**Create Eureka Discovery Server and register**

**microservices**

# Java Code

## EurekaDiscoveryServerApplication.java

@SpringBootApplication  
@EnableEurekaServer  
public class EurekaDiscoveryServerApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(EurekaDiscoveryServerApplication.class, args);  
 }  
}

## AccountServiceApplication.java

@EnableDiscoveryClient  
@SpringBootApplication  
public class AccountServiceApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(AccountServiceApplication.class, args);  
 }  
}

## LoanServiceApplication.java

@EnableDiscoveryClient  
@SpringBootApplication  
public class LoanServiceApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(LoanServiceApplication.class, args);  
 }  
}

## GreetServiceApplication.java

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

## controller/GreetController.java

@RestController  
public class GreetController {  
 @GetMapping("/greet")  
 public String greet() {  
 return "Hello World";  
 }  
}

## ApiGatewayApplication.java

@EnableDiscoveryClient  
@SpringBootApplication  
public class ApiGatewayApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(ApiGatewayApplication.class, args);  
 }  
}

## LogFilter.java

@Component  
public class LogFilter implements GlobalFilter, Ordered {  
 private static final Logger log = LoggerFactory.getLogger(LogFilter.class);  
  
 @Override  
 public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {  
 log.info("Incoming request → {}", exchange.getRequest().getURI());  
 return chain.filter(exchange);  
 }  
  
 @Override  
 public int getOrder() {  
 return -1; // highest precedence  
 }  
}

# Configuration Files

## application.properties (Eureka Discovery Server)

server.port=8761  
eureka.client.register-with-eureka=false  
eureka.client.fetch-registry=false  
logging.level.com.netflix.eureka=OFF  
logging.level.com.netflix.discovery=OFF

## application.properties (Account Service)

server.port=8080  
spring.application.name=account-service  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

## application.properties (Loan Service)

server.port=8081  
spring.application.name=loan-service  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka

## application.properties (Greet Service)

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

## application.properties (API Gateway)

server.port=9090  
spring.application.name=api-gateway  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka  
spring.cloud.gateway.discovery.locator.enabled=true  
spring.cloud.gateway.discovery.locator.lower-case-service-id=true

## pom.xml (Eureka Client Dependency)

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

# Output







