

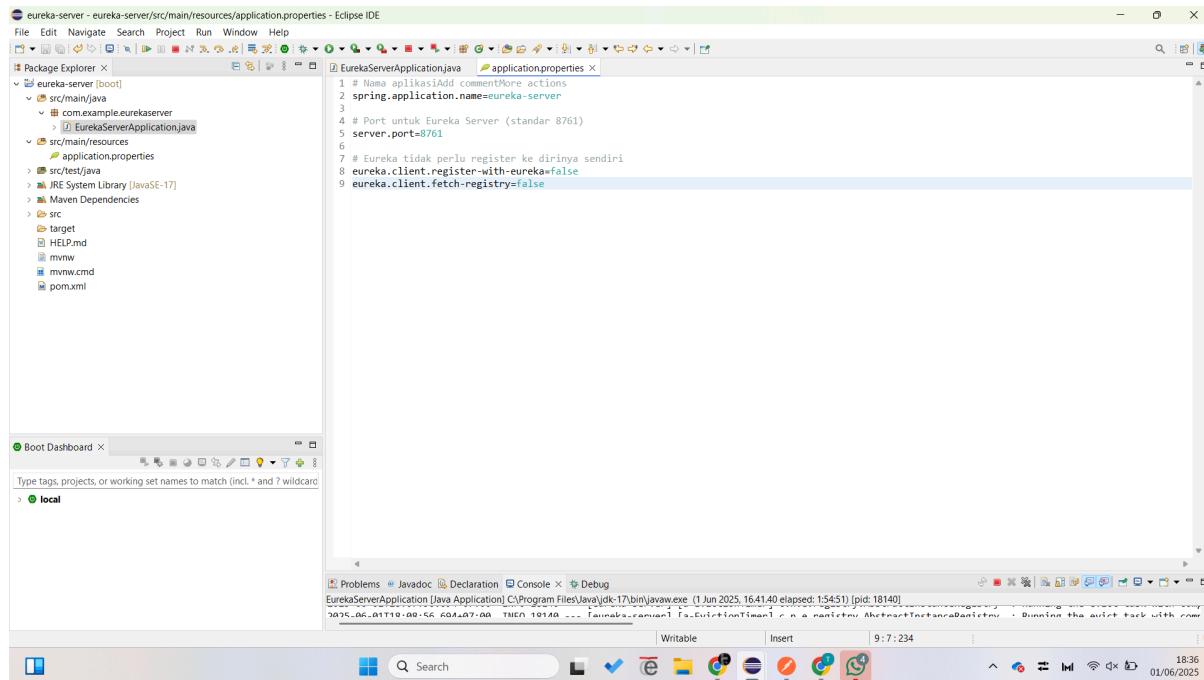
## Microservices dengan Spring Boot

Microservices adalah cara membangun aplikasi dengan membaginya menjadi bagian-bagian kecil yang independen. Setiap bagian bertanggung jawab atas satu fungsi spesifik dan dapat berjalan sendiri.

### microservices-ecommerce

#### eureka-server

##### EurekaServerApplication.java



The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer:** Shows the project structure for "eureka-server [boot]". It includes a src/main/java folder containing "com.example.eurekaserver" with "EurekaServerApplication.java", and a src/main/resources folder containing "application.properties".
- EurekaServerApplication.java:** The code defines a Spring Boot application with the following properties:

```
1 # Nama aplikasiAdd commentMore actions
2 spring.application.name=eureka-server
3
4 # Port untuk Eureka Server (standar 8761)
5 server.port=8761
6
7 # Eureka tidak perlu register ke dirinya sendiri
8 eureka.client.register-with-eureka=false
9 eureka.client.fetch-registry=false
```
- application.properties:** Contains configuration for the application, including port and registry settings.
- Boot Dashboard:** Shows a local workspace entry.
- Console:** Displays the output of the Java application run command, showing the application is running on port 8761.
- Bottom Status Bar:** Shows the current time as 18:36 and the date as 01/06/2025.

##### application.properties

Eureka Server Application Code (EurekaServerApplication.java):

```

1 package com.example.eurekaserver;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;
6
7 @SpringBootApplication
8 @EnableEurekaServer // Ini yang bikin jadi Eureka Server
9 public class EurekaServerApplication {
10    public static void main(String[] args) {
11        SpringApplication.run(EurekaServerApplication.class, args);
12    }
13 }

```

Console Output:

```

EurekaServerApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 Jun 2025, 16:41:40 elapsed: 1:55:05) [pid: 18140]
2025-06-01T17:34:03.899+07:00 INFO 29616 --- [user-service] (rap-executor-<4>) c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2025-06-01T17:34:39.099+07:00 INFO 29616 --- [user-service] (beatExecutor-<4>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1 : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1
2025-06-01T17:34:39.113+07:00 INFO 29616 --- [user-service] (beatExecutor-<4>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1 : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1

```

## user-service

### application.properties

User Service Application Code (UserServiceApplication.java):

```

1 # Port untuk User ServiceAdd commentMore actions
2 server.port=8081
3
4 # Nama service (harus unik)
5 spring.application.name=user-service
6
7 # Alamat Eureka Server untuk register
8 eureka.client.service-url.defaultZone=http://localhost:8761/eureka/

```

Console Output:

```

UserServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 Jun 2025, 16:43:44 elapsed: 1:53:53) [pid: 29616]
2025-06-01T17:34:03.899+07:00 INFO 29616 --- [user-service] (rap-executor-<4>) c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2025-06-01T17:34:39.099+07:00 INFO 29616 --- [user-service] (beatExecutor-<4>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1 : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1
2025-06-01T17:34:39.113+07:00 INFO 29616 --- [user-service] (beatExecutor-<4>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1 : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1

```

### UserServiceApplication.java

user-service - user-service/src/main/java/com/example/demo/UserServiceApplication.java - Eclipse IDE

```

1 package com.example.demo;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
6 import org.springframework.cloud.client.loadbalancer.LoadBalanced;
7 import org.springframework.context.annotation.Bean;
8 import org.springframework.web.client.RestTemplate;
9
10 @SpringBootApplication
11 @EnableDiscoveryClient
12 public class UserServiceApplication {
13
14     public static void main(String[] args) {
15         SpringApplication.run(UserServiceApplication.class, args);
16     }
17
18     @Bean
19     @LoadBalanced // ⚡ TAMBAHKAN INI - PENTING!
20     public RestTemplate restTemplate() {
21         return new RestTemplate();
22     }
23 }

```

Boot Dashboard

Type tags, projects, or working set names to match:

local

com.example.demo.UserServiceApplication.java - user-service/src/main/java

## model/User.java

user-service - user-service/src/main/java/com/example/demo/model/User.java - Eclipse IDE

```

1 package com.example.demo.model;
2
3 public class User {
4     private Long id;
5     private String name;
6     private String email;
7     private String phone;
8
9     // Constructor
10    public User() {}
11
12    public User(Long id, String name, String email, String phone) {
13        this.id = id;
14        this.name = name;
15        this.email = email;
16        this.phone = phone;
17    }
18
19    // Getters dan Setters
20    public Long getId() { return id; }
21    public void setId(Long id) { this.id = id; }
22
23    public String getName() { return name; }
24    public void setName(String name) { this.name = name; }
25
26    public String getEmail() { return email; }
27    public void setEmail(String email) { this.email = email; }
28
29    public String getPhone() { return phone; }
30    public void setPhone(String phone) { this.phone = phone; }
31 }

```

Boot Dashboard

Type tags, projects, or working set names to match:

local

com.example.demo.UserServiceApplication.java - user-service/src/main/java

## controller/UserController.java

user-service - user-service/src/main/java/com/example/demo/controller/UserController.java - Eclipse IDE

```

1 package com.example.demo.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.*;
5 import org.springframework.web.client.RestTemplate;
6 import org.springframework.http.ResponseEntity;
7 import com.example.demo.model.User;
8 import java.util.*;
9
10 @RestController
11 @RequestMapping("/users")
12 public class UserController {
13
14     @Autowired
15     private RestTemplate restTemplate;
16
17     // Data dummy untuk simulasi database
18     private List<User> users = Arrays.asList(
19         new User(1L, "John Doe", "john@email.com", "081234567890"),
20         new User(2L, "Jane Smith", "jane@email.com", "081234567891"),
21         new User(3L, "Bob Johnson", "bob@email.com", "081234567892")
22     );
23
24     // GET /users - Ambil semua user
25     @GetMapping
26     public List<User> getAllUsers() {
27         return users;
28     }
29
30     // GET /users/{id} - Ambil user berdasarkan ID
31     @GetMapping("/{id}")
32     public User getUserId(@PathVariable Long id) {
33         return users.stream()
34             .filter(user -> user.getId().equals(id))
35             .findFirst();
36     }
37 }

```

Boot Dashboard

Type tags, projects, or working set names to match:

local

Problems Javadoc Declaration Console

UserServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe [1 Jun 2025, 16:43:44 elapsed: 1:54:11] [pid:29616]

```

2025-06-01T17:34:03.899+07:00 INFO 29616 --- [user-service] (rap-executor-<id>) c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2025-06-01T17:34:39.098+07:00 INFO 29616 --- [user-service] (beatExecutor-<id>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1
2025-06-01T17:34:39.099+07:00 INFO 29616 --- [user-service] (beatExecutor-<id>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1
2025-06-01T17:34:39.113+07:00 INFO 29616 --- [user-service] (beatExecutor-<id>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1

```

Writable Smart Insert 7:24:280

18:37 01/06/2025

user-service - user-service/src/main/java/com/example/demo/controller/UserController.java - Eclipse IDE

```

1 package com.example.demo.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.*;
5 import org.springframework.web.client.RestTemplate;
6 import org.springframework.http.ResponseEntity;
7 import com.example.demo.model.User;
8 import java.util.*;
9
10 @RestController
11 @RequestMapping("/users")
12 public class UserController {
13
14     @Autowired
15     private RestTemplate restTemplate;
16
17     // GET /users/info - Info service
18     @GetMapping("/info")
19     public String getServiceInfo() {
20         return "User Service is running on port 8081";
21     }
22
23     // Method yang diperbaiki - komunikasi dengan Product Service
24     @GetMapping("/products")
25     public ResponseEntity<Object> getUserProducts() {
26         try {
27             // Panggil Product Service melalui nama service (bukan localhost)
28             Object products = restTemplate.getForObject(
29                 "http://product-service/products", // ⚡ UBAH INI
30                 Object.class
31             );
32
33             Map<String, Object> response = new HashMap<>();
34             response.put("message", "Products available for users");
35             response.put("products", products);
36             response.put("source", "Called from User Service");
37
38             return ResponseEntity.ok(response);
39         } catch (Exception e) {
40             Map<String, Object> errorResponse = new HashMap<>();
41             errorResponse.put("error", "Cannot connect to Product Service");
42             errorResponse.put("details", e.getMessage());
43             errorResponse.put("suggestion", "Make sure Product Service is running and registered to Eureka");
44
45             return ResponseEntity.status(500).body(errorResponse);
46         }
47     }
48 }

```

Boot Dashboard

Type tags, projects, or working set names to match:

local

Problems Javadoc Declaration Console

UserServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe [1 Jun 2025, 16:43:44 elapsed: 1:54:24] [pid:29616]

```

2025-06-01T17:34:03.899+07:00 INFO 29616 --- [user-service] (rap-executor-<id>) c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2025-06-01T17:34:39.098+07:00 INFO 29616 --- [user-service] (beatExecutor-<id>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1
2025-06-01T17:34:39.099+07:00 INFO 29616 --- [user-service] (beatExecutor-<id>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1
2025-06-01T17:34:39.113+07:00 INFO 29616 --- [user-service] (beatExecutor-<id>) com.netflix.discovery.DiscoveryClient : DiscoveryClient_USER-SERVICE/LAPTOP-KKVRM5P:user-1

```

Writable Smart Insert 7:24:280

18:38 01/06/2025

Eclipse IDE - user-service - user-service/src/main/java/com/example/demo/controller/UserController.java

```

68     return ResponseEntity.status(500).body(errorResponse);
69   }
70 }
71 // Method dashboard yang diperbaiki
72 @GetMapping("/{id}/dashboard")
73 public ResponseEntity<UserDashboard> getUserDashboard(@PathVariable Long id) {
74   Map<String, Object> dashboard = new HashMap<>();
75
76   try {
77     // Ambil data user
78     User user = getUserById(id);
79     if (user == null) {
80       return ResponseEntity.notFound().build();
81     }
82     dashboard.put("user", user);
83
84     // Ambil info dari Product Service
85     String productInfo = restTemplate.getForObject(
86       "http://product-service/products/info", // UBAH INI JUGA
87       String.class
88     );
89     dashboard.put("productServiceStatus", productInfo);
90     dashboard.put("status", "All services connected");
91
92     return ResponseEntity.ok(dashboard);
93   } catch (Exception e) {
94     dashboard.put("user", getUserById(id));
95     dashboard.put("productServiceStatus", "Product Service unavailable");
96     dashboard.put("error", e.getMessage());
97     dashboard.put("status", "Partial service failure");
98
99     return ResponseEntity.ok(dashboard);
100   }
101 }
102

```

Boot Dashboard

Problems Javadoc Declaration Console

UserServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe [1 Jun 2025, 16:43:44 elapsed: 1:54:39] [pid:29616]

2025-06-01T17:34:03.899+07:00 INFO 29616 --- [user-service] [rap-executor-<id>] com.netflix.eureka.discovery.EurekaClient : DiscoveryClient\_USER-SERVICE/LAPTOP-KKVRM5P:user-1 : Resolving eureka endpoints via configuration

2025-06-01T17:34:39.098+07:00 INFO 29616 --- [user-service] [beatExecutor-<id>] com.netflix.discovery.DiscoveryClient : DiscoveryClient\_USER-SERVICE/LAPTOP-KKVRM5P:user-1 : DiscoveryClient\_USER-SERVICE/LAPTOP-KKVRM5P:user-1

2025-06-01T17:34:39.113+07:00 INFO 29616 --- [user-service] [beatExecutor-<id>] com.netflix.discovery.DiscoveryClient : DiscoveryClient\_USER-SERVICE/LAPTOP-KKVRM5P:user-1

2025-06-01T17:34:39.113+07:00 INFO 29616 --- [user-service] [beatExecutor-<id>] com.netflix.discovery.DiscoveryClient : DiscoveryClient\_USER-SERVICE/LAPTOP-KKVRM5P:user-1

Writable Smart Insert 7:24:280

18:38 01/06/2025

## product-service

### application.properties

Eclipse IDE - product-service - product-service/src/main/resources/application.properties

```

server.port=8082
spring.application.name=product-service
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/

```

Boot Dashboard

Problems Javadoc Declaration Console

ProductServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe [1 Jun 2025, 16:47:46 elapsed: 1:52:34] [pid:23964]

2025-06-01T17:27:58.238+07:00 INFO 23964 --- [product-service] [rap-executor-<id>] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration

Writable Insert 3:68:124

18:40 01/06/2025

### ProductServiceApplication.java

product-service - product-service/src/main/java/com/example/demo/ProductServiceApplication.java - Eclipse IDE

```

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

```

Boot Dashboard

ProductServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 Jun 2025, 16:47:46 elapsed: 152:03) [pid: 23964]
2025-06-01T17:27:58.238+07:00 INFO 23964 --- [prod-executor-0] c.h.e.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration

18:39 01/06/2025

## model/Product.java

product-service - product-service/src/main/java/com/example/demo/model/Product.java - Eclipse IDE

```

1 package com.example.demo.model;
2
3 public class Product {
4     private Long id;
5     private String name;
6     private String description;
7     private Double price;
8     private Integer stock;
9
10    // Constructor
11    public Product() {}
12
13    public Product(Long id, String name, String description, Double price, Integer stock) {
14        this.id = id;
15        this.name = name;
16        this.description = description;
17        this.price = price;
18        this.stock = stock;
19    }
20
21    // Getters dan Setters
22    public Long getId() { return id; }
23    public void setId(Long id) { this.id = id; }
24
25    public String getName() { return name; }
26    public void setName(String name) { this.name = name; }
27
28    public String getDescription() { return description; }
29    public void setDescription(String description) { this.description = description; }
30
31    public Double getPrice() { return price; }
32    public void setPrice(Double price) { this.price = price; }
33
34    public Integer getStock() { return stock; }
35    public void setStock(Integer stock) { this.stock = stock; }
36}

```

Boot Dashboard

ProductServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 Jun 2025, 16:47:46 elapsed: 152:16) [pid: 23964]
2025-06-01T17:27:58.238+07:00 INFO 23964 --- [prod-executor-0] c.h.e.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration

18:40 01/06/2025

## controller/ProductController.java

product-service - product-service/src/main/java/com/example/demo/controller/ProductController.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer X
product-service [root]
src/main/java
com.example.demo
 ProductServiceApplication.java
ProductController.java
Product.java
application.properties
src/main/resources
static
templates
application.properties
src/test/java
IJE System Library [JavaSE-17]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Boot Dashboard X
Type tags, projects, or working set names to match (incl.* and ? wildc
local
com.example.demo.ProductServiceApplication.java - product-service/src/main/java
Problems Javadoc Declaration Console X
ProductServiceApplication Java Application C:\Program Files\Java\jdk-17\bin\javaw.exe (1 Jun 2025, 16:47:46 elapsed: 151ms) [pid:23964]
2023-06-01T17:27:59.238+07:00 INFO 23964 --- [product-service] [rap-executor-0] c.n.e.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via c
18:39 01/06/2025

```

## api-gateway

pom.xml (tambahkan di dalam <dependencies>)

api-gateway - api-gateway/pom.xml - Eclipse IDE

```

File Edit Navigate Search Project Run Window Help
application.properties ApiGatewayApplication.java pom.xml
https://maven.apache.org/xsd/maven-4.0.0.xsd (xi:schemaLocation)
<xsd:version>1.0</xsd:version>
<xsd:encoding>UTF-8</xsd:encoding>
<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/POM/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>3.1.8</version> Upgrade to the Latest Patch
    <relativePath/>
  </parent>
  <groupId>com.example</groupId>
  <artifactId>api-gateway</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>api-gateway</name>
  <description>API Gateway project for Spring Boot and Spring Cloud</description>
  <properties>
    <java.version>17</java.version>
    <spring-cloud.version>2022.0.4</spring-cloud.version>
  </properties>
  <dependencies>
    <!-- Actuator -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-actuator</artifactId>
    </dependency>
    <!-- Eureka Client -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>
  </dependencies>

```

Boot Dashboard X

Type tags, projects, or working set names to match (incl.\* and ? wildc  
local

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Problems Javadoc Declaration Console X Debug

ApiGatewayApplication Java Application C:\Program Files\Java\jdk-17\bin\javaw.exe (1 Jun 2025, 18:40:50 elapsed: 0:02:20) [pid:26400]

18:43 01/06/2025

api-gateway - api-gateway/pom.xml - Eclipse IDE

```

<!-- Spring cloud Gateway -->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-gateway</artifactId>
</dependency>
<!-- WebFlux (dibutuhkan oleh Gateway) -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-webflux</artifactId>
</dependency>
<!-- Test -->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-dependencies</artifactId>
    <version>${spring-cloud.version}</version>
    <type>pom</type>
    <scope>import</scope>
</dependency>
</dependencies>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-dependencies</artifactId>
            <version>${spring-cloud.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
<build>
    <plugins>
        <plugin>
</plugin>
    </plugins>
</build>

```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Problems | Javadoc | Declaration | Console | Debug

ApiGatewayApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 Jun 2025, 16:52:17 elapsed: 1:51:11) [pid: 14524]

Writable | Insert | 14 : 25 : 524

18:43 01/06/2025

api-gateway - api-gateway/pom.xml - Eclipse IDE

```

<dependency>
    <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>
<repositories>
    <repository>
        <id>spring-milestones</id>
        <name>Spring Milestones</name>
        <url>https://repo.spring.io/milestone</url>
        <snapshots>
            <enabled>false</enabled>
        </snapshots>
    </repository>
</repositories>
</project>

```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

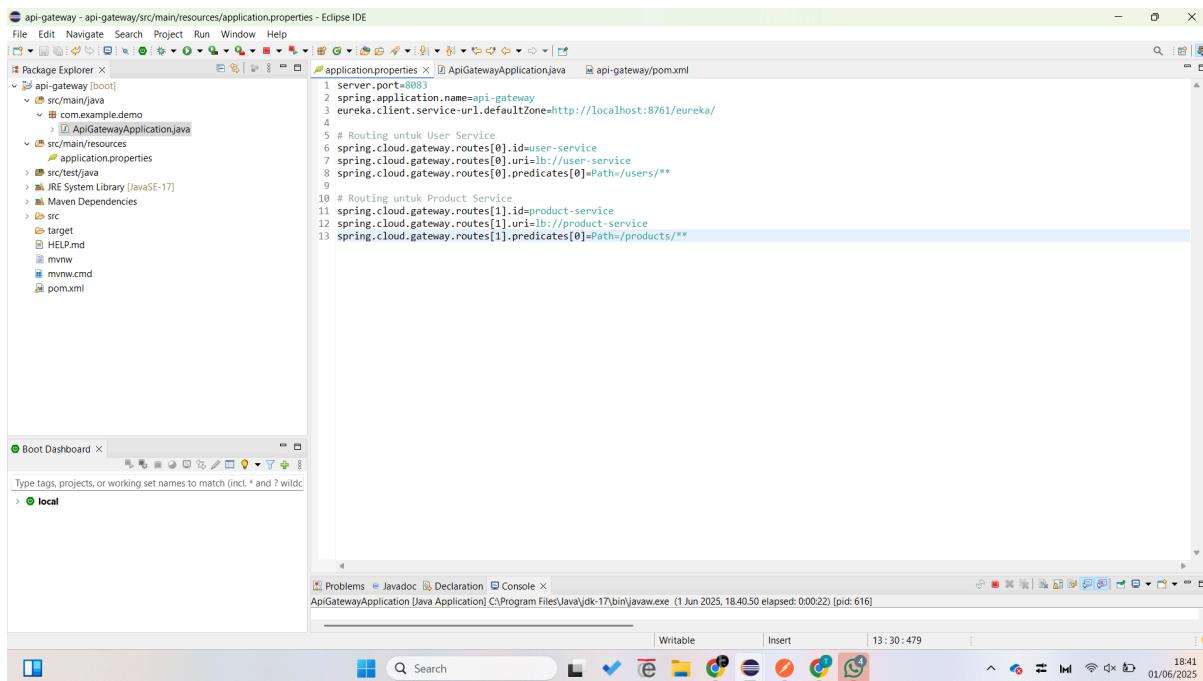
Problems | Javadoc | Declaration | Console | Debug

ApiGatewayApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 Jun 2025, 16:52:17 elapsed: 1:51:20) [pid: 14524]

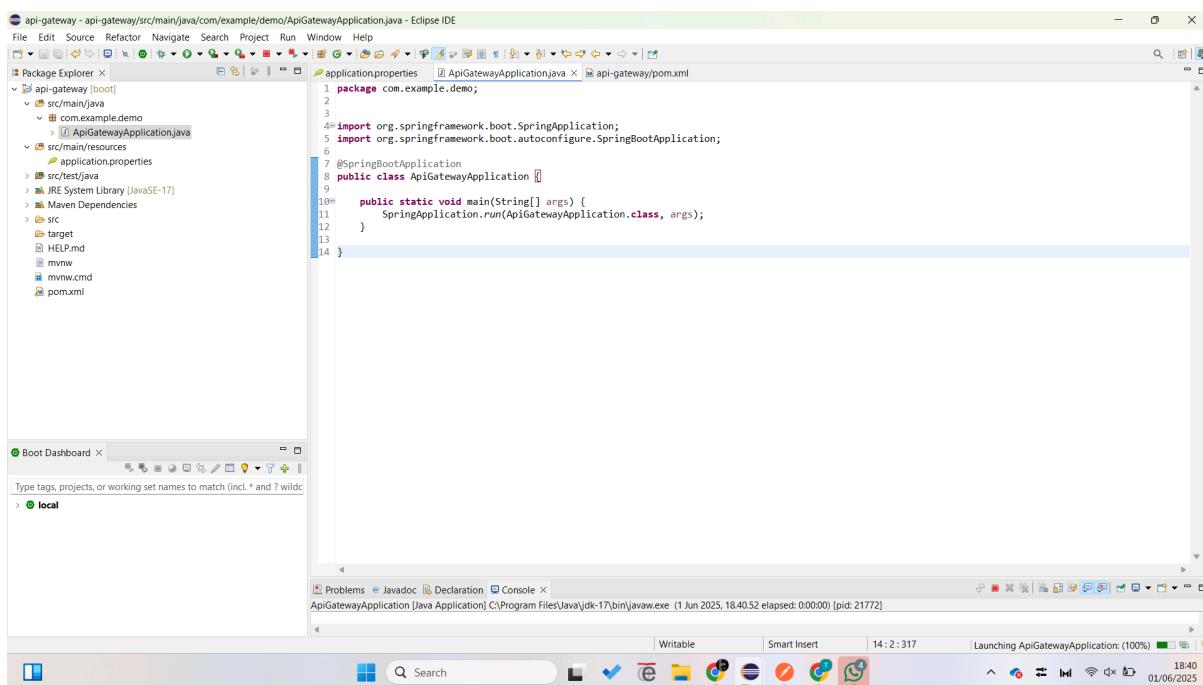
Writable | Insert | 14 : 25 : 524

18:43 01/06/2025

application.properties



## ApiGatewayApplication.java



Buka browser: <http://localhost:8761>

The screenshot shows the Spring Eureka dashboard at [localhost:8761](http://localhost:8761). The top navigation bar includes tabs for 'HOME' and 'LAST 1000 SINCE STARTUP'. Below the header, there's a 'System Status' section with environment details like 'Environment: test', 'Data center: default', and system metrics such as 'Uptime: 00:10', 'Lease expiration enabled: true', and 'Renews threshold: 6'. The 'DS Replicas' section lists registered instances: 'API-GATEWAY' (UP), 'PRODUCT-SERVICE' (UP), and 'USER-SERVICE' (UP). A 'General Info' section provides a summary of the cluster.

GET <http://localhost:8083/users>

The screenshot shows a Postman collection named 'Mylan's Workspace' containing a 'GET /users' request. The request URL is <http://localhost:8083/users>. The response body is a JSON array of three user objects:

```

1 [
2   {
3     "id": 1,
4     "name": "John Doe",
5     "email": "john@email.com",
6     "phone": "081234567890"
7   },
8   {
9     "id": 2,
10    "name": "Jane Smith",
11    "email": "jane@email.com",
12    "phone": "081234567891"
13  },
14  {
15    "id": 3,
16    "name": "Bob Johnson",
17    "email": "bob@email.com",
18    "phone": "081234567892"
19  }
20 ]

```

GET <http://localhost:8083/users/1>

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (empty), 'Environments' (with entries: ADPL, ADPL Monolithic, KPL\_10, Microservice, TP\_10), 'Flows' (empty), and 'History' (empty). The main workspace shows a request to 'http://localhost:8083/users/1'. The method is 'GET', and the URL is 'http://localhost:8083/users/1'. Under 'Headers', there are three entries: 'Content-Type: application/json', 'Accept: application/json', and 'User-Agent: PostmanRuntime/7.29.0'. Under 'Body', the response is displayed as JSON:

```
1 {  
2   "id": 1,  
3   "name": "John Doe",  
4   "email": "john@email.com",  
5   "phone": "881234567899"  
6 }
```

The status bar at the bottom indicates '200 OK' with a response time of 35 ms and a size of 190 B. The bottom navigation bar shows various icons for Postman features like Postbot, Runner, Start Proxy, Cookies, Vault, and Trash.

GET http://localhost:8083/users/info

The screenshot shows the Postman application interface. The sidebar is identical to the previous one. The main workspace shows a request to 'http://localhost:8083/users/info'. The method is 'GET', and the URL is 'http://localhost:8083/users/info'. Under 'Headers', there are three entries: 'Content-Type: application/json', 'Accept: application/json', and 'User-Agent: PostmanRuntime/7.29.0'. Under 'Body', the response is displayed as Raw text:

```
1 User Service is running on port 8081
```

The status bar at the bottom indicates '200 OK' with a response time of 27 ms and a size of 152 B. The bottom navigation bar shows various icons for Postman features like Postbot, Runner, Start Proxy, Cookies, Vault, and Trash.

GET http://localhost:8083/products

Mylan's Workspace

HTTP http://localhost:8083/products

GET http://localhost:8083/products/

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description	Bulk Edit
Body	Cookies	Headers (3)	Test Results

200 OK 157 ms 392 B

```
1 [  
2 {  
3   "id": 1,  
4   "name": "Laptop",  
5   "description": "Gaming Laptop 16GB RAM",  
6   "price": 1.5E7,  
7   "stock": 10  
8 },  
9 {  
10   "id": 2,  
11   "name": "Mouse",  
12   "description": "Wireless Gaming Mouse",  
13   "price": 500000.0,  
14   "stock": 25  
15 },  
16 {  
17   "id": 3,  
18   "name": "Keyboard",  
19   "description": "Mechanical RGB Keyboard",  
20   "price": 1200000.0,  
21   "stock": 15  
22 }  
23 ]
```

Online Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash

18:45 01/06/2025

GET http://localhost:8083/products/1

Mylan's Workspace

HTTP http://localhost:8083/products/1

GET http://localhost:8083/products/1

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description	Bulk Edit
Body	Cookies	Headers (3)	Test Results

200 OK 53 ms 204 B

```
1 {  
2   "id": 1,  
3   "name": "Laptop",  
4   "description": "Gaming Laptop 16GB RAM",  
5   "price": 1.5E7,  
6   "stock": 10  
7 }
```

Online Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash

18:45 01/06/2025

GET http://localhost:8083/products/info

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8083/products/info`. The response status is `200 OK` with a response time of 32 ms and a body size of 155 B. The response content is: "1 Product Service is running on port 8082".

GET http://localhost:8083/users/products

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8083/users/uproducts`. The response status is `200 OK` with a response time of 16 ms and a body size of 392 B. The response content is a JSON array of three products:

```
[{"id": 1, "name": "Laptop", "description": "Gaming Laptop 16GB RAM", "price": 1.5E7, "stock": 10}, {"id": 2, "name": "Mouse", "description": "Wireless Gaming Mouse", "price": 500000.0, "stock": 25}, {"id": 3, "name": "Keyboard", "description": "Mechanical RGB Keyboard", "price": 1200000.0, "stock": 15}]
```

GET http://localhost:8083/users/1/dashboard

Home Workspaces API Network

Mylan's Workspace

New Import

Collections

Environments

Flows

History

http://localhost:8083/users/1/dashboards

GET http://localhost:8083/users/1/dashboards

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description	Bulk Edit
Body	{} JSON	Preview Visualize	200 OK 24 ms 190 B
Cookies			
Headers (3)			
Test Results			

```
1 {  
2   "id": 1,  
3   "name": "John Doe",  
4   "email": "john@email.com",  
5   "phone": "881234567890"  
6 }
```

Save Share Send

200 OK 24 ms 190 B

Postbot Runner Start Proxy Cookies Vault Trash

Search

18:47 01/06/2025

