

Created by: Rajmohan Sakkarai

Employee Id: 863706

Assignment: Microservice Assignment1

## 1. Get All Products:

### i)PostMan Result:

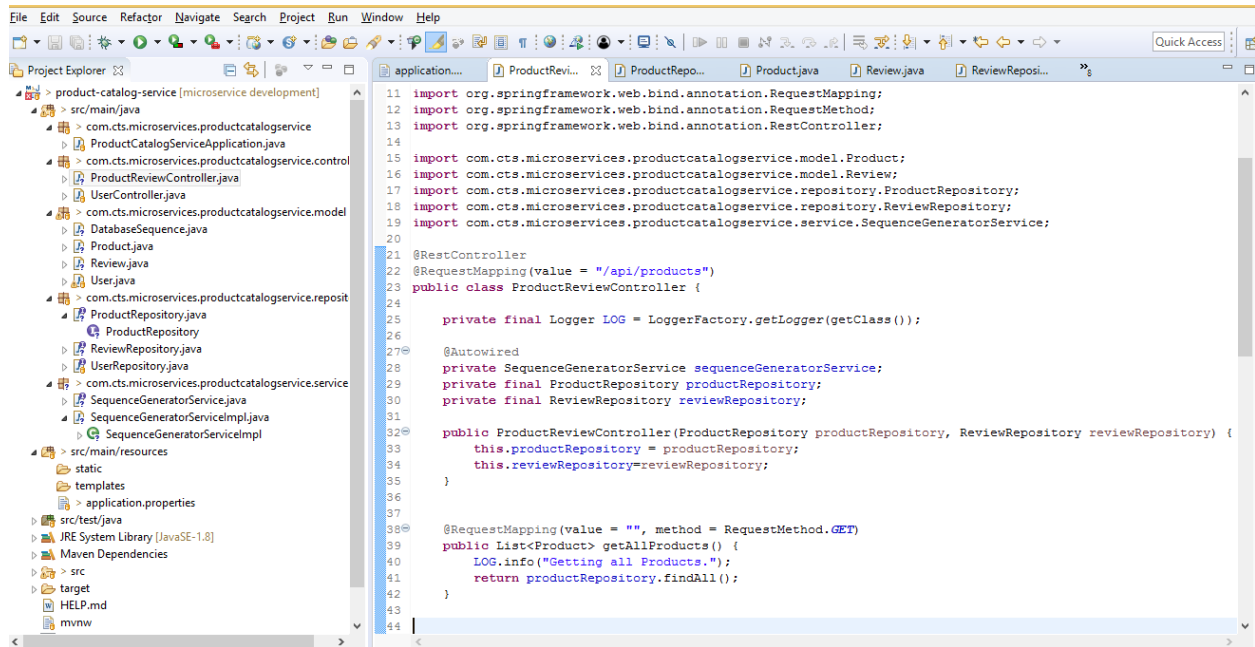
The screenshot shows the Postman interface with a GET request to `localhost:8081/api/products`. The response status is 200 OK, and the body contains a JSON array of two products:

```
[{"id": 1, "name": "moto_e", "price": 5000.0, "description": "Moto E Smart phone..! "}, {"id": 2, "name": "samsung_m31", "price": 10000.0, "description": "Samsung M31 Smart phone..! "}]
```

### ii)Mongo DB Result:

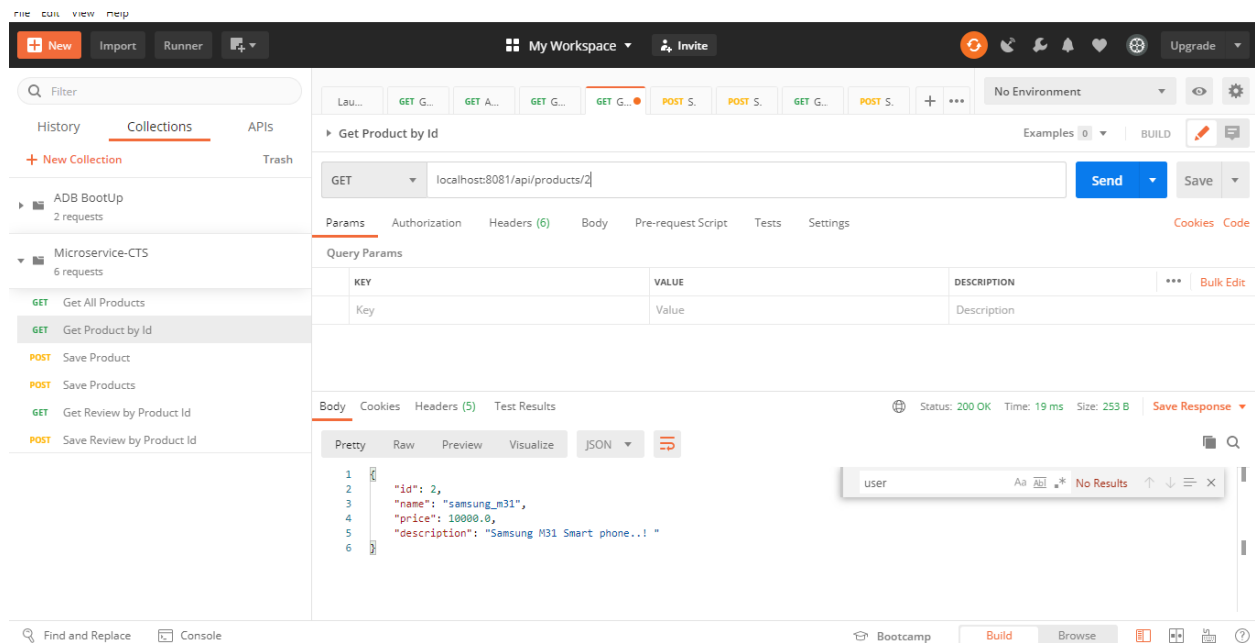
```
C:\Windows\system32\cmd.exe - mongo
> db.products.find(<>).pretty()
{
  "_id" : 1,
  "name" : "moto_e",
  "price" : 5000,
  "description" : "Moto E Smart phone..! ",
  "_class" : "con.cts.microservices.productcatalogservice.model.Product"
}
{
  "_id" : 2,
  "name" : "samsung_m31",
  "price" : 10000,
  "description" : "Samsung M31 Smart phone..! ",
  "_class" : "con.cts.microservices.productcatalogservice.model.Product"
}
{
  "_id" : 3,
  "name" : "Nokia",
  "price" : 15000,
  "description" : "Nokia Smart phone..! ",
  "_class" : "con.cts.microservices.productcatalogservice.model.Product"
}
```

### lii) Code:



## 2. Get One Product:

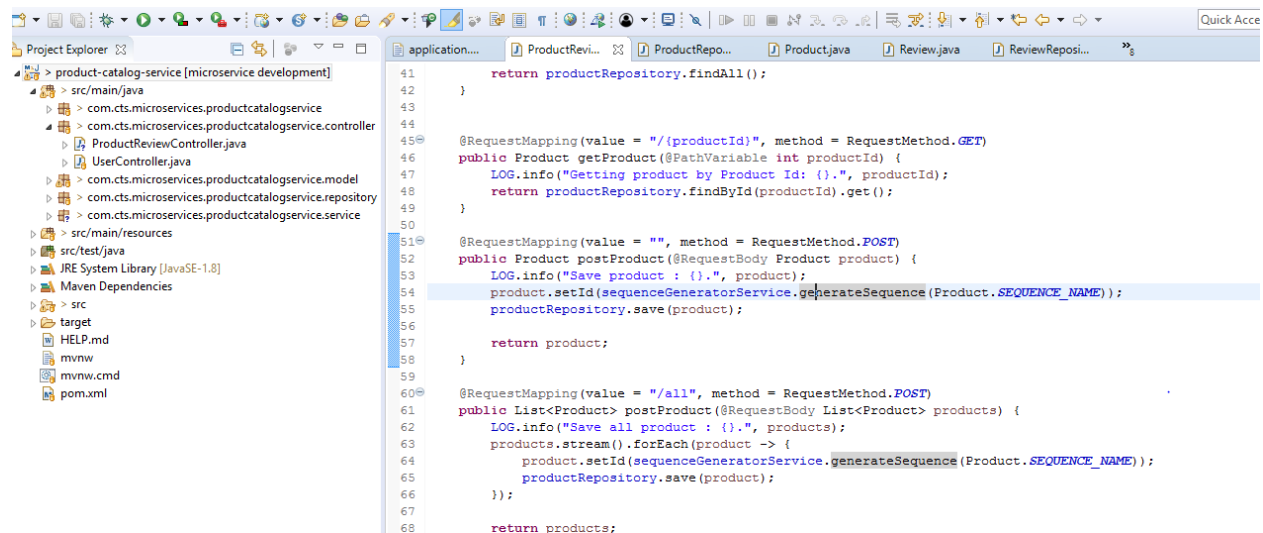
### i) PostMan



### ii) MongoDB:

```
CA. C:\Windows\system32\cmd.exe - mongo
> db.products.find(<{_id:2}>).pretty()
{
  "_id" : 2,
  "name" : "samsung_m31",
  "price" : 10000,
  "description" : "Samsung M31 Smart phone..!",
  "_class" : "com.cts.microservices.productcatalogservice.model.Product"
}
```

### iii) Code:



The screenshot shows an IDE with a project explorer on the left and a code editor on the right. The project explorer shows a project named 'product-catalog-service [microservice development]' with the following structure:

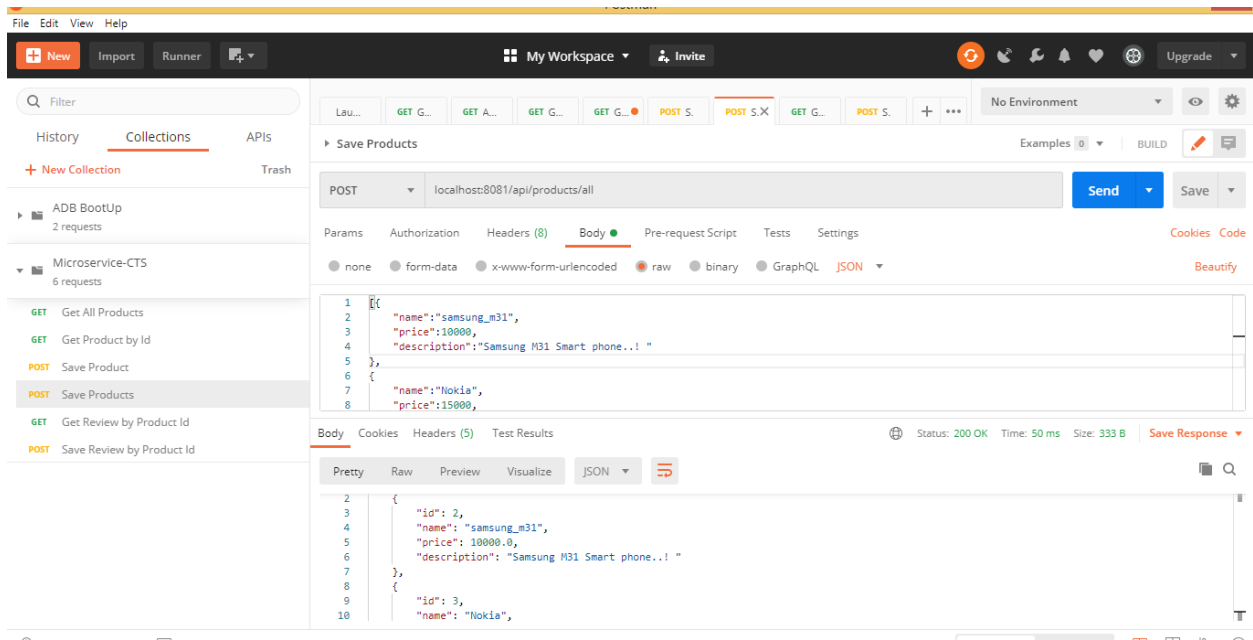
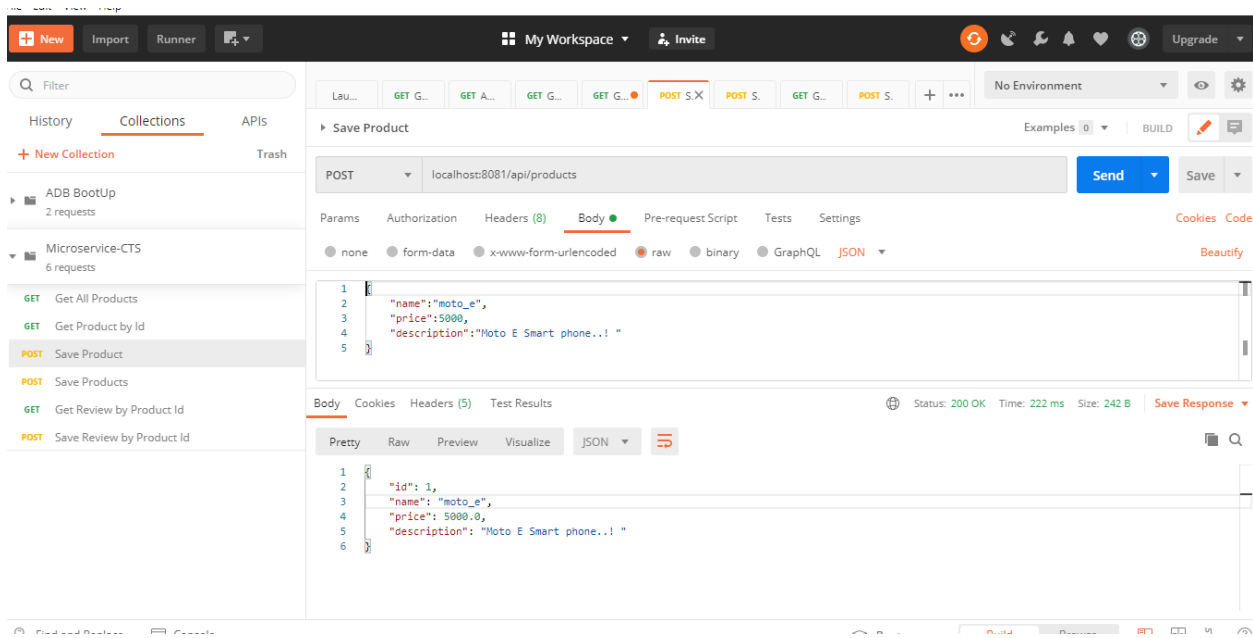
- src/main/java
  - com.cts.microservices.productcatalogservice
    - com.cts.microservices.productcatalogservice.controller
      - ProductReviewController.java
      - UserController.java
    - com.cts.microservices.productcatalogservice.model
    - com.cts.microservices.productcatalogservice.repository
    - com.cts.microservices.productcatalogservice.service
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml

The code editor shows the following Java code:

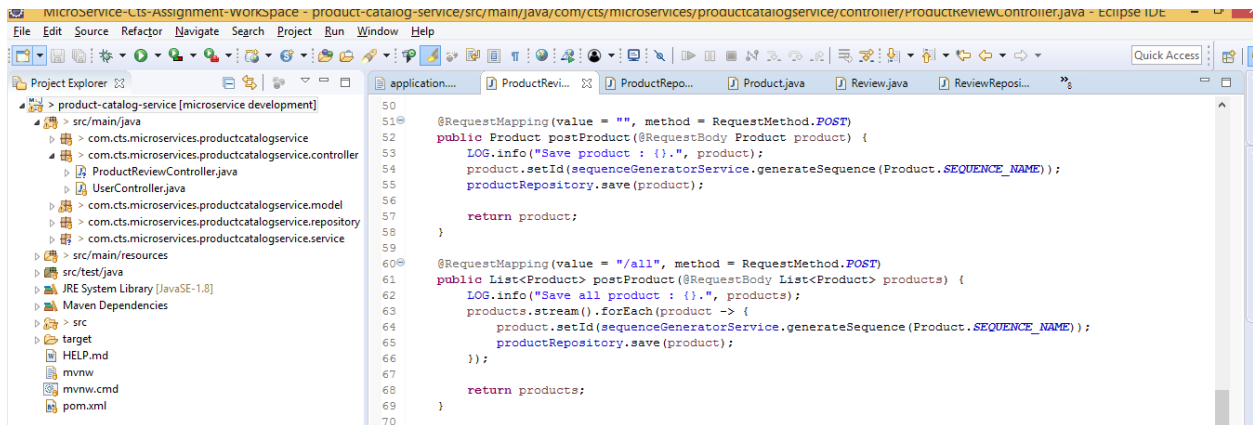
```
41     return productRepository.findAll();
42 }
43
44
45 @RequestMapping(value = "/{productId}", method = RequestMethod.GET)
46 public Product getProduct(@PathVariable int productId) {
47     LOG.info("Getting product by Product Id: {}. ", productId);
48     return productRepository.findById(productId).get();
49 }
50
51 @RequestMapping(value = "", method = RequestMethod.POST)
52 public Product postProduct(@RequestBody Product product) {
53     LOG.info("Save product : {}. ", product);
54     product.setId(sequenceGeneratorService.generateSequence(Product.SEQUENCE_NAME));
55     productRepository.save(product);
56
57     return product;
58 }
59
60 @RequestMapping(value = "/all", method = RequestMethod.POST)
61 public List<Product> postProduct(@RequestBody List<Product> products) {
62     LOG.info("Save all product : {}. ", products);
63     products.stream().forEach(product -> {
64         product.setId(sequenceGeneratorService.generateSequence(Product.SEQUENCE_NAME));
65         productRepository.save(product);
66     });
67
68     return products;
```

### 3.Save one Product and Save Multiple Products:

#### i)PostMan:

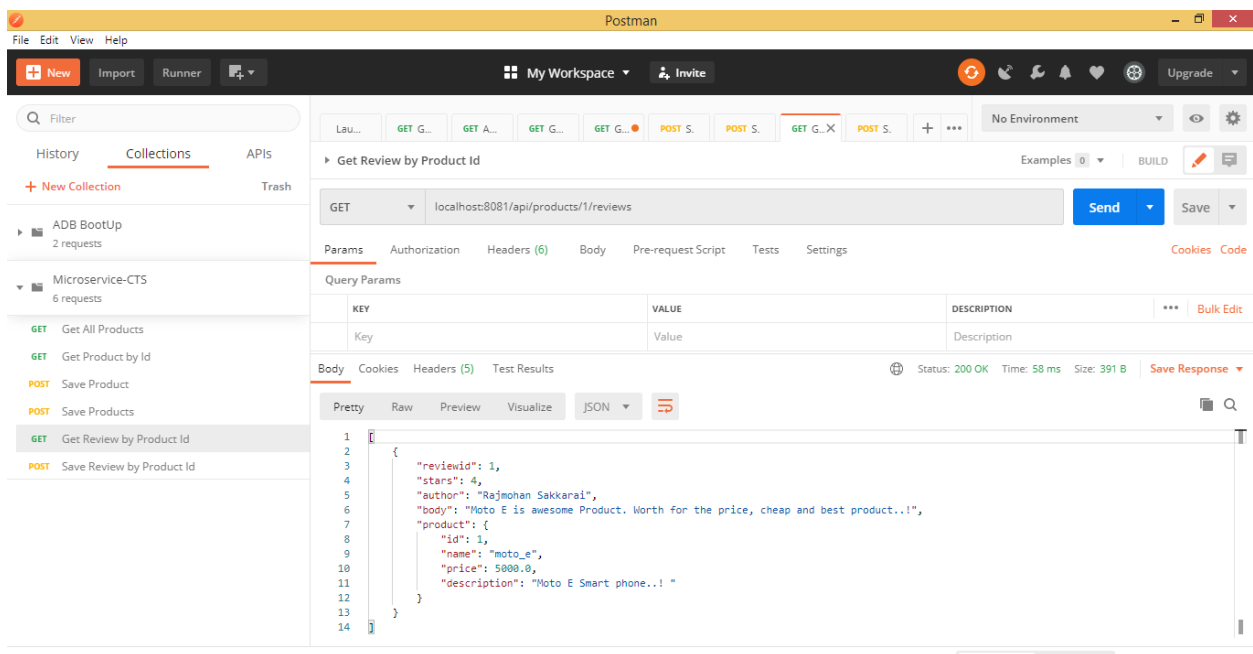


ii) Code:

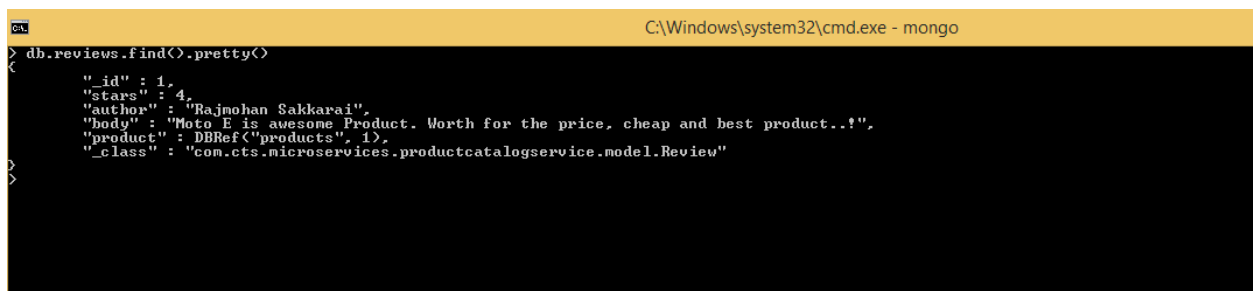


#### 4)Get Review:

##### i)PostMan:



##### ii)Mongo DB:



### iii) Code:

```
70  
71  
72 @RequestMapping(value = "/{productId}/reviews", method = RequestMethod.GET)  
73 public Collection<Review> getReview(@PathVariable int productId) {  
74     LOG.info("Getting product by Product Id: {}.", productId);  
75     return reviewRepository.findByProjectId(productId);  
76 }
```

## 5. Save Review:

### i) PostMan:

The screenshot shows the Postman interface with a workspace named 'My Workspace'. A collection named 'Microservice-CTS' is selected, containing a request named 'Save Review by Product Id'. The request is a POST to 'localhost:8081/api/products/1/reviews'. The body is a JSON object: 

```
{  "stars": 4,  "author": "Rajmohan Sakkarai",  "body": "Moto E is awesome Product. Worth for the price, cheap and best product..!"}
```

. The response is a 200 OK status with a JSON body: 

```
{  "reviewid": 1,  "stars": 4,  "author": "Rajmohan Sakkarai",  "body": "Moto E is awesome Product. Worth for the price, cheap and best product..!",  "product": {    "id": 1,    "name": "moto_e",    "price": 5000.0,    "description": "Moto E Smart phone..!"  },  "id": 1}
```

### ii) MongoDB:

```
> db.reviews.find(<<'product.$id':1>>).pretty()
{
  "_id" : 1,
  "stars" : 4,
  "author" : "Rajmohan Sakkarai",
  "body" : "Moto E is awesome Product. Worth for the price, cheap and best product..!",
  "product" : DBRef("products", 1),
  "_class" : "com.cts.microservices.productcatalogservice.model.Review"
}
```

### iii) Code:

