

초격차 패키지 Online.

아는 사람만 아는 Boot 기능

PART1 | Spring Native [Experimental]

더 빠른 Boot

PART2 | Testcontainers

Docker의 활용

PART3 | Config Server

외부 설정을 전담하는 독립적인 서버를 만들어보자

PART4 | Config Client

외부 설정 서버에 접근하기

아는 사람만 아는 Boot 기능

3 Config Server

설정값을 관리하는 방법

3. Config Server

서비스 로직과 독립적인 외부 설정값의 관리 방법

- 소스 코드 안에 삽입
 - 설정값이 소스 코드에 노출됨
 - 설정을 바꾸고 싶으면 빌드/배포 해야 함
- 외부 설정 파일 (*.properties)
 - 좀 나은 방법, 설정값을 덮어쓸 수 있는 방법이 존재 (ex: command line argument)
 - 패키징 되어 있기 때문에, 변경점을 확정하면 결국 빌드/배포
- DB에 관리
 - 설정을 바꾸고 싶으면 DB값을 update
 - 설정값이 몇 개 없는데 테이블 설계하긴 애매..?

Spring Cloud Config

3. Config Server

외부 환경 설정을 관리하는 독립적인 서버

- 설정을 바꾸고 싶으면 파일 내용만 바꾸면 끝
- git repository 지원
- 분산 환경에서 다양한 스프링 부트 애플리케이션이 접근하기 용이

Spring Cloud Config: 사용

3. Config Server

- application.properties 에 설정 파일을 저장하는 git repository 등록
 - spring.cloud.config.server.git.uri=https://github.com/{my}/{project}
- properties endpoint
 - /{application}/{profile}[/{label}]
 - /{application}-{profile}.yml
 - /{label}/{application}-{profile}.yml
 - /{application}-{profile}.properties
 - /{label}/{application}-{profile}.properties
- endpoint 설명
 - application: 스프링 부트 앱 이름 (spring.config.name) (default: application)
 - profile: 스프링 부트 프로파일 (ex: alpha, beta, real, test, ...)
 - label: git 브랜치명 (default: master)

실습

3.
Config Server

Reference

3. Config Server

- <https://cloud.spring.io/spring-cloud-config/reference/html/>