

BACK-END 요구사항

1. Database

초기 user 정보를 Table에 Insert하였다.

src/main/resources 의 import.sql

```
import
Limit to 1000 rows
1  -- 요구 명세서에서 쓰라는 것을 insert --
2
3  • INSERT INTO `user` VALUES (1, 'SSAFY', '홍길동', '$2a$10$XtfiAWWiEjcRY8qXMRZmfOkIeXk7aXZeXziyS.IyMjJMvCt48gBsO', '교육생', 'test-1');
4  • INSERT INTO `conference_category` (`id`, `name`) VALUE (1, '업무'), (2, '교육'), (3, '기타');
```

MySQL 에서 확인한 user과 conference_category 테이블

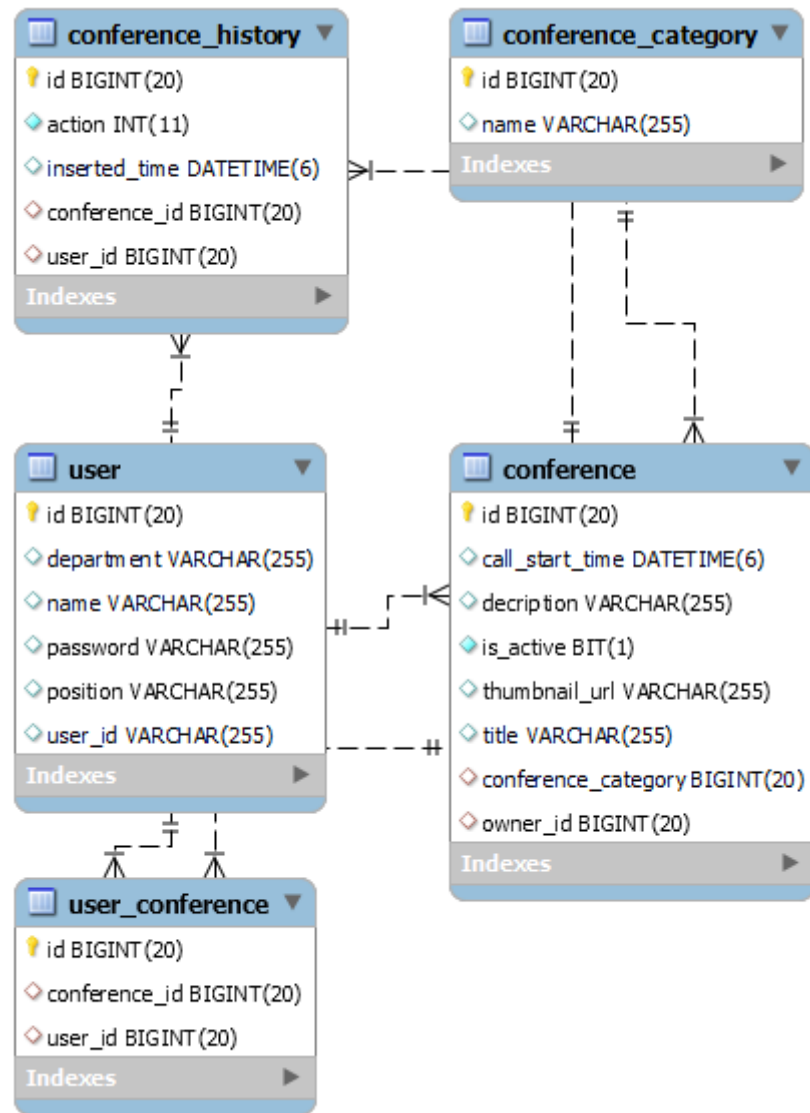
password의 12345가 암호화 되어 들어갔음을 확인할 수 있다.

id	department	name	password	position	user_id
1	SSAFY	홍길동	\$2a\$10\$XtfiAWWiEjcRY8qXMRZmfOkIeXk7aXZeXziyS.IyMjJMvCt48gBsO	교육생	test-1
NULL	NULL	NULL	NULL	NULL	NULL

id	name
1	업무
2	교육
3	기타
NULL	NULL

2. ERD 개요

MySQL에서 생성된 ERD



아쉬운점: 실선이 아닌 점선 형태로 구현되었다.

3. JPA

Entity

상기 ERD에서 확인 가능

db.entity 파일 마다 아래와 같이 column 지정

```
User.java  UserServiceIm...  Conference_Cat...  Conference_Hist...
1 package com.ssafy.db.entity;
2
3 import com.fasterxml.jackson.annotation.JsonIgnore;
4
10
11 /**
12  * 유저 모델 정의.
13  */
14 @Entity
15 @Getter
16 @Setter
17 // extends는 상속받았다는 뜻
18 // 상속이므로 애도 primary key가 id이다
19 public class User extends BaseEntity{
20     // MySQL에서 column 확인 가능(변수명 = column 명)
21     // user 말고 테이블 또 만들고 싶으면 class 명이랑 내부 바뀌서 복불하면 됨
22     // id는 만들지 않아도 있는데 그 이유가 BaseEntity를 상속받았기 때문
23     String position;
24     String department;
25     String name;
26     String userId;
27
28     // 암호화 된 column이라 Ignore 한다
29 @JsonIgnore
30 @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
31     String password;
32 }
33
```

Repository

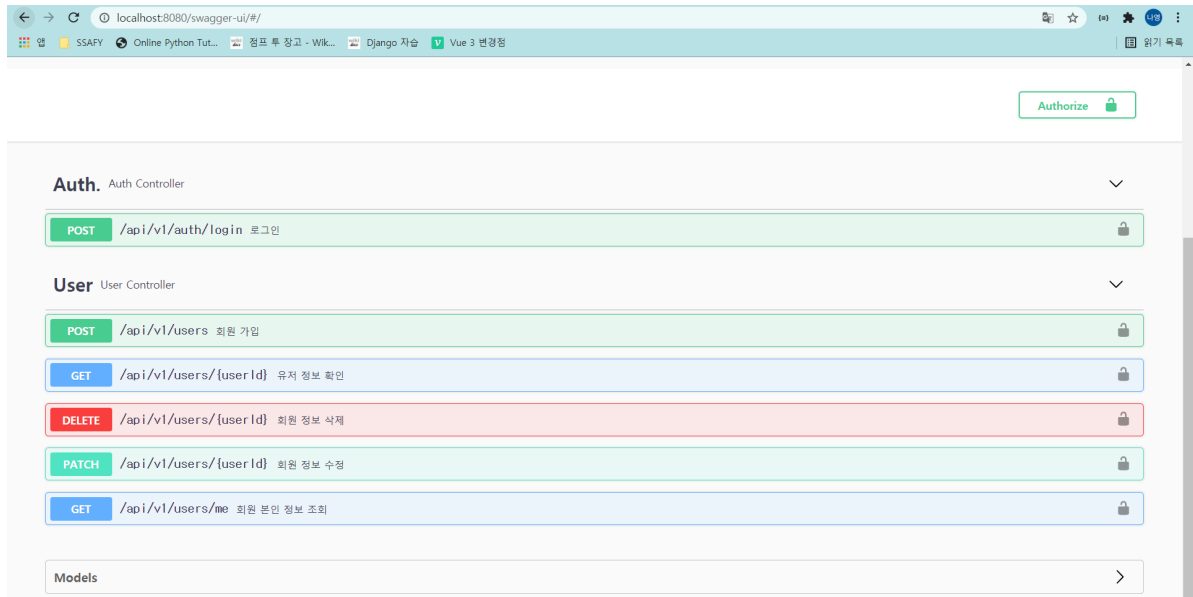
RepositorySupport 대신 Repository를 이용하였다.

예시: UserRepository

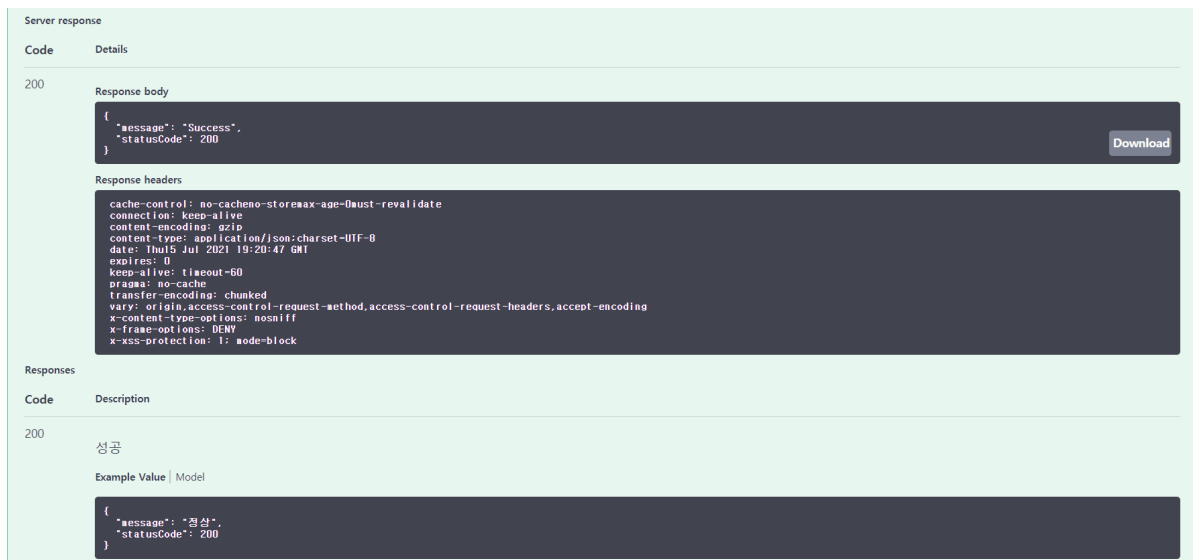
```
User.java  UserServiceImp...  Conference_Cate...  Conference_Histo...  BaseEntity.java  UserControllerJ...  UserRepositoryJ...
1 package com.ssafy.db.repository;
2
3 import com.ssafy.db.entity.User;
4
9 /**
10  * 유저 모델 관련 디비 쿼리 생성을 위한 JPA Query Method 인터페이스 정의.
11  */
12
13 // 두가지 방식이 있는데 이게 더 쉬워보임
14 @Repository
15 public interface UserRepository extends JpaRepository<User, Long> {
16     // 아래와 같이, Query Method 인터페이스(반환값, 메소드명, 인자) 정의를 하면 자동으로 Query Method 구현됨.
17     Optional<User> findById(String userId);
18 }
```

4. Swagger

swagger를 이용해 API를 시각화 하고 빠른 테스트를 진행할 수 있었다.



User 의 회원가입



swagger를 이용해 요청을 보내고 응답을 받아보았다.

server response와 responses 모두 200 성공이 나와야만 성공이었다.