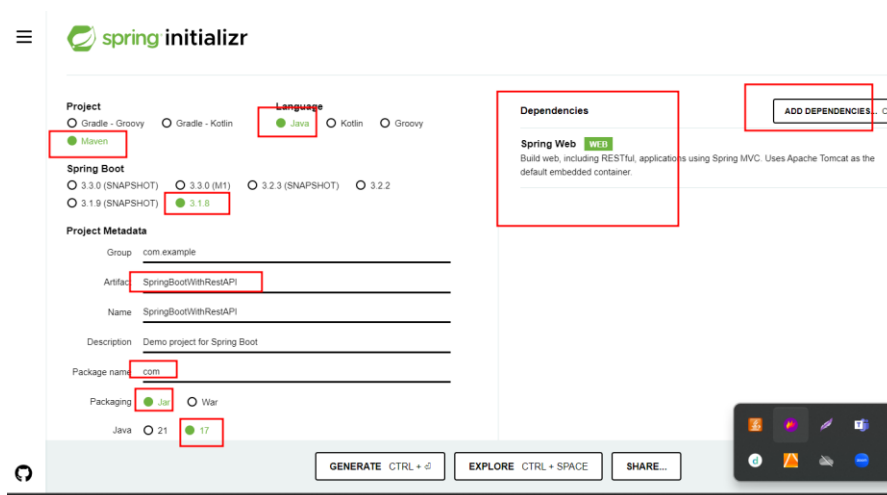


## Spring boot Rest full Web Service

In Spring boot we need to add web starter and in place @Controller

We need to write @RestController annotation.

When we write @RestController annotation we can write return type of method string or user defined object with produce attribute.



@Controller annotation is use to return view as jsp or thymeleaf

@RestController annotation is use to return rest api.

Before @RestController we if want to use rest api concept

@RestController = @Controller + @ResponseBody;

## Spring boot with Rest API interacting with database using Spring JPA Data.

Entity can be Employee, Customer, Manager, Product, order etc.

ProductManagementRestAPI -> project name

Web starter -> to create web service

Jpa starter --> to use core jpa or spring jpa data.

Mysql connector -> to connect mysql database

Dev starter -> it is use to refresh the project whenever we do any changes.

(only in development mode).

The screenshot displays the Spring Initializr web application interface. The 'Project' section has 'Maven' selected under 'Project' and 'Java' under 'Language'. The 'Spring Boot' section has '3.1.8' selected. The 'Project Metadata' section includes fields for 'Group' (com.example), 'Artifact' (ProductManagementRestAPI), 'Name' (ProductManagementRestAPI), 'Description' (Demo project for Spring Boot), 'Package name' (com), 'Packaging' (Jar), and 'Java' version (17). The 'Dependencies' section lists 'Spring Web', 'Spring Data JPA', 'MySQL Driver', and 'Spring Boot DevTools'. A red box highlights the 'Dependencies' section and the 'ADD DEPENDENCIES...' button. The bottom of the page features buttons for 'GENERATE', 'EXPLORE', and 'SHARE...'.

## Spring JPA Data.

JPA Data provided two interface ie CrudRepository and JpaRepository.

These interface base upon java 8 features. Which provided implementation of dao layer.

Java 8 onward interface can contains method with body.

CrudRepository → Basic Repository

JpaRepository → Jpa specific repository

JpaRepository interface extends CrudRepository.

We need to create normal interface and that implements may be extends CrudRepository or JpaRepository.

Which takes generic type. 1 class name and 2<sup>nd</sup> data type of primary key.

**@RequestBody** annotation is use to extract the data from request body part.

Post, put and patch data must be pass through request body part

## Product

Pid → auto increment PK

Pname →

Price →

Qty --→

|   |    |       |    |
|---|----|-------|----|
| 1 | TV | 56000 | 10 |
|---|----|-------|----|

## Orders

Oid → auto\_increment PK

Pid --→ FK

Datetime →

Qty --→

totalAmount --→

|     |   |             |   |         |
|-----|---|-------------|---|---------|
| 100 | 1 | 25-Jan-2024 | 2 | 2*56000 |
|-----|---|-------------|---|---------|

Don't create table name as Order because **order** by clause keyword present in database. **Orders**.

## Product with Orders relationship

### One to many relationship

In orders table

Oid → auto increment.

Pid → FK it can 1 or 2 or 3.

Ldt → datetime (system time).

Qty → 1 or 2 or 3 or 4

Total → qty \* price

We need to pass only pid, qty etc or (pname and qty)

