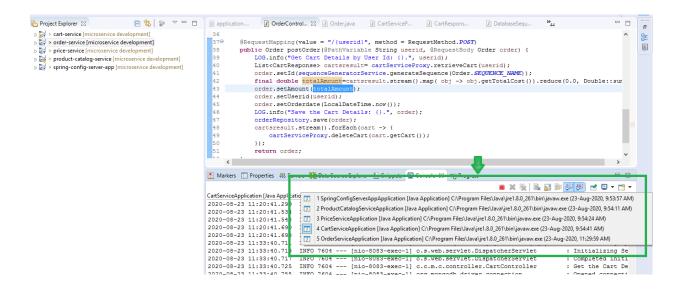
Created by: Rajmohan Sakkarai

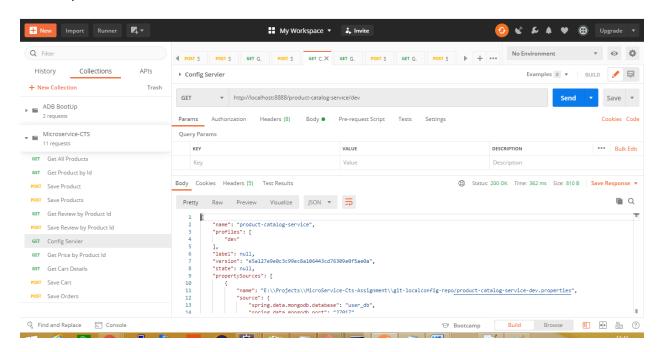
Employee Id: 863706

**Assignment: Microservice Assignment2 and Assignment3** 

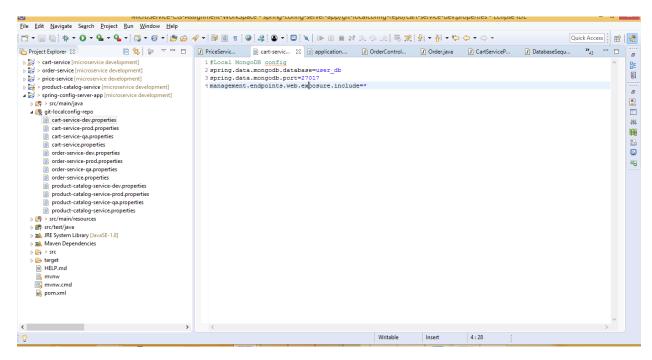


# Config Server:

### i)PostMan



## ii)Code Configuration files from GitHub with Environment based Configuration



## iii) GitHub

```
Rajmohan Sakkaraijūkjamban Miloūdā /e/Projects/MicroService-Cts-Assignment/git-1
Siglis tabus
On branch master
Untracked files:
(use "git and drilow," to include in what will be committed)
(use "git and drilow," to include in what will be committed)

(use "git and drilow," to include in what will be committed)

(use "git and drilow," to include in what will be committed)

(use "git and drilow," to commit but untracked files present (use "git add" to track)

Kajmahan Sakkaraijūkajamban Miloūdā /e/Projects/MicroService-Cts-Assignment/git-1

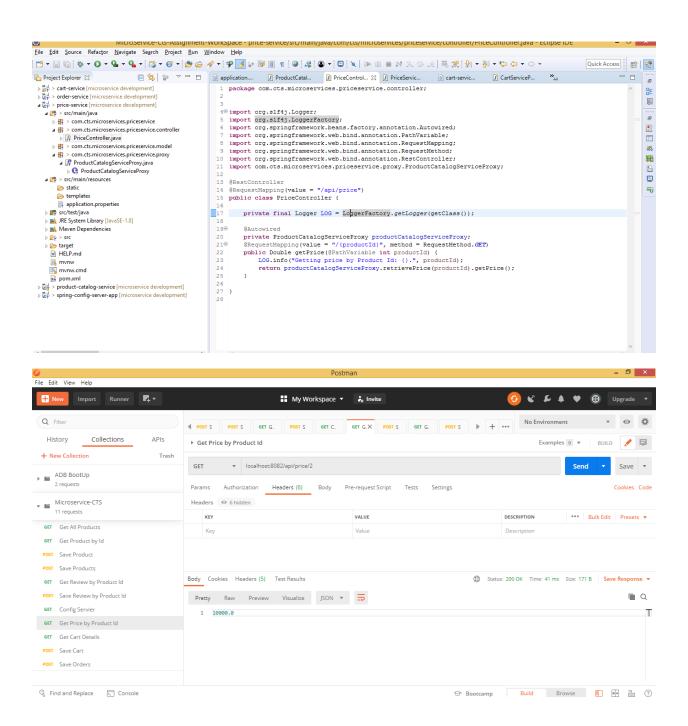
socialcomis-group (master)

Siglia di "

socialcomis-group (master)

socialcomis-group (m
```

Price Service with Feighn Client:



```
MicroService-Cts-Assignment-WorkSpace - price-service/src/main/java/com/cts/microservices/priceservice/proxy/ProductCatalogServiceProxy,java - Eclipse IDE
<u>File Edit Source Refactor Navigate Search Project Run Window Help</u>
Quick Access : P
Project Explorer 🛭
                                                                 🖹 🥞 💆 🗖 📋 application.... 🚺 ProductCatal... 🗯 📝 PriceControl...
 > Moderate | Service | Discrete 
                                                                                                                             1 package com.cts.microservices.priceservice.proxy;
                                                                                                                                                                                                                                                                                                                                                                                                                             8
                                                                                                                                                                                                                                                                                                                                                                                                                              △ Mil > price-service [microservice development]
                                                                                                                              3@ import org.springframework.cloud.openfeign.FeignClient;
                                                                                                                                   import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
      ▲ 🚜 > src/main/java
                                                                                                                                                                                                                                                                                                                                                                                                                              ē

→ # > com.cts.microservices.priceservice

                                                                                                                                                                                                                                                                                                                                                                                                                             <u>•</u>
           import com.cts.microservices.priceservice.model.Product;
                                                                                                                                                                                                                                                                                                                                                                                                                             @FeignClient(name="product-catalog-service".url="http://localhost:8081")
            > com.cts.microservices.priceservice.model
                                                                                                                                                                                                                                                                                                                                                                                                                              뭐
                                                                                                                                   public interface ProductCatalogServiceProxy {

    > com.cts.microservices.priceservice.proxy
    ProductCatalogServiceProxy.java

                                                                                                                                                                                                                                                                                                                                                                                                                             H
                                                                                                                                              @GetMapping("/api/products/{id}")
                                                                                                                                                                                                                                                                                                                                                                                                                             ▲ (# > src/main/resources
                                                                                                                                            public Product retrievePrice(@PathVariable("id") int id);
                                                                                                                                                                                                                                                                                                                                                                                                                             -
              templates
                                                                                                                                                                                                                                                                                                                                                                                                                              =6
                application.properties
      Mayen Dependencies
       HELP.md
           mvnw
          mvnw.cmd
          pom.xml
           > product-catalog-service [microservice development]
 > spring-config-server-app [microservice development]
```

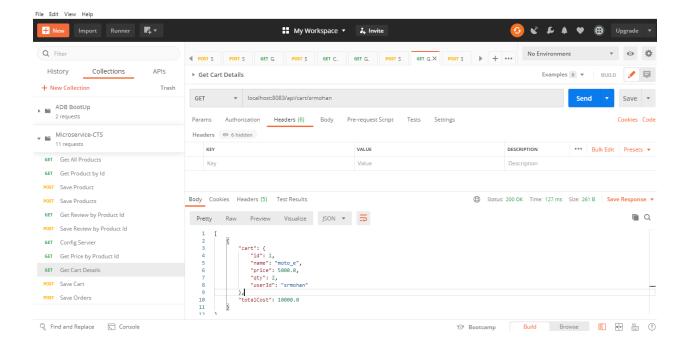
## Cart Service with fieghn Client:

```
cart-service/src/main/java/com/cts/microservices/cartservice/controller/CartController.java - Eclipse IL
<u>File Edit Source Refactor Navigate Search Project Run Window Help</u>
Quick Access
                                    😑 😘 👺 🔻 🗀 📋 application.... 🗓 ProductCatal... 📗 PriceControl... 🔯 CartControll... 🕱 📄 cart-servic... 🗓 CartServiceP...
Project Explorer 🖂
 △ Mid > cart-service [microservice development]
                                                                     38
                                                                                                                                                                                                                                   ^ -
                                                                                @RequestManning(value = "/{userid}", method = RequestMethod.GFT)
    ⊿ 🚜 > src/main/java
                                                                                                                                                                                                                                          gsequestmapping(value = "/(useria/", method = kequestmathod.deff)
public ListCartResponse> postCart(@PathVariable String userid) {
    LOG.info("Get the Cart Details by User Id: {}.", userid);
    Collection<Cart> carts= cartRepository.findByUserId(userid);
                                                                      40
41
42

→ B > com.cts.microservices.cartservice.controller

         ☐ CartController.java

→ 😭 CartController
                                                                                                                                                                                                                                         43
                                                                                      List<CartResponse> cartsRes=new ArrayList<>();
                                                                                      Cartxs.forEach(obj -> {
    Cartxseponse cartxses=new Cartxseponse();
    double totalCost=obj.getPrice()*obj.getQty();
                                                                      44
45
46
47
48
      > = > com.cts.microservices.cartservice.model
       W
                                                                                           cartRes.setCart(obj);
        # > com.cts.microservices.cartservice.service
                                                                                           cartRes.setTotalCost(totalCost);
    49
50
51
52
53
54
556
57
58
59
60
61
62
63
64
65
66
67
68
                                                                                            cartsRes.add(cartRes);
                                                                                                                                                                                                                                          =6
      Maven Dependencies
                                                                                     return cartsRes;
    @RequestMapping(value = "/(productId)/(userid)", method = RequestMethod.FOST)
public CartResponse postCart(@PathVariable int productId, @PathVariable String userid, @RequestBody Cart of LOG.info("Get the Price by Product Id: {\}.", productId);
double price= priceServiceProxy.retrievePrice(productId);
cart.setId(sequenceGeneratorService.generateSequence(Cart.SEQUENCE_NAME));
cart.setId(sequenceGeneratorService.generateSequence(Cart.SEQUENCE_NAME));
      W HELP.md
      mvnw.cmd
 cart.setPrice(price);
  > price-service [microservice development]
        product-catalog-service [microservice dev
                                                                                      cart.setUserId(userid);
                                                                                     Coart.Secostriquetrial, LOG.info("Save the Cart Details: ().", cart); cartRepository.save(cart); CartResponse response=new CartResponse();
 > spring-config-server-app [microservice development]
                                                                                      response.setCart(cart);
                                                                                      response.setTotalCost(cart.getPrice()*cart.getQty());
                                                                                      return response;
                                                                                    equestMapping(value = "", method = RequestMethod.DELETE)
                                                                                public ResponseEntity<Object> deleteCart(@RequestBody Cart cart) {
                                                                                                                                         Writable
                                                                                                                                                          Smart Insert 71:6
```



### Order Service with fieght client:

```
MicroService-Cts-Assignment-WorkSpace - order-service/src/main/java/com/cts/microservices/orderservice/controller/OrderController.java - Eclipse IDE
<u>File Edit Source Refactor Navigate Search Project Run Window Help</u>
🍖 Project Explorer 🕄 🕒 😘 🗽 💆 🗖 📗 application.... 📗 ProductCatal... 📗 PriceControl... 🚇 CartControll... 📦 cart-servic... 🖟 OrderControl... 🖫 2
 22 @RequestMapping(value = "/api/orders")
                                                            23 public class OrderController {
                                                                                                                                                                                                        private final Logger LOG = LoggerFactory.getLogger(getClass());
      # # > com.cts.microservices.orderservice.controller
                                                                                                                                                                                                        <u>...</u>
        ☐ OrderController.java

→ ☐ OrderController
                                                           28
29
30<sup>©</sup>
31
32
                                                                    private CartServiceProxy cartServiceProxy;
                                                                                                                                                                                                        ▶ # > com.cts.microservices.orderservice.model
                                                                    @Autowired private OrderRepository;
      W
                                                                                                                                                                                                        33⊖
                                                                                                                                                                                                        ▣
                                                                     private SequenceGeneratorService sequenceGeneratorService;
     =0
                                                                    @RequestMapping (value = "/(userid)", method = RequestMethod.POST)
public Order postOrder(@PathVariable String userid, @RequestBody Order order) {
   LOG.info("Get Cart Details by User Id: {}.", userid);
   ListCartResponse> cartsresult= cartServiceProxy.retrieveCart(userid);
   order.setId(sequenceGeneratorService.generateSequence(Order.SEQUENCE_NAME));
         SequenceGeneratorServiceImpl.java
    > src/main/resources
    Maven Dependencies
   > Eq. > src
> E target
W HELP.md
                                                                         final double totalAmount=cartsresult.stream().map( obj -> obj.getTotalCost()).reduce(0.0, Double::sur order.setAmount(totalAmount); order.setUserid(userid); order.setUserid(userid);
     mvnw
mvnw.cmd
                                                                         LOG.info("Save the Cart Details: {}.", order);
                                                                         carteresult.stream() forEach(cart -> {
    cartServiceProxy.deleteCart(cart.getCart());
});
                                                                         orderRepository.save(order);
     pom.xml
  | price-service [microservice development] | price-service [microservice development] | product-catalog-service [microservice development]
  > spring-config-server-app [microservice development]
                                                                         return order:
: O 👺 com.cts.microservices.orderservice.controller.OrderController.java - order-service/src/main/java
```

