

Spring Boot 3 REST API with Spring Data JPA CRUD

Same entity as the previous project, but in this occasion we will make use of that Spring Data Dependency. In my cas I used an in-memory database “H2” but I will also show you how to make use of a persistent database with a simple change.

The project was created using Spring Initializer with the following settings:

- Project: Maven
- Language: Java
- Spring Boot: 3.3.3
- Project Metadata:
 - Group: com.example
 - Artifact: jpa-crud-api
 - Name: jpa-crud-api
 - Package Name: com.example.jpacrudapi
 - Packaging: Jar
 - Java: 17

The following dependencies were added:

- Spring Web: For REST APIs.
- Spring Data JPA: To work with JPA and repositories.
- H2 Database: For an in-memory database (or you can choose MySQL/Postgres)
- Lombok: To simplify entity and DTO classes

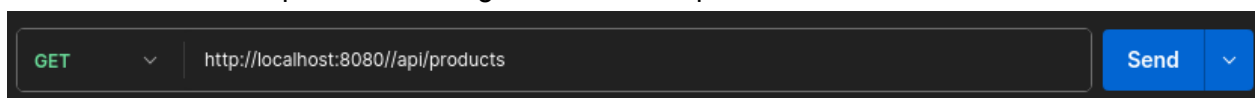
You can test the application by running the project(JpaCrudApiApplication.java) and trying out the Get, Post, Put, Delete operations using a client like postman and accessing them via localhost.

To change to a persisten database you just have to modify the “application.properties” file inside of the folder resources, delete or comment the h2 configuration lines and put the following for MySQL:

```
spring.datasource.url=jdbc:mysql://localhost:3306/products_db
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

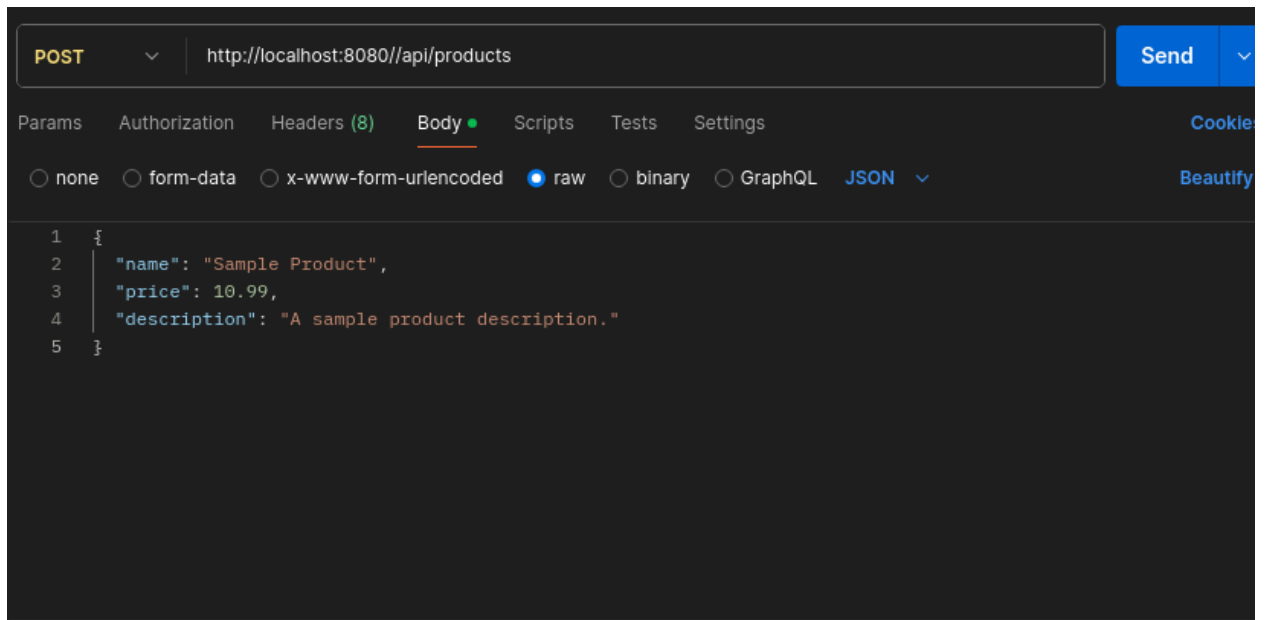
Here is an example of the API being tested:

- First in Postman let's put the following in a new workspace:



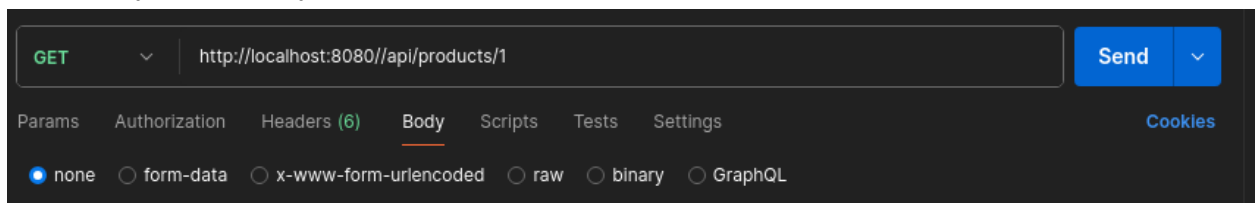
We have not added products yet so it will just show “[]”

- Then we can try POST by putting the following:



It will automatically assign id to 1

- So then if you GET it by that id 1:



You get the Sample Product