

แบบฝึกปฏิบัติ ครั้งที่ 8

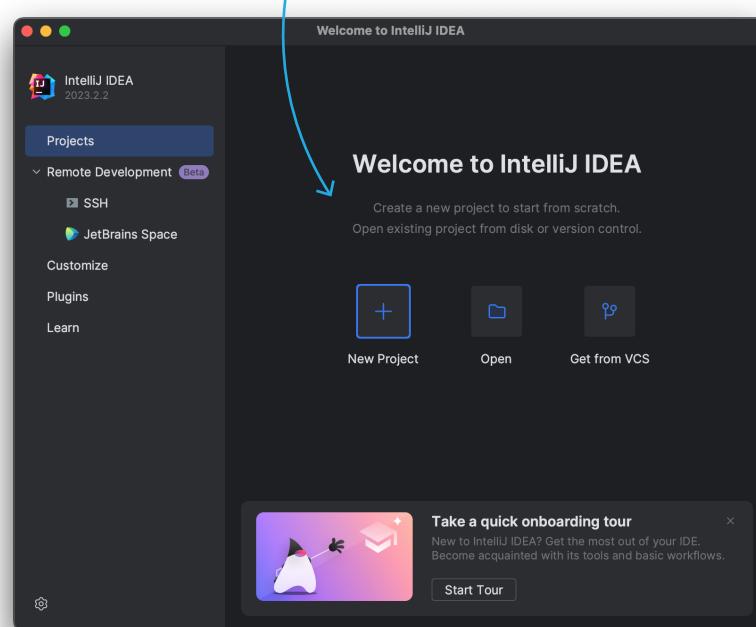
โดย ผศ.ดร.ธราภิเชษฐ์ อิจิจูรูญโรจน์ และ อาจารย์สัญชัย น้อยจันทร์ วิชา 06016325 การโปรแกรมเชิงบริการ

เรื่อง API Gateway และ Service Discovery

- วัตถุประสงค์
- เพื่อฝึกฝนการสร้าง API Gateway ด้วย Spring Cloud API Gateway
 - เพื่อฝึกฝนการทำ service discovery ด้วย Eureka Discovery

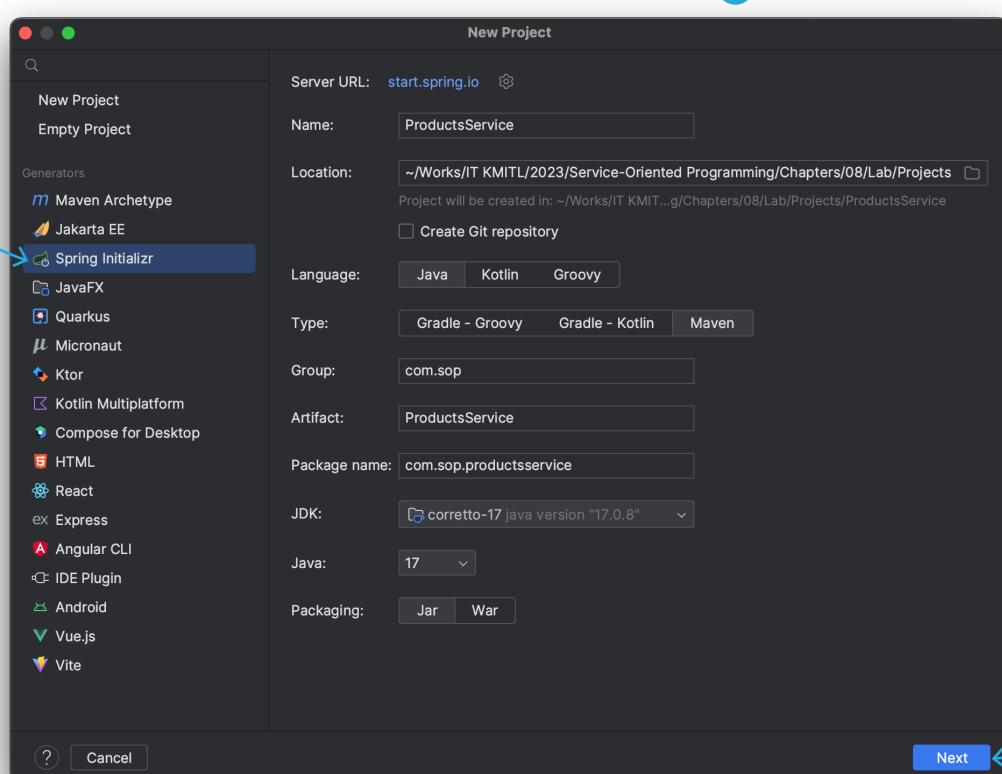
ข้อที่ 1 : สร้าง Products Service

1 เปิดโปรแกรม IntelliJ IDEA ขึ้นมา



2 กดปุ่ม New Project เพื่อสร้างโปรเจกต์ใหม่

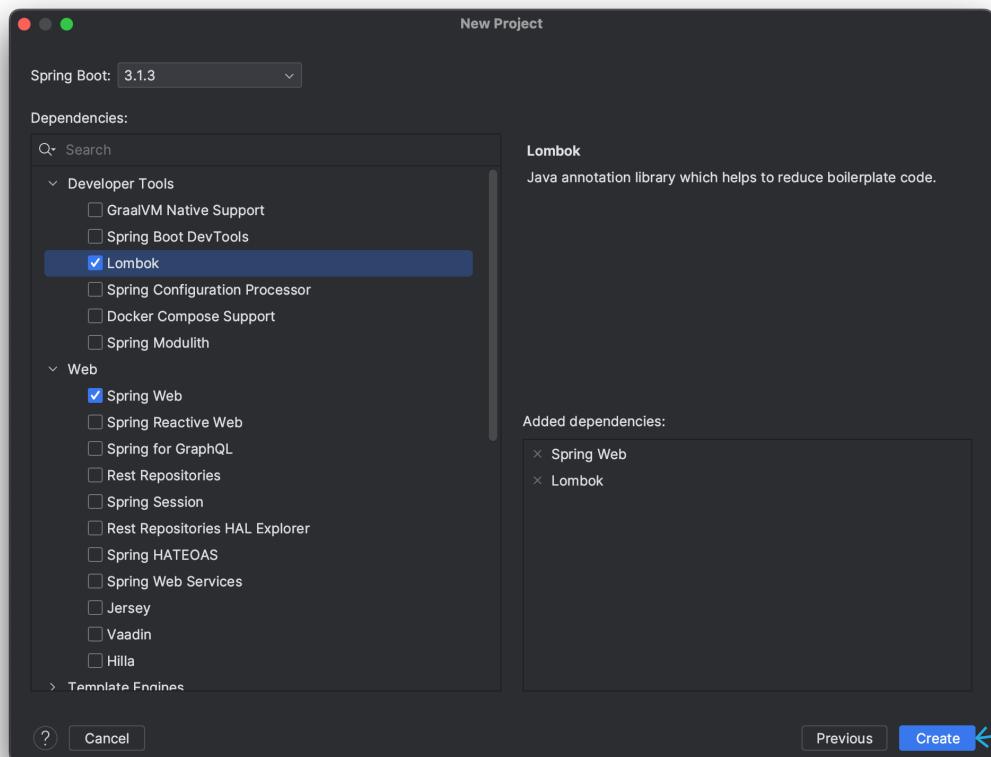
3 เลือก Spring Initializr



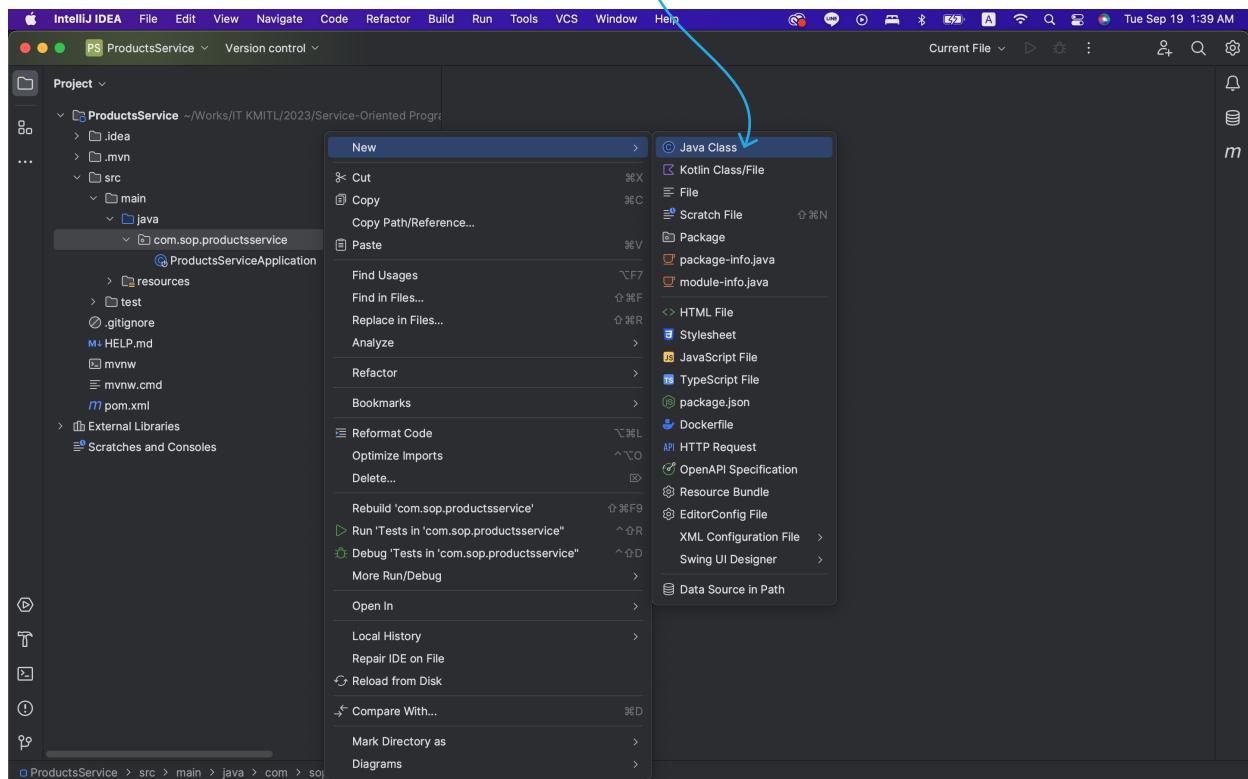
4 กำหนดค่าต่าง ๆ ตามภาพ

5 คลิกปุ่ม Next

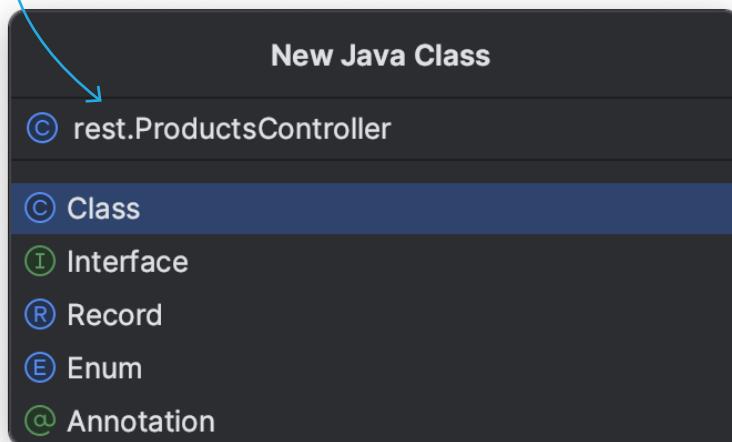
6 คลิกเลือก dependencies ชื่อ Spring Web และ Lombok



8 คลิกขวาที่ com.sop.productservice แล้วเลือก New > Java Class



9 พิมพ์ชื่อ rest.ProductsController และกดปุ่ม Enter



10 เขียนโค้ดตามภาพ

```
package com.sop.productsservice.rest;

import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/products")
public class ProductsController {

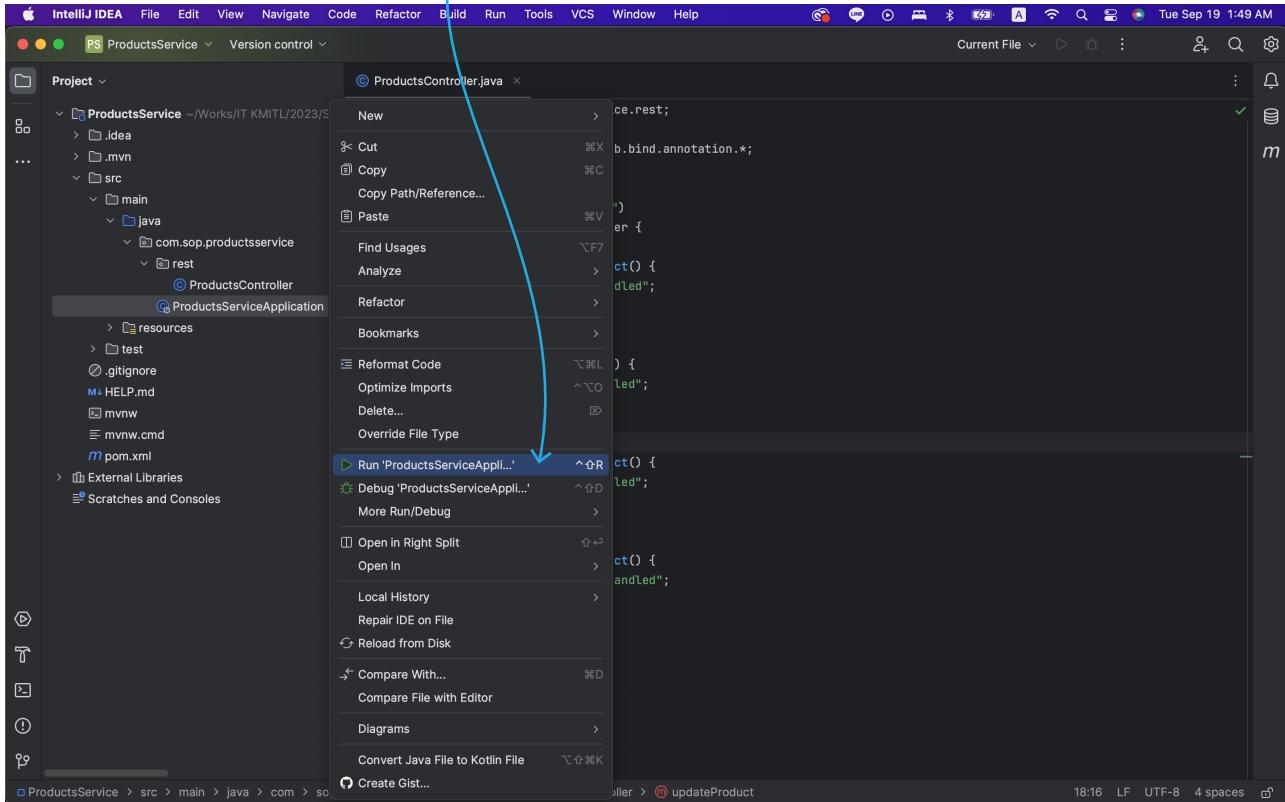
    @PostMapping
    public String createProduct() {
        return "HTTP POST Handled";
    }

    @GetMapping
    public String getProduct() {
        return "HTTP GET Handled";
    }

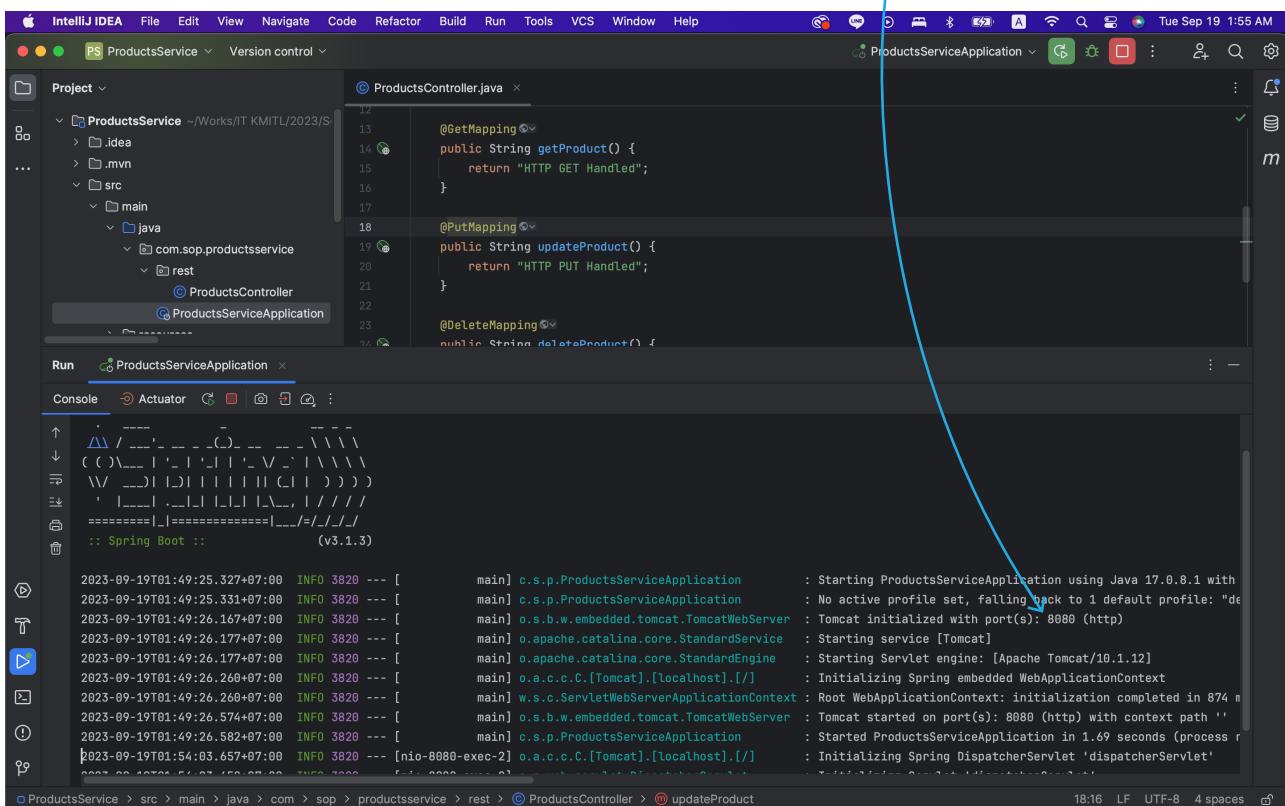
    @PutMapping
    public String updateProduct() {
        return "HTTP PUT Handled";
    }

    @DeleteMapping
    public String deleteProduct() {
        return "HTTP DELETE Handled";
    }
}
```

11 คลิกขวาที่ ProductServiceApplication แล้วเลือก Run 'ProductServiceAppl...'



12 สังเกตหมายเลข port ที่ service รันอยู่



13 ลองเรียก services ทั้งสี่ด้วย Postman

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/products`. The 'Params' tab is selected. The response status is 200 OK, time 179 ms, size 181 B.

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/products`. The 'Params' tab is selected. The response status is 200 OK, time 7 ms, size 180 B.

The screenshot shows the Postman interface with a PUT request to `http://localhost:8080/products`. The 'Params' tab is selected. The response status is 200 OK, time 8 ms, size 180 B.

The screenshot shows the Postman interface with a DELETE request to `http://localhost:8080/products`. The 'Params' tab is selected. The response status is 200 OK, time 3 ms, size 183 B.

แบบฝึกปฏิบัติ ครั้งที่ 8

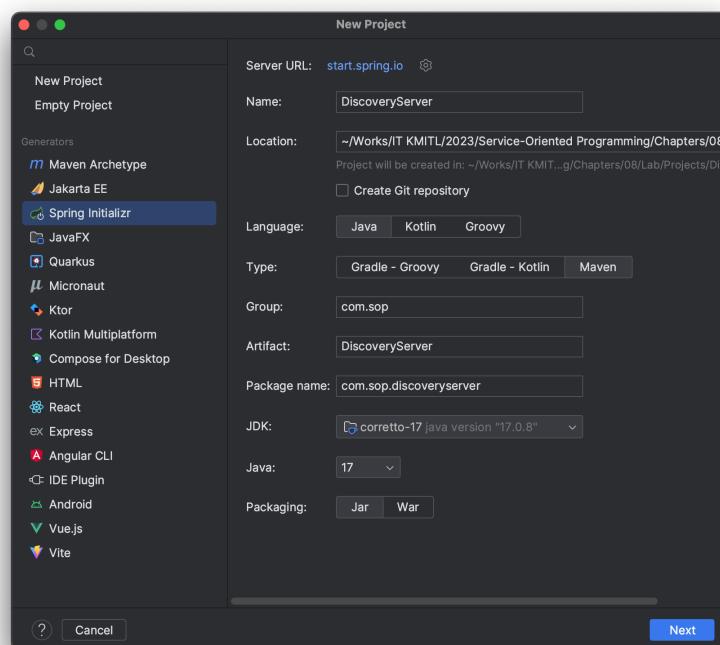
โดย ผศ.ดร.ธราภิเชษฐ์ อธิจักรุณโรจน์ และ อาจารย์สัญญา น้อยจันทร์ วิชา 06016325 การโปรแกรมเชิงบริการ

เรื่อง API Gateway และ Service Discovery

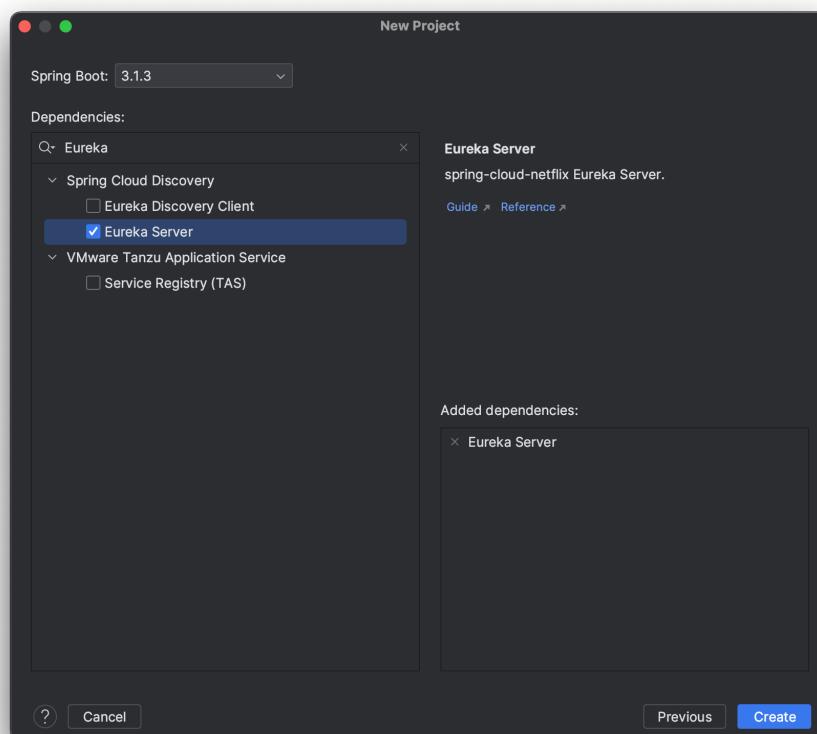
- วัตถุประสงค์
- เพื่อฝึกฝนการสร้าง API Gateway ด้วย Spring Cloud API Gateway
 - เพื่อฝึกฝนการทำ service discovery ด้วย Eureka Discovery

ข้อที่ 2 : สร้าง Discovery Server ด้วย Eureka Server

1 สร้าง project ใหม่ตามภาพ



2 คลิกเลือก dependencies ชื่อ Eureka Server



3 เขียนโค้ดตามภาพ

The screenshot shows the IntelliJ IDEA interface with the project 'DiscoveryServer' open. The code editor displays the `DiscoveryServerApplication.java` file:

```
package com.sop.discoveryserver;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@EnableEurekaServer
@SpringBootApplication
public class DiscoveryServerApplication {

    public static void main(String[] args) {
        SpringApplication.run(DiscoveryServerApplication.class, args);
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the project 'DiscoveryServer' open. The code editor displays the `application.properties` file:

```
server.port=8761
eureka.client.register-with-eureka=false
eureka.client.fetch-registry=false
eureka.instance.prefer-ip-address=true
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
```

4 Run project

5 เปิด `http://localhost:8761` บน web browser

The screenshot shows a web browser displaying the Spring Eureka dashboard at `http://localhost:8761`. The page includes sections for System Status, DS Replicas, General Info, and Instance Info. A blue arrow points from the number 6 in the steps below to the 'No instances available' message in the DS Replicas section.

System Status

Environment	test	Current time	2023-09-19T02:09:13 +0700
Data center	default	Uptime	00:00
		Lease expiration enabled	false
		Renews threshold	1
		Renews (last min)	0

DS Replicas

No instances available

General Info

Name	Value
total-avail-memory	94mb
num-of-cpus	8
current-memory-usage	59mb (62%)
server-uptime	00:00
registered-replicas	http://localhost:8761/eureka/
unavailable-replicas	http://localhost:8761/eureka/ ,
available-replicas	

Instance Info

Name	Value
ipAddr	192.168.1.16
status	UP

6 สังเกตว่าตอนนี้เรายังไม่มี service instance มา register กับ Eureka Server เลย

ข้อที่ 3 : ทำให้ Products Service เป็น Discovery Client

1 เปิดเว็บไซต์ <https://mvnrepository.com>

2 ค้นหา Spring Cloud Starter Netflix Eureka Client

Maven Repository: Eureka client

Found 29923 results

Sort: relevance | popular | newest

1. Eureka Client
com.netflix.eureka > eureka-client
eureka-client
Last Release on Jun 21, 2023

2. Spring Cloud Starter Netflix Eureka Client
org.springframework.cloud > spring-cloud-starter-netflix-eureka-client
Spring Cloud Starter Netflix Eureka Client
Last Release on Jul 28, 2023

3. Spring Cloud Netflix Eureka Client
org.springframework.cloud > spring-cloud-netflix-eureka-client
Spring Cloud Netflix Eureka Client
Last Release on Jul 28, 2023

HITACHI Inspire the Next
The Green Recovery Path in ASEAN
Discover More

3 คลิก

4 คลิก

Maven Repository: org.springframework.cloud > spring-cloud-starter-netflix-eureka-client

Spring Cloud Starter Netflix Eureka Client

Apache 2.0

spring | netflix | cloud | client | starter

#1255 in MvnRepository (See Top Artifacts)

368 artifacts

Version	Vulnerabilities	Repository	Usages	Date
4.0.3		Central	5	Jul 28, 2023
4.0.2		Central	9	May 25, 2023
4.0.1		Central	5	Mar 29, 2023
4.0.0		Central	8	Dec 16, 2022

Web Hosting from GoDaddy

Speedy and reliable service. 99.9% uptime guaranteed. Trust your site to GoDaddy

A woman in a red top and black pants is doing a yoga pose.

Maven Repository: org.springframework.cloud > spring-cloud-starter-netflix-eureka-client/4.0.3

Spring Cloud Starter Netflix Eureka Client » 4.0.3

Apache 2.0

spring | netflix | cloud | client | starter

Pivotal Software, Inc.

https://projects.spring.io/spring-cloud

Jul 28, 2023

pom (6 KB) | jar (2 KB) | View All

Central

#1255 in MvnRepository (See Top Artifacts)

368 artifacts

```
<!-- https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-netflix-eureka-client -->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    <version>4.0.3</version>
</dependency>
```

Include comment with link to declaration

Copied to clipboard!

publicis sapient

Accelerate your digital

Get the report

5 Copy

6 เปิดไฟล์ pom.xml ใน ProductService project ขึ้นมา แล้ววางโค้ดลงไปในตำแหน่งตามภาพ

```

<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    <version>4.0.3</version>
</dependency>

```

7 ลบบรรทัดนี้ออก

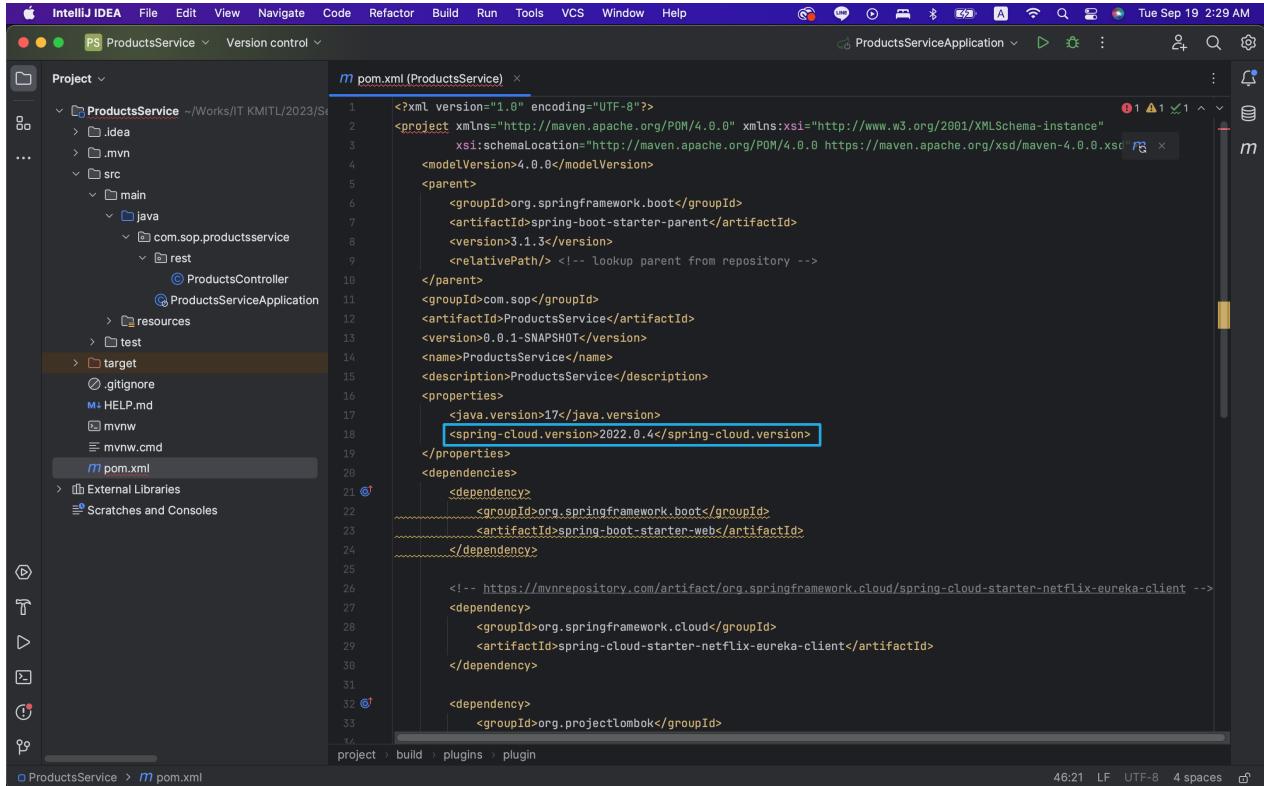
8 Copy บรรทัดนี้จากไฟล์ pom.xml ของ DiscoveryServer project

```

<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
</dependencyManagement>

```

9 วางโค้ดลงไว้ในไฟล์ pom.xml ของ ProductService ในตำแหน่งตามภาพ

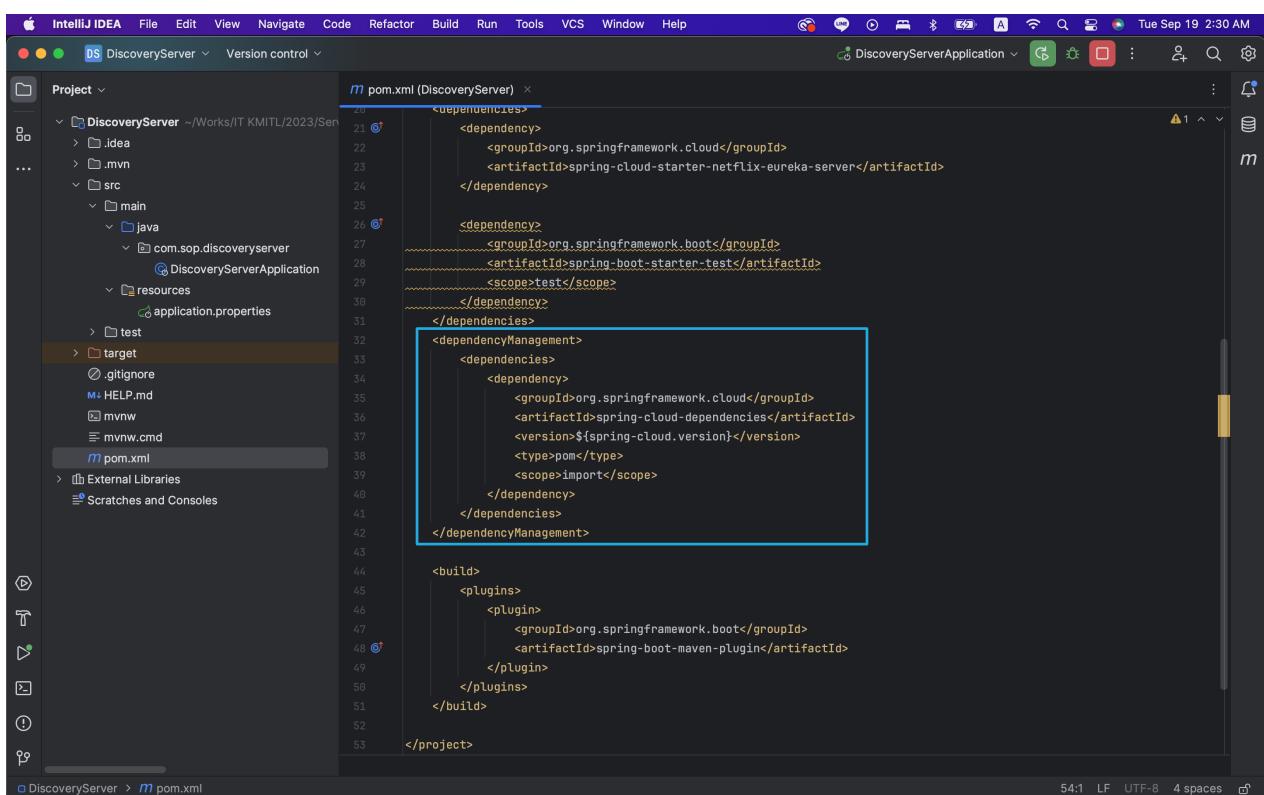


```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.1.3</version>
        <relativePath/> <!-- lookup parent from repository --&gt;
    &lt;/parent&gt;
    &lt;groupId&gt;com.sop&lt;/groupId&gt;
    &lt;artifactId&gt;ProductsService&lt;/artifactId&gt;
    &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;
    &lt;name&gt;ProductsService&lt;/name&gt;
    &lt;description&gt;ProductsService&lt;/description&gt;
    &lt;properties&gt;
        &lt;java.version&gt;17&lt;/java.version&gt;
        &lt;spring-cloud.version&gt;2022.0.4&lt;/spring-cloud.version&gt;
    &lt;/properties&gt;
    &lt;dependencies&gt;
        &lt;dependency&gt;
            &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;
            &lt;artifactId&gt;spring-boot-starter-web&lt;/artifactId&gt;
        &lt;/dependency&gt;

        &lt;!-- https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-netflix-eureka-client --&gt;
        &lt;dependency&gt;
            &lt;groupId&gt;org.springframework.cloud&lt;/groupId&gt;
            &lt;artifactId&gt;spring-cloud-starter-netflix-eureka-client&lt;/artifactId&gt;
        &lt;/dependency&gt;

        &lt;dependency&gt;
            &lt;groupId&gt;org.projectlombok&lt;/groupId&gt;
        &lt;/dependency&gt;
    &lt;/dependencies&gt;
&lt;/project&gt;</pre>
```

10 Copy โค้ดในไฟล์ pom.xml ของ DiscoveryServer ในตำแหน่งตามภาพ



```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
        <version>3.1.3</version>
    </parent>
    <groupId>com.sop.discoveryserver</groupId>
    <artifactId>DiscoveryServerApplication</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>DiscoveryServerApplication</name>
    <description>DiscoveryServerApplication</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>org.springframework.cloud</groupId>
                <artifactId>spring-cloud-dependencies</artifactId>
                <version>${spring-cloud.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

11 วางโค้ดลงไปในไฟล์ pom.xml ของ ProductService ในตำแหน่งตามภาพ

```

<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-dependencies</artifactId>
            <version>${spring-cloud.version}</version>
            <type>pom</type>
            <scope>imports</scope>
        </dependency>
    </dependencies>
</dependencyManagement>

```

13 เขียนโค้ดตามภาพแล้ว run ProductService project ใหม่อีกครั้ง

```

package com.sop.productsservice;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@EnableDiscoveryClient
@SpringBootApplication
public class ProductsServiceApplication {

    public static void main(String[] args) { SpringApplication.run(ProductsServiceApplication.class, args); }
}

```

```

eureka.client.service-url.defaultZone=http://localhost:8761/eureka
spring.application.name=product-service

```

14 Refresh หน้า Spring Eureka Console

The screenshot shows the Spring Eureka console interface. At the top, it displays system status with environment 'test', data center 'default', current time '2023-09-19T02:36:54 +0700', uptime '00:28', lease expiration enabled 'false', renew threshold '3', and renew count '0'. A red warning message states: 'EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.' Below this, the 'DS Replicas' section shows a single instance 'localhost' under 'General Info'. A blue arrow labeled '15' points to the 'localhost' entry, and another blue arrow labeled '16' points to the 'General Info' section.

System Status

Environment	test	Current time	2023-09-19T02:36:54 +0700
Data center	default	Uptime	00:28
		Lease expiration enabled	false
		Renew threshold	3
		Renews (last min)	0

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

DS Replicas

localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
PRODUCT-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.16:product-service

General Info

Name	Value

15 สังเกตว่า products service ปรากฏขึ้นมาแล้ว

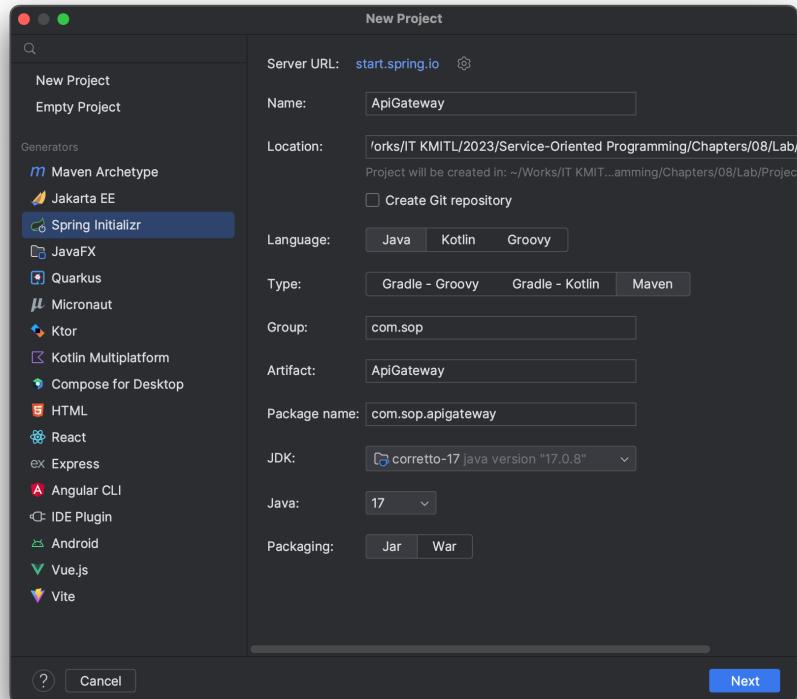
16 เปิดลิงค์นี้ในแท็บใหม่

17 URL จะลงทะเบียนด้วย /actuator/info ให้ลับส่วนนี้ออกแล้วแทนที่ด้วย /products

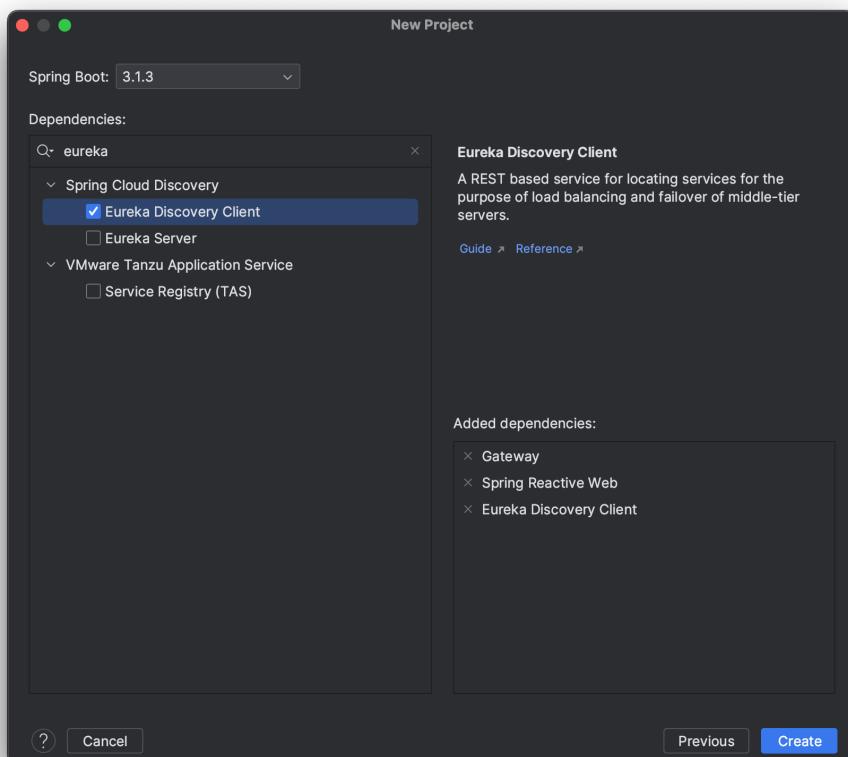
The screenshot shows a browser window with the URL '92.168.1.16:8080/products'. The page content is 'HTTP GET Handled'. A blue arrow labeled '17' points to the URL bar.

ข้อที่ 4 : สร้าง API Gateway ด้วย Spring Cloud API Gateway

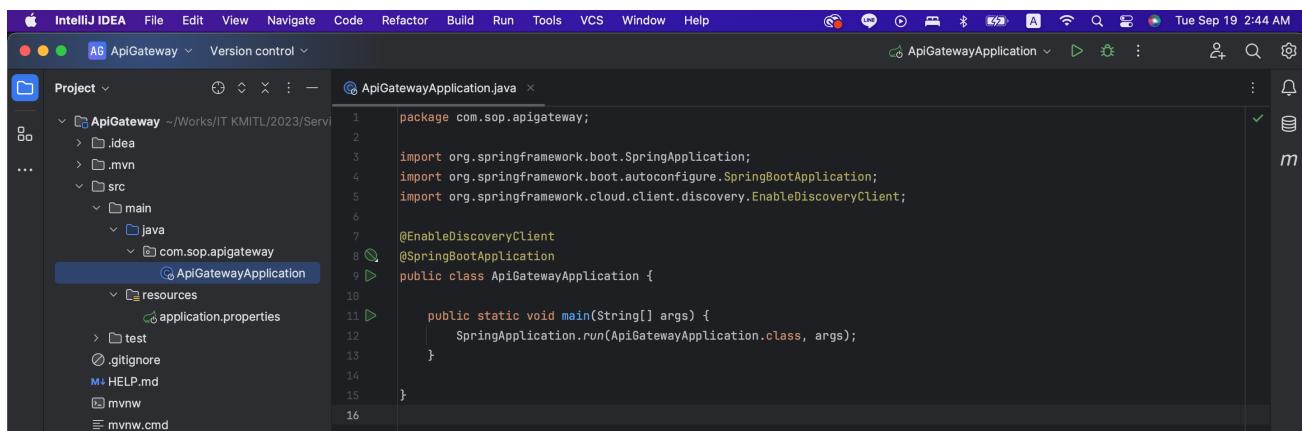
1 สร้าง project ใหม่ด้วยรายละเอียดตามภาพ



2 เลือก dependencies 3 ตัว ได้แก่ Gateway, Spring Reactive Web และ Eureka Discovery Client

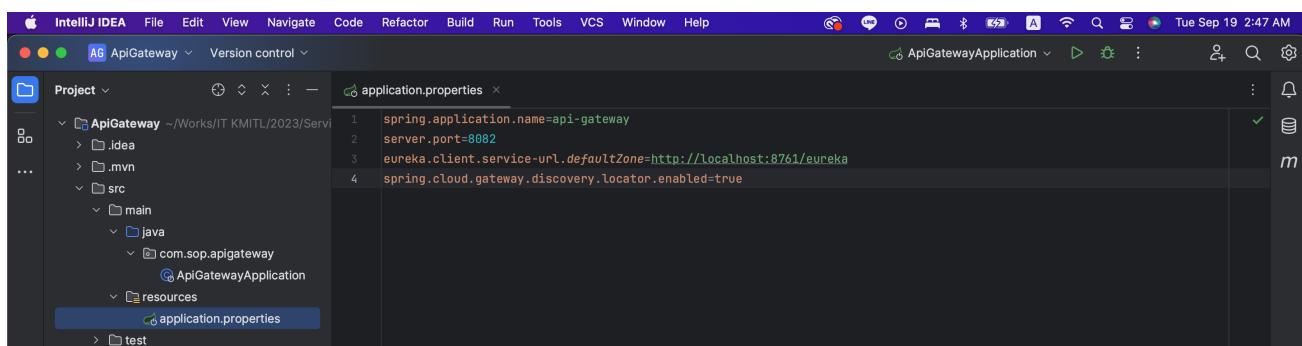


3 เขียนโค้ดตามภาพแล้ว run project



The screenshot shows the IntelliJ IDEA interface with the project 'ApiGateway' open. The code editor displays the file 'ApiGatewayApplication.java' containing the following code:

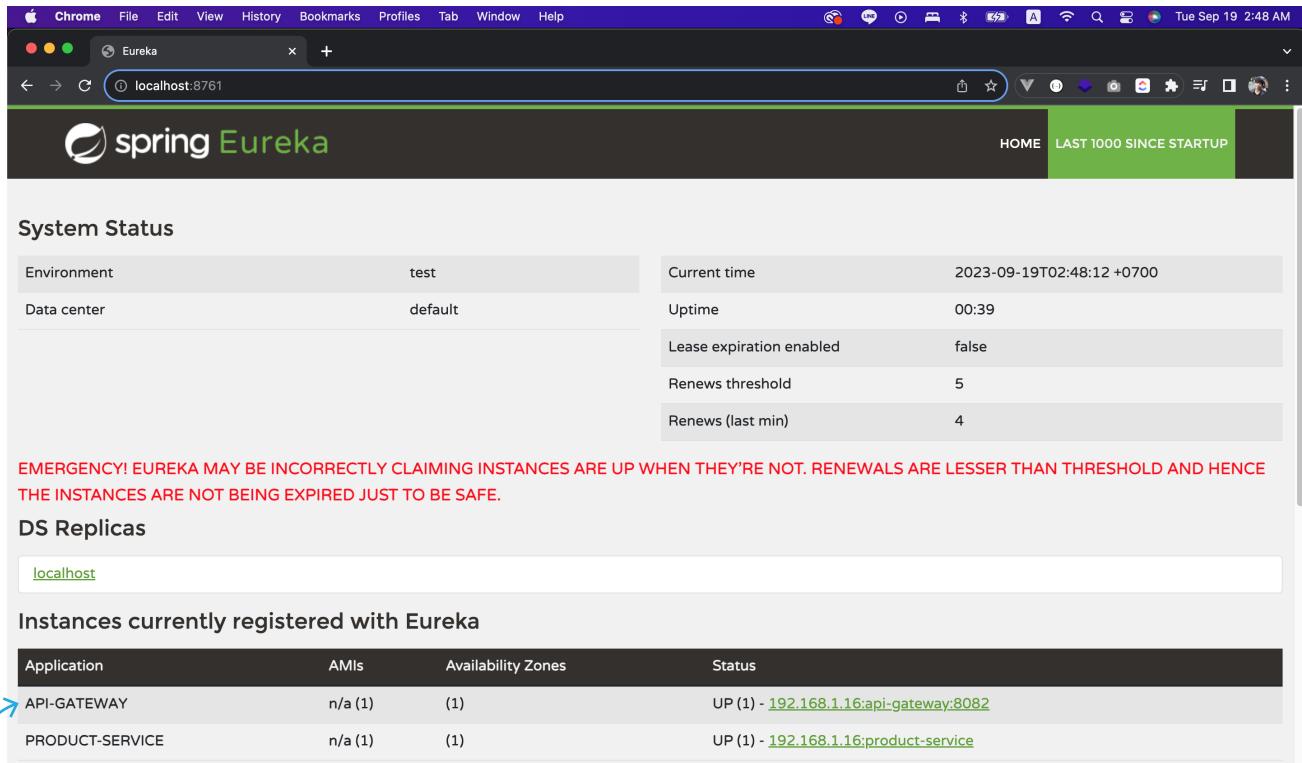
```
1 package com.sop.apigateway;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
6
7 @EnableDiscoveryClient
8 @SpringBootApplication
9 public class ApiGatewayApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(ApiGatewayApplication.class, args);
13     }
14
15 }
```



The screenshot shows the IntelliJ IDEA interface with the project 'ApiGateway' open. The code editor displays the file 'application.properties' containing the following configuration:

```
1 spring.application.name=api-gateway
2 server.port=8082
3 eureka.client.service-url.defaultZone=http://localhost:8761/eureka
4 spring.cloud.gateway.discovery.locator.enabled=true
```

4 Refresh หน้า Spring Eureka Console



The screenshot shows a browser window displaying the Spring Eureka console at 'localhost:8761'. The page has a green header with the text 'spring Eureka' and 'HOME LAST 1000 SINCE STARTUP'. Below the header is a section titled 'System Status' with the following table:

Environment	test	Current time	2023-09-19T02:48:12 +0700
Data center	default	Uptime	00:39
		Lease expiration enabled	false
		Renew threshold	5
		Renew (last min)	4

Below the status table is a red warning message: 'EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.'

The next section is 'DS Replicas' with a table:

localhost			
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - 192.168.1.16:api-gateway:8082
PRODUCT-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.16:product-service

5 สังเกตว่า API Gateway ปรากฏขึ้นมาแล้ว

6 ลองเรียก products service ผ่าน API gateway

The screenshot shows the Postman interface. On the left, there's a sidebar with 'Service-Oriented Programming' and 'Microservices' sections. Under 'Microservices', there's a 'Products' section with four items: 'POST Create Product', 'GET Get Products' (which is selected), 'PUT Update Product', and 'DELETE Delete Product'. In the main area, a 'Get Products' request is being made to 'http://localhost:8082/PRODUCT-SERVICE/products'. The 'Params' tab is active, showing a single parameter 'Key' with value 'Value'. Below the request, the status bar indicates 'Status: 200 OK Time: 692 ms Size: 132 B'. A blue arrow points from the number 6 to the 'Get Products' item in the sidebar.

7 ข้อมูลของ service ที่ต้องการเรียก (ดูจากหน้า Spring Eureka Console)

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - 192.168.1.16:api-gateway:8082
PRODUCT-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.16:product-service

8 ลองเปลี่ยนเป็นตัวพิมพ์เล็กๆ เรียกแล้วจะได้ Error

The screenshot shows the Postman interface. The 'Get Products' request is now directed to 'http://localhost:8082/product-service/products'. The 'Body' tab is active, showing the response body which includes a timestamp, a path, an error code, an error message, and a request ID. The status bar at the bottom indicates 'Status: 404 Not Found Time: 42 ms Size: 232 B'. A blue arrow points from the number 8 to the URL field in the request header.

9 เขียนโค้ดตามภาพแล้ว run API Gateway อีกครั้ง

The screenshot shows the IntelliJ IDEA interface with a project named 'ApiGateway'. In the 'src/main/resources' directory, there is a file named 'application.properties'. The content of the file is as follows:

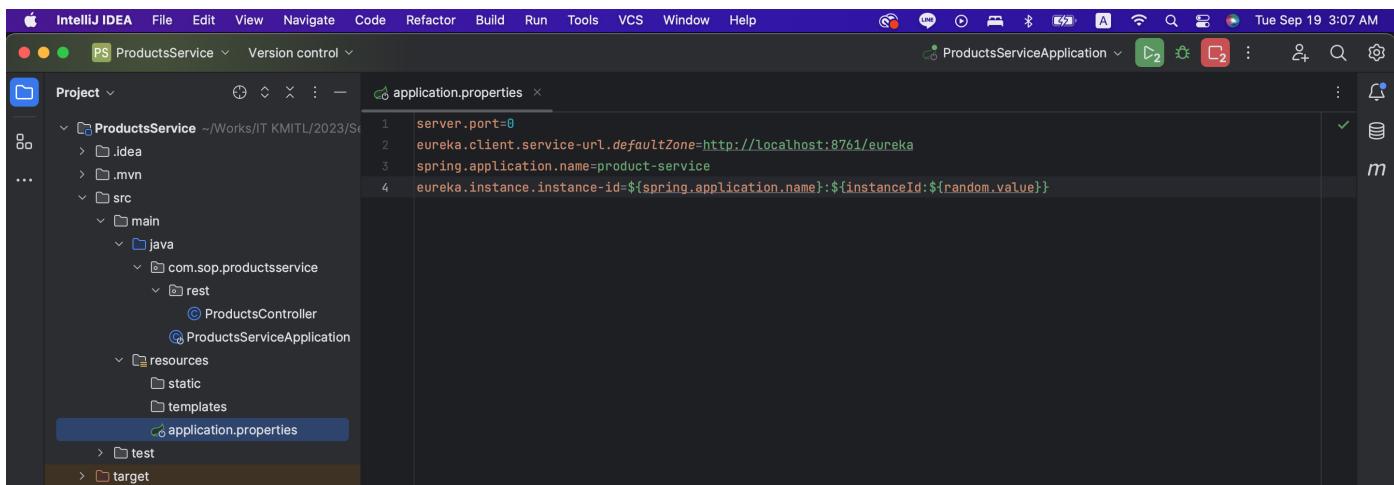
```
spring.application.name=api-gateway
server.port=8082
eureka.client.service-url.defaultZone=http://localhost:8761/suska
spring.cloud.gateway.discovery.locator.enabled=true
spring.cloud.gateway.discovery.locator.lower-case-service-id=true
```

10 ลองเรียก products service ดูใหม่

The screenshot shows the Postman interface. The 'Get Products' request is now directed to 'http://localhost:8082/product-service/products'. The 'Body' tab is active, showing the response body which includes a timestamp, a path, an error code, an error message, and a request ID. The status bar at the bottom indicates 'Status: 200 OK Time: 412 ms Size: 132 B'. A blue arrow points from the number 10 to the 'Send' button.

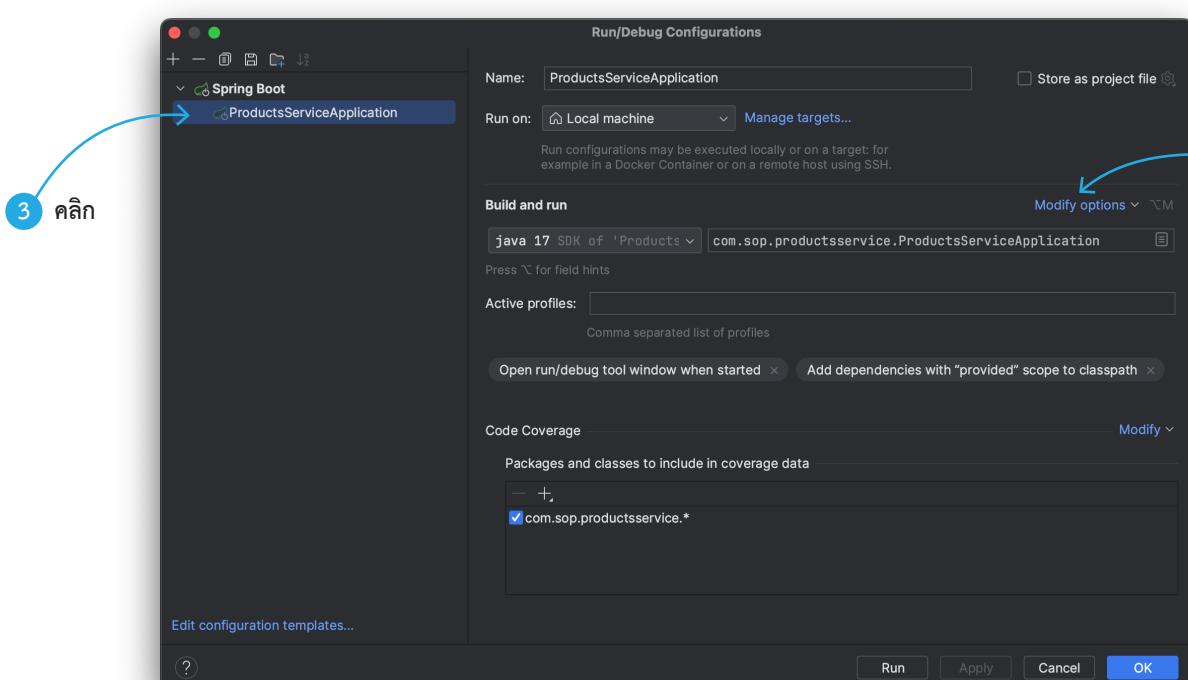
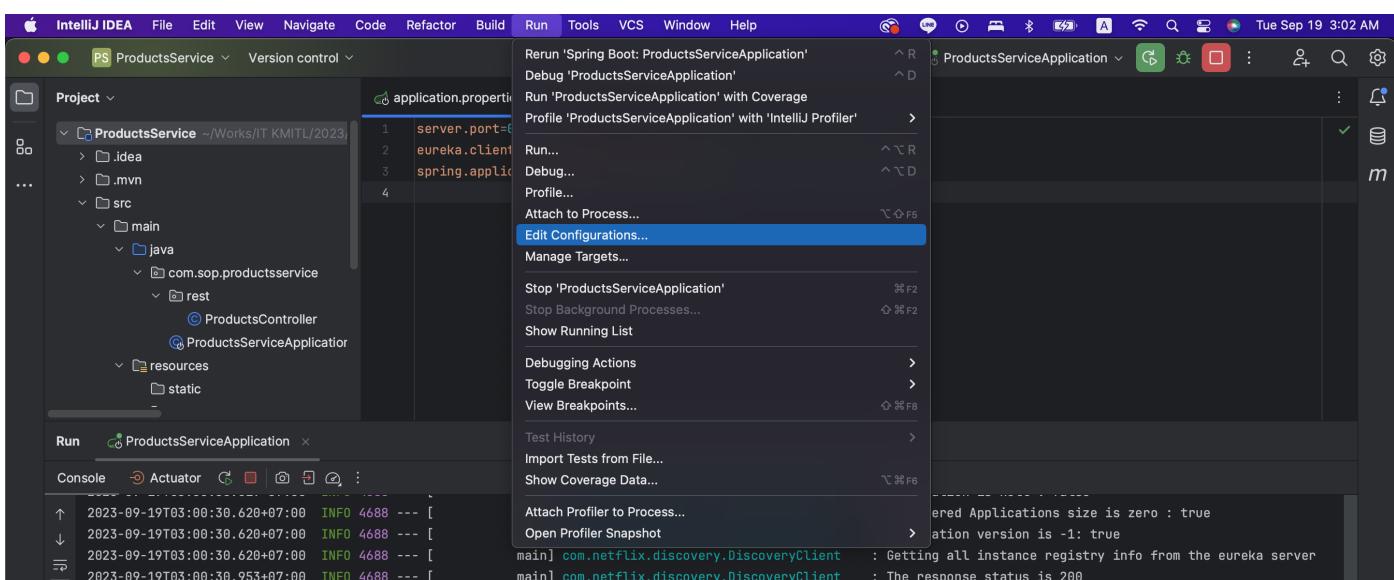
ข้อที่ 4 : ทำ Load Balancing (สร้าง instance ของ Test Service มากกว่า 1 ตัว)

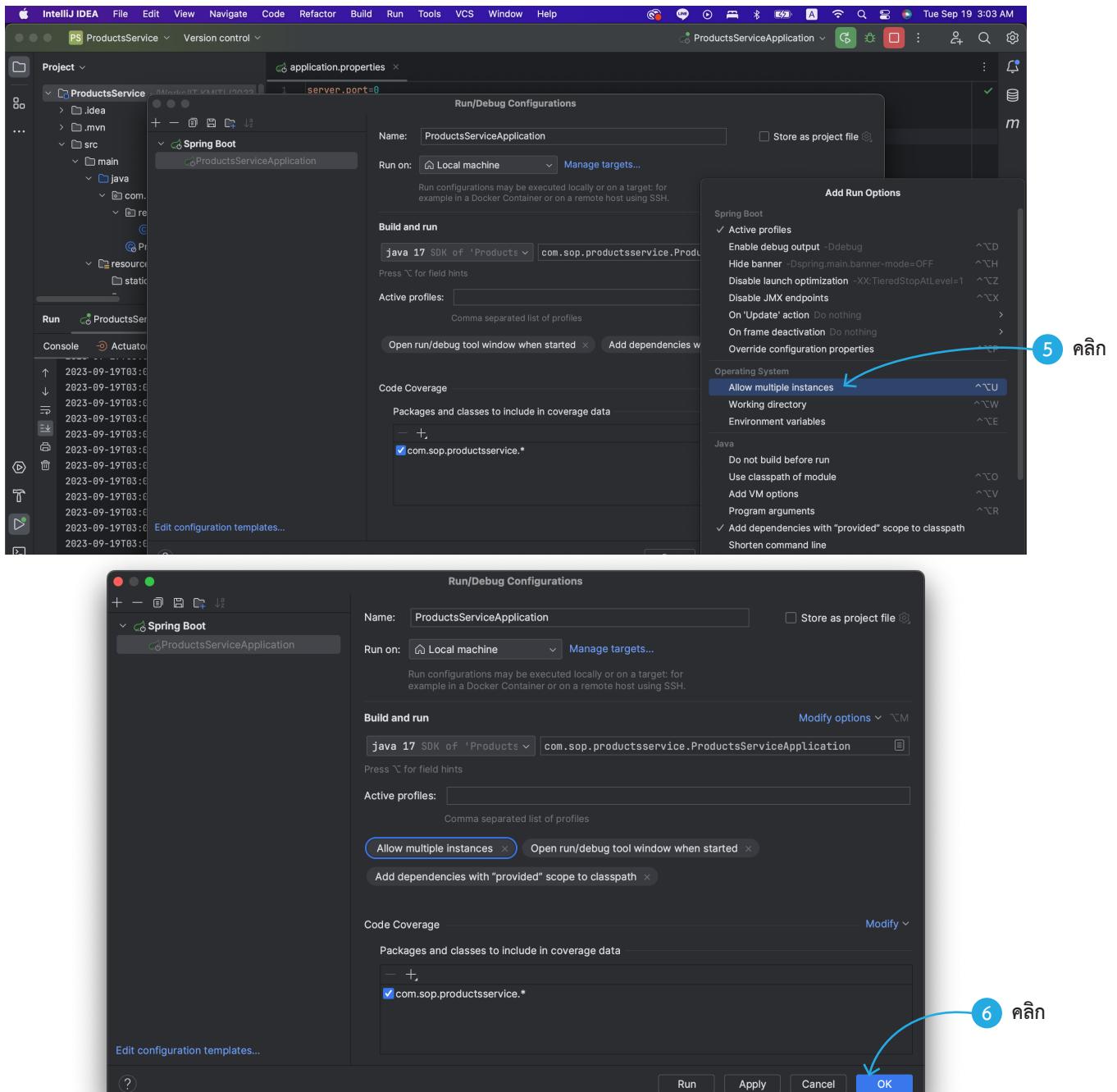
1 แก้ไขไฟล์ application.properties ใน ProductService ตามภาพ



```
server.port=0
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
spring.application.name=product-service
eureka.instance.instance-id=${spring.application.name}:${instanceId:${random.value}}
```

2 ตั้งค่าให้ IntelliJ IDEA สามารถรัน instance ของ project เดิมได้มากกว่า 1 instance โดยการคลิกที่ Run > Edit Configurations...





7 กด run ProductService project หลาย ๆ ครั้ง

8 ล็อกเกอร์ product service ปรากฏขึ้นมาหลาย instances

Application	Availability		
	AMIs	Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - 192.168.1.16:api-gateway:8082
PRODUCT-SERVICE	n/a (3)	(3)	UP (3) - product-service:105dfdc1f3c32c6ac082a697b0bc58f , product-service:dc48e0ead13cab71b6a02382c226002 , product-service:0e2d1b8ee47d39f0ae699fa936dd63c0

9. เขียนโค้ดเพิ่มตามภาพ

```

1 package com.sop.productsservice.rest;
2
3 import org.springframework.core.env.Environment;
4 import org.springframework.web.bind.annotation.*;
5
6 @RestController
7 @RequestMapping("/products")
8 public class ProductsController {
9
10    private final Environment env;
11
12    public ProductsController(Environment env) {
13        this.env = env;
14    }
15
16    @PostMapping
17    public String createProduct() {
18        return "HTTP POST Handled";
19    }
20
21    @GetMapping
22    public String getProduct() {
23        return "HTTP GET Handled " + env.getProperty("local.server.port");
24    }
25
26    @PutMapping
27    public String updateProduct() {
28        return "HTTP PUT Handled";
29    }
30
31    @DeleteMapping
32    public String deleteProduct() {
33        return "HTTP DELETE Handled";
34    }
}

```

10. เรียก product service ผ่าน API gateway ดูหลาย ๆ ครั้ง สังเกตว่าหมายเลข port ที่แสดงเป็นผลลัพธ์ของ service จะเปลี่ยนไปเรื่อย ๆ เป็นการกระจาย load ไปให้แต่ละ instance ของ product service

The screenshots show the Postman interface with the following details:

- Screenshot 1:** Request URL: `http://localhost:8082/product-service/products`. Response status: 200 OK, Time: 19 ms, Size: 138 B.
- Screenshot 2:** Request URL: `http://localhost:8082/product-service/products`. Response status: 200 OK, Time: 24 ms, Size: 138 B.
- Screenshot 3:** Request URL: `http://localhost:8082/product-service/products`. Response status: 200 OK, Time: 25 ms, Size: 138 B.