MySQL DATABASE를 어떻게 운영할 지에 대한 이야기

Sql언어 사용법: <http://leechoong.com/posts/2018/mysql_basic/>

한글 저장을 위한 mysql 구성: <https://www.dante2k.com/526>

\*database 접속 주소: C:\Bitnami\wampstack-8.0.6-0\mysql\bin

\*database id: mysql -u internet\_project\_1 -p

그 이후 password 치면 됨(pw: JongbinKyuu12)

\*database 보기: show databases;

\*database 선택: use 데이터베이스이름(inpro\_db);

\*table 보기: show tables;

\*table 안 구성요소: DESC 테이블이름;

\*mysql 계정 만들기: create user 'internet\_project\_1'@localhost identified by 'JongbinKyuu12';

\*권한주기: 관리자 계정에서

create schema inpro\_db; ->

grant select, insert, update on inpro\_db.\* to 'internet\_project\_1'@localhost; 또는

grant all privileges on inpro\_db.\* to 'internet\_project\_1'@localhost;

\*권한 확인법: show grants for 'internet\_project\_1'@localhost;

\*bitnami에서 port는 3307로 하기!

우선 database 생성해야함:

CREATE DATABASE inpro\_DB DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;

: //database 생성, 한글 인식을 위해 위와 같은 명령어를 덧붙히기

USE inpro\_DB; //database 사용 선언

사용할 table:

1. user\_db: User의 data를 보관할 DB: 보안에 대해선 엄중하게 처리해야 할 듯.  
   사용할 field:  
   1) us\_ID: 회원번호, int, AI, PK  
   2) user\_ID: txt, 영소문자+숫자, max 15글자, 중복허락X  
   3) user\_PW: txt, 영소문자+숫자+특수기호(이건 고민해보자), max 20글자  
   4) signup\_date: 가입일(날짜): node.js에서 처리할 예정  
   사용할 객체 형식: object/Json string:  
    user{user\_id: 아이디, user\_pw: 패스워드, signup\_date: 가입날짜(“YYYY-MM-DD” 형태의 string), logined: false}: 초기에 회원가입 할 때, 아이디 login 할 때  
    user{user\_id: 아이디, user\_pw: NULL, signup\_date: NULL, logined: true}: 로그인 된 상태에서 분실물을 검색/등록 할 때  
   ==🡺 이제 이걸 sql code로 표현해보자:   
   mysql> create table user\_DB  
    -> (  
    -> `us\_id` int NOT NULL auto\_increment primary key,  
    -> `user\_ID` VARCHAR(15) NOT NULL UNIQUE,  
    -> `user\_PW` VARCHAR(20) NOT NULL,  
    -> `signup\_date` DATE NOT NULL  
    -> );
2. item\_DB: User가 찾은 item을 저장하는 DB  
   1) it\_ID: int AI PK  
   2) name: 자세한 명칭 – 모를 수도 있으므로 null도 가능, 최대 20글자  
   3) category: 경찰청 API를 이용하여 특정 요소들만 고를 수 있게 하기  
   4-1) acq\_lat: 습득한 GPS의 좌표 – double  
   4-2) acq\_long: double  
   5) acq\_time: 습득한 시간 – 사용자가 직접 입력 or 사용자의 현재 시간 자동으로 받아서  
   6) acq\_user: 습득한 user를 id로 연결: FOREIGN KEY이다!  
   7) real\_user: 그 물건의 실제 주인을 의미하며, Foreign key이지만 null이 가능하다: 여기서 아직 주인을 못찾은 것과 주인이 이 프로그램의 이용자가 아닌 것을 구분하기 위한 단서가 필요->  
   8) found: 물건의 주인을 찾았는지 찾지 못했는지 저장하는 int  
   9) place: 현재 그 물건이 존재하는 위치  
   10) spec: 물건에 대한 설명. 외형, 모습, 상태 이런거  
   11) img\_src: 사진의 저장소 위치를 연결  
     
   client⬄server의 object 이동 간의 데이터 형식: item object의 배열  
   - item{name: 분실물 이름, category: 분실물 카테고리(text값), acq\_gps: (위도, 경도: 소수점 7자리까지(ex: 37.5635937,126.938414 이런식), acq\_time: “YYYY-MM-DD HH:MM:SS”(24시간제), acq\_user: NULL(어짜피 user도 같이 send 하는데, 그 때 user object를 보고 확인할 예정), real\_user:NULL(필요없음), found:0(필요없는데이터), place: string(현재 보관하고 있는 장소: ex)”종비니집”), spec: text(물건에 대한 설명), img\_src: NULL(어짜피 필요x 파일이름 server에서 정할거임)}: 분실물을 등록할 때  
   - item{name: 분실물 이름, category: 분실물 카테고리(text값), acq\_gps: (위도, 경도: 소수점15자리?까지(ex: 37.5635937,126.938414 이런식), acq\_time: “YYYY-MM-DD HH:MM:SS”(24시간제), acq\_user: user\_id(string형식), found:0(필터링해서 나오는 데이터), place: string(현재 보관하고 있는 장소: ex)”종비니집”), spec: text(물건에 대한 설명), img\_src: server에서 저장된 image의 링크를 app.get을 통해 return하는 함수를 등록할 예정, 이 링크를 적어서 보내줌, 그리고 front에서 이 링크를 src에 넣음으로서 이미지 표현}: 분실물 정보를 server에서 불러와서 분실물 검색게시판에 load할 때, 핀포인트로 표현할 때 등등   
     
   ==🡺 이제 이걸 sql code로 표현해보자:   
   mysql> create table item\_db(  
    -> `it\_id` int not null auto\_increment primary key,  
    -> `name` varchar(30),  
    -> `category` varchar(30) not null,  
    -> `acq\_lat` double,  
    -> `acq\_long` double,  
    -> `acq\_time` datetime,  
    -> `acq\_user` int,  
    -> `real\_user` int,  
    -> `found` int(1),  
    -> `place` varchar(100) not null,  
    -> `spec` varchar(1000) not null,  
    -> `img\_src` varchar(100),  
    -> foreign key(acq\_user)  
    -> references user\_db(us\_id),  
    -> foreign key(real\_user)  
    -> references user\_db(us\_id));

전체적인 파일 디렉토리가 어떤 식으로 구성이 될 것인지:

Public

ㄴ(여기는 server에서 통신하는 것: static으로 구성할까? https://expressjs.com/ko/starter/static-files.html)

Private

ㄴdb들을 여기에 저장:

Index.js