# Spring Boot Project Documentation

Created by: **Anand R Diwatai**

### Date : 29/12/2024

## Introduction:

This document outlines the steps to set up and run a Spring Boot project that performs CRUD operations on Categories and Products, including testing with Postman. The project also demonstrates server-side pagination and linking of products with categories.

## Steps to Run the Code:

1. Step 1: Create a Spring Boot project using Spring Initializr or your preferred IDE.

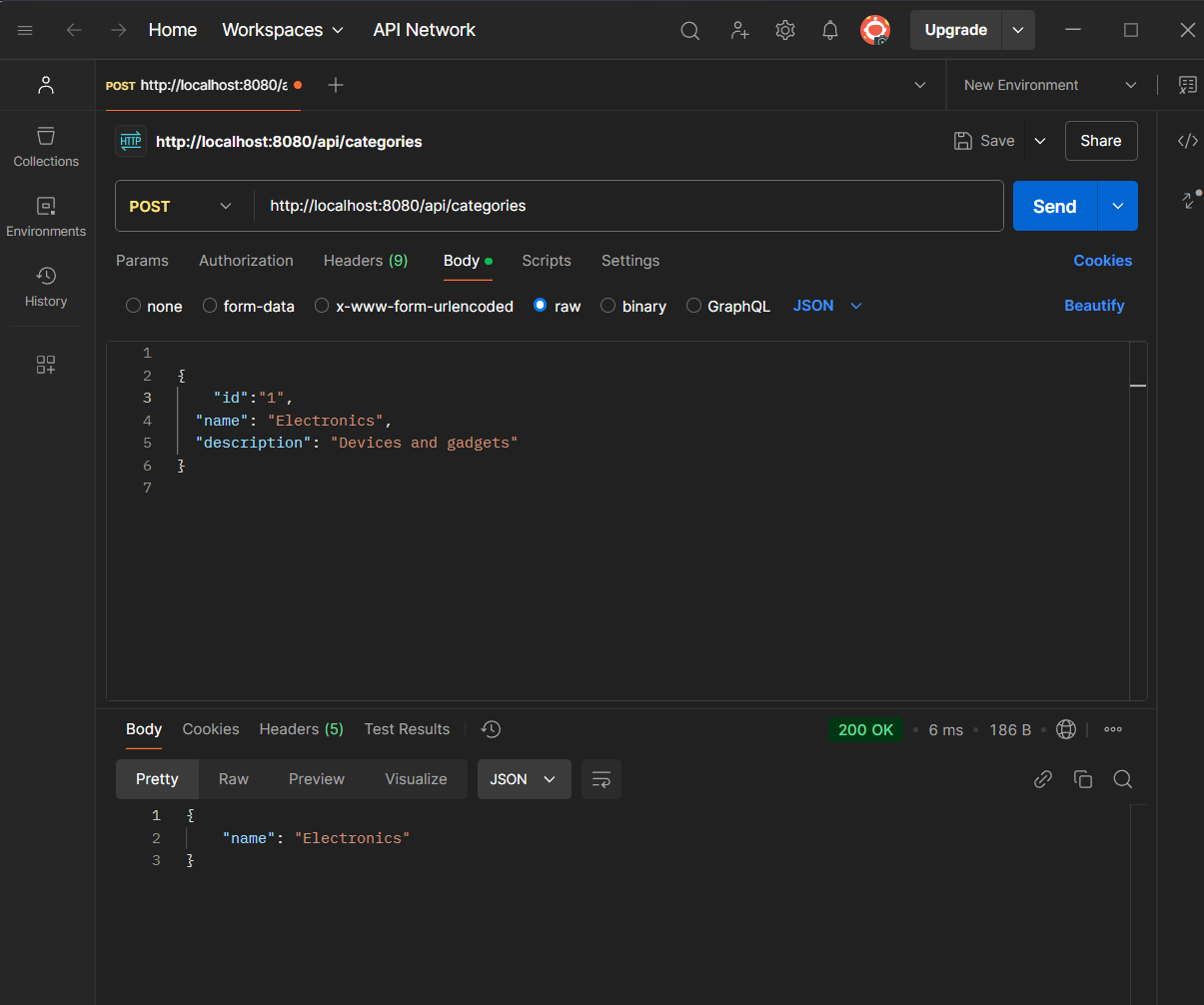
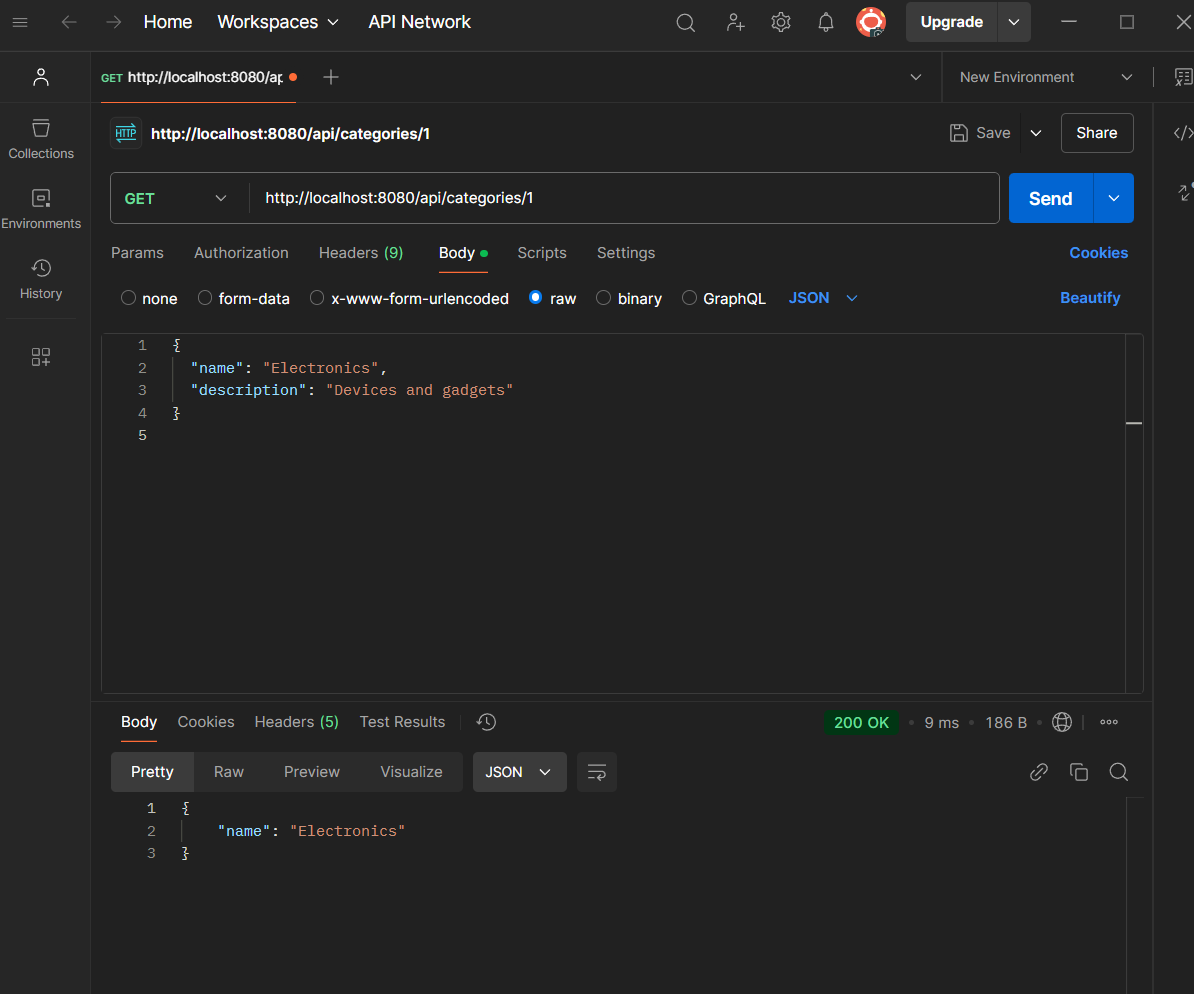
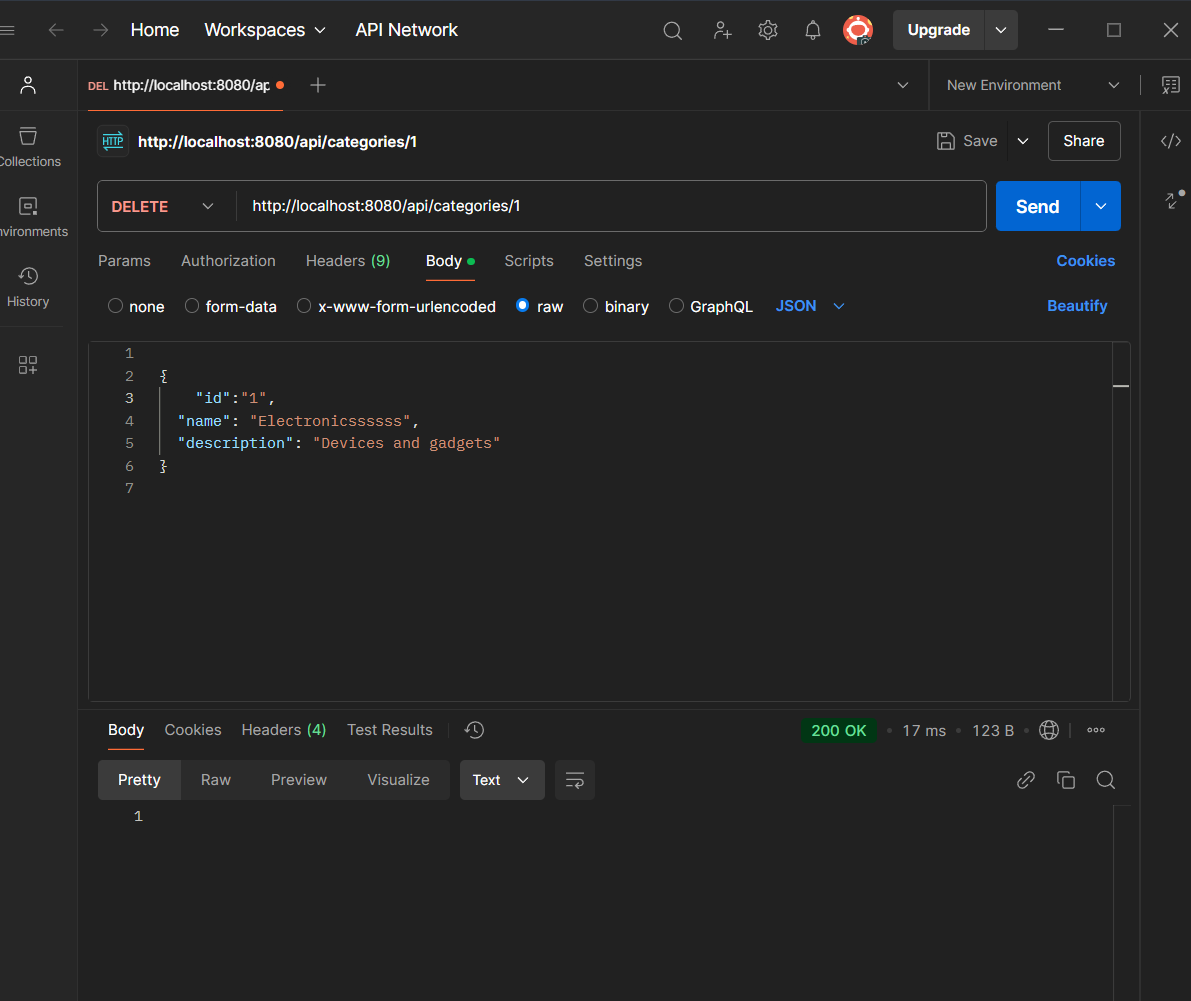
* **Dependencies**:
  + Spring Web
  + Spring Data JPA
  + MySQL Driver
  + Spring Boot DevTools
  + Hibernate

1. Step 2: Set up application.properties to connect with MySQL or H2 for testing(I have Used H2).
2. Step 3:Create Category and Product Entities - Define Category and Product entities.
3. Step 4: Create Repository Interfaces - Set up repository interfaces for Category and Product.
4. Step 5:Create Controller for CRUD Operations - Create a controller for performing Create, Read, Update, Delete operations for both categories and products.
5. Step 6: Implement Server-Side Pagination - Implement pagination for fetching categories and products.
6. Step 7:Fetch Product with Category Details - Ensure products are linked to their categories and can be fetched together
7. Step 8: Testing with Postman.

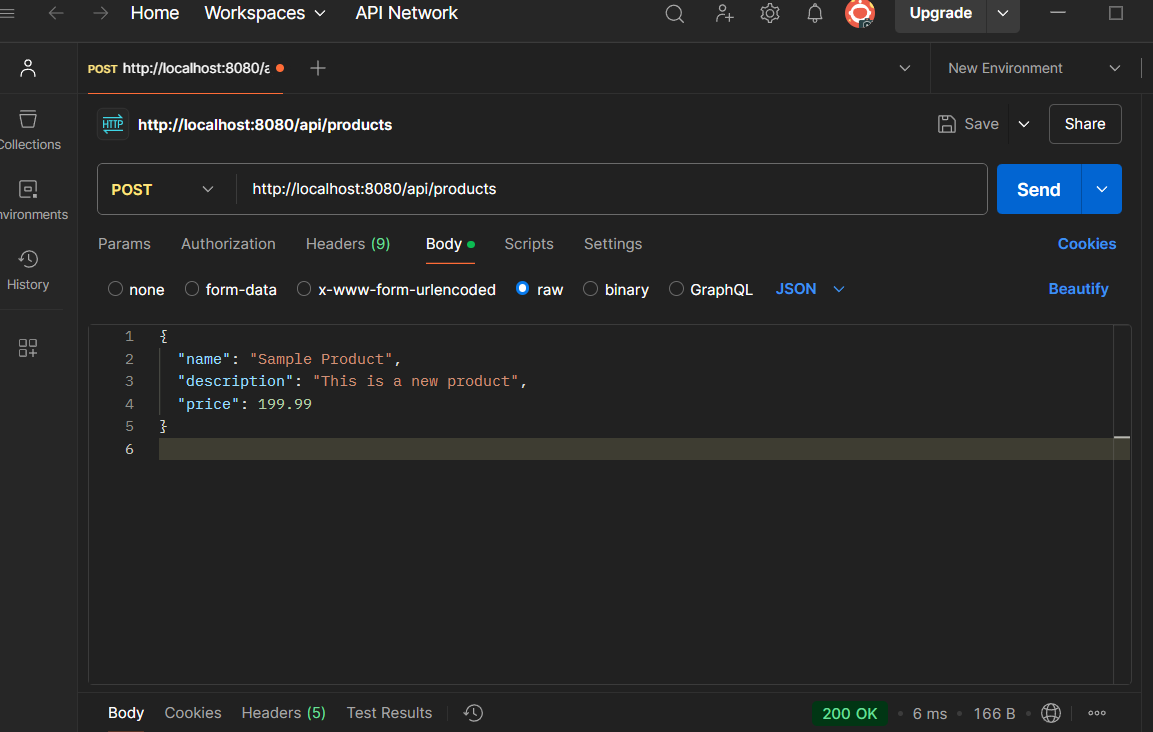
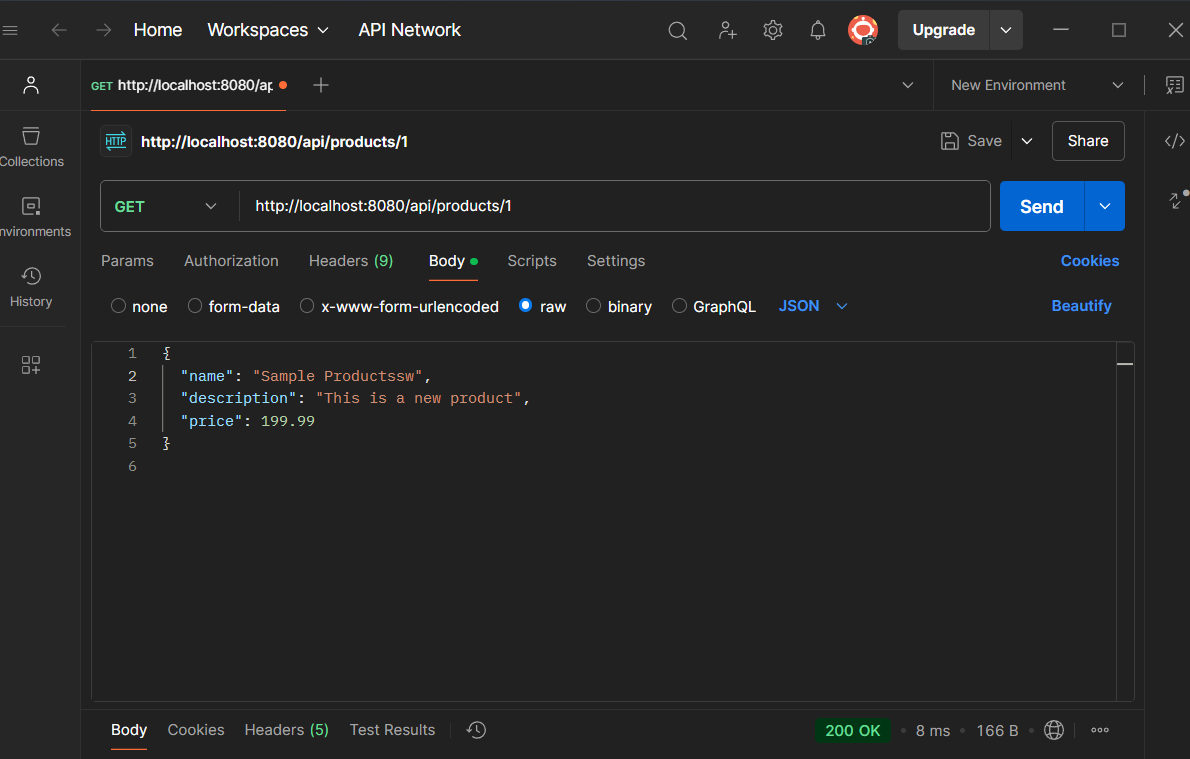
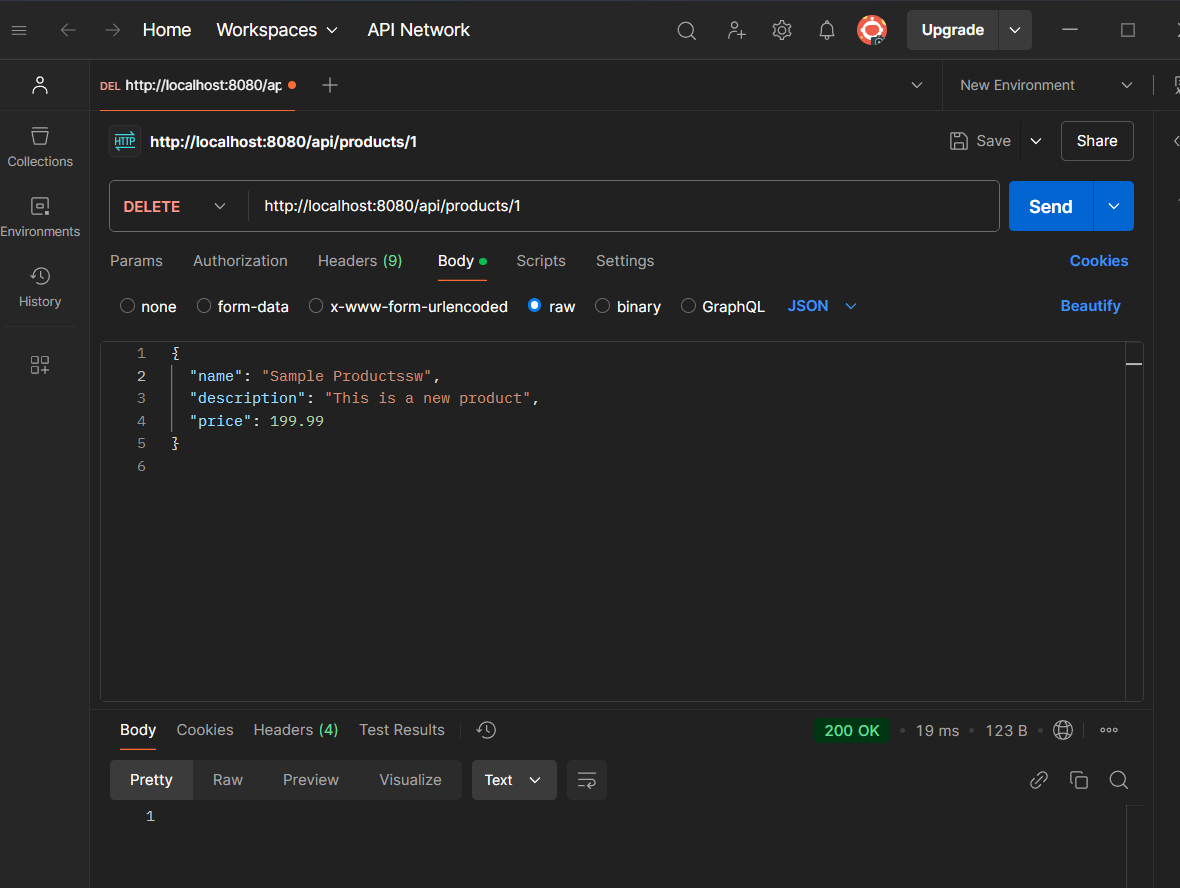
## Postman Testing:

Use Postman to test the following endpoints for Categories and Products.

Categories Endpoints:

* URL: http://localhost:8080/api/categories/1
* Method: POST
* 
* URL: http://localhost:8080/api/categories/1
* Method: GET
* 
* URL: http://localhost:8080/api/categories/1
* Method: DELETE
* 

Products Endpoints:

* URL: http://localhost:8080/api/products
* Method: POST
* 
* URL: http://localhost:8080/api/products/1
* Method: GET
* 
* URL: http://localhost:8080/api/products/1
* Method: DELETE
* 

## Machine Test:

To run a machine test for an API, use JUnit with Spring Boot Test. Here's the outline:

1. 1. Set up the test: Use @SpringBootTest for integration tests.
2. 2. Create a test class: Write a test class using MockMvc to simulate API calls.
3. 3. Write test logic: Use mockMvc.perform() to test endpoints and validate responses.
4. 4. Test for Categories and Products: Write separate tests for both Categories and Products.