**Spring 2025 – SWER007**

**Lab 12: Centralized Configuration using Spring Cloud Config**

**✅ Objective**

Set up a **centralized configuration server** using **Spring Cloud Config** to manage all microservices' properties from a single Git-backed source.

**🔧 Prerequisites**

You should have:

* employee-service, department-service, auth-service, api-gateway, and discovery-service running
* A GitHub or local Git repo ready to store configuration files

**1️ Create a Spring Cloud Config Server**

Use Spring Initializr or your IDE:

spring init --dependencies=cloud-config-server config-service

**Add to ConfigServiceApplication.java:**

@EnableConfigServer

@SpringBootApplication

public class ConfigServiceApplication {

public static void main(String[] args) {

SpringApplication.run(ConfigServiceApplication.class, args);

}

}

**Example application.yml:**

server:

port: 8888

spring:

application:

name: config-service

cloud:

config:

server:

git:

uri: https://github.com/your-org/your-config-repo

✅ Make sure your Git repo contains property files like:

* employee-service.yml
* department-service.yml
* auth-service.yml
* api-gateway.yml

**2️ Update Client Services to Use Config Server**

In each service's bootstrap.yml (create if missing):

spring:

application:

name: employee-service # or department-service, etc.

cloud:

config:

uri: http://localhost:8888

# Optional Eureka config here

Remove most content from the service's application.yml — it will be fetched from Git now.

**3️ Config Repo Structure**

In your Git repo, create service-specific files:

**employee-service.yml**:

server:

port: 8081

spring:

application:

name: employee-service

Repeat for other services.

✅ Push the repo to GitHub and test with:

http://localhost:8888/employee-service/default

**4️ Restart Services and Test**

Start config-service first, then the rest.

✅ Each service should fetch its config from the Git repo.

Use logs to verify config is loaded from the remote location.

**📝 Lab Tasks**

1. Create config-service project and enable Git-backed config
2. Set up a public or private Git repo with service configs
3. Update each service to load config via bootstrap.yml
4. Restart services and verify centralized config loading

**🎯 Expected Outcome**

* All services fetch their configs from a single Git repo
* Config changes can be made centrally without touching service code