

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

## 2 @Entity 디자인

## 엔티티 클래스 애노테이션

데이터베이스에 저장(persist)할 자바 객체를 정의

- 다양한 애노테이션을 이용해 보다 자세한 테이블 스키마 정보를 표현
- 애노테이션으로 표현한 스키마 정보와 실제 테이블 스키마가 완벽히 일치해야 할 필요는 없음
- 하나의 도메인(domain) 으로 간주

## @Entity: JPA 애노테이션

## 2. @Entity 디자인

@Entity 클래스 안에서 사용되는 주요 JPA 애노테이션

- @Table, @Index, @UniqueConstraint: 테이블 기본 정보와 인덱스, unique 키를 설정
- @Id, @GeneratedValue: primary key 설정
- @Column: 각 컬럼 설정
- @Enumerated: enum 을 처리하는 방법을 설정
- @Transient: 특정 필드를 DB 영속 대상에서 제외
- @OneToOne, @OneToMany, @ManyToOne, @ManyToMany: 연관 관계 설정
- @MappedSuperClass: 상속을 이용한 공통 필드 정의
- @Embedded, @Embeddable: 클래스 멤버를 이용한 공통 필드 정의
- @DateFormat: 스프링에서 제공하는 애노테이션, 날짜 입력의 포맷을 지정

## @Entity: JPA 엔티티의 lifecycle event 를 활용한 Auditing 테크닉

### 2. @Entity 디자인

JPA 엔티티에 생성일시, 수정일시 같이 일정하게 작성하는 메타데이터를 처리 가능

- @PrePersist
- @PostPersist
- @PreRemove
- @PostRemove
- @PreUpdate
- @PostUpdate
- @PostLoad

## @Entity: Spring JPA Auditing 애노테이션

## 2. @Entity 디자인

엔티티의 생성일시, 수정일시, 생성자, 수정자를 자동으로 관리해주는 애노테이션

- 설정
  - @EnableJpaAuditing
  - @EntityListeners(AuditingEntityListener.class)
- 활용
  - @CreatedBy
  - @CreatedDate
  - @LastModifiedBy
  - @ LastModifiedDate

## Reference

2.  
@Entity 디자인

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