

초격차 패키지 Online.

Spring Data JPA 와 테크닉

PART1 | @Repository

Spring Data JPA 의 시작점

PART2 | @Entity 디자인

객체를 정의하고 DB 데이터와 연결짓는 테크닉

PART3 | DataSource, TransactionManager

Spring Data JPA 기반 기술 정리

PART4 | DB 바꾸기 (MySQL -> PostgreSQL)

정말 특정 DB에 종속되지 않았을까?

PART5 | JPA 테스트

Repository 테스트 방법 살펴보기

Spring Data JPA 와 테크닉

1 @Repository

@Repository

1. @Repository

스프링 스테레오타입 애노테이션

persistence layer 를 구현하는 클래스에 사용

- @Component 와 마찬가지로 해당 클래스를 빈으로 등록
- DAO 패턴을 적용한 클래스에도 사용 가능
- persistence layer 에서 발생하는 예외를 잡아서 DataAccessException 으로 처리해줌
 - PersistenceExceptionTranslationPostProcessor
- Spring Data JPA 를 사용한다면, "직접 사용할 일은 없다" 고 봐도 무방함

@Repository

1.

@Repository

```
7 @NoRepositoryBean
8 public interface JpaRepository<T, ID> extends PagingAndSortingRepository<T, ID>, QuerydslPredicateSupportingRepository<T> {
9     // ...
10     /*
11      * (non-Javadoc)
12      * @see org.springframework.data.repository.CrudRepository#findAll()
13      */
14     @Override
15     List<T> findAll();
16     // ...
17 }

org.springframework.data.jpa.repository.support
@Repository
@Transactional(readOnly = true)
public class SimpleJpaRepository<T, ID>
extends Object
implements org.springframework.data.jpa.repository.support.JpaRepositoryImplementation<T, ID>

Default implementation of the org.springframework.data.repository.CrudRepository
interface. This will offer you a more sophisticated interface than the plain
EntityManager .
```

```
Default implementation of the org.springframework.data.repository.
CrudRepository interface. This will offer you a more sophisticated
interface than the plain EntityManager .

Author:      Oliver Gierke, Eberhard Wolff, Thomas Darimont,
             Mark Paluch, Christoph Strobl, Stefan Fussenegger,
             Jens Schauder, David Madden, Moritz Becker, Sander
             Krabbenborg, Jesse Wouters

Type parameters: <T> – the type of the entity to handle
                 <ID> – the type of the entity's identifier

@Repository
@Transactional(readOnly = true)
public class SimpleJpaRepository<T, ID> implements JpaRepositoryImplementation<T, ID> {
```

Spring Data JPA 와 주요 인터페이스들

1. @Repository

Spring Data JPA 인터페이스

단계별로 필요한 기능까지만 사용 가능

- Repository: 기본 repository 인터페이스. 어떤 메소드도 제공하지 않음
- CrudRepository: Repository + CRUD 기능 제공
- PagingAndSortingRepository: CrudRepository + 페이징, 정렬 기능 제공
- JpaRepository: PagingAndSortingRepository + Spring Data JPA repository 전체 기능

Query method

1.
@Repository

인터페이스에 작성한 메소드 이름이 곧 쿼리 표현이 됨

- ex: `List<Event> findByEventStatusAndEventNameOrCapacity(String eventStatus, String eventName, Integer capacity);`
- 다이나믹 쿼리를 만들 수는 없음
- 사용 가능한 키워드
 - `distinct, and, or, is, not, between, lessThan, lessThanEqual, greaterThen, greaterThenEqual`
 - `null, isNotNull, like, startingWith, endingWith, containing, orderBy, in, true, false, ignoreCase`
- <https://docs.spring.io/spring-data/jpa/docs/2.5.5/reference/html/#jpa.query-methods.query-creation>
- join 등 복잡한 표현은 불가

몇가지 애노테이션들

1. @Repository

- @Param: 쿼리 메소드 입력 파라미터에 사용하여 애노테이션 기반 파라미터 바인딩할 때 사용
- @Query: 직접 JPQL을 작성하고 싶을 때 사용
- @NoRepositoryBean: 빈으로 등록하고 싶지 않은 인터페이스를 지정할 수 있음
 - 특정 쿼리 메소드를 기본 메소드로 지정하는 방식으로 운영 가능
 - 특정 메소드를 선택적으로 사용하거나 api에 노출하고자 할 때도 사용하는 테크닉

Reference

1.
@Repository

- <https://docs.spring.io/spring-data/jpa/docs/2.5.5/reference/html/#reference>
- <https://spring.io/guides/gs/accessing-data-jpa/>