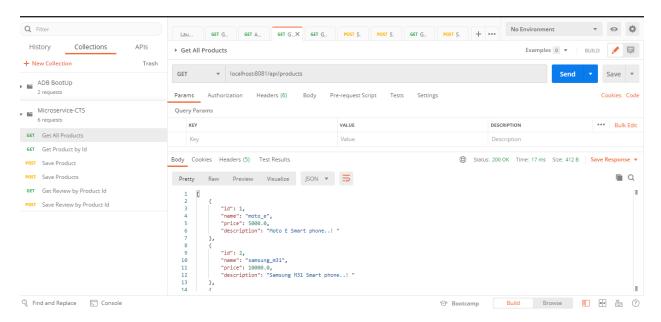
Created by: Rajmohan Sakkarai

Employee Id: 863706

**Assignment: Microservice Assignment1** 

# 1. Get All Products:

i)PostMan Result:



## ii)Mongo DB Result:

```
db.products.find().pretty()

".id": 1,
"name": "moto_e",
"ywice": 5999;
"description": "Moto E Smart phone..!",
"_class": "com.cts.microservices.productcatalogservice.model.Product"

".id": 2,
"name": "sansung_m31",
"price": 19090,
"description": "Sansung M31 Smart phone..!",
"class": "com.cts.microservices.productcatalogservice.model.Product"

".id": 3,
"name": "Nokia",
"price": 15000,
"description": "Nokia Smart phone..!",
"class": "com.cts.microservices.productcatalogservice.model.Product"

".id": 3,
"name": "Nokia",
"price": 15000,
"description": "Nokia Smart phone..!",
"class": "com.cts.microservices.productcatalogservice.model.Product"
```

lii) Code:

```
<u>File Edit Source Refactor Navigate Search Project Run Window Help</u>
Quick Access
Project Explorer 💢
                               🖹 😘 🏻 🔛 🗖 🗎 📵 application.... 🚺 ProductRevi... 🐰 🚺 ProductRepo... 🗓 Productjava 🗓 Review.java 📝 ReviewReposi...
                                                             inport org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RequestMethod;
△ Mi-J > product-catalog-service [microservice development]
   ⊿ 📇 > src/main/java

▲ # > com.cts.microservices.productcatalogservice

                                                             13 import org.springframework.web.bind.annotation.RestController;
         > 🔝 ProductCatalogServiceApplication.java
                                                                 import com.cts.microservices.productcatalogservice.model.Product;
      # => com.cts.microservices.productcatalogservice.control
         import com.cts.microservices.productcatalogservice.model.Review;
                                                                 import com.cts.microservices.productcatalogservice.repository.ProductRepository;
                                                                 import com.cts.microservices.productcatalogservice.repository.ReviewRepository, import com.cts.microservices.productcatalogservice.sepository.ReviewRepository; import com.cts.microservices.productcatalogservice.service.SequenceGeneratorService;
      ▲ 🚜 > com.cts.microservices.productcatalogservice.model
        @RestController
          Review.java
                                                                 @RequestMapping(value = "/api/products")
public class ProductReviewController {
         User.java

    com.cts.microservices.productcatalogservice.reposit
    ProductRepository.java
                                                                       private final Logger LOG = LoggerFactory.getLogger(getClass());
            ProductRepository
          ReviewRepository.java
                                                                      private SequenceGeneratorService sequenceGeneratorService;
private final ProductRepository productRepository;

☑ UserRepository.iava

        > com.cts.microservices.productcatalogservice.service

R SequenceGeneratorService.java
                                                                      private final ReviewRepository reviewRepository;
        public ProductReviewController(ProductRepository productRepository, ReviewRepository reviewRepository) {

■ ② Src/main/resources

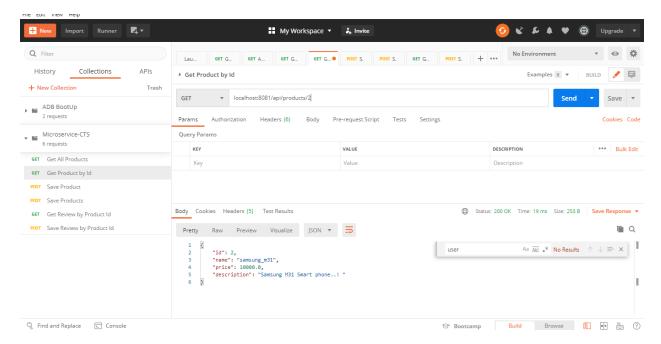
                                                                           this.productRepository = productRepository;
this.reviewRepository=reviewRepository;
                                                             34
35
36
       templates
        🔒 > application.properties
                                                             37
38⊖
39

→ m src/test/java

                                                                      @RequestMapping(value = "", method = RequestMethod.GET)
public List<Product> getAllProducts() {
    LOG.info("Getting all Products.");
   ि > src
                                                                           return productRepository.findAll();
   W HELP.md
                                                             43
44
```

## 2.Get One Product:

# i)PostMan



# ii) MongoDb:

```
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:

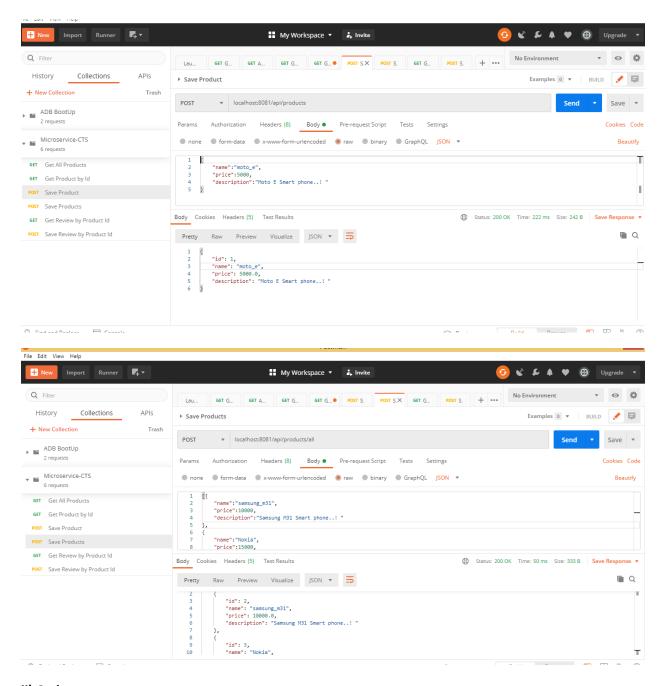
```
🖹 😘 🔝 🗢 🗆 🗋 👔 application.... 🖟 ProductRevi... 💢 📝 ProductRepo... 🗍 ProductJava 📝 Review.java 📝 ReviewReposi... 🐾
Project Explorer ≅
return productRepository.findAll();
                                                      41

    Connects microservices product catalog service.controller
    Product Review Controller, java

                                                              @RequestMapping(value = "/(productId)", method = RequestMethod.GET)
public Product getProduct(@PathVariable int productId) (
    LOG.info("Getting product by Product Id: ().", productId);
    return productRepository.findById(productId).get();
       46
    > tom.cts.microservices.productcatalogservice.model
    → ♣ > com.cts.microservices.productcatalogservice.repository
                                                       49
    ▶ # > com.cts.microservices.productcatalogservice.service
  > src/main/resources
                                                               @RequestMapping(value = "", method = RequestMethod.POST)
                                                      b @ src/test/iava
  ⊳ 🕞 > src
  HELP.md
   mvnw
mvnw.cmd
                                                      59
60<sup>⊕</sup>
61
62
63
64
65
66
67
                                                               @RequestMapping (value = "/all", method = RequestMethod.POST)
public List<Product> postProduct(@RequestBody List<Product> products) {
    pom.xml
                                                                   LOG.info("Save all product: {}:", products);
products.stream().forEach(product -> {
    products.setId(sequenceGeneratorService.generateSequence(Product.SEQUENCE_NAME));
    product.setId(sequenceGeneratorService.generateSequence(Product.SEQUENCE_NAME));
                                                                 product.setId(sequenceGeneratorS
productRepository.save(product);
});
                                                                   return products;
```

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

i)PostMan:



ii) Code:

```
File Edit Source Refactor Navigate Search Project Run Window Help
[ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T | [ ] T
                                                                         🗎 😘 🤝 🔻 🗖 📗 application.... 🗓 ProductRevi... 🐰 🗓 ProductRepo... 🗋 Productjava 🚺 Review.java 🚺 ReviewReposi...
Project Explorer 🛭

▲ Note: A product-catalog-service [microservice development]

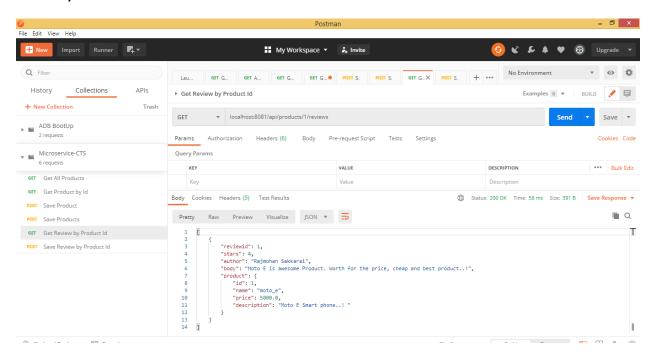
                                                                                                                                                                    @RequestMapping(value = "", method = RequestMethod.POST)
         public Product postFroduct(@RequestBody Product product) {
   LOG.info("Save product : {}\.", product);
   product.setId(sequenceGeneratorService.generateSequence(Product.SEQUENCE_NAME));

    → Com.cts.microservices.productcatalogservice.cont
    → ProductReviewController.java

              productRepository.save(product);
                                                                                                                                                                              return product;
             > com.cts.microservices.productcatalogservice.repository
                                                                                                                                              58
59
60<sup>©</sup>
                   > com.cts.microservices.productcatalogservice.service
         > cm > src/main/resources
                                                                                                                                                                   @RequestMapping(value = "/all", method = RequestMethod.POST)
         public ListFroduct> postProduct(@RequestBody List<Product> products) {
   LOG.info("Save all product: {}.", products);
   products.stream().forEach(product -> {
                                                                                                                                             61
62
63
64
65
66
67
68
69
          JRE System Library [JavaSE-1.8]
         ⊳ 🛵 > src
⊳ 🗁 target
                                                                                                                                                                                         product.setId(sequenceGeneratorService.generateSequence(Product.SEOUENCE NAME));
                                                                                                                                                                                           productRepository.save(product);
            W HELP.md
            mvnw
mvnw.cmd
                                                                                                                                                                              return products;
             pom.xml
```

### 4)Get Review:

# i)PostMan:



# ii)Mongo DB:

```
C:\Windows\system32\cmd.exe - mongo

> db.reviews.find().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"

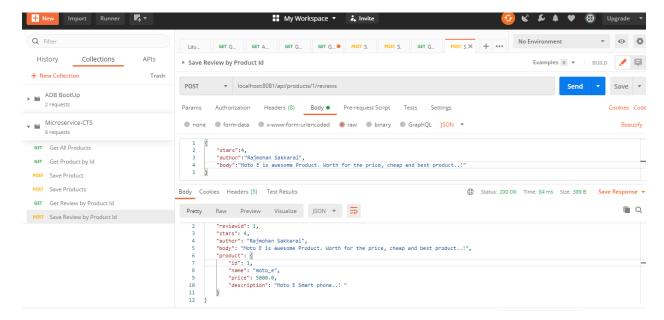
>
```

### iii) Code:

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

#### 5.Save Review:

# i)PostMan:



# ii)MongoDB:

iii)Code:

```
🧶 MicroService-Cts-Assignment-WorkSpace – product-catalog-service/src/main/java/com/cts/microservices/productcatalogservice/controller/ProductReviewController.java - Eclipse IDE 👚 🗖 📙
<u>File Edit Source Refactor Navigate Search Project Run Window Help</u>
Quick Access
Project Explorer 🖂
                                  😑 😘 🦻 🔻 🗀 📔 application.... 🗓 ProductRevi... 🗯 📝 ProductRepo... 🕽 ProductJava 🗓 Review.java 🗓 ReviewReposi... 🐾
                                                                                                                                                                                                                                                 - -

■ Sproduct-catalog-service [microservice development]

                                                                           56
57
     return product;
        @RequestMapping(value = "/all", method = RequestMethod.POST)
public List<Product> postProduct(@RequestBody List<Product> products) {
    LOG.info("Save all product : ().", products);
    products.stream().forEach(product -> {
       product.setId(sequenceGeneratorService.generateSequence(Product.SEQUENCE_NAME));
productRepository.save(product);
});

→ ∰ > com.cts.microservices.productcatalogservice.service

     > # > src/main/resources
     ▷ # src/test/java

    ▶ May JRE System Library [JavaSE-1.8]
    ▶ Mayen Dependencies

                                                                                            return products;
     > See > src
> See target
W HELP.md
                                                                                      8RequestMapping(value = "/{productId}/reviews", method = RequestMethod.GET)
public CollectioncReview> getReview(8PathVariable int productId) {
    LOG.info("Getting product by Product Id: {}.", productId);
    return reviewRepository.findByProjectId(productId);
        mvnw
       pom.xml
                                                                                      @RequestMapping (value = "/(productId)/reviews", method = RequestMethod.FOST)
public Review postReview(@PathVariable int productId, @RequestBody Review review) {
   LOG.info("Save Review : {}.", review);
   Product product= productRepository.findById(productId).get();
   review.setFroduct(product);
   review.setFreviewid(sequenceGeneratorService.generateSequence(Review.SEQUENCE_NAME));
   review.setReviewid(sequenceGeneratorService.generateSequence(Review.SEQUENCE_NAME));
                                                                                             reviewRepository.save(review);
```