

섹션 3의 학습 우선 순위

✓ 설명

프로젝트를 남겨둔 시점에서의 학습 우선 순위를 위에서부터 아래대로 순서대로 나열해 보았습니다.

★ **무엇부터 학습 해야할지 도저히 모르겠다고 생각된다면 이 문서를 참고하세요.**

- ★ **필수:** 프로젝트 병행하면서 필수로 학습을 해야됨을 의미합니다.
- **선택 사항:** 프로젝트 때 학습해서 적용하면 좋지만 필수는 아닌 내용을 의미합니다.
- **프로젝트 이 후:** 프로젝트 때 반드시 학습하지 않아도 프로젝트를 진행할 수 있음을 의미합니다.

✓ 필수

1 API 계층 1차 구현

- Controller 1차 구현
 - @RestController
 - 애너테이션 사용 방법
 - @RequestMapping 애너테이션 사용 방법
 - @PostMapping, @PatchMapping, @GetMapping 애너테이션 사용 방법
 - 핸들러 메서드 작성 방법
 - @RequestBody 애너테이션을 이용해 DTO 객체를 파라미터로 전달 받는 방법
 - DTO 클래스 작성
 - @Getter와 @Setter의 사용방법
 - JSON 문자열과 DTO 클래스 필드의 매핑 개념

- Sub 데이터베이스를 이용하여 API 호출 테스트가 가능하게 하는 방법
 - Github Repository
 - [be-reference-api-documentation](#)

2 Service 계층 1차 구현

- Service 클래스 및 메서드 생성 방법
- @Service 애너테이션 사용 방법

3 API 계층 2차 구현

- Controller의 2차 구현
 - Service 클래스를 DI 받는 방법
 - Service 클래스의 메서드 호출 방법
 - Mapstruct를 이용한 Mapper 적용 방법
 - DTO와 Entity 간의 상호 변환 방법

4 데이터 액세스 계층 1차 구현

- JpaRepository를 상속하는 Repository 작성 방법

5 Service 계층 2차 구현

- Service 클래스에 Repository DI 받는 방법
- Service 클래스의 비즈니스 로직 구현
- 데이터 액세스 계층의 Repository의 메서드를 호출 방법
- @Transactional 애너테이션을 이용한 트랜잭션 처리 방법

6 API 계층 3차 구현

- DTO 유효성 검증 애너테이션을 이용한 유효성 검증 방법

7 Service 계층 3차 구현

- 비즈니스 로직에 대한 예외 throw 로직 구현 방법

8 데이터 액세스 계층 2차 구현

- 엔티티 컬럼 매핑 방법
- 엔티티 간의 연관관계 매핑 방법

9 예외 처리

- GlobalExceptionHandler 구현
- @RestControllerAdvice 애너테이션을 이용한 Exception 처리 방법
- ErrorResponse 구현 방법

10 애플리케이션 빌드/실행/배포

- application-xxxx.yml을 이용한 프로파일 적용

✓ 선택 사항

- 선택 사항에 있는 내용은 **프로젝트 때까지 역량이 안된다면 역량이 되는 팀원에게 부탁**해야 합니다.
- ★ **다른 팀원이 구현하는 동안 여러분은 학습을 병행해야 합니다.**
 - 슬라이스 테스트 케이스 작성 방법
 - API 문서화 로직 작성 방법
 - Rest Client 사용 방법
 - 프로젝트 요구 사항에 따라 다르겠지만 외부 API 연동이 필요하다면 학습이 필요함니다.
 - 파일 업로드
 - Github Repository
 - [be-reference-file-upload](#)
 - 이메일 전송
 - Github Repository
 - [be-reference-send-email](#)

✓ 프로젝트 이 후

- 프로젝트 이 후 더 학습해야 될 내용입니다.

1 서비스 계층

- Mapstruct
 - @Mapping 애너테이션을 이용한 매핑 방법
 1. <https://mapstruct.org/>

2 데이터 액세스 계층

- Spring Data JDBC

- JPA
 - Github Repository
 - [be-reference-jpa](#)
 - 학습 내용
 - Fetch 전략
 - N + 1 문제
- Spring Data JPA
 - Github Repository
 - [be-reference-spring-data-jpa](#)
 - [be-solution-jpa](#)
 - 학습 내용
 - Value Object 사용
 - static 멤버 클래스를 이용한 DTO 클래스 리팩토링

3 트랜잭션

- Github Repository
 - [be-reference-tx](#)
 - [be-reference-distributed-tx](#)
- 학습 내용
 - 분산 트랜잭션
 - AOP를 이용한 트랜잭션 적용
 - Application Event Publishing 사용법
 - Proxy
 - Dynamic Proxy
 - Reflection
 - ProxyFactoryBean

4 테스트

- Github Repository
 - [be-reference-testing](#)
- 학습 내용
 - 단위 테스트 방법
 - 슬라이스 테스트 방법
 - Mockito 사용 방법

5 API 문서화

- Github Repository
 - [be-reference-api-documentation](#)
- 학습 내용
 - API 문서 생성 방법
 - AOP를 이용한 Stub 사용 방법

6 애플리케이션 빌드/실행/배포

- Github Repository
 - [be-reference-build](#)
- 학습 내용
 - @Profile 사용 방법

7 기타

- 디자인 패턴
 - Github Repository
 - [be-reference-design-pattern](#)

- Java 실무
 - Github Repository
 - be-reference-advanced-java
 - Future 사용 방법
 - 람다와 Stream API 기본