



# E-commerce - Catalog Service & Order Service

🕒 작성 일시	@2022년 11월 22일 오후 10:53
🕒 최종 편집 일시	@2022년 12월 3일 오후 4:54
📌 유형	Spring Cloud로 개발하는 마이크로서비스 애플리케이션(MSA)
👤 작성자	
👥 참석자	

## Catalog Service

[목표](#)

[API](#)

[프로젝트 생성](#)

[프로젝트 내용 추가](#)

## Order Service

[목표](#)

[API](#)

[프로젝트 생성](#)

## Catalog Service

### 목표

- 상품 목록 조회

### API

기능	URL (API Gateway)	URL (API Gateway 미사용)	HTTP Method
상품 목록 조회	/catalog-service/catalogs	/catalogs	GET

### 프로젝트 생성

(강의는 H2 를 사용했지만 나는 postgresql 함으로 dependency 에 postgresql driver 를 추가한다.)

**Project**

Language: ☒ Java ☐ Kotlin ☐ Groovy

Spring Boot: ☐ 3.0.1 (SNAPSHOT) ☐ 3.0.0 ☒ 2.7.7 (SNAPSHOT) ☐ 2.7.6

**Project Metadata**

Group:

Artifact:

Name:

Description:

Package name:

Packaging: ☒ Jar ☐ War

Java: ☐ 19 ☐ 17 ☒ 11 ☐ 8

**Dependencies**

- Lombok** (DEVELOPER TOOLS): Java annotation library which helps to reduce boilerplate code.
- Spring Web** (WEB): Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
- Spring Data JPA** (SQL): Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
- PostgreSQL Driver** (SQL): A JDBC and R2DBC driver that allows Java programs to connect to a PostgreSQL database using standard, database independent Java code.
- Eureka Discovery Client** (SPRING CLOUD DISCOVERY): A REST based service for locating services for the purpose of load balancing and failover of middle-tier servers.
- Spring Boot DevTools** (DEVELOPER TOOLS): Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

## 프로젝트 내용 추가

- 데이터 적재
  - src\main\resources\data.sql 에 insert 될 내용을 작성한다.
- yml(user-service 에서 아래 내용 추가)

```
jpa:
  hibernate:
    ddl-auto: create-drop # SessionFactory 생성시 DDL 적용, 제거시 drop 즉, 메모리에 테이블 내용이 담긴다는 뜻.
    generate-ddl: true # DDL 문을 생성하여 DB 적용하라는 뜻
    ### Spring boot 2.5 이상 부터 아래 properties 가 있어야 data.sql 를 실행함.
    defer-datasource-initialization: true
  sql:
    init:
      mode: always
    ###
```

- catalog 테이블
  - 해당 데이터를 외부 자바 시스템에서도 사용할 수 있도록 byte형태로 변환하는 기술로서, DB에 데이터를 저장하기 용이하기 위해, Serializable 을 impl 한다.

```
public class CatalogEntity implements Serializable {  
  
}
```

- 나머지 소스는 git 에서 확인 (User-service 와 크게 다를 것이 없음)
  - VO, DTO, Service, Repository, Controller 등

## Order Service

### 목표

- 사용자 별 상품 주문
- 사용자 별 주문 내역 조회

### API

기능	URL (API Gateway)	URL (API Gateway 미사용)	HTTP Method
사용자 별 상품 주문	/order-service/{user_id}/orders	/orders	POST
사용자 별 주문 내역 조회	/order-service/{user_id}/orders	/orders	GET

### 프로젝트 생성

(강의는 H2 를 사용했지만 나는 postgresql 함으로 dependency 에 postgresql driver 를 추가한다.)

Project

☐ Gradle - Groovy
☐ Gradle - Kotlin
☒ Java
☐ Kotlin
☐ Groovy

Maven

Spring Boot

☐ 3.0.1 (SNAPSHOT)
☐ 3.0.0
☐ 2.7.7 (SNAPSHOT)
☒ 2.7.6

Project Metadata

Group

com.example

Artifact

order-service

Name

order-service

Description

Demo project for Spring Boot

Package name

com.example.order-service

Packaging

☒ Jar
☐ War

Java

☐ 19
☐ 17
☒ 11
☐ 8

Dependencies

ADD DEPENDENCIES... CTRL + B

Lombok

DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

Spring Boot DevTools

DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

PostgreSQL Driver

SQL

A JDBC and R2DBC driver that allows Java programs to connect to a PostgreSQL database using standard, database independent Java code.

Eureka Discovery Client

SPRING CLOUD DISCOVERY

A REST based service for locating services for the purpose of load balancing and failover of middle-tier servers.

Spring Data JPA

SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

- 내용은 user-service, catalog-service 와 비슷한 CRUD 로직임으로, git 에서 확인하도록 한다.
- model mapper 를 웬만하면 STRICT 로 설정하고 적용하는게 좋다. (default 값인 STANDARD로 하면 mapping error가 나올 수 있다.)
- <https://devwithpug.github.io/java/java-modelmapper/>