# 1. Account Microservice

## AccountController.java

@RestController  
@RequestMapping("/accounts")  
public class AccountController {  
  
 @GetMapping("/{number}")  
 public Map<String, Object> getAccount(@PathVariable String number) {  
 Map<String, Object> response = new HashMap<>();  
 response.put("number", number);  
 response.put("type", "savings");  
 response.put("balance", 234343);  
 return response;  
 }  
}

## application.properties

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

## Main Class

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

# 2. Loan Microservice

## LoanController.java

@RestController  
@RequestMapping("/loans")  
public class LoanController {  
  
 @GetMapping("/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 Map<String, Object> response = new HashMap<>();  
 response.put("number", number);  
 response.put("type", "car");  
 response.put("loan", 400000);  
 response.put("emi", 3258);  
 response.put("tenure", 18);  
 return response;  
 }  
}

## application.properties

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

## Main Class

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

## application.properties

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

## Main Class

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

# 4. Greet Microservice

## GreetController.java

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

## application.properties

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

## Main Class

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

# 5. API Gateway

## application.properties

server.port=9090  
spring.application.name=api-gateway  
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka/  
spring.cloud.gateway.routes[0].id=greet-service  
spring.cloud.gateway.routes[0].uri=lb://greet-service  
spring.cloud.gateway.routes[0].predicates[0]=Path=/greet-service/\*\*  
spring.cloud.gateway.discovery.locator.enabled=true  
spring.cloud.gateway.discovery.locator.lowerCaseServiceId=true

## Main Class

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

# 6. Global Filter

## LogFilter.java

@Component  
public class LogFilter implements GlobalFilter {  
  
 private static final Logger logger = LoggerFactory.getLogger(LogFilter.class);  
  
 @Override  
 public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {  
 logger.info("Request received: " + exchange.getRequest().getURI());  
 return chain.filter(exchange);  
 }  
}