Angular Java Assignment:

Technology Stack:
Angular 11, Spring Boot, PostgreSQL

Git Repository (Clone respository):
https://github.com/amitshirke123/angular_mockapi_assignment.git

Installations and Steps to run application:

1. **Angular installations:**

   - Prerequisite: Angular 11, Node.js 12.19.0 or latest.
   - Run **npm install** in Angular project root directory (FESource).
   - Run **ng serve** in Angular project root directory.

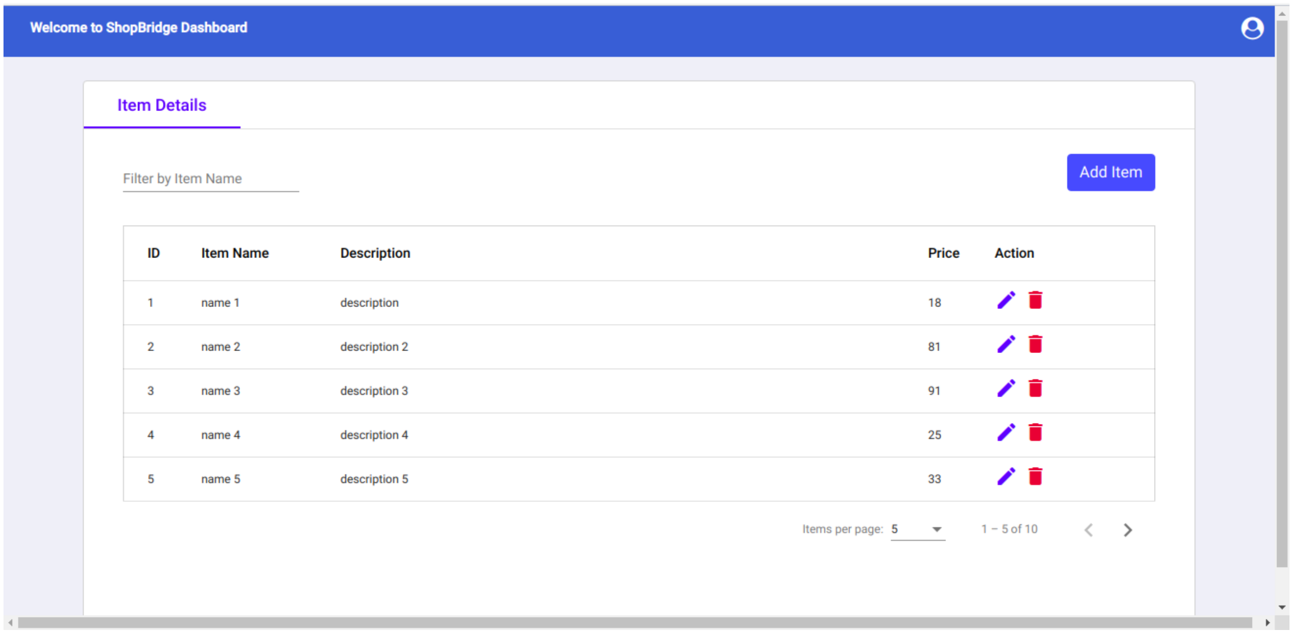
2. **Java backend:**
   - Prerequisite: Java 8
   - Run **InventoryApplication.java** file from Spring boot application.

3. **PostgreSQL:**
   - Prerequisite: PostgreSQL installed.
   - Create database inventorydb.
   - Table will get created in database when you run java application.

Project Modules:
Item details dashboard

Module functions:
1. Display items list
2. Edit item details
3. Add new item
4. Search item from the list
5. Delete item

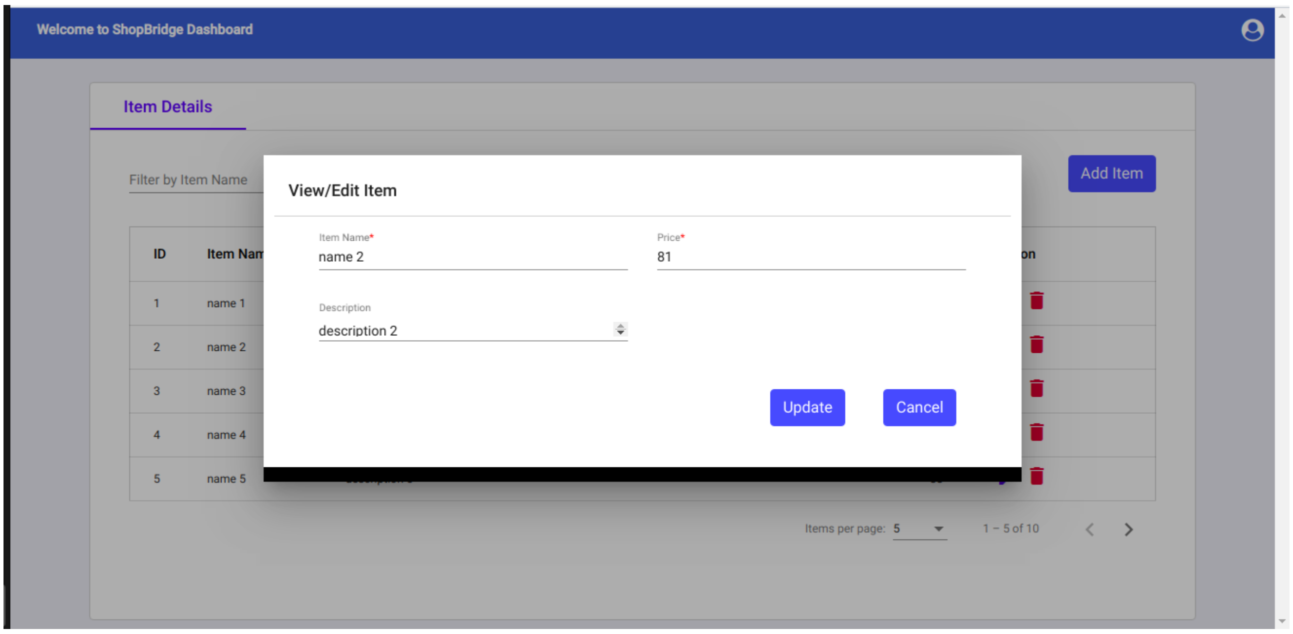
1. Display items list:



ID	Item Name	Description	Price	Action
1	name 1	description	18	
2	name 2	description 2	81	
3	name 3	description 3	91	
4	name 4	description 4	25	
5	name 5	description 5	33	

Items per page: 5   1 – 5 of 10

2. Edit Item Details:



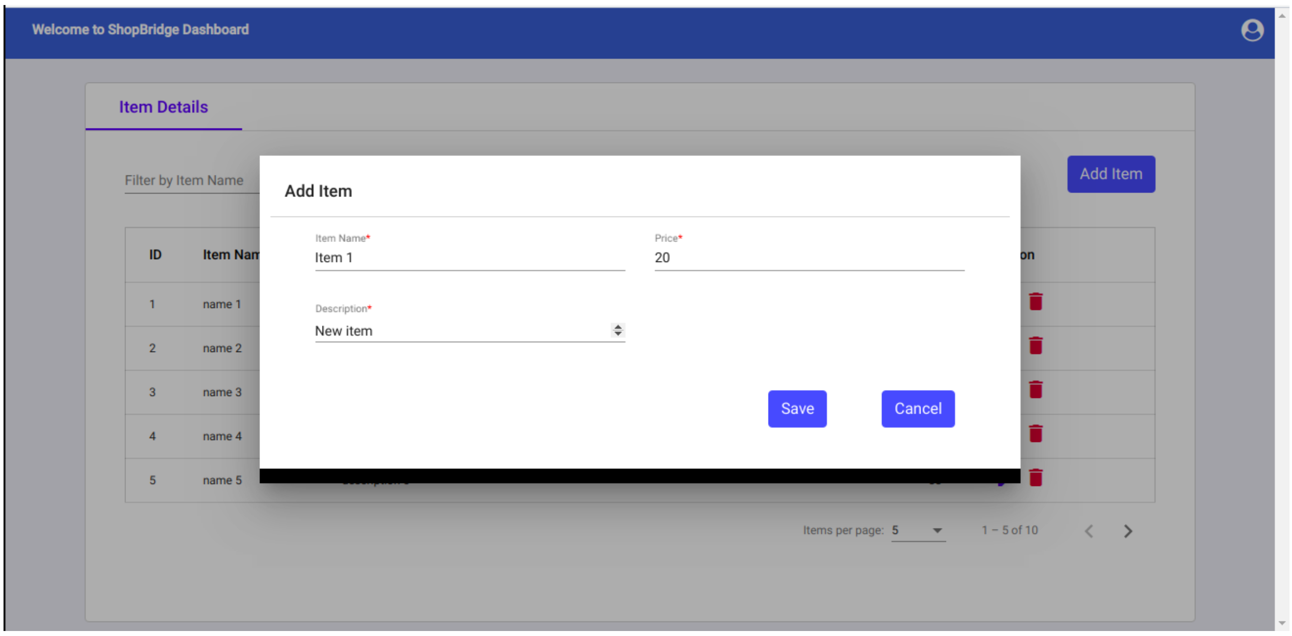
Item Name*: name 2

Price*: 81

Description: description 2

Items per page: 5   1 – 5 of 10

3. Add new item:


Welcome to ShopBridge Dashboard

Item Details

Filter by Item Name

Add Item

ID	Item Nam	on
1	name 1	
2	name 2	
3	name 3	
4	name 4	
5	name 5	

Items per page: 5 1 – 5 of 10

Add Item

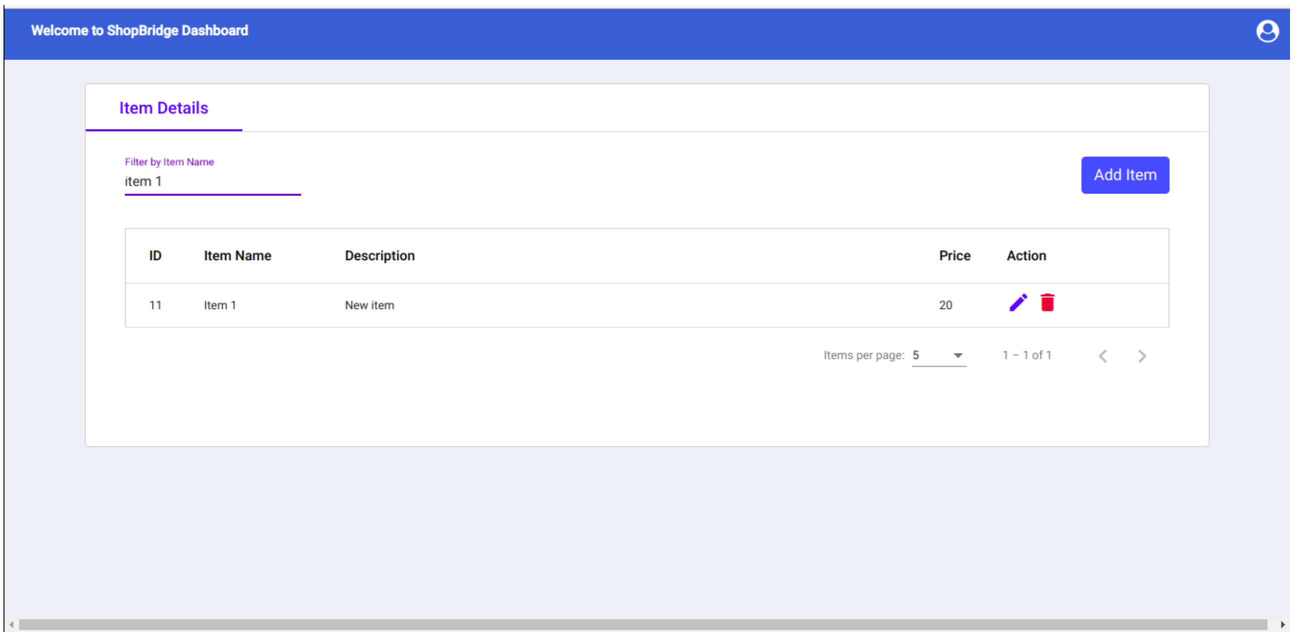
Item Name*
Item 1

Price*
20

Description*
New item

Save Cancel

4. Search item from the list:


Welcome to ShopBridge Dashboard

Item Details

Filter by Item Name
item 1

Add Item

ID	Item Name	Description	Price	Action
11	Item 1	New item	20	

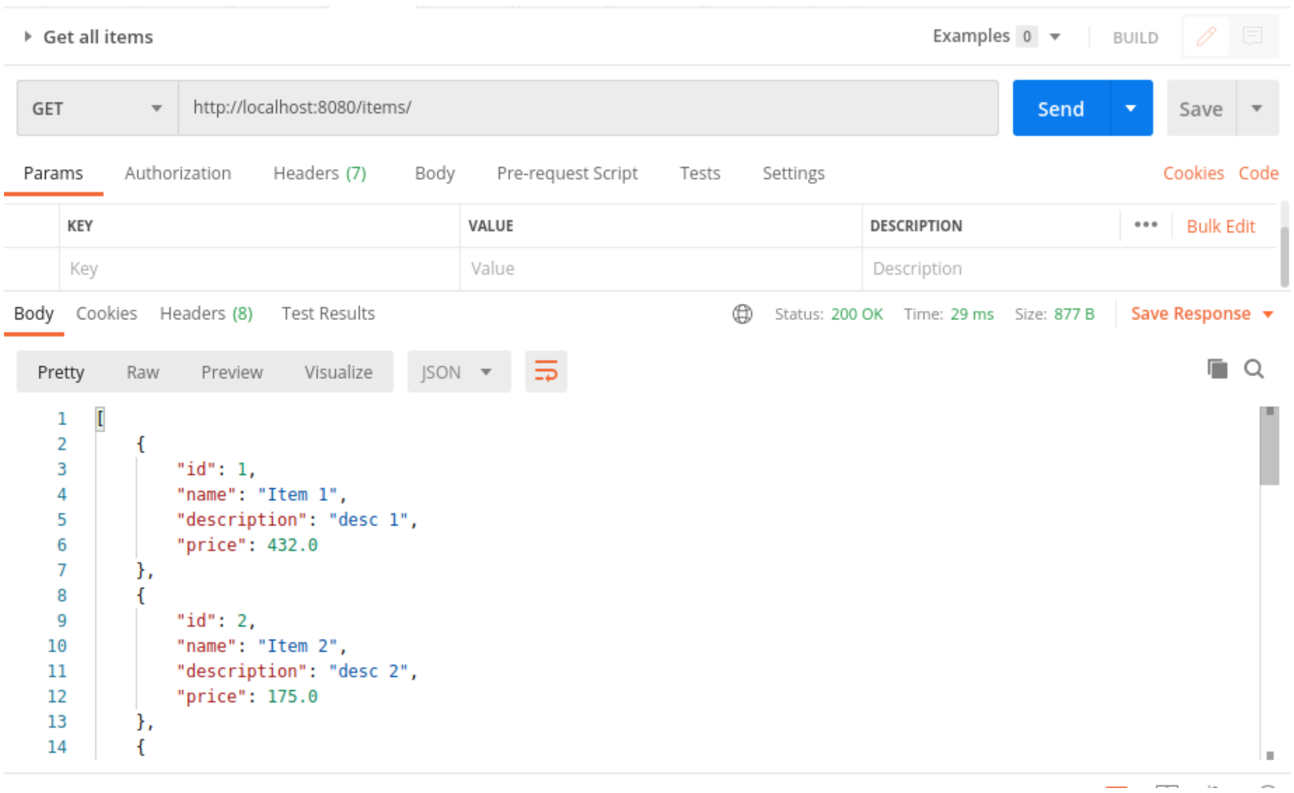
Items per page: 5 1 – 1 of 1

5. Delete the item:

Red button in the list indicates the delete functionality.

Postman Results of Backend APIs:

1. To get all items:

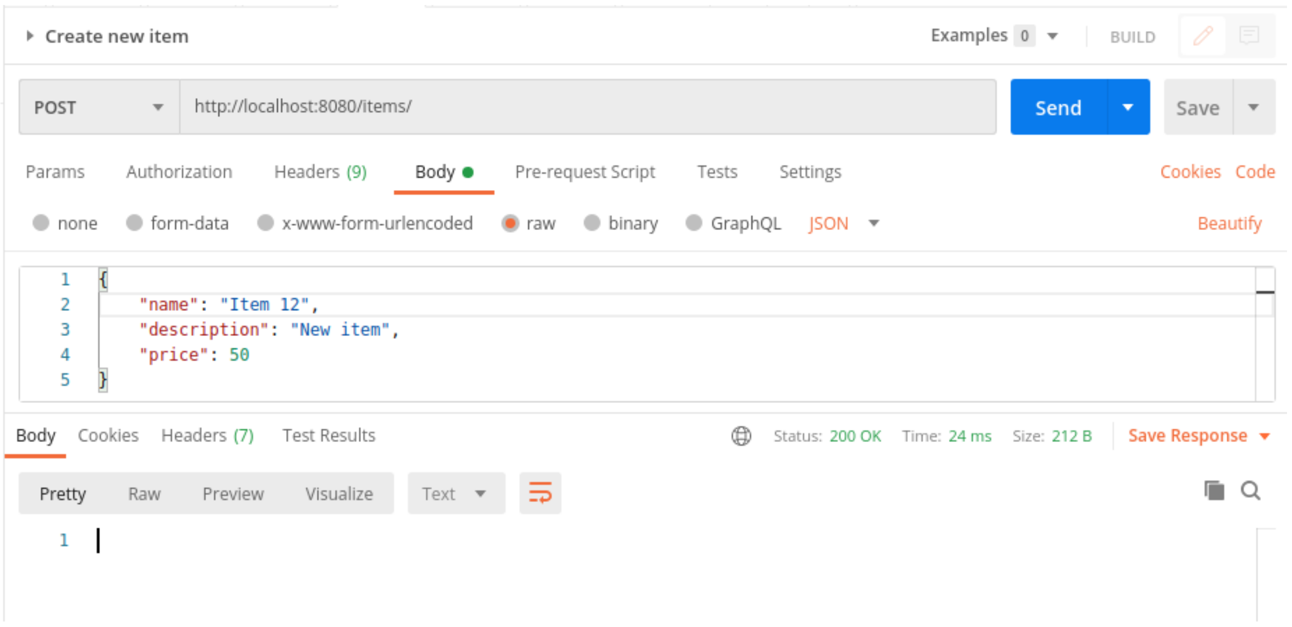


GET http://localhost:8080/items/

Status: 200 OK Time: 29 ms Size: 877 B

```
[
    {
        "id": 1,
        "name": "Item 1",
        "description": "desc 1",
        "price": 432.0
    },
    {
        "id": 2,
        "name": "Item 2",
        "description": "desc 2",
        "price": 175.0
    },
    {
```

2. To create new item:



POST http://localhost:8080/items/

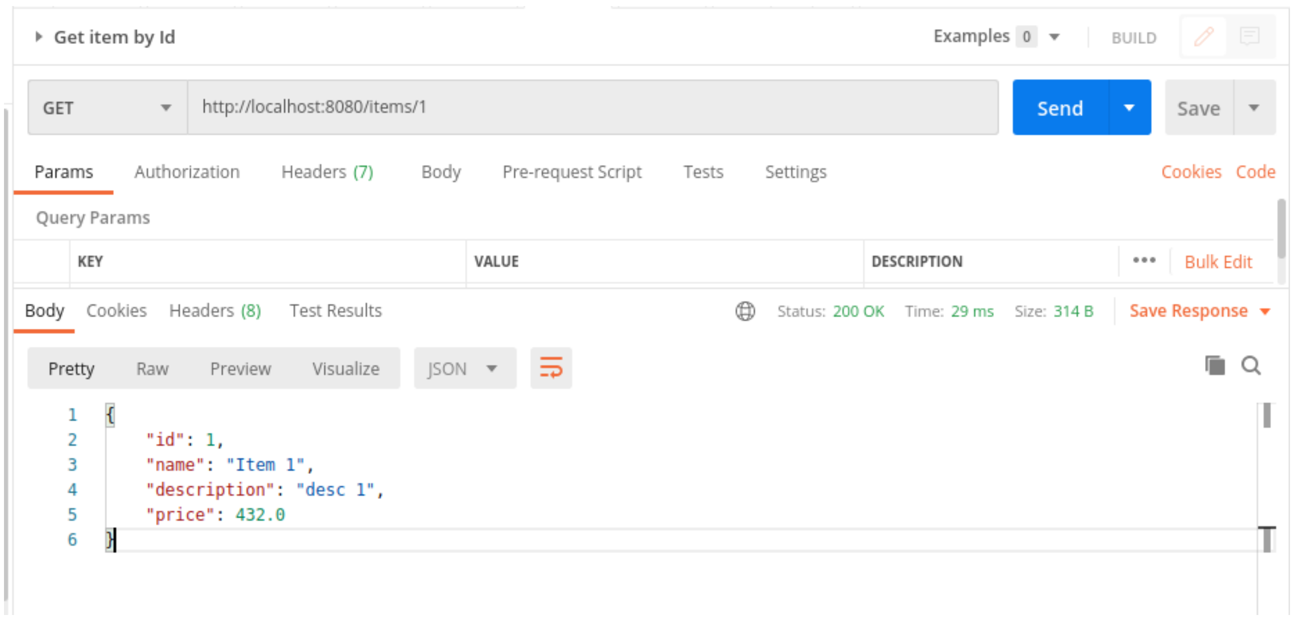
```
{
    "name": "Item 12",
    "description": "New item",
    "price": 50
}
```

Status: 200 OK Time: 24 ms Size: 212 B

```

```

3. To get item details by Id:

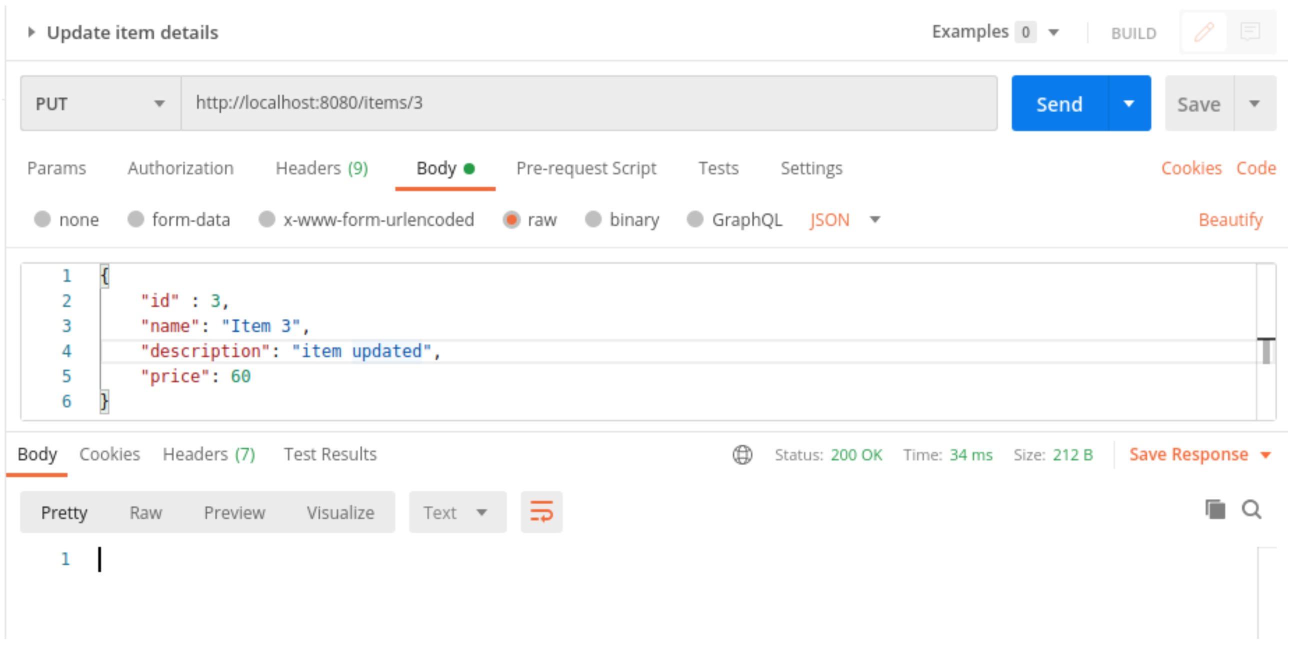


GET http://localhost:8080/items/1

Status: 200 OK Time: 29 ms Size: 314 B

```
{
    "id": 1,
    "name": "Item 1",
    "description": "desc 1",
    "price": 432.0
}
```

4. To update the item details:

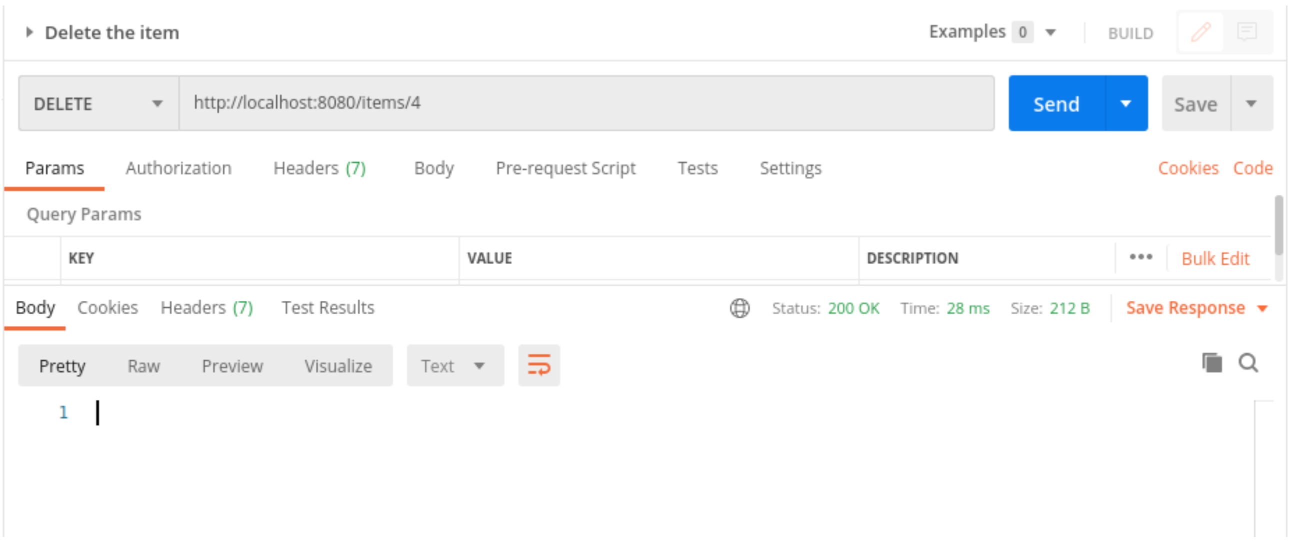


PUT http://localhost:8080/items/3

```
{
    "id" : 3,
    "name": "Item 3",
    "description": "item updated",
    "price": 60
}
```

Status: 200 OK Time: 34 ms Size: 212 B

5. To detele the item:



http://localhost:8080/items/4

Status: 200 OK Time: 28 ms Size: 212 B

Angular Project files description:

1. **Item details component:**
    - It shows Items list which includes edit and delete functionality.
    - Add Item button to add new item.
    - Search box to search item on item name.

2. **Add/Edit component:**
    - To add and edit the item.

3. **Item data service:**
    - To call all mock backend APIs

4. **Loader component:**
    - To show loader on some functionalities backend API process.

5. **Pipe component:**
    - To filter data on user inputed item name.

Java Spring boot application files:

1. **Inventory Application java file:**
    - It is a main java file from where spring boot application gets started.

2. **Item Controller java file:**
    - It includes controller class that accepts http requests from frontend.

3. **Item repository java file:**
    - It includes interface that extends the Crud repository interface for crud operations.

4. **Item java file:**
    - It is a POJO class that includes attributes and getter setter methods.

5. **Item service java file:**
    - It is class that implements the crud operations.

6. **Application properties file:**
    - This is the database related configuration file.

7. **Schema postgres sql file:**
    - It is the sql file that creates the table in database when we run java application.

8. **Data postgres sql file:**
    - It is the sql file that inserts the data into table when we run java application.

Postgres database schema:

Database name: inventorydb

```
TABLE item(
id serial PRIMARY KEY,
name VARCHAR(255),
price float,
description VARCHAR(255)
)
```