# 4장 – HTTP Method / Get



사용자들로부터 접속을 받기 위하여 주소를 설정해야 한다.

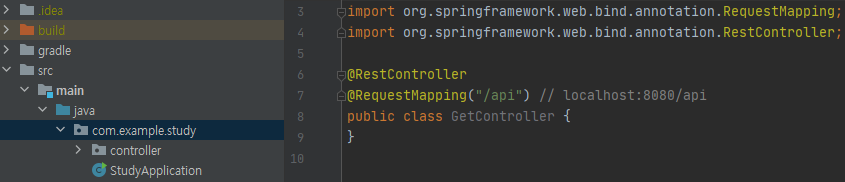
그런 주소들의 묶음을 모아 놓는 것을 Controller라고 부른다.

Spring에게 이곳은 Controller로 활용할 것이라고 알려주는 @RestController라는 Annotation을 달아준다.

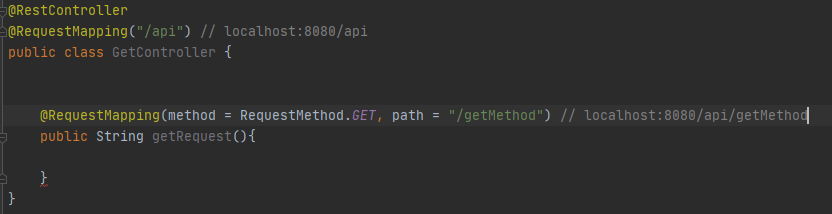
그리고 이곳으로 들어올 API 주소를 Mapping 하기 위하여 @RequestMapping을 적어준다.

(url 주소가 class에서 RequestMapping 이 겹치는 것은 정상 실행된다,

단, method 단에서 url 주소가 RequestMapping이 겹치면 에러 발생한다. 어떤 주소로 매칭을 시켜야 할 지 모르기 때문.)



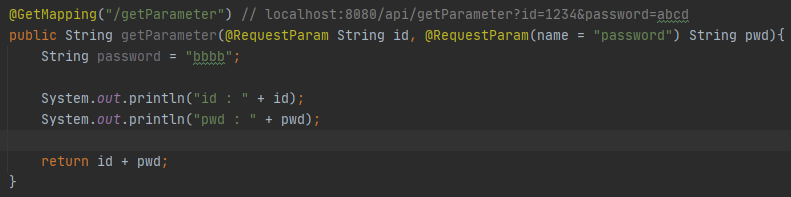
사용자의 요청에 대해서 한 가지 Method로 받아들일 수 있다.



Get 파라미터의 인자를 @RequestParam 으로 받아올 수 있다.

RequestParameter의 이름과 method에서 받아야 하는 변수 이름이 다를 때,

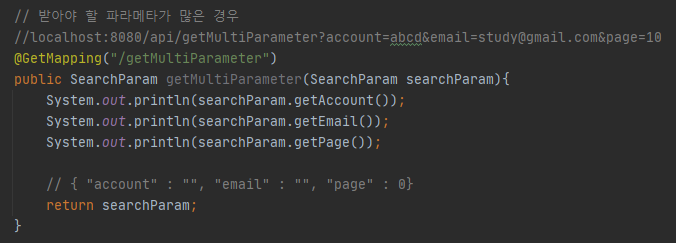
‘@RequestParam(name = “password”) String pwd’ 처럼 사용하여 Mapping하여 사용할 수 있다.

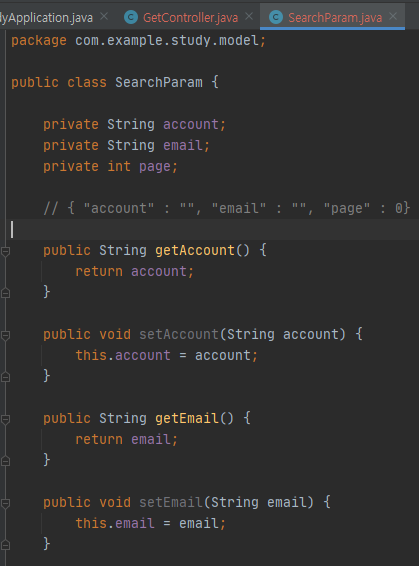


인자를 객체로 받는 방법 (받아야 할 파라메타가 많은 경우)

인자 클래스(SearchParam.java)를 만들고 get,set method와 생성자가 있어야 한다.

객체를 return하면 내장된 Jackson 라이브러리를 통하여 json 형태로 바꿔서 return하게 된다.

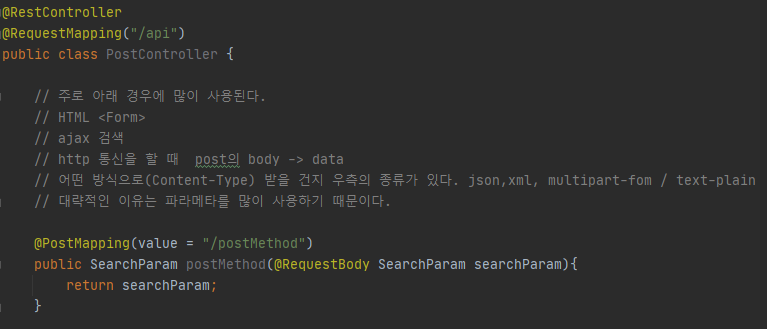






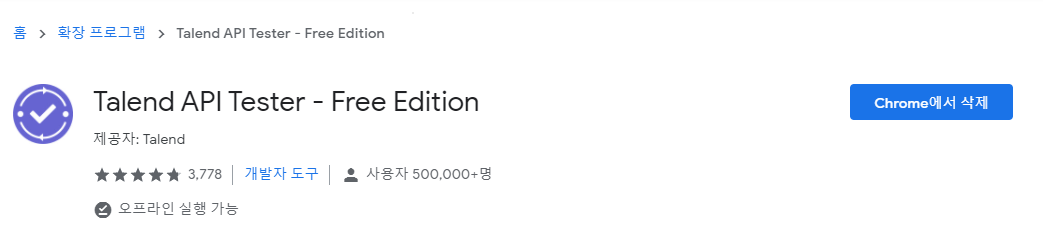
# 5장 Post Method



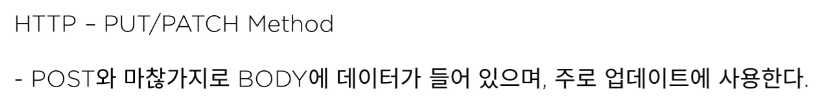


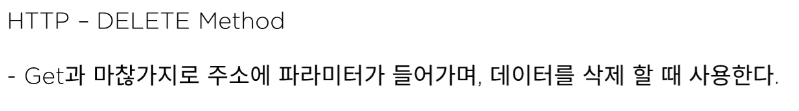
@PostMapping 으로 url mapping 하고, get과 다르게 method에서 @RequestBody annotation과 body값의 값을 받을 객체를 정의하여 Post method로 보낸 값을 받을 수 있다.



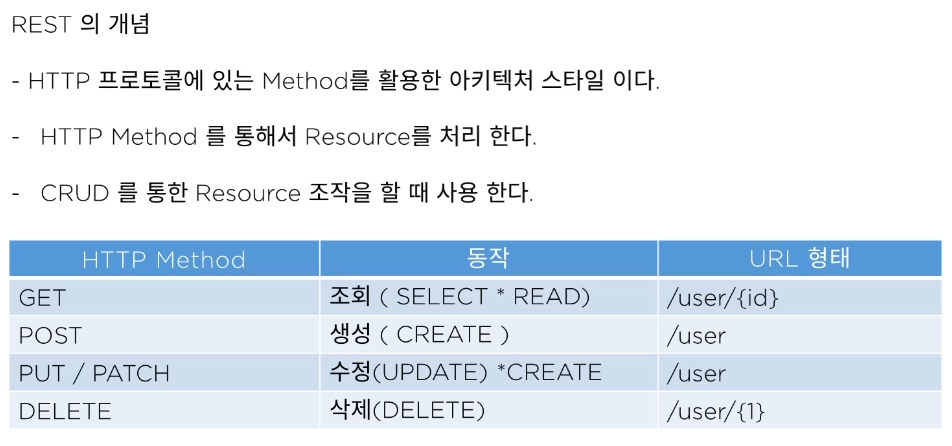


위는 크롬 웹스토어에서 RestAPI Test 위하여 사용하는 프로그램





## REST의 개념



# 7장 Lombok 플러그인

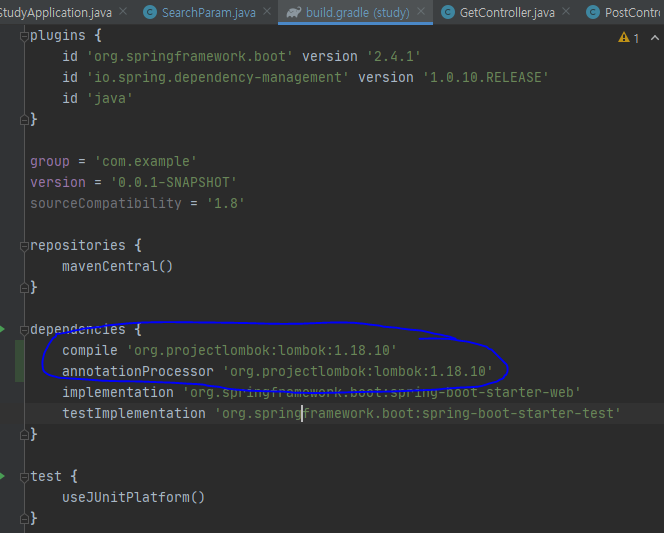
AS-IS : 기존 클래스를 만든 후 생성자 + get/set method를 IDE의 generator 기능으로 생성하거나 직접 코딩하여야 했음.

더불어 생성자 + get/set 메서드 때문에 코드 라인이 길어져서 가독성이 떨어짐.

TO-BE : Lombok 플러그인을 설치하면 @Data 만 붙이면 기본생성자+get/set 메서드를 생성해준다. 더불어 코드상에서 보이지 않아서 가독성이 좋아진다.



market에서 Lombok 설치하고, build.gradle에서 dependencies에 추가 해야 라이브러리 정상 사용 가능하다.



# 애먹였던 에러 - Process finished with exit code 1

9장 JPA Annotation가 import가 안되서 문제 찾다가 위 에러까지 만나서 한참 방법 찾았음.(git reset, revert도 하면서…)

해결법 블로그에 정리해 놓았음.

<https://charactermail.tistory.com/285>

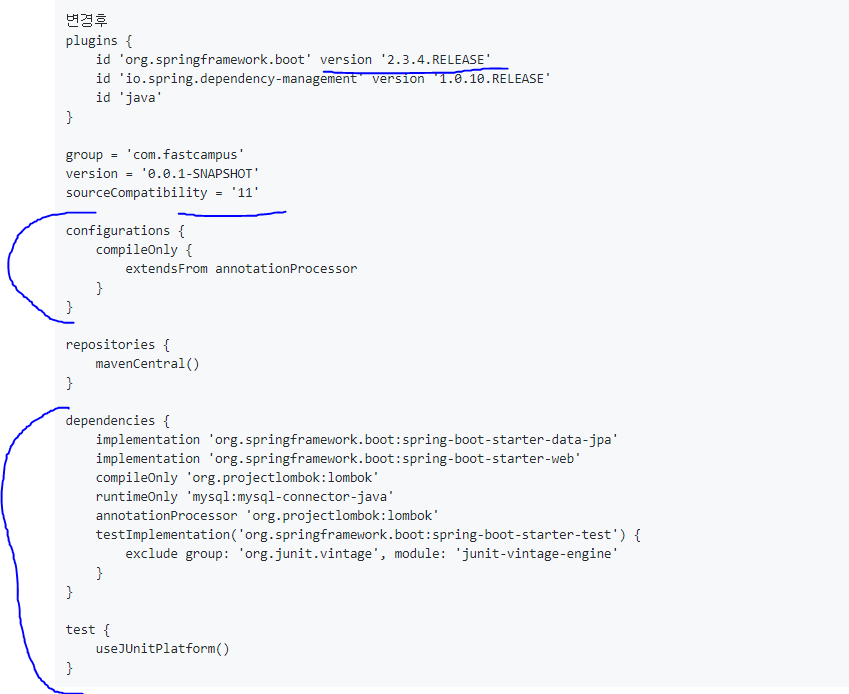
# 7장 이후 적용한 개발환경 변경사항

<https://github.com/steve-developer/fastcampus-java/tree/master/04-jpa-crud>

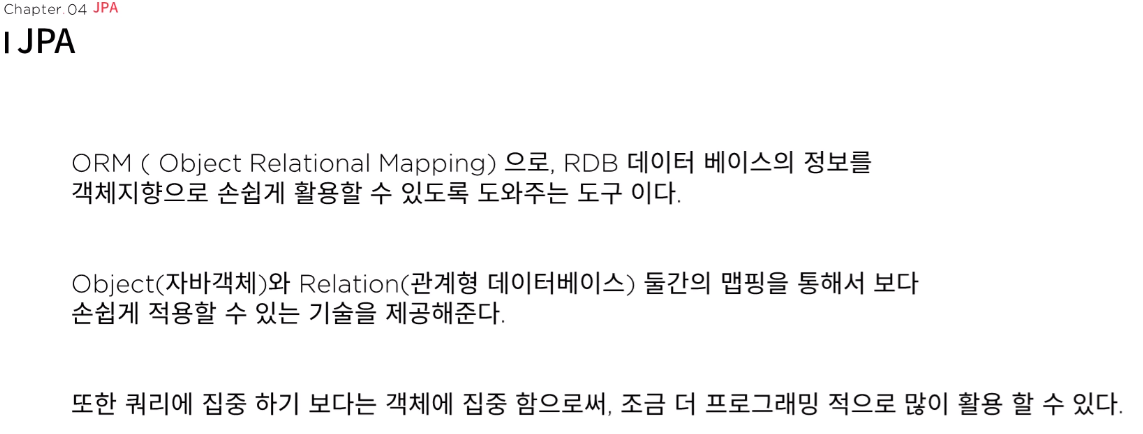
강의하시는분이 2020 최신 설정으로 업데이트사항 위 깃허브에 업로드 해놓았고,

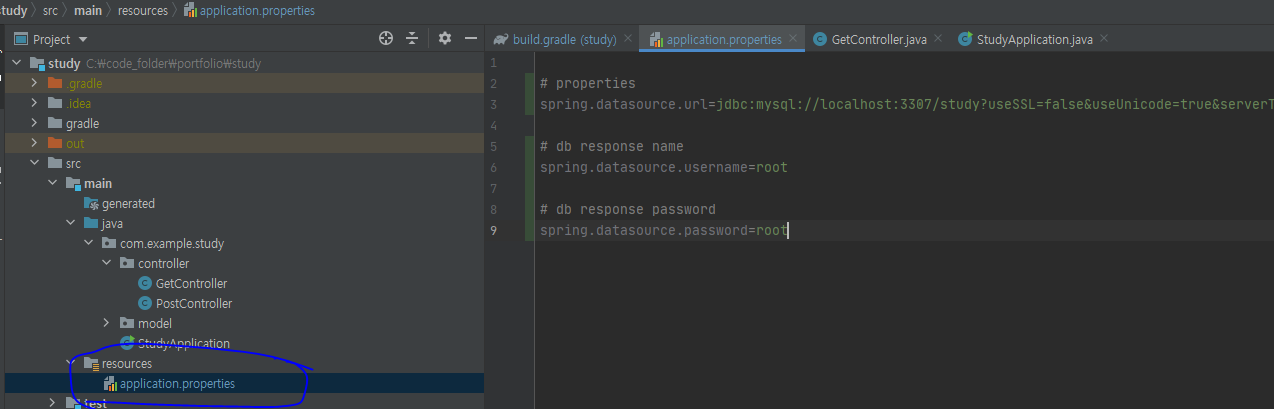
아래 체크한 부분 설정 변경하였음.

스프링부트 2.3.4으로, java 버전 11, dependencies등 아래처럼 변경사항 적용 하였음.



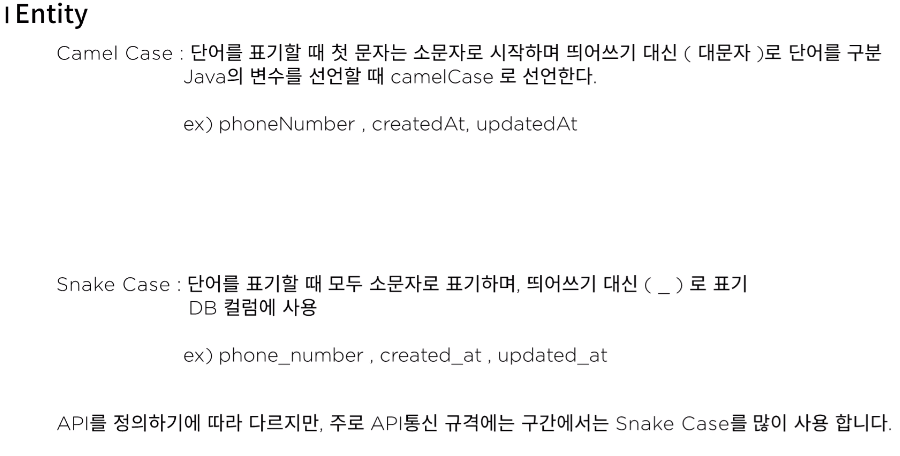
# 8장 JPA

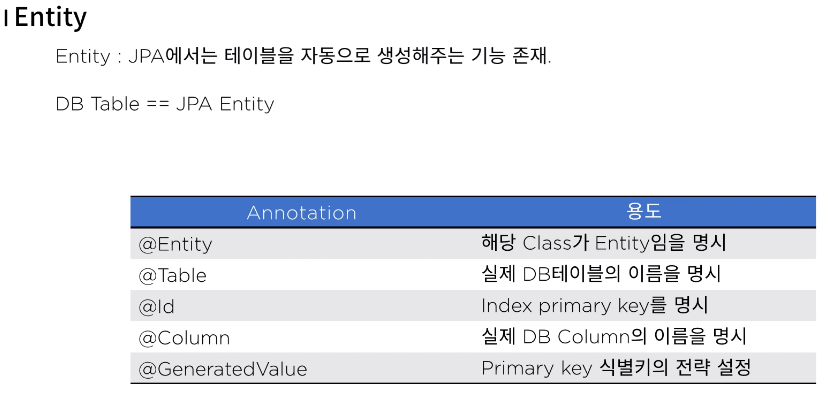


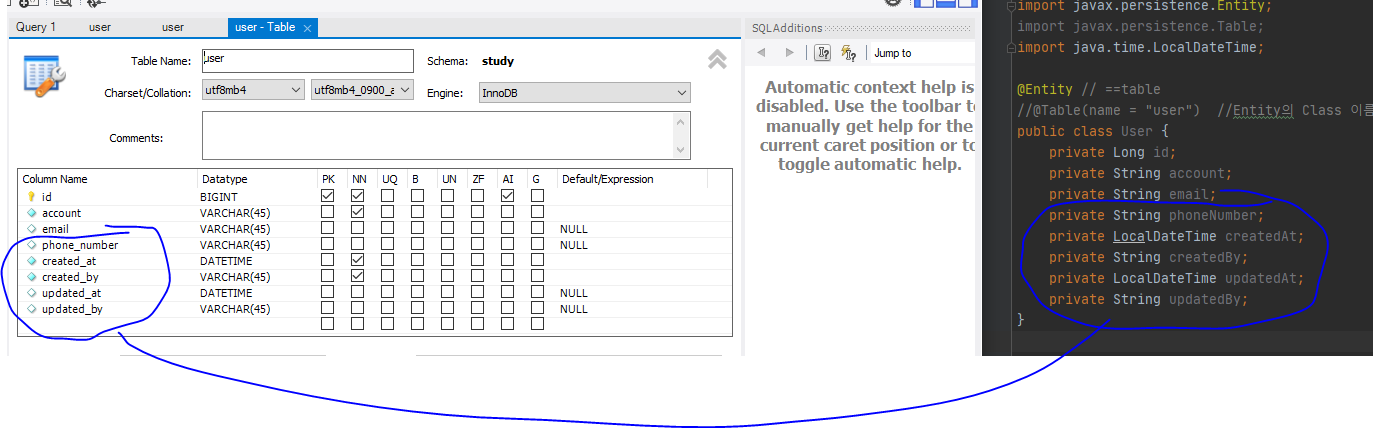


## Resource/application.properties 는 스프링 부트 프로젝트에 추가된 라이브러리들의 설정들을 관리하는 곳이다.

# 9장 – Entity 생성과 Repository

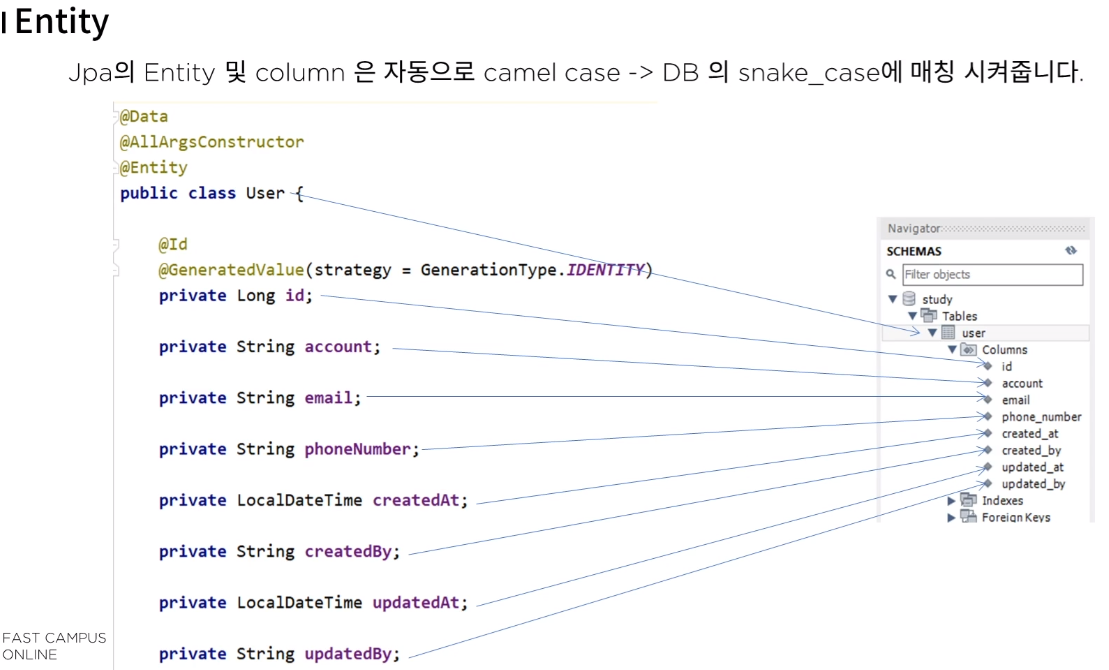






Entity의 속성에서 Java에서는 CamelCase로 선언하지만 Database에서는 SnakeCase이다.

JPA에서는 이런 것을 자동으 Mapping해주는 로직이 있어서 표기법이 달라도 문제가 발생하지 않는다.

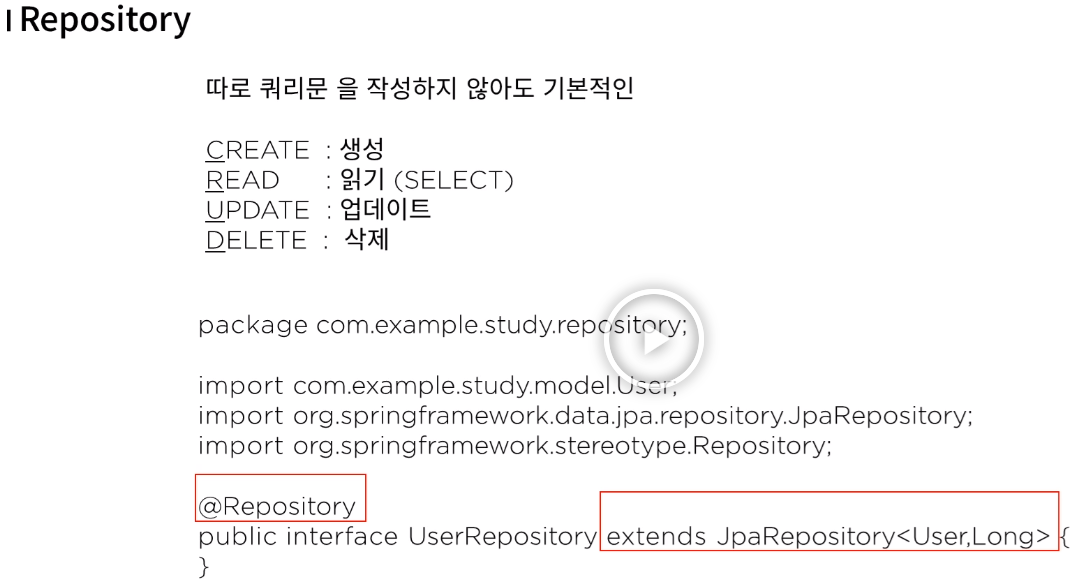


## Repository

따로 쿼리문을 작성하지 않아도 기본적인 인터페이스를 통해서 CRUD를 사용할 수 있게 해주는 것

사용법 심플함.

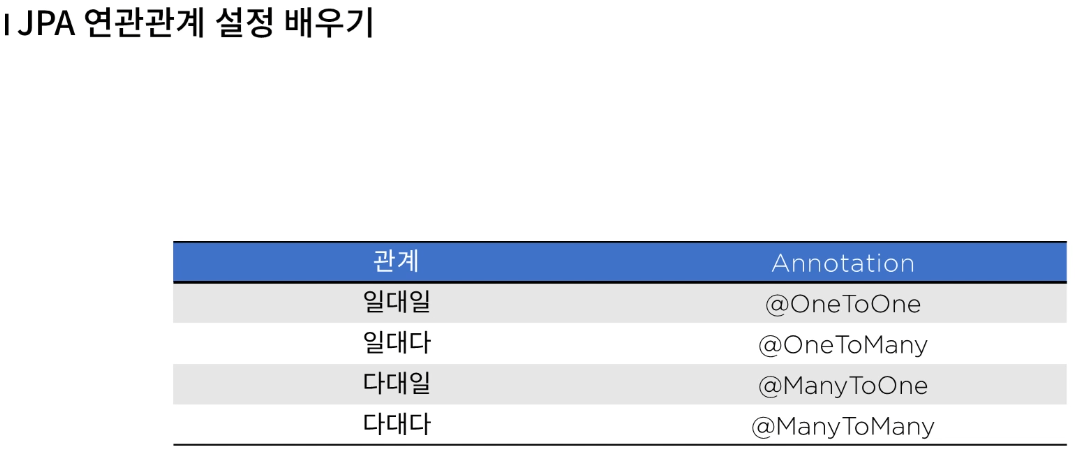
1. 인터페이스 먼저 만들고
2. Annotation을 @Repository 달아준다.
3. 이미 개발되어있는 JpaRepository를 상속받아 준다.
4. JpaRepository<어떠한 타입의 오브젝트 클래스가 넘어갈 건지 제네릭으로 넘겨줌 , 기본 키의 식별자>

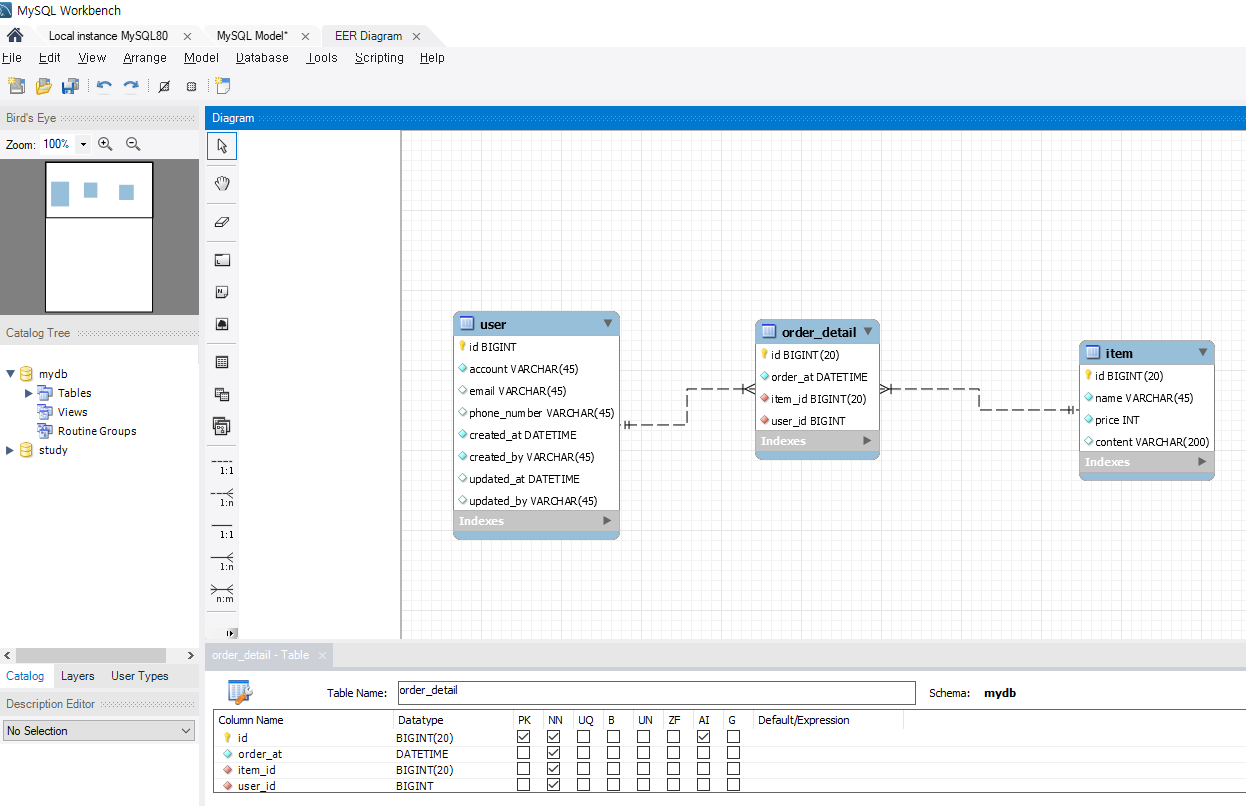


# 13. JPA 연관관계 설정

Work bench 의 ERD Diagram 그리는 tool로 User, item, order\_detail ERD 생성(연관관계 포함) 그리고 DB에 저장하는 것 하였음.

Database -> Reverse Engineer / Forward Engineer 탭 눌러서 ERD 생성 페이지 접근 / 생성한 ERD로 테이블 만들기 수행 하였음

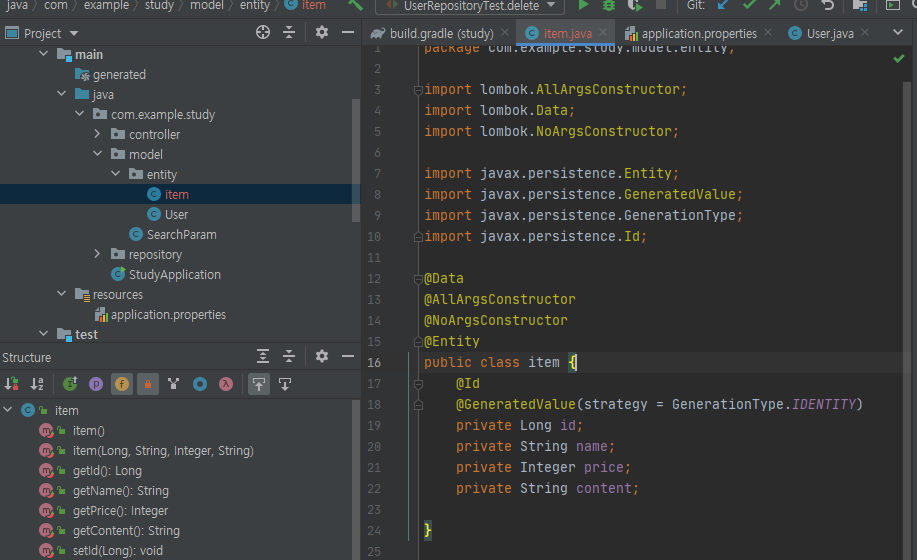




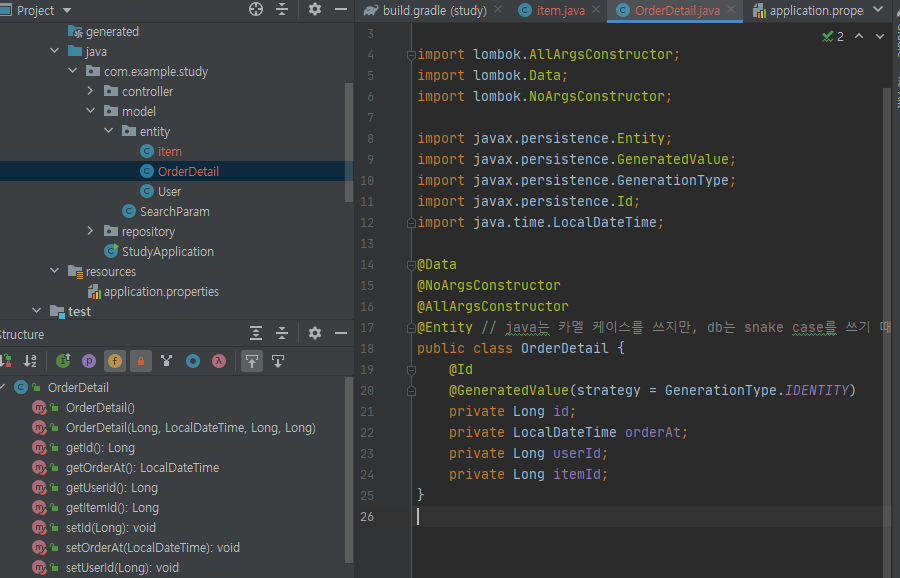
# 14 JPA 연관관계 설정 배우기 – 코딩하기

## TABLE을 설계하게 되면 거기에 맞는 Entitiy를 만들어 주어야 한다.

ERD에서 item table을 만들었으니, 아래처럼 Item table을 만들었다.



마찬가지로 OrderDetail Table도 만든다.



더불어 Repository와 Test도 만들어서 진행하였음.

## 주의사항 : test 라이브러리(다른거에러) org.junit.jupiter.api.Test;

import org.junit.jupiter.api.Test;

