

WEEK - 5

Microservices with Spring Boot 3 and Spring Cloud

Api-gateway

microservices-project\api-gateway\src\main\java\com\example\apigateway\ApiGatewayApplication.java

```
package com.example.apigateway;

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

@SpringBootApplication
@EnableDiscoveryClient
public class ApiGatewayApplication {

    public static void main(String[] args) {

        SpringApplication.run(ApiGatewayApplication.class, args);

    }

}
```

microservices-project\api-gateway\src\main\java\com\example\apigateway\LogFilter.java

```
package com.example.apigateway;

import org.springframework.cloud.gateway.filter.GlobalFilter;
import org.springframework.stereotype.Component;
import org.springframework.web.server.ServerWebExchange;
import reactor.core.publisher.Mono;
import org.springframework.cloud.gateway.filter.GatewayFilterChain;

@Component
public class LogFilter implements GlobalFilter {

    @Override
    public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {

        System.out.println("Request Path: " + exchange.getRequest().getURI());

    }

}
```

```

        return chain.filter(exchange);    }
    }

microservices-project\api-gateway\src\main\resources\application.properties

spring.application.name=api-gateway

server.port=9090

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

# Let Gateway discover services automatically

spring.cloud.gateway.discovery.locator.enabled=true

spring.cloud.gateway.discovery.locator.lower-case-service-id=true

spring.main.web-application-type=reactive

```

src\microservices-project\api-gateway\pom.xml

```

<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
        http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.example</groupId>
    <artifactId>api-gateway</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>
    <name>api-gateway</name>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.1.2</version>
        <relativePath />
    </parent>

```

```
<properties>
  <java.version>17</java.version>
  <spring-cloud.version>2022.0.4</spring-cloud.version>
</properties>

<dependencies>
  <!-- Spring Boot core -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter</artifactId>
  </dependency>
  <!-- Spring Cloud Gateway (Reactive) -->
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-gateway</artifactId>
  </dependency>
  <!-- Eureka Client -->
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
  </dependency>
</dependencies>

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

```

        <scope>import</scope>
    </dependency>
</dependencies>
</dependencyManagement>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

Eureka-server

microservices-project\eureka-server\src\main\java\com\example\eureka-server\EurekaServerApplication.java

```

package com.example.eureka-server;

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

@SpringBootApplication
@EnableEurekaServer

public class EurekaServerApplication {

    public static void main(String[] args) {

        SpringApplication.run(EurekaServerApplication.class, args);

    }

}

```

microservices-project\eureka-server\src\main\resources\application.properties

server.port=8761

spring.application.name=eureka-server

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

microservices-project\eureka-server\pom.xml

```
<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
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>eureka-server</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>eureka-server</name>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.1.2</version> <!-- Match this with your Spring Boot version -->
    <relativePath/>
  </parent>
  <properties>
    <java.version>17</java.version>
    <spring-cloud.version>2022.0.4</spring-cloud.version>
  </properties>
  <dependencies>
    <!-- Spring Boot Core -->
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter</artifactId>
</dependency>
<!-- Eureka Discovery Server -->
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
</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>

Greet-service

microservices-project\greet-service\src\main\java\com\example\greet-service\GreetController.java

```
package com.example.greet-service;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController

public class GreetController {

    @GetMapping("/greet")
    public String greet() {
        return "Hello World";
    }
}
```

microservices-project\greet-service\src\main\java\com\example\greet-service\GreetServiceApplication.java

```
package com.example.greet-service;

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

@SpringBootApplication
@EnableDiscoveryClient

public class GreetServiceApplication {

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

microservices-project\greet-service\src\main\resources\application.properties

spring.application.name=greet-service
server.port=8082
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true

microservices-project\greet-service\pom.xml

```
<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
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>greet-service</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>greet-service</name>
  <description>Greet Microservice with Eureka Client</description>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.1.2</version>
    <relativePath />
  </parent>
  <properties>
    <java.version>17</java.version>
    <spring-cloud.version>2022.0.4</spring-cloud.version>
  </properties>
  <dependencies>
```



```
<!-- Spring Boot Core -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter</artifactId>
</dependency>
<!-- Spring Boot Web -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<!-- Eureka Client -->
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</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>
```

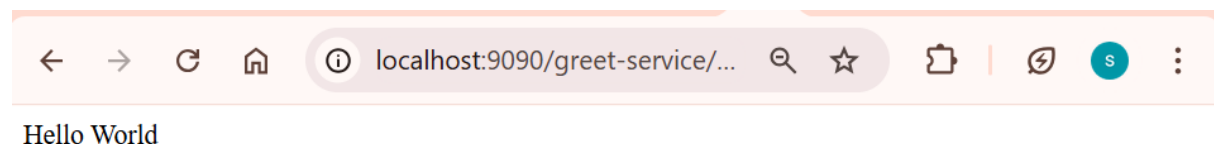
```
</plugin>
```

```
</plugins>
```

```
</build>
```

```
</project>
```

OUTPUT :



System Status

Environment	test	Current time	2025-07-19T22:15:07 +0530
Data center	default	Uptime	00:15
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	8

DS Replicas

[localhost](#)

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - DESKTOP-F3H8O64:api-gateway:9090
GREET-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-F3H8O64:greet-service:8082

General Info

← → ↺ ↻	localhost:8761	🔍 ☆ 🗂 🔄 ⚙
GREET-SERVICE	n/a (1)	(1) UP (1) - DESKTOP-F3H8O64:greet-service:8082

General Info

Name	Value
total-avail-memory	83mb
num-of-cpus	8
current-memory-usage	68mb (81%)
server-uptime	00:15
registered-replicas	http://localhost:8761/eureka/
unavailable-replicas	http://localhost:8761/eureka/
available-replicas	

Instance Info

Name	Value
ipAddr	192.168.206.7
status	UP

The screenshot shows an IDE with the following components:

- EXPLORER:** Displays the project structure for 'microservices project'. The 'greet-service' directory is expanded, showing 'src/main/resources' and 'pom.xml'.
- EDITOR:** Shows the 'application.properties' file for 'greet-service' with the following content:

```
1 spring.application.name=greet-service
2 server.port=8082
3
4 eureka.client.service-url.defaultZone=http://localhost:8761/eureka
5 eureka.client.register-with-eureka=true
6 eureka.client.fetch-registry=true
7
```
- PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS:** The 'TERMINAL' tab is active, showing the output of the 'greet-service' application. The logs indicate that the application is successfully registering with Eureka and initializing the Spring DispatcherServlet.

The terminal output includes the following log entries:

```
2025-07-19T22:08:32.107+05:30 INFO 28556 --- [main] o.s.c.n.e.s.EurekaServiceRegistry : Registering application GR
EET-SERVICE with eureka with status UP
2025-07-19T22:08:32.108+05:30 INFO 28556 --- [main] com.netflix.discovery.DiscoveryClient : Saw local status change ev
ent StatusChangeEvent [t2025-07-19T22:08:32.177+05:30 INFO 28556 --- [nfoReplicator-0] com.netflix.discovery.DiscoveryClient : Di
scoveryClient.GREET-SERVICE/DESKTOP-F3H8O64:greet-service:8082 - registration status: 204
2025-07-19T22:13:00.998+05:30 INFO 28556 --- [nio-8082-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring Dispat
2025-07-19T22:08:32.177+05:30 INFO 28556 --- [nfoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient.GREET-SERV
ICE/DESKTOP-F3H8O64:greet-service:8082 - registration status: 204
2025-07-19T22:13:00.998+05:30 INFO 28556 --- [nio-8082-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring Dispat
cherServlet 'dispatcherServlet'
2025-07-19T22:13:00.999+05:30 INFO 28556 --- [nio-8082-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'disp
atcherServlet'
2025-07-19T22:13:01.003+05:30 INFO 28556 --- [nio-8082-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization i
n 3 ms
2025-07-19T22:13:31.570+05:30 INFO 28556 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints
via configuration
```