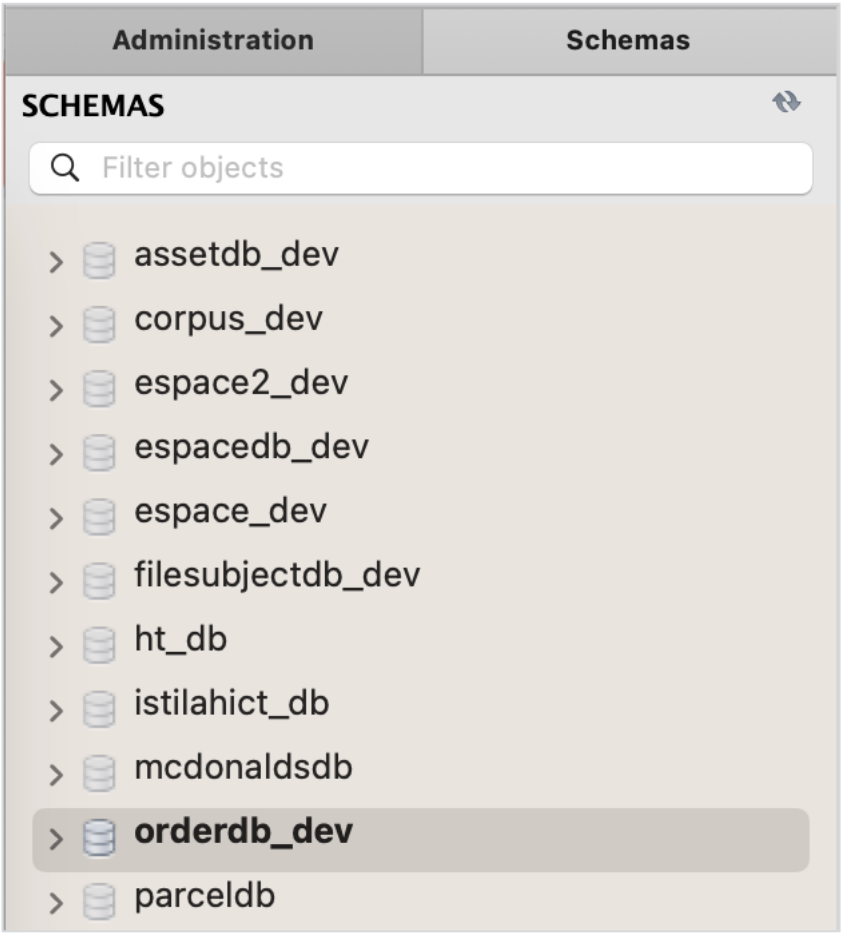

Figure 5.4: orderdb_dev in the Schemas

5.2 Create Tables

1. Download orderdb_dev.sql from ulearn.
2. Click the button shown in Figure 5.5.

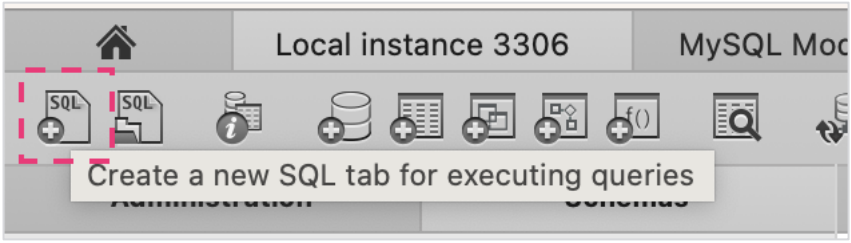

Figure 5.5: Button to create a new SQL tab

A new Query tab will be displayed.

3. Double-click **orderdb_dev** from the **Schemas** to activate it.

4. Click the button shown in Figure 5.6 to open an SQL script that was downloaded from step 1. The script will be loaded in the query tab.

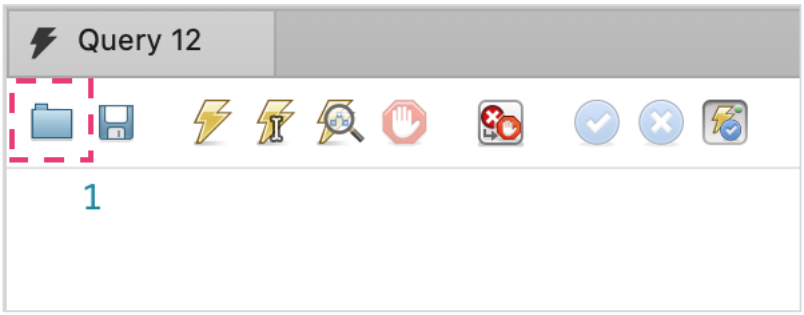

Figure 5.6: Button to load a new SQL script

5. Click the button shown in Figure 5.7 to execute the script.

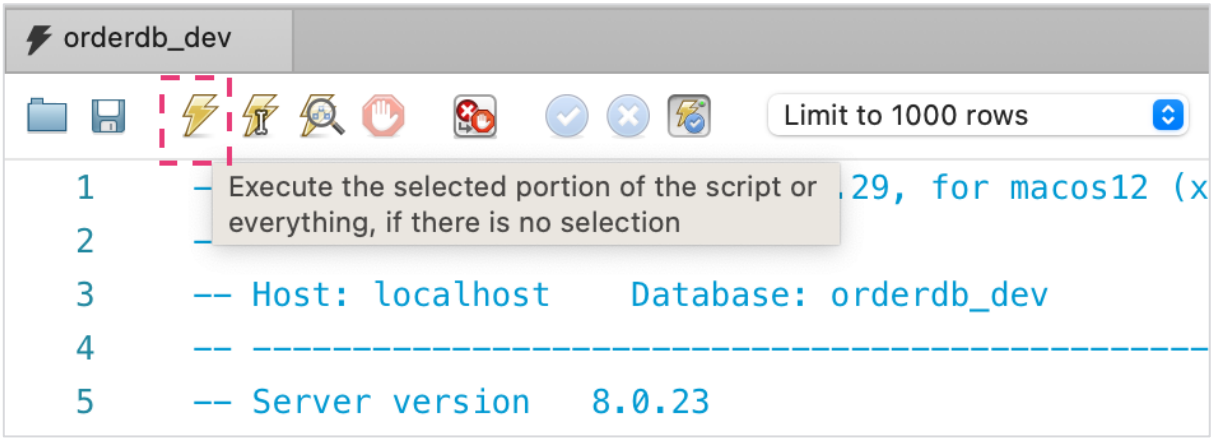

Figure 5.7: Button to execute the SQL script

The tables and their data will be created.

6. Expand Tables of **orderdb_dev** from **Schemas**. The list of tables should be similar as shown in Figure 5.8.

Figure 5.8: List of tables in orderdb_dev

5.3 View Data in Tables

1. Click **ProductType** from **Tables**. Three (3) button icons, as shown in Figure 5.9 will appear beside the table name.

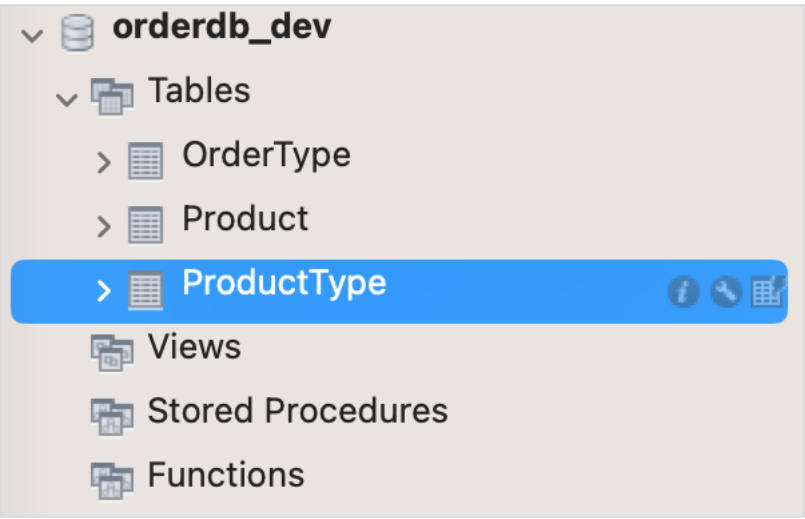

Figure 5.9: The button icons for a table

2. Click the **third button icon**. The data of **ProductType** will be displayed in a new tab.
3. Repeat steps 1 and 2 to view data from other tables.

6 Development Environment Preparation

There are seven activities in the development environment preparation. The activities are: -

1. Create a new Eclipse workspace.
2. Create a new Spring Boot project.
3. Import a new Spring Boot project.
4. Update a Maven Project.
5. Execute the new Spring Boot project.
6. Configure the REST Provider Application
7. Re-execute the Spring Project

The relationship between these activities is shown in Figure 6.1.

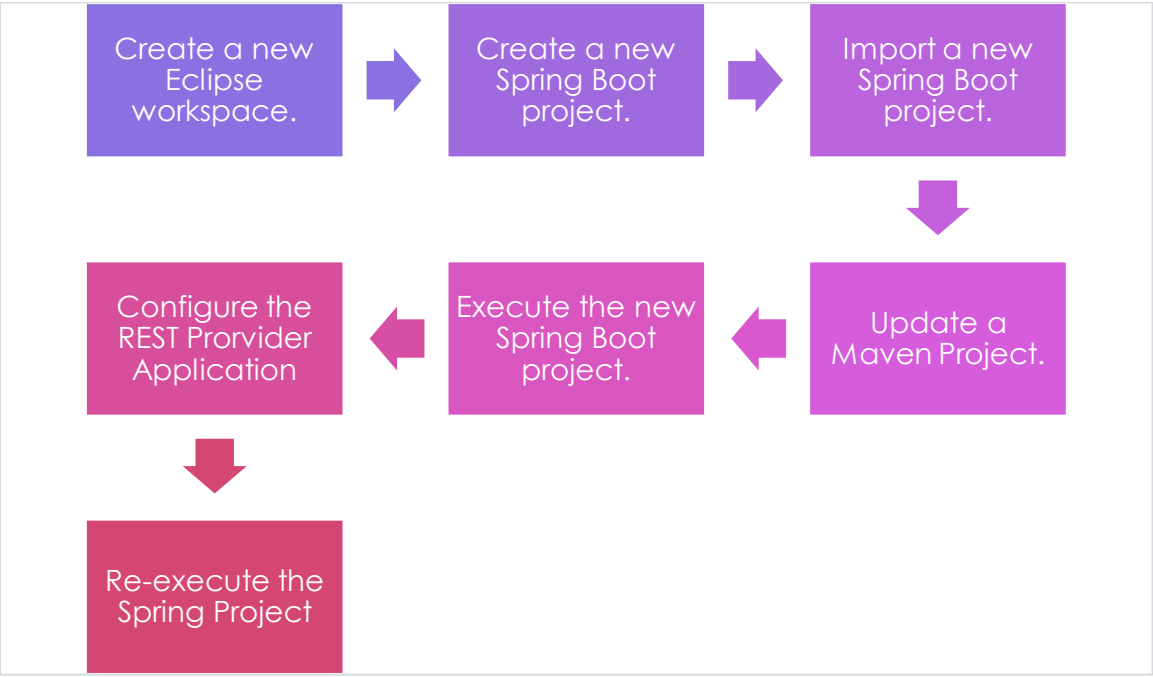

Figure 6.1: Relation of activities in the data layer implementation

6.1 Create a new Eclipse Workspace

1. Open **Eclipse**.
2. Create a new workspace to develop a REST provider application.
3. Name the workspace as **workspace_bitp3123_restorderapp**.
4. Refer to the previous lab to create and switch into a new workspace in Eclipse.

6.2 Create a new Spring Boot Project

1. Click https://start.spring.io. A page like Figure 6.2 will be displayed.

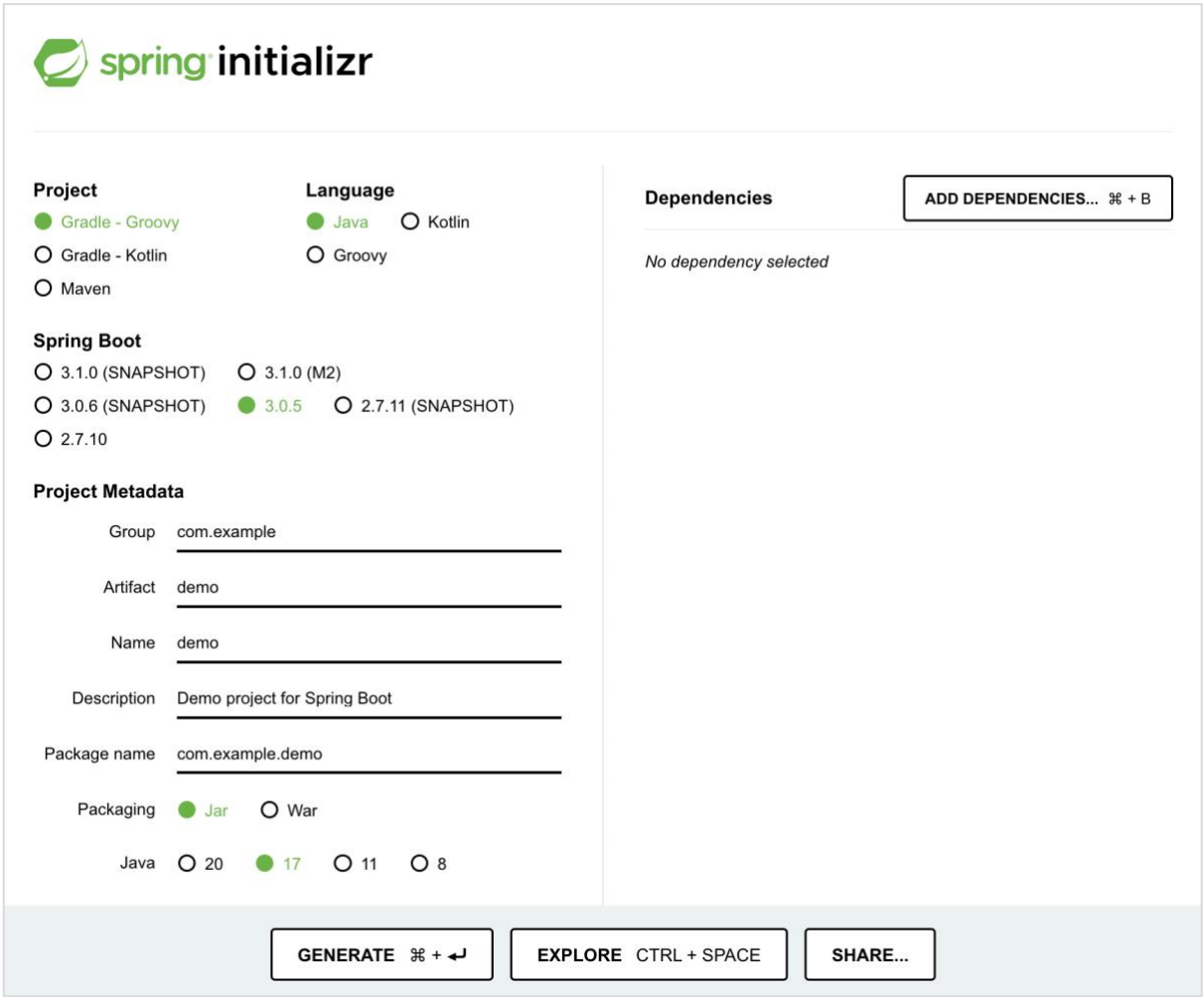

Figure 6.2: Spring Intializr web page

The Spring Initializr page consist of 5 components which are Project, Language, Spring Boot, Project Metadata and Dependencies.

2. Select the data shown in Table 6.1 for the first three components.

Table 6.1: Project component values

Component	Value
Project	Maven Project
Language	Java
Spring Boot	3.1.0 (Snapshot) or the latest

3. Enter and select the data shown in Table 6.2 for the Project Metadata fields.

Table 6.2: Project Metadata values

Field	Value
Group	my.edu.utem.ftmk.dad
Artifact	restorderapp
Name	restorderapp
Description	Lab exercise for REST provider application
Package name	my.edu.utem.ftmk.dad.restorderapp
Packaging	Jar
Java	17

4. Click the button **Add Dependencies**. A modal window as shown in Figure 6.3 will be displayed.

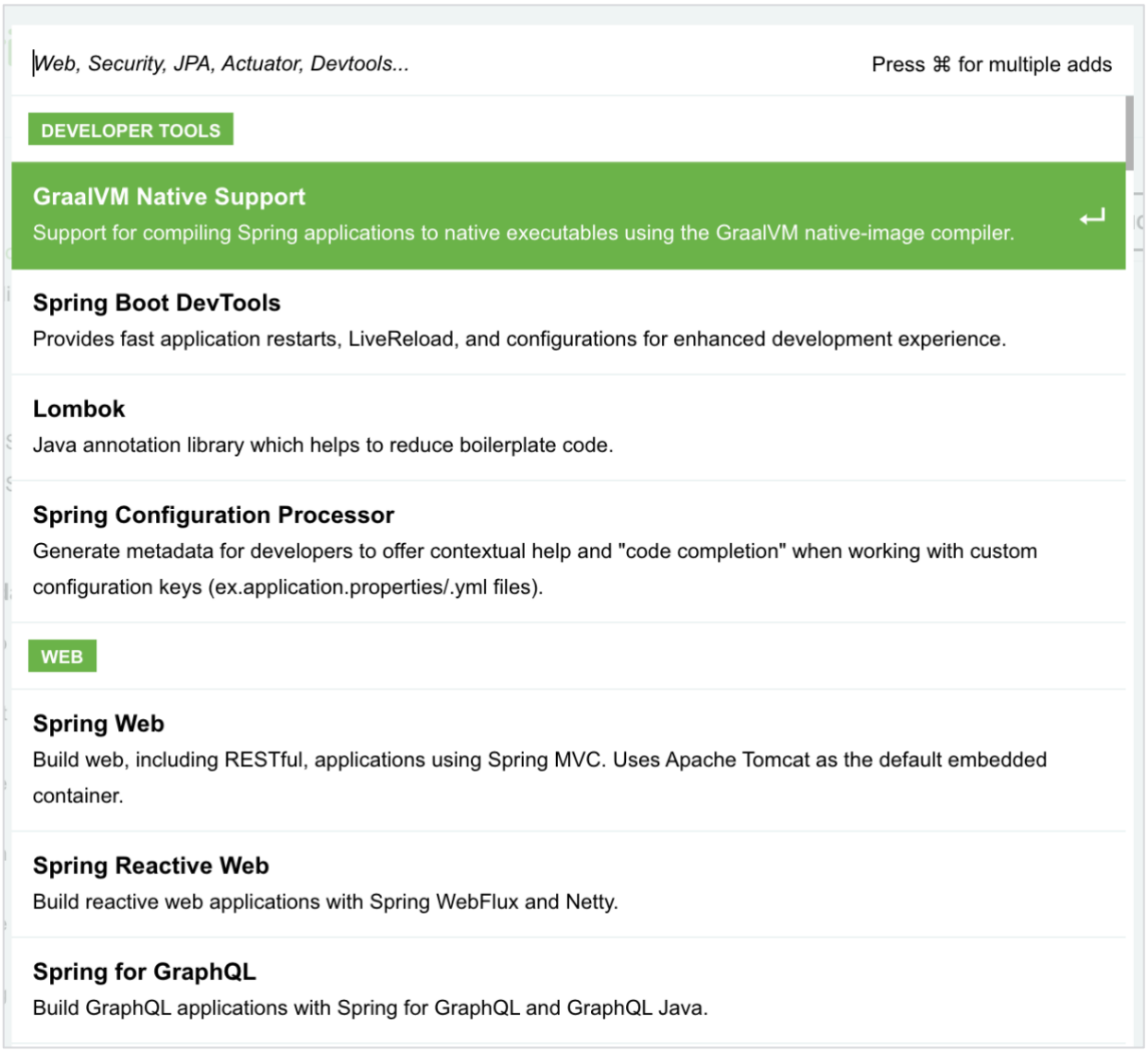

Figure 6.3: A window to add project dependencies

5. Type web in the text box shown in Figure 6.4. A list of result will be displayed.

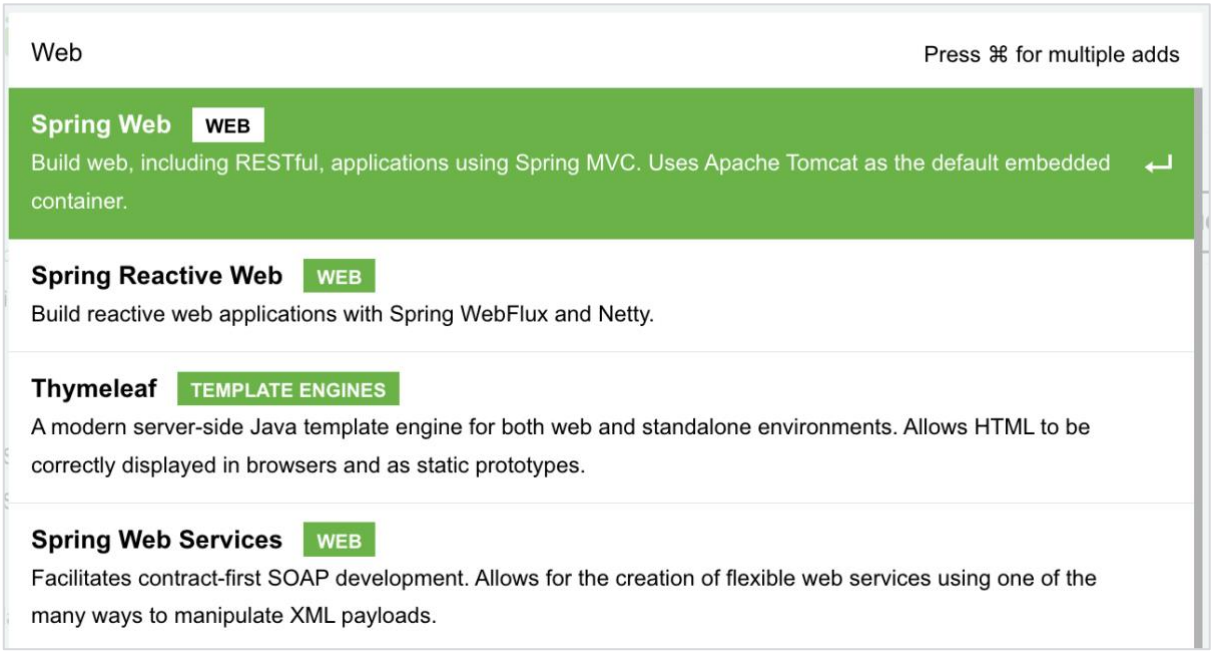

Figure 6.4: Result from web

6. Select **Spring Web** from the list. **Spring Web** will be added into the **Dependencies** list as shown in Figure 6.5.

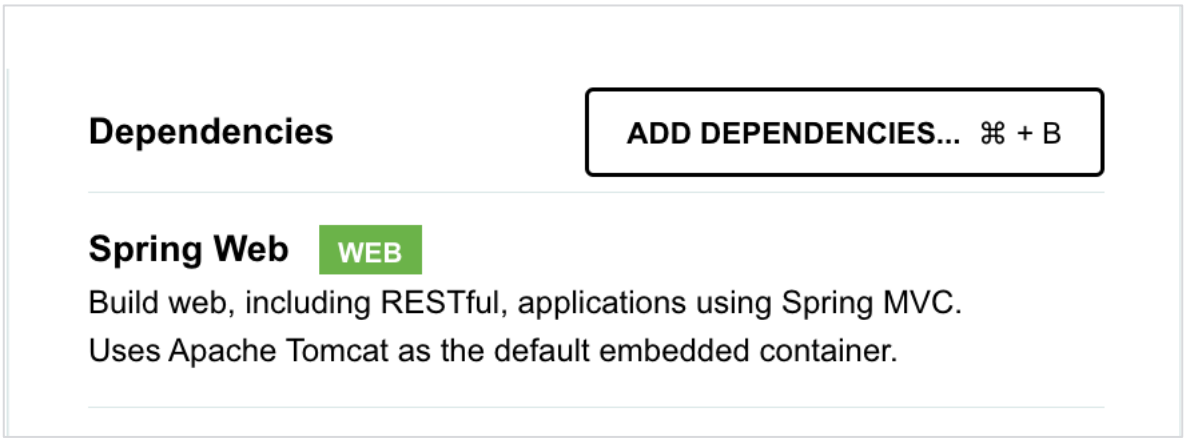

Figure 6.5: Spring Web is added into the Dependencies list

7. Repeat steps 5 and 6 for the following dependency artefacts.
   a. Spring Data JPA
   b. MySQL Driver
   c. Spring Boot Dev tools
8. The final project specification shall look like Figure 6.6.

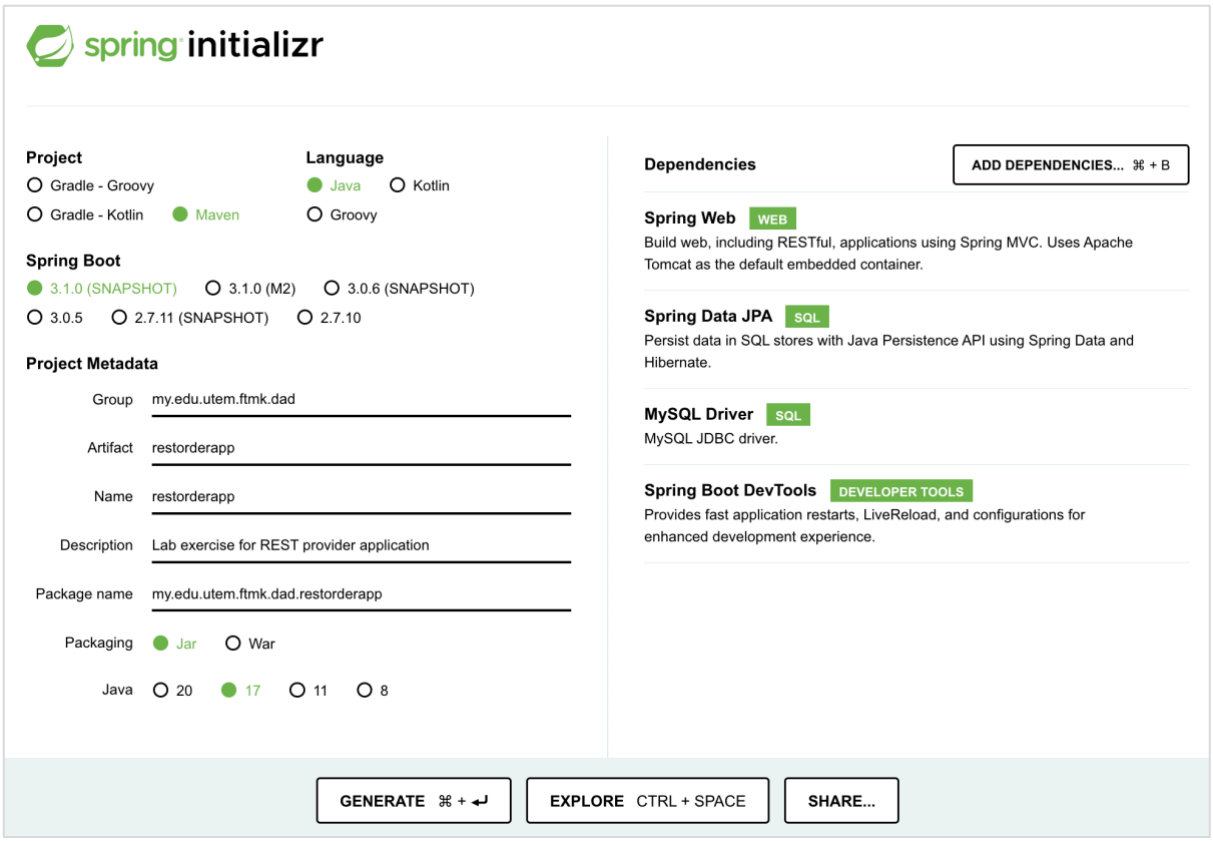

Figure 6.6: Spring Boot specification for restorderapp

9. Click the **Generate** button. A file named **restoderapp.zip** will be downloaded into your computer. Wait until it has finished downloading.
10. Locate the downloaded file on your computer.
11. Unzip the project. The content in the **restorderapp** folder shall be similar as in Figure 6.7.

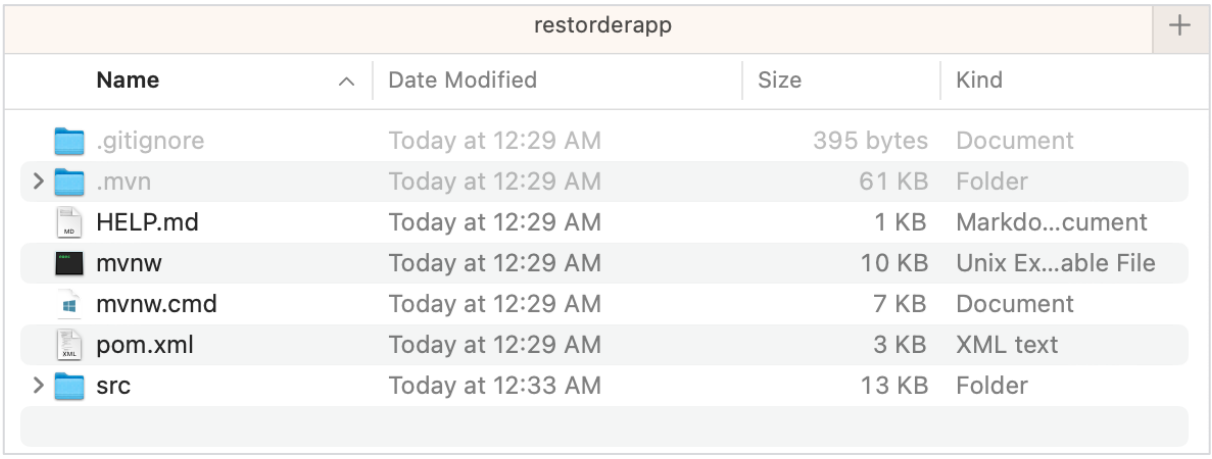

Figure 6.7: Content of restorderapp folder

6.3 Import a new Spring Boot Project

1. Open **Eclipse**.
2. Click **File** from the Eclipse menu.
3. Select **Import**, as shown in Figure 6.8.

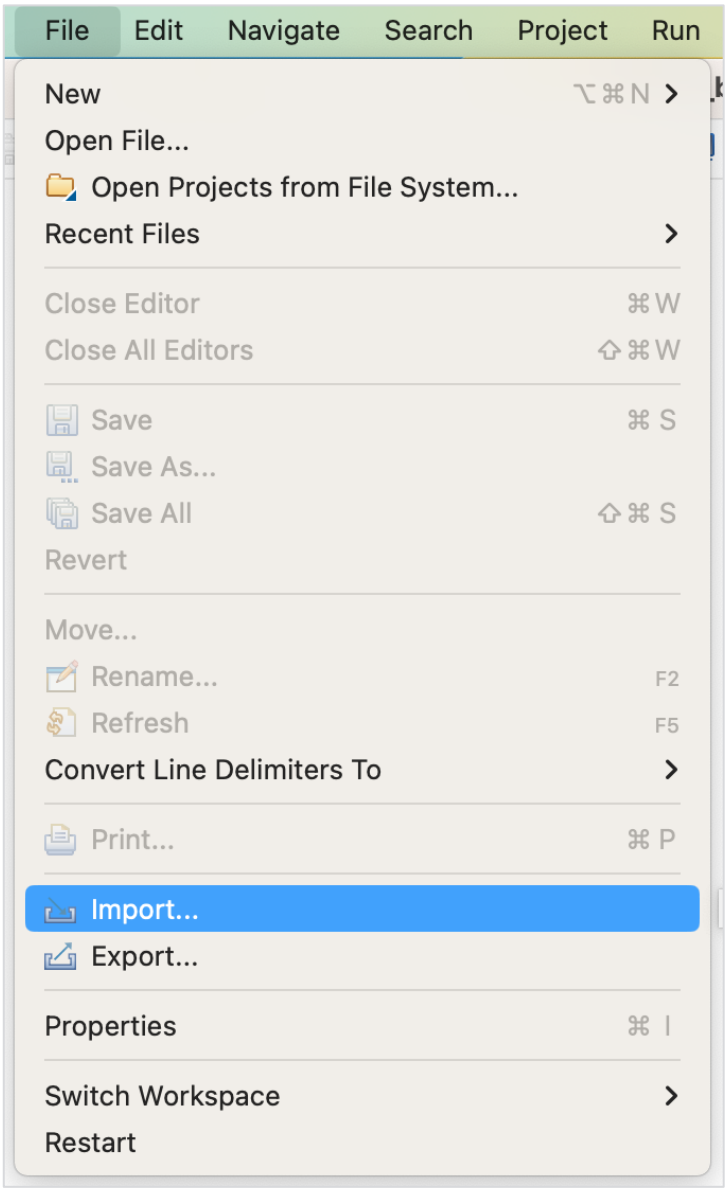

Figure 6.8: Menu to import a project

A window named **Import**, as shown in Figure 6.9 will be displayed.

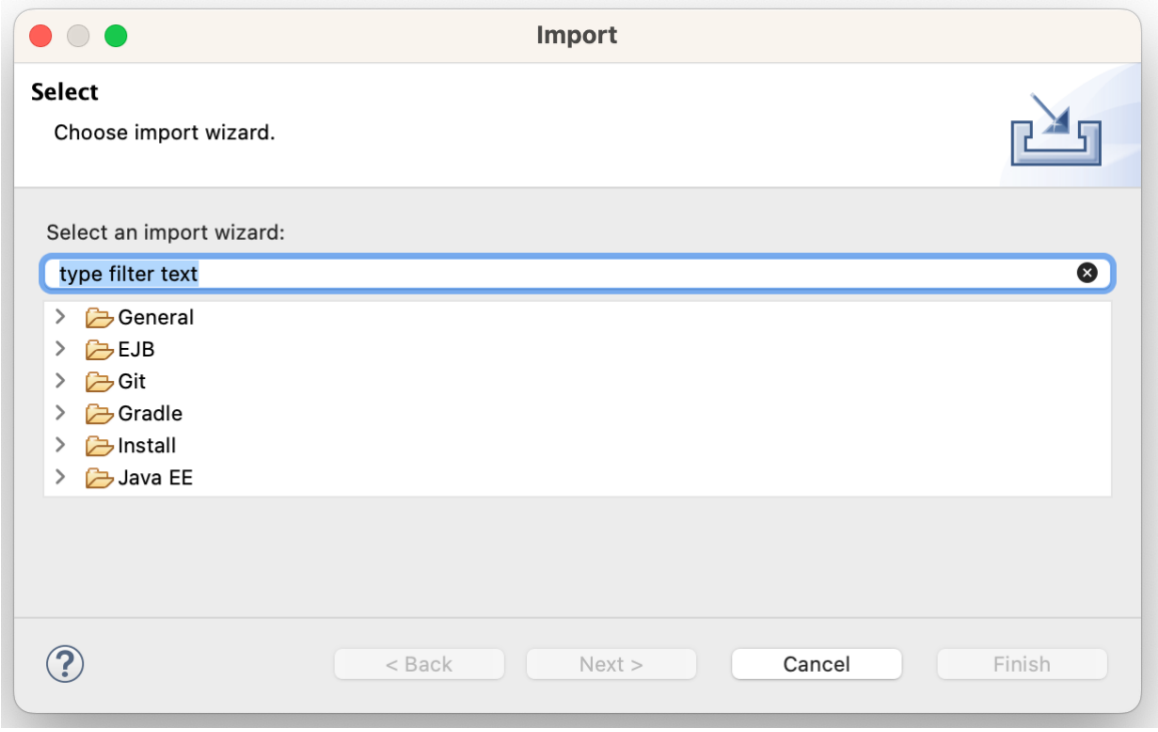

Figure 6.9: A window to import a project

4. Scroll down the list until a folder named **Maven** is found.
5. Select **Maven** > **Existing Maven Project**s, as shown in Figure 6.10.

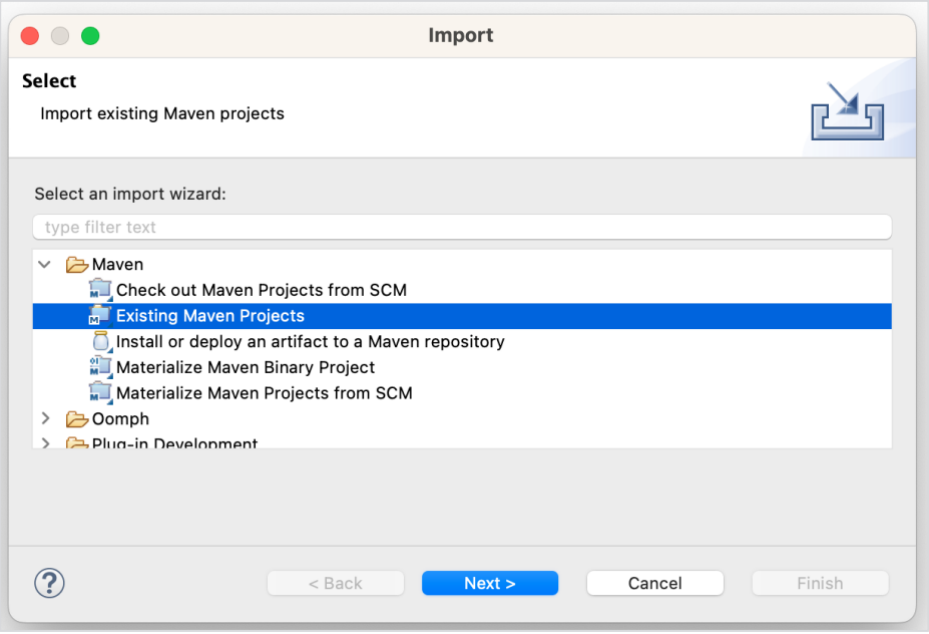

Figure 6.10: Selection to import existing Maven project

6. Then, click the **Next** button. A window named **Import Maven Projects**, as shown in Figure 6.11will be displayed.

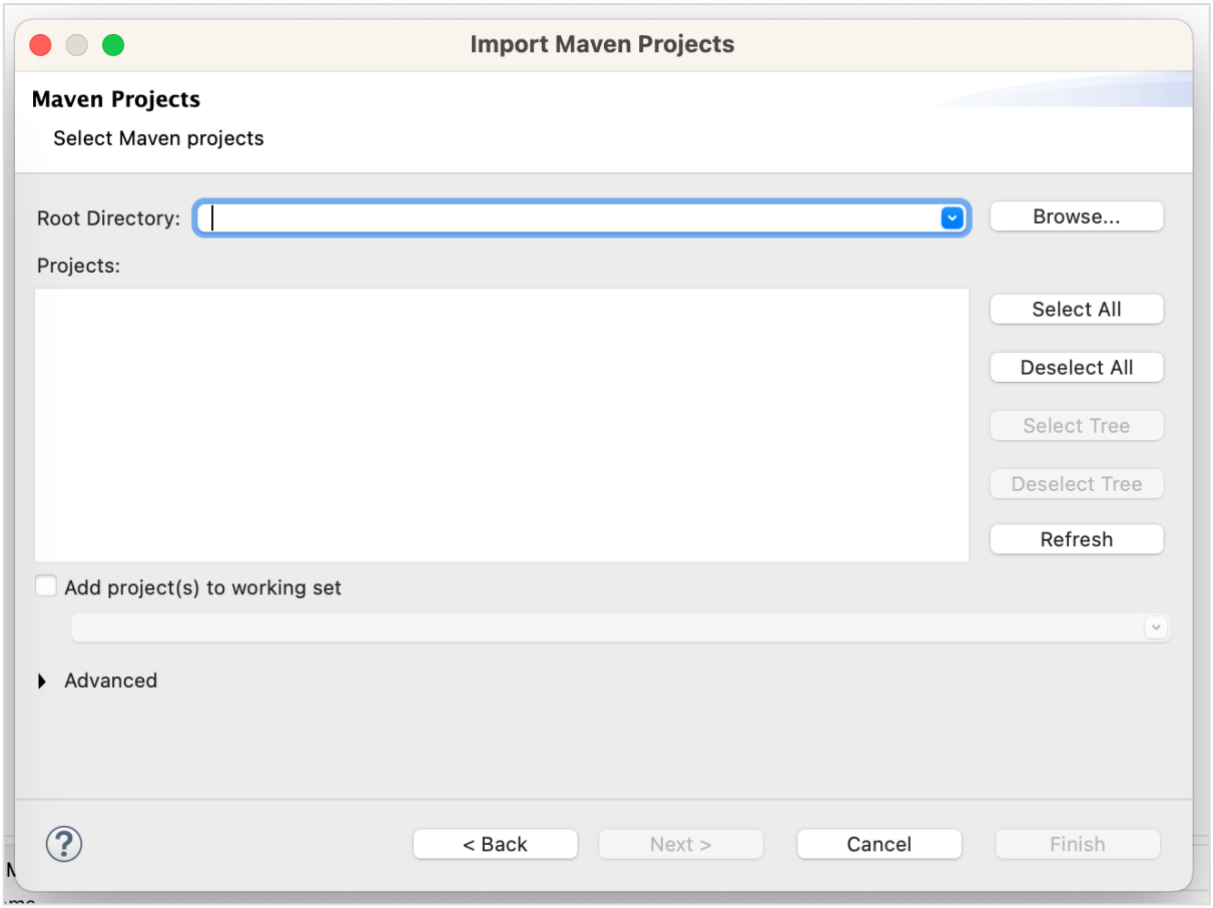

Figure 6.11: A window to import a Maven project

7. Click the **Browse** button. Locate the **restorderapp** folder that was unzipped in the previous exercise. Figure 6.12 the outcome from locating the folder.

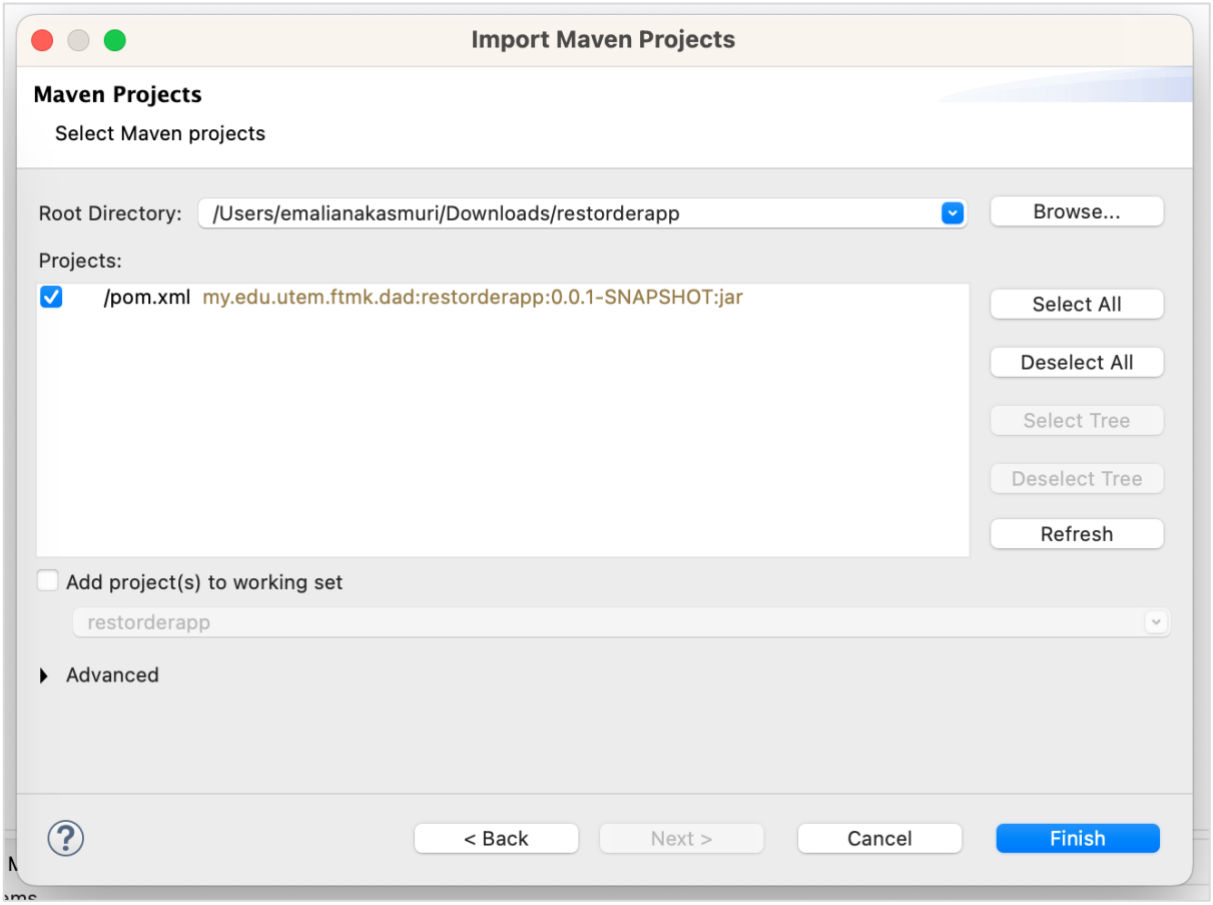

Figure 6.12: Importing restorderapp into Eclipse

8. Click the **Finish** button. The project will be imported into Eclipse. It will take some time for Eclipse to download the files relevant to the project.

6.4 Update a Maven Project

1. Right-click on **restorderapp** from the **Project Explorer**.
2. Select **Maven** > **Update Project**, as shown in Figure 6.13.

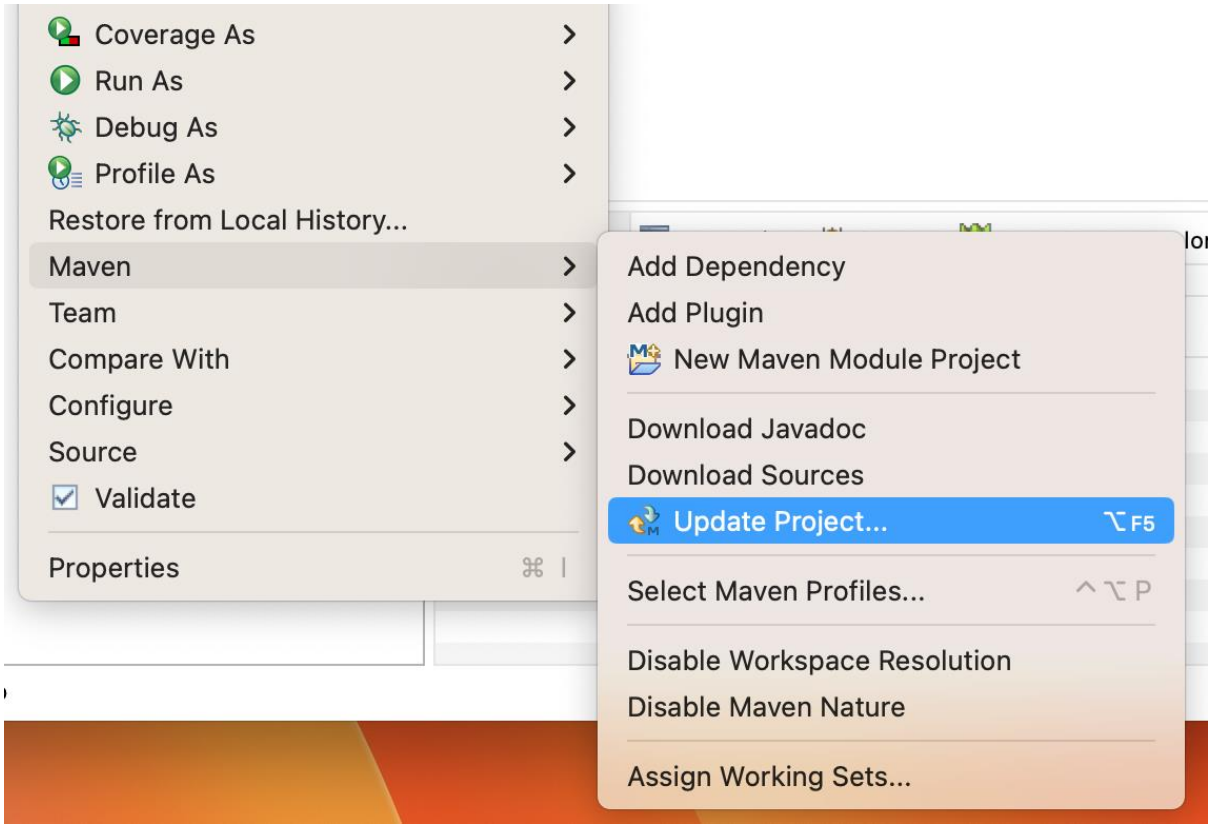

Figure 6.13: Menu to update a Maven project

3. A window named Update Maven Project, as shown in Figure 6.14 will be displayed.

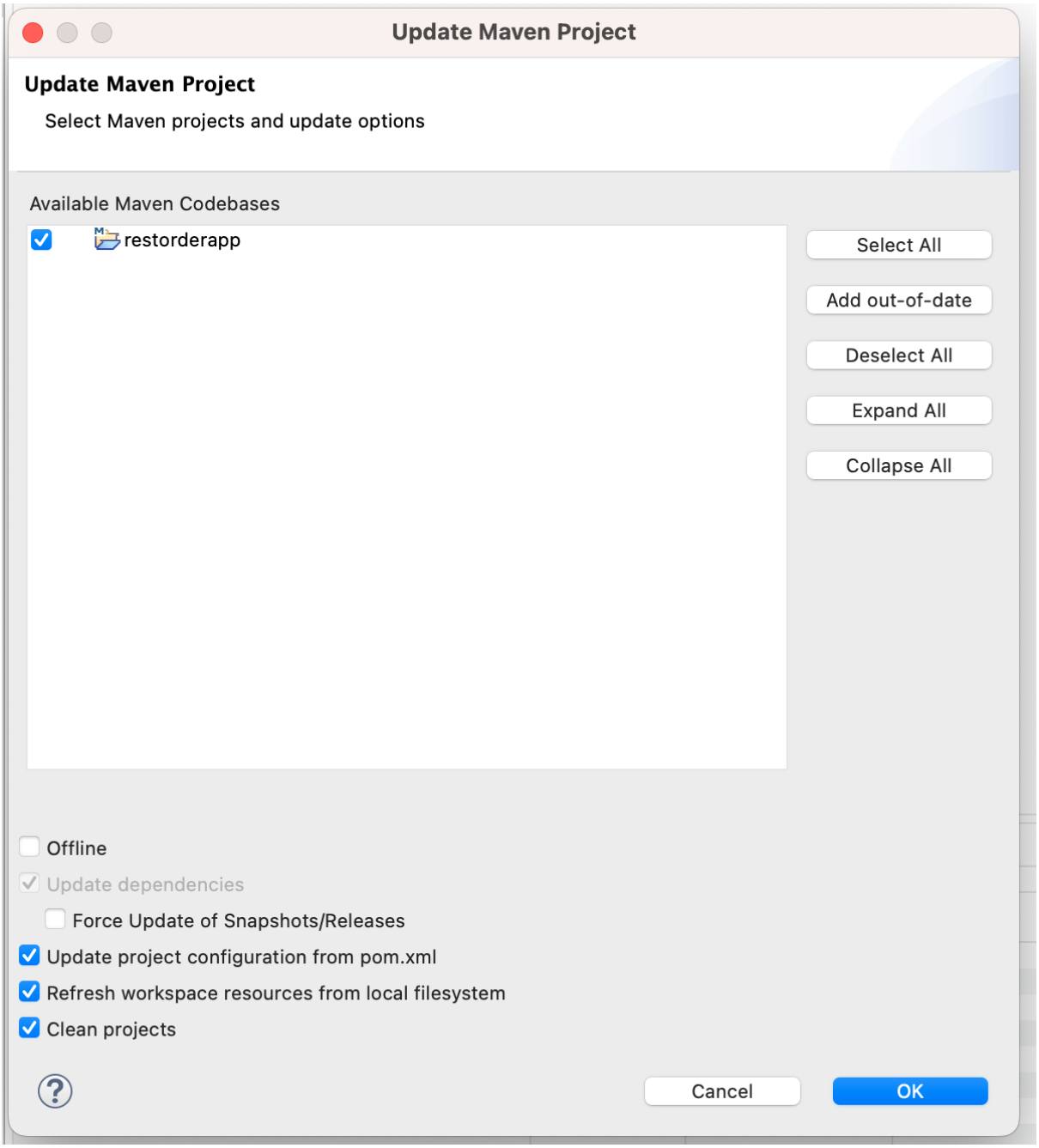

Figure 6.14: A window to update a Maven project

4. Check the following checkboxes.
    a. Force Update of Snapshots/Releases
    b. Update project configuration from pom.xml
    c. Refresh workspace resources form local filesystem
    d. Clean projects
5. Then, click the OK button. Eclipse will be updating **restorderapp** project. It might take some time, depending on the speed of the network.

6.5 Execute the new Spring Boot Project

1. Right-click on **restorderapp** from the **Project Explorer**.
2. Select **Run As** > **Maven Build** as shown in Figure 6.15.

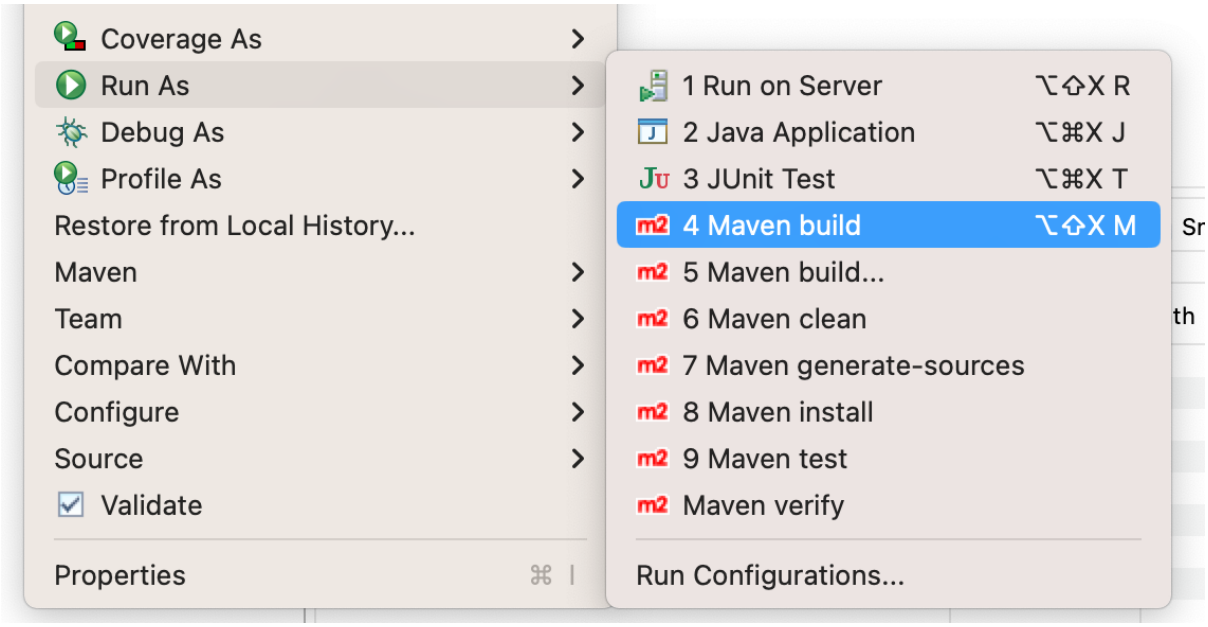

Figure 6.15: Menu to build a Maven project

3. A window named **Edit Configuration**, as shown in Figure 6.16 will be displayed.

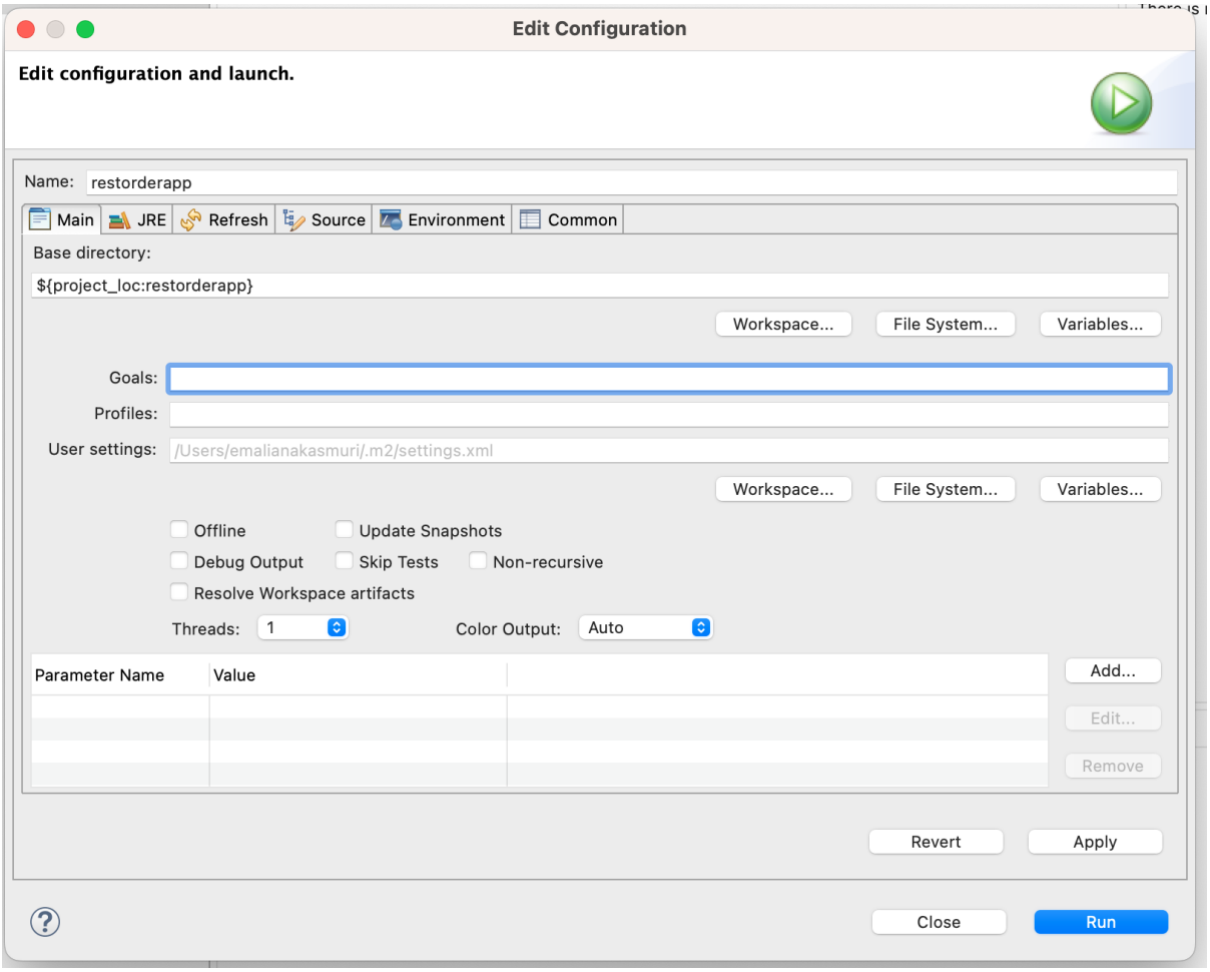

Figure 6.16: A window to configure project build and execution

4. Enter the spring-boot:run in the **Goal** text box.
5. Then, click **Apply**.
6. After that, click **Run**. The console should display Application Failed to Start and Build Success, as shown in Figure 6.17.

```
***************************
APPLICATION FAILED TO START
***************************

Description:

Failed to configure a DataSource: 'url' attribute is not specified and no embedded datasource could be configured.

Reason: Failed to determine a suitable driver class


Action:

Consider the following:
        If you want an embedded database (H2, HSQL or Derby), please put it on the classpath.
        If you have database settings to be loaded from a particular profile you may need to activate it (no profiles are currently

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  9.389 s
[INFO] Finished at: 2023-04-10T01:12:05+08:00
[INFO] ------------------------------------------------------------------------
```

Figure 6.17: Some messages from the execution

6.6 Configure REST Provider Application

1. Expand **restorderprovider** from the **Project Explorer**.
2. Expand **src/main/resources**. The content of the folder should be similar as shown in Figure 6.18.

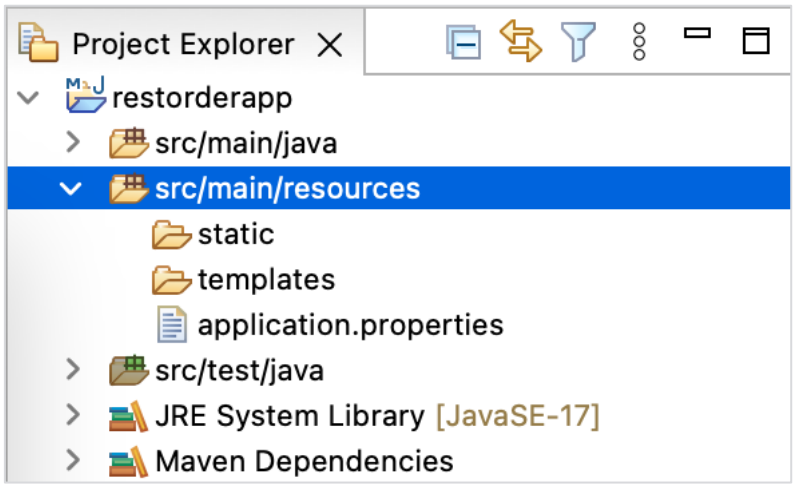

Figure 6.18: The content of src/main/resources

3. Double-click **application.properties** from the **Project Explorer**. The file will be opened in an editor.

4. Enter the configuration properties shown in Figure 6.19 into **application.properties**.

```
# Database configuration
spring.datasource.url =
jdbc:mysql://127.0.0.1:3306/orderdb_dev

spring.datasource.username = [MySQL Username]
spring.datasource.password = [MySQL Password]
spring.jpa.properties.hibernate.dialect =
org.hibernate.dialect.MySQLDialect

spring.jpa.hibernate.naming.physical-strategy=
org.hibernate.boot.model.naming.PhysicalNamingStrategyStandar
dImpl

# Context path to access the application -> eg:
localhost:8080/orderapp
server.servlet.context-path = /orderapp
```

Figure 6.19: Configuration for application.properties

5. Save the file.

6.7 Re-Execute the Spring Boot Project

1. Update the Maven project (Refer to the steps in 5.4).
2. Execute the Maven project (Refer to the steps in 5.5).
3. Locate message Tomcat started on port(s): 8080 (http) with context path '/orderapp' from the **Console**. This message indicates that the project has been successfully executed.

7 Testing environment preparation

There are three activities in the testing environment preparation. The activities are: -

1. Create a new Postman's workspace.
2. Create a new web service collection.
3. Document the new web service collection.

The relationship between these processes is shown in Figure 7.1.

Figure 7.1: Relation of activities in the testing environment preparation

7.1 Create a new Postman's Workspace

1. Open **Postman**.
2. Click the menu **Workspace** from the Postman menu bar, as shown in Figure 7.2.

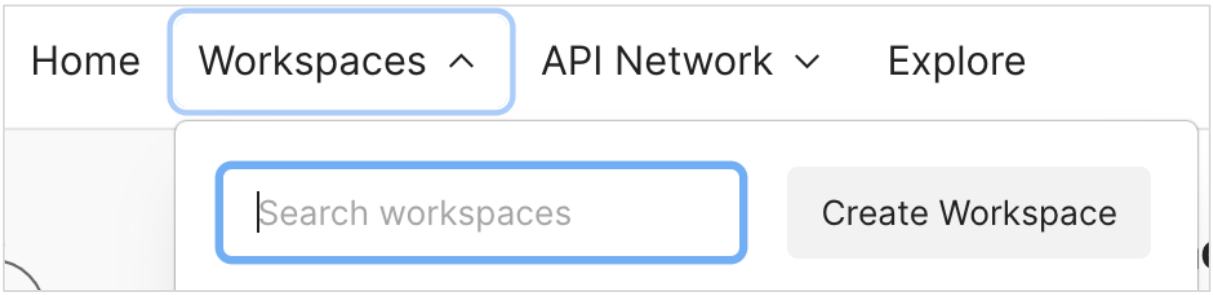

Figure 7.2: Menu to create a new workspace

3. Click the button **Create Workspace** from the menu in Figure 7.2.
4. A screen named **Create workspace** as shown in Figure 7.3 will be displayed.

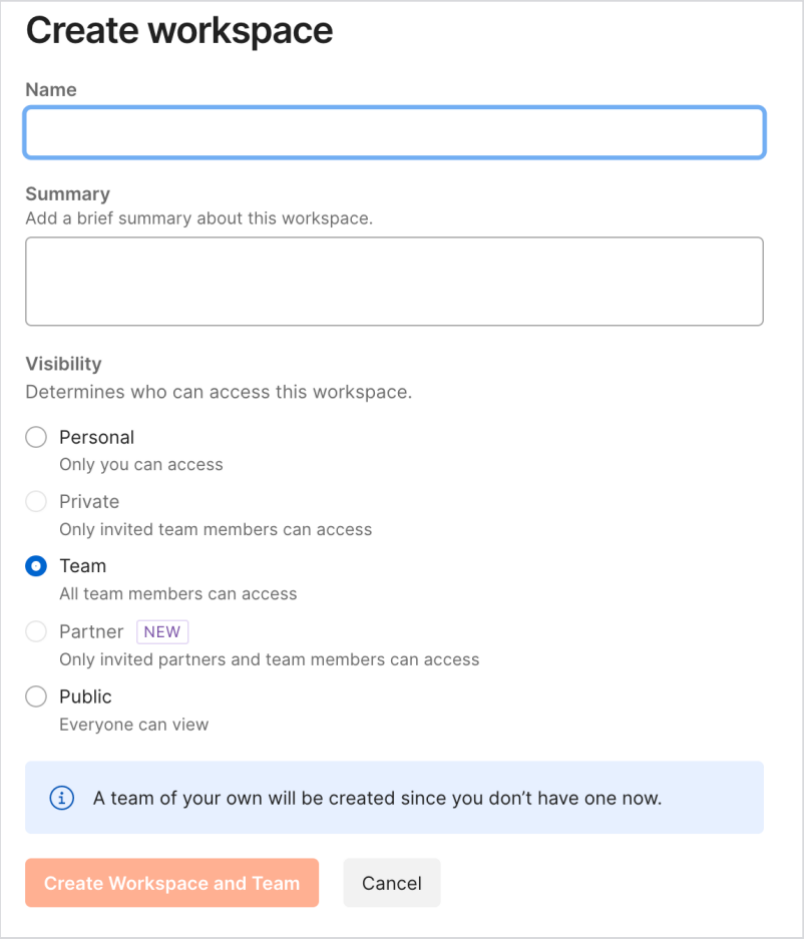

Figure 7.3: A screen to create a new workspace

5. Enter the information shown in Table 7.1 to create a new workspace.

Table 7.1: Field value to create a new workspace

Field	**Value**
Name	workspace_bitp3123_rest_orderapp
Summary	A workspace to test REST web service for orderapp.
Visibility	Personal

The outcome should be similar as shown in Figure 7.4.

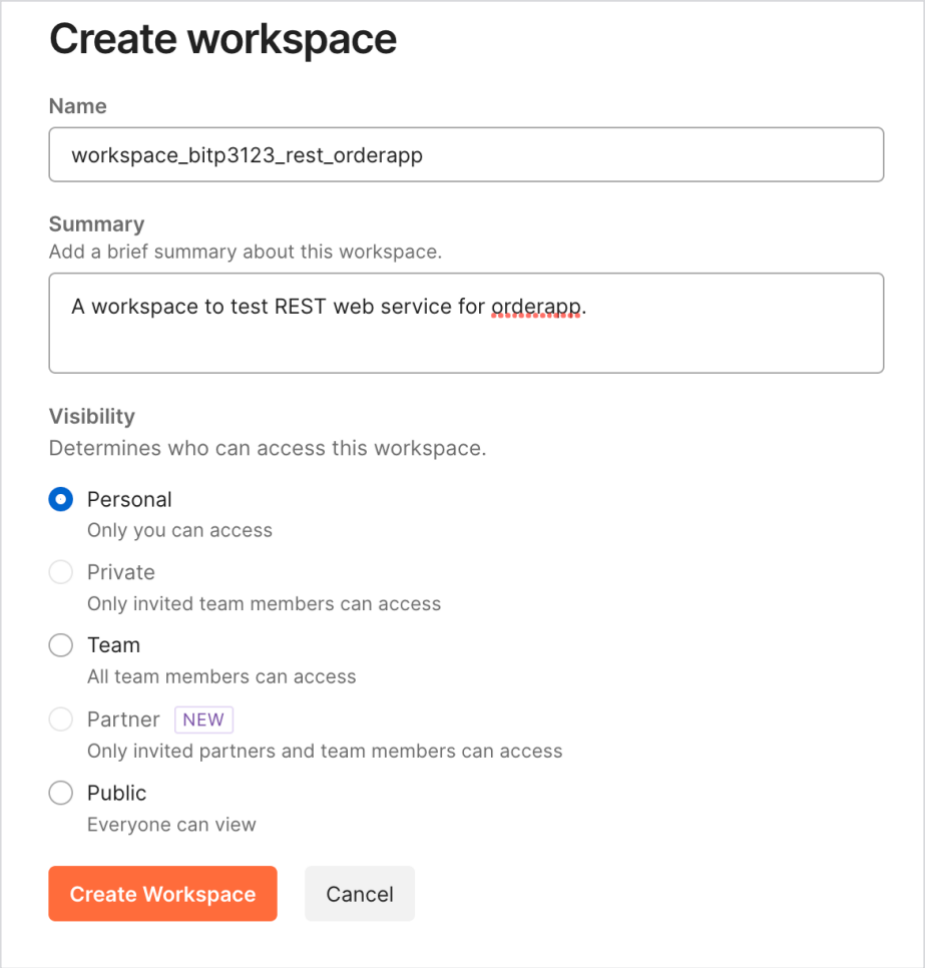

Figure 7.4: Final specification to create a new workspace

6. Click the **Create Workspace** button. A new workspace similar to Figure 7.5 will be created.

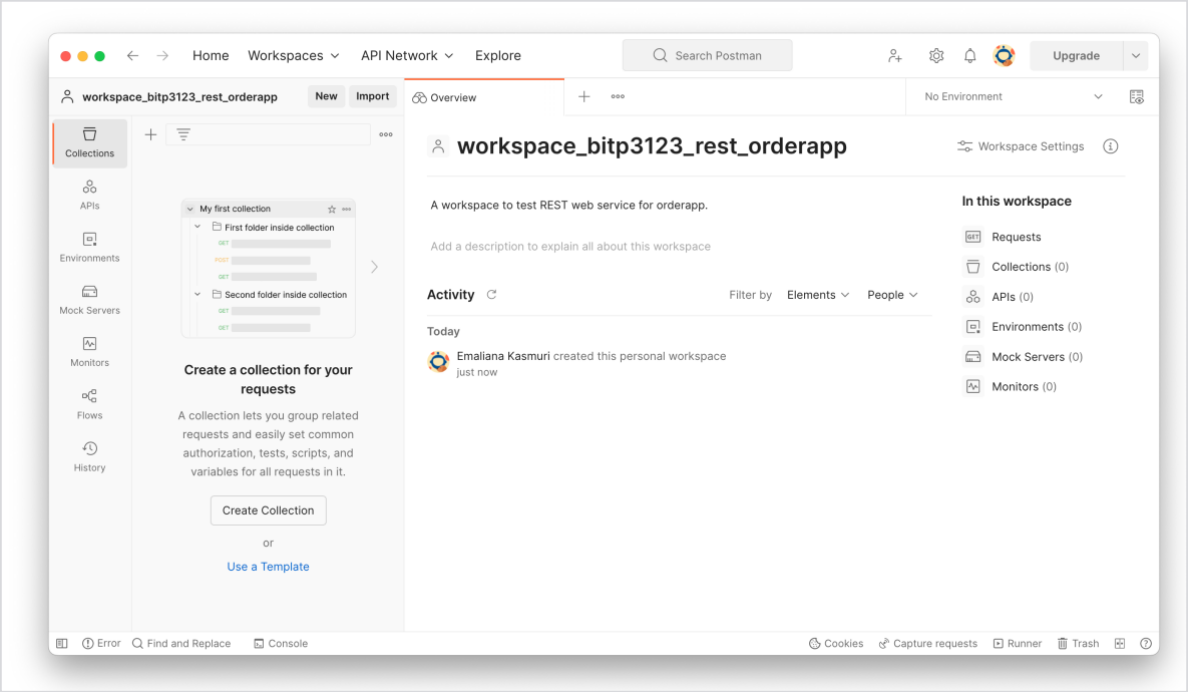

Figure 7.5: A new Postman workspace for Lab Week 04

7.2 Create a New Request Collection

1. Click the button '**+**' as shown in Figure 7.6, to create a new collection.

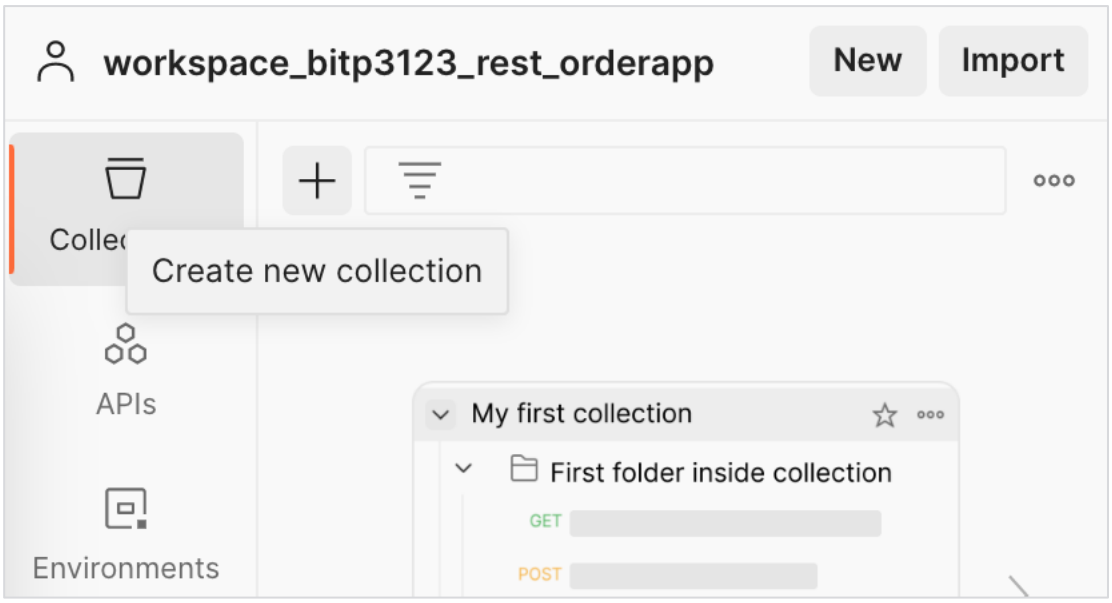

Figure 7.6: Button to create a new request collection

2. A tab named **New Collection**, as shown in Figure 7.7 will be displayed at the right-hand side of the workspace.

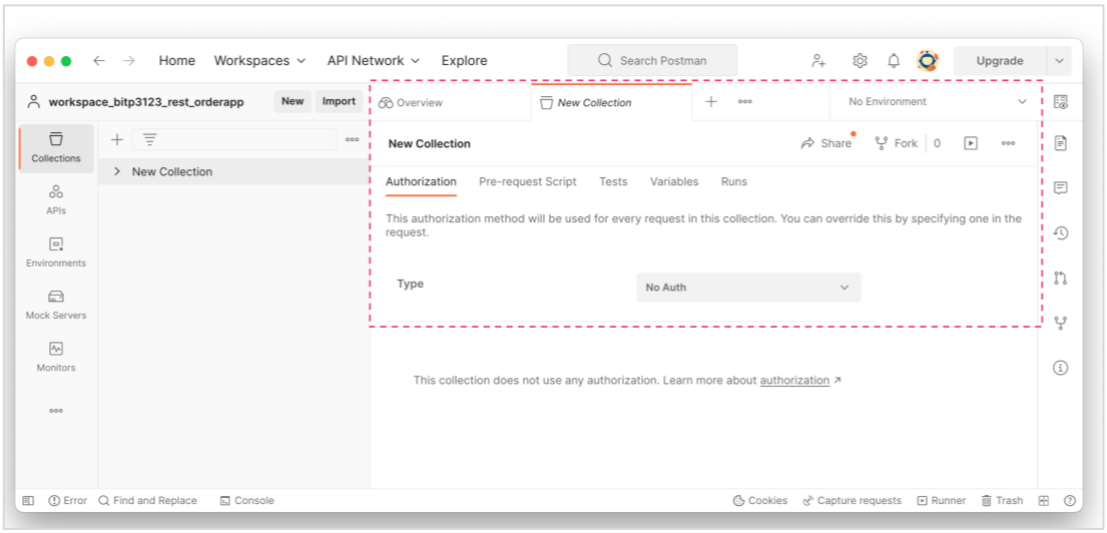

Figure 7.7: A new collection is created (red dotted box)

3. Bring the mouse to the **New Collection**. Two icons, as shown in Figure 7.8 will be displayed.

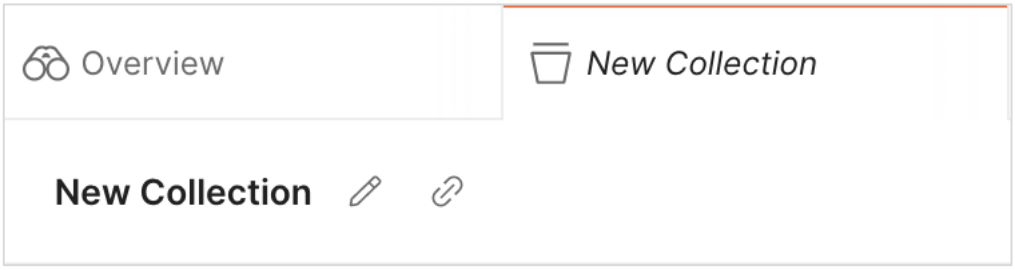

Figure 7.8: Icons to manage the new collection

4. Click the **pencil icon**. The text New Collection will change to an editable text box, as shown in Figure 7.9.

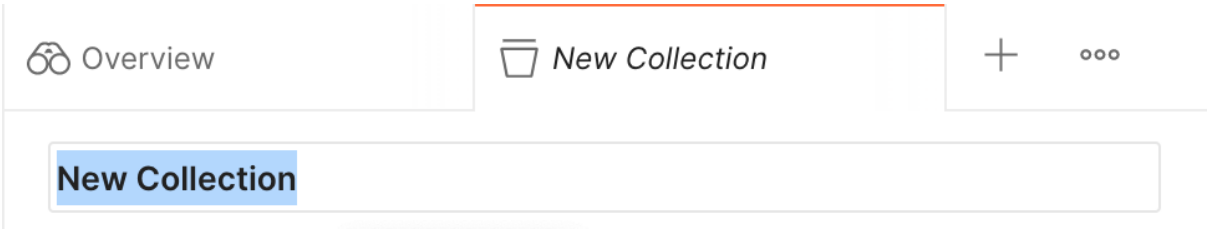

Figure 7.9: Text box to rename the New Collection.

5. Enter **REST Product Type** into the text box.
6. Then press Enter. A new collection will be created. It should appear in the collection list, as shown in Figure 7.10.

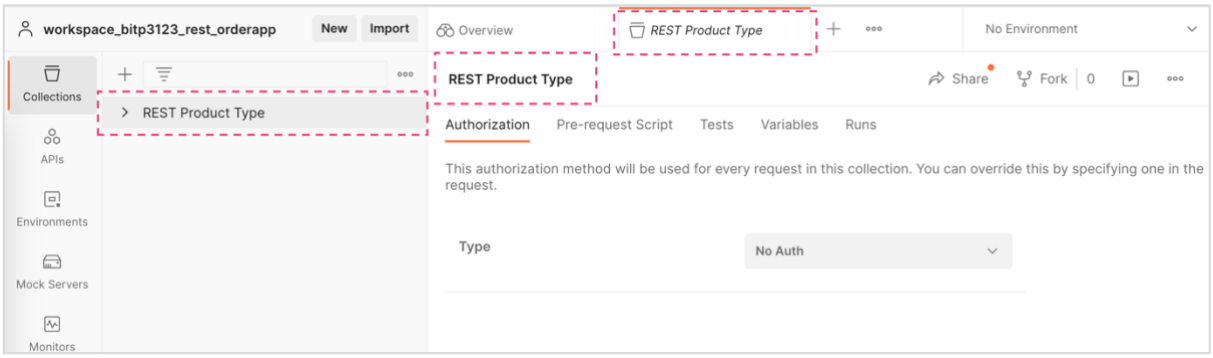

Figure 7.10: REST Product Type in Collection List

7.3 Document the Collection

1. Click the **Documentation** button from the right-hand side menu as shown in Figure 7.11, to document the collection.

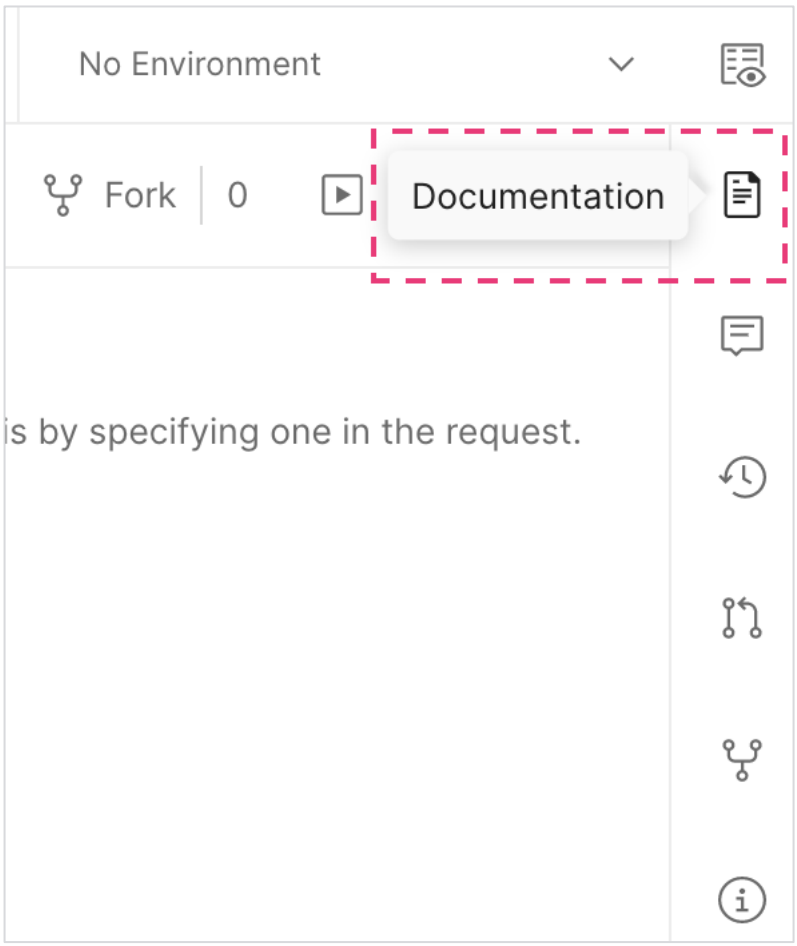

Figure 7.11: Button to open Documentation window for the collection

A screen named Documentation as shown in Figure 7.12 will be displayed.

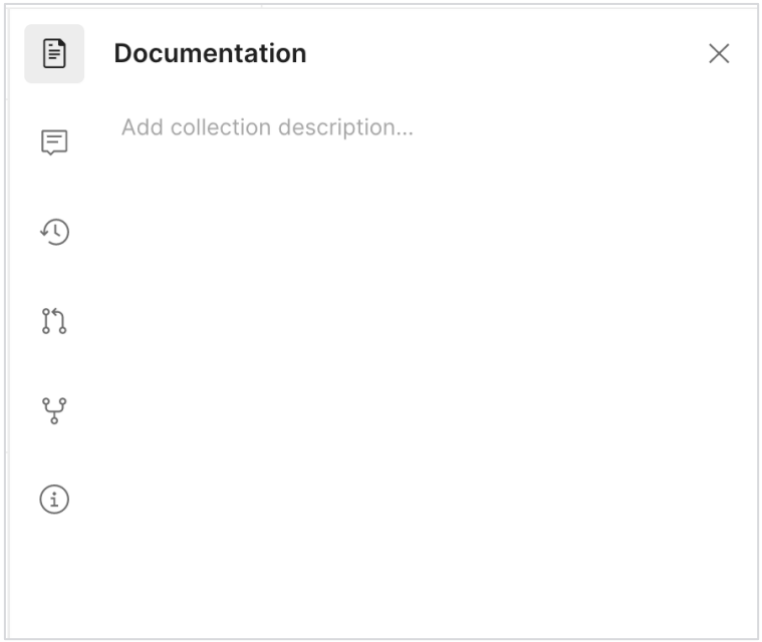

Figure 7.12: A screen to describe the purpose of the collection

2. Point the mouse pointer into the area labelled **Add collection description**.
3. A pencil icon will appear. Click the icon. The look of the screen will change, similar to Figure 7.13.

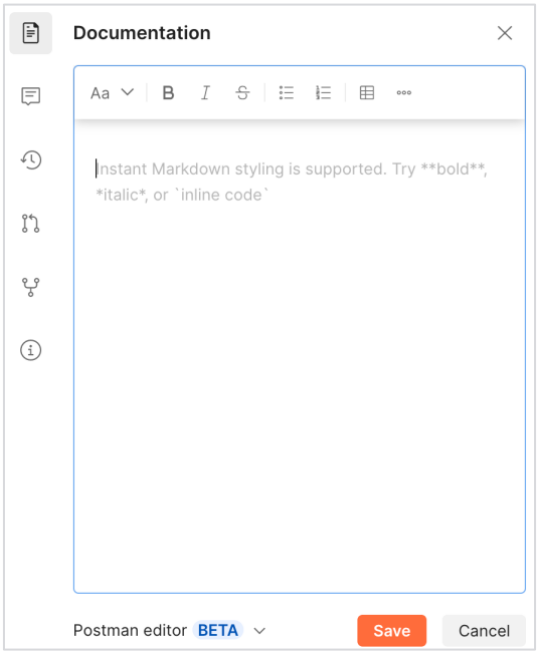

Figure 7.13: An editable documentation

4. Enter This collection contains a list of REST web services to access the product types' of information in the editor.

5. Then, click the **Save** button.

END OF DOCUMENT