**Spring 2025 – SWER007**

**Lab 11: Inter-Service Communication with OpenFeign**

**Objective**

Integrate inter-service communication using **OpenFeign** and containerize all services with **Docker Compose**. This finalizes the architecture so everything can be launched and tested with a single command.

**🔧 Prerequisites**

You should have:

* employee-service, department-service, auth-service, api-gateway, and discovery-service running
* Services registered in Eureka and secured with JWT

**Add OpenFeign to Employee Service**

OpenFeign simplifies REST calls between services.

**1 Add to pom.xml:**

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-openfeign</artifactId>

</dependency>

**2 Enable Feign:**

@EnableFeignClients

@SpringBootApplication

public class EmployeeServiceApplication { }

**3 Create a Feign client to call Department Service:**

@FeignClient(name = "department-service")

public interface DepartmentClient {

@GetMapping("/departments/{id}")

DepartmentDto getDepartment(@PathVariable("id") Long id);

}

**4 Use this client in EmployeeServiceImpl to enrich responses with department info.**

@Autowired

private DepartmentClient departmentClient;

In method newEmployee and save employee.

 // check if the department exists by calling department service

        try {

            DepartmentDto departmentDto = departmentClient.one(newEmployee.getDepartmentId()).getContent();

        } catch (FeignException e) {

            log.error("Error calling department-service: {}", e.getMessage());

            // Handle the case when the department service is unavailable or the department is not found

            if (e.status() == 404) {

                return ResponseEntity.status(HttpStatus.NOT\_FOUND).body("Department not found");

            }

            return ResponseEntity.status(HttpStatus.SERVICE\_UNAVAILABLE).body("Department service is unavailable");

        }