

초격차 패키지 Online.

복잡한 쿼리의 작성과 응용

PART1 | Querydsl vs. Jooq

Querydsl 과 Jooq 기본 살펴보기

PART2 | Querydsl 활용

Querydsl + Spring Boot 2.5

PART3 | Jooq 활용

Jooq, 정말 쓸만할까요?

PART4 | eager fetch, lazy fetch, N+1 문제

JPA 사용하면서 가장 흔히 겪는 문제 1탄

PART5 | 순환 참조 문제

JPA 사용하면서 가장 흔히 겪는 문제 2탄

복잡한 쿼리의 작성과 응용

3 Jooq 활용

Jooq

3. Jooq 활용

Jooq

테이블 스키마로부터 자바 코드를 만들어주는 라이브러리

- 시스템의 설계가 자바 코드(엔티티)가 아닌, DB 에서 시작될 때 유용한 구조

Jooq

3. Jooq 활용

Jooq, 좋은 이유

- Jooq 방식으로 사용한다면, 엔티티를 작성할 필요가 없다
- 로그가 보기 예뻐
 - query 로그 안에 binding parameter 가 함께 포함됨
- 결과 로그가 예뻐

Jooq

3. Jooq 활용

Jooq, 불편한 이유

ORM 기술이 아니기 때문에, Spring Data JPA 와 결이 잘 안 맞음

- JPA 와 정반대의 매커니즘: Jooq 클래스가 엔티티 클래스를 방해
- JPA 기술이 아니어서 오는 문제점
 - Spring Data JPA 트랜잭션 연동이 힘들: 자체 트랜잭션 기술 -> Jooq 코드가 서비스에 노출
 - 하이버네이트 auto-ddl 사용 불가, DB 스키마는 이미 준비되어 있어야 함
- 스프링과 연동되어있지 않아서 오는 불편함
 - 스프링 Pageable 정보로부터 Jooq 쿼리를 조립하기가 상당히 까다로움
- gradle 플러그인을 쓸 경우, DB 정보가 build.gradle 에 침투함
 - DB 기본 정보, 로깅, 타입 변환(enum 등) 정보, 패키지 정보, DB dialect, JDBC 드라이버 정보...
- 그리고 매뉴얼도 꽤 어려움.. 레퍼런스도 적고..

Jooq

3. Jooq 활용

스프링이 초기에 밀어줬지만 Querydsl 에 밀리는 또 하나의 이유

유료.

- Open Source 플랜만 무료
 - open source DB dialect 지원: derby, firebird, h2, hsqldb, ignite, mariadb, mysql, postgresql, sqlite
 - 지원 안되는 대표적인 DB: ms access, oracle, sql server, aurora, bigquery, db2, snowflake 등
- Express, Professional, Enterprise 플랜에 따라 단계별 과금

Reference

3. Jooq 활용

- <https://www.jooq.org/doc/3.14/manual/>
- <https://github.com/etiennestuder/gradle-jooq-plugin>
- https://github.com/etiennestuder/gradle-jooq-plugin/tree/master/example/configure_jooq_version_from_spring_boot