

# Assignment on Microservice Communications

## Create 2 Microservice applications as follows

1. product-service (assume this service is running on port number 8081)
2. customer-service (assume this service is running on port number 8082)

local  
↑ customer-service [devtools] [:8082]  
↑ product-service [devtools] [:8081]

## Create following classes in product-service microservice application

1. Create Product class

```
package com.cts.product.model;

public class Product {
    private int productId;
    private String productName;
    private double price;
    private String description;

    //default constructor
    //parameterized constructor
    // getters and setters
    // generate toString method

    public Product() {}

    public Product(int productId, String productName, double price, String description) {
        this.productId = productId;
        this.productName = productName;
        this.price = price;
        this.description = description;
    }
}
```

2. Create ProductData class to maintain Products information

```
package com.cts.product.dao.util;
import java.util.ArrayList;
import java.util.List;
import com.cts.product.model.Product;

public class ProductData {
    private static List<Product> products;
    static {
        products=new ArrayList<>();
        products.add(new Product(10, "Laptop", 35000, "Dell I3"));
        products.add(new Product(11, "Laptop", 250000, "Mac Book Pro"));
        products.add(new Product(12, "Laptop", 75000, "Lenovo I5"));
        products.add(new Product(13, "Mobile", 135000, "Samsung S22 Ultra"));
        products.add(new Product(14, "Mobile", 156000, "Iphone 14 Pro Max 256GB"));
        products.add(new Product(15, "Mobile", 45000, "Oneplus 10 Pro"));
        products.add(new Product(16, "Tablet", 120000, "Ipad 5th Gen"));
        products.add(new Product(17, "Tablet", 59000, "Samsung A7 Tablet"));
        products.add(new Product(18, "Keyboard", 12000, "Apple Magic Keyboard"));
        products.add(new Product(19, "Keyboard", 3500, "Dell Wireless keyboard"));
    }
    public static List<Product> findAll(){
        return products;
    }
    public static void addProduct(Product product) {
        products.add(product);
    }
    public static Product findById(int id) {
        return products.stream()
            .filter(product->product.getProductId()==id)
            .findAny()
            .orElse(null);
    }
}
```

```

public static List<Product> findByProductName(String productName){
    return products.stream()
        .filter(product->product.getProductName().equals(productName))
        .toList();
}

public static void removeProduct(int productId) {
    products.removeIf(product->product.getId()==productId);
}

public static Product updateProduct(Product newProduct) {
    products.replaceAll(product->product.getId()==newProduct.getId()?newProduct:product);
    System.out.println(products);
    if(products.contains(newProduct))
        return newProduct;
    else
        return null;
}
}

```

3. Now Create ProductRestServiceResource class in product-service application and implement following REST end points to perform Product operation from above class

**@RestController**

**@CrossOrigin**

**@RequestMapping("/api/products")**

**public class ProductServiceResource { ... }**

- a. Find All products GET mapping
- b. FindById GET Mapping
- c. FindByName GET Mapping
- d. AddNewProduct POST mapping with request body
- e. UpdateProduct PUT mapping
- f. DeleteProduct DELETE Mapping

4. Make sure user able to use all above services using rest end point

5. Now create customer-service microservice on port **8082**

6. Create **CustomerServiceResource** class in customer-service application and define same REST endpoint methods defined in above product-service application in **ProductServiceResource** class.

**@RestController**

**@CrossOrigin**

**@RequestMapping("/api/customer")**

**public class CustomerServiceResource { ... }**

7. When user sends rest end point request to <http://localhost:8082/api/customers/> , then customer-service should able to communicate with product-service on port number 8081 and load all products and display them as JSON array. And You should provide all services to access product-service resources from customer-service application.
8. You may use **RestTemplate** or **WebClient** for Microservice communication

