**WEEK 5: ADDITIONAL HANDS-ON**

**2. Microservices with API gateway**

**Create Eureka Discovery Server and register**

SOLUTION:

**Eureka Discovery Server Setup**

Use [Spring Initializr](https://start.spring.io/):

* **Group**: com.cognizant
* **Artifact**: eureka-discovery-server
* **Dependency**: Spring Cloud Discovery → *Eureka 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

In main class:

@EnableEurekaServer

@SpringBootApplication

public class EurekaDiscoveryServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaDiscoveryServerApplication.class, args);

}

}

Use [Spring Initializr](https://start.spring.io/):

* **Group**: com.cognizant
* **Artifact**: account
* **Dependencies**:
  + Spring Web
  + Spring Boot DevTools
  + Eureka Discovery Client

spring.application.name=account-service

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

In main class:

@EnableDiscoveryClient

@SpringBootApplication

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

}

@RestController

public class GreetController {

@GetMapping("/greet")

public String greet() {

return "Hello World";

}

}

Configure application.properties:

spring.application.name=greet-service

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

spring.application.name=api-gateway

server.port=9090

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

**LogFilter.java**

@Component

public class LogFilter implements GlobalFilter, Ordered {

private static final Logger logger = LoggerFactory.getLogger(LogFilter.class);

@Override

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

logger.info("Incoming request: {}", exchange.getRequest().getURI());

return chain.filter(exchange);

}

@Override

public int getOrder() {

return -1;

}

}