K-Digital Training 웹 풀스택 과정

Sequelize

Sequelize 란?

- 자바스크립트 구문을 알아서 SQL로 변환해준다.
- DB 작업을 쉽게 할 수 있도록 도와주는 ORM 라이브러리 중 하나

Object – Relation Mapping

https://sequelize.org/api/v6/identifiers.html

Sequelize 설치

```
npm install sequelize sequelize-cli mysql2
```

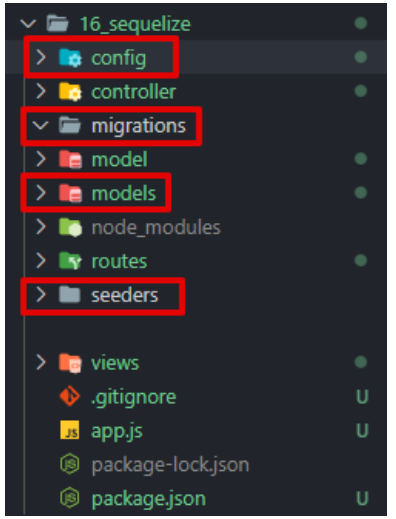
- sequelize: 시쿼라이즈 패키지
- sequelize-cli: 시퀄라이즈 명령어 실행

```
npx sequelize init
```

- sequelize init 명령어 호출
  (프로젝트를 처음 시작하는 경우에 유용)

```
"dependencies": {
  "ejs": "^3.1.8",
  "express": "^4.18.2",
  "mysql2": "^2.3.3",
  "sequelize": "^6.25.3",
  "sequelize-cli": "^6.5.1"
}
```

pakage.json
패키지 설치 확인



몇몇 새로운 폴더가 생성됨

Sequelize 폴더 구조

(1) config 폴더 생성

- 기존 방식은 모델 폴더 안에 파일이 여러 개 있다면 아래 데이터베이스 연결 코드를 모든 파일에 적어줘야 함

=> 데이터 베이스 명을 바꾸게 된다면 모든 파일을 다 바꿔줘야 함..!!

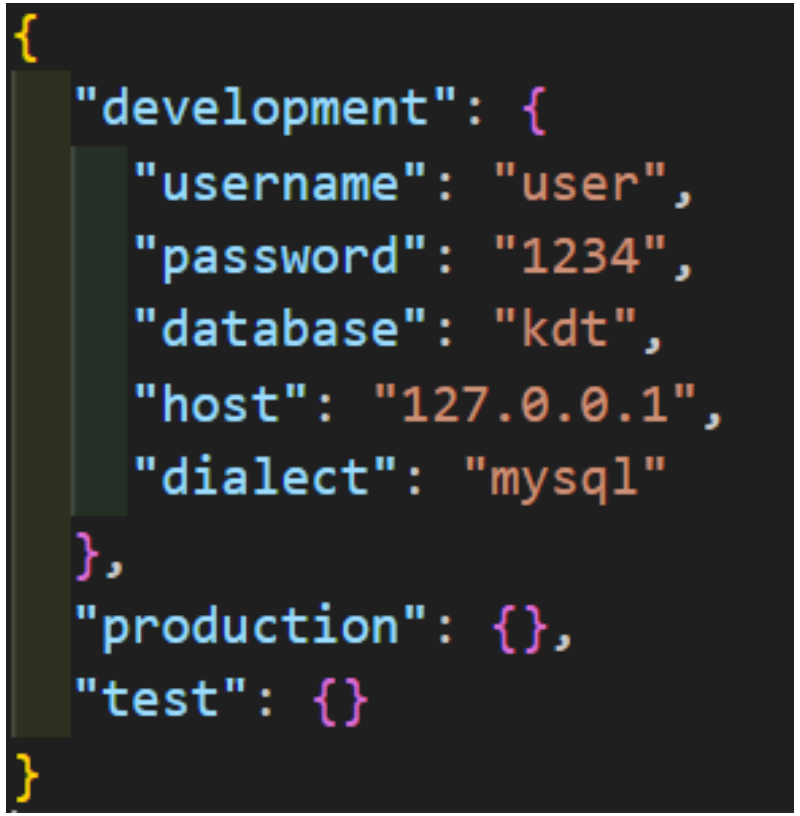
- config.json 파일로 한번에 처리하기!

```
const mysql = require('mysql2');

// DB 연결
const conn = mysql.createConnection({
  host: 'localhost',
  user: 'user',
  password: '1234',
  database: 'kdt',
});
```

Sequelize 이전 데이터베이스 연결 코드

(1) config 폴더 생성



```
{
  "development": {
    "username": "user",
    "password": "1234",
    "database": "kdt",
    "host": "127.0.0.1",
    "dialect": "mysql"
  },
  "production": {},
  "test": {}
}
```

config.json

** 참고! localhost란?

• localhost: 현재 컴퓨터의 내부 주소

→ 외부에서 접근 불가. 본인 컴퓨터로만 접근 가능

→ 개발시 테스트용으로 많이 사용함

• localhost를 IP 주소로 바꾸면 127.0.0.1

(2) models/index.js 파일 생성

```
const Sequelize = require('sequelize');
const config = require(__dirname + '/../config/config.json')['development’];

const db = {};
const sequelize = new Sequelize(
  config.database, // 데이터베이스 명
  config.username, // 사용자
  config.password, // 비밀번호
  config, // 정보 전체
);

db.sequelize = sequelize;
db.Sequelize = Sequelize;

db.Visitor = require('./Visitor')(sequelize, Sequelize); // 모델 정의 후 추가하기
module.exports = db;
```

index.js

Sequelize 모델 정의

mysql에서 정의한 테이블을 sequelize에서도 정의 필요 => mysql 테이블과 sequelize의 모델이 대응!!!!

```
Sequelize.define(param1, param2, param3);
// param1: 모델(테이블) 이름 설정
// param2: 컬럼 정의
// param3: 모델 옵션 정의
```

models / Visitor.js

Sequelize.define() 첫 번째 인자 - 모델 이름 설정

```
const model = Sequelize.define(
  // 모델 정의 -> sequelize 객체의 define 함수를 사용
  'visitor', // 인자1: 모델 이름 설정
  {...
  }, // 인자2: 컬럼 정의
  {...
  } // 인자3: 모델의 옵션 정의
);
```

Sequelize.define() 두 번째 인자 - 컬럼 정의

type : 데이터 타입을 정의 (문자, 숫자, 날짜 등등)

primaryKey : 기본키 설정 (default : false)

autoIncrement : 숫자 자동 증가 (default : false)

allowNull : NOT NULL 허용 여부 (default : true)

comment : column에 대한 설명을 작성한다.

validate : 데이터 유효성 검사를 하는 속성

Sequelize.define() 두 번째 인자 - 컬럼 정의

Sequelize 자료형 (DataTypes) vs. MySQL 자료형

```
Sequelize.STRING                    // VARCHAR(255)
Sequelize.STRING(1234)              // VARCHAR(1234)

Sequelize.TEXT                      // TEXT
Sequelize.TEXT('tiny')              // TINYTEXT

Sequelize.INTEGER                   // INTEGER

Sequelize.DATE                      // DATETIME for mysql / sqlite, TIMESTAMP
```

더 많은 데이터 종류는? 공식문서를 참고하자!
(https://sequelize.org/docs/v6/other-topics/other-data-types/)

Sequelize.define() 세 번째 인자 - 모델 옵션 정의

charset : "utf8"

collate : "utf8_general_ci" (만약 여기서 설정을 안 하면 DB를 생성할 때 설정해야 한다.)

tableName : 테이블 이름 설정

freezeTableName : true로 설정하면 이름을 복수로 설정하지 않는다.

timestamps : 기본적인 설정은 true

Sequelize 쿼리문

- findAll() - select
- findOne() - select
- create() - insert
- update() - update
- destroy() - delete

Sequelize에서 SQL문에 상응하는 메서드가 존재!! (즉, SQL문을 JavaScript로 생성한다!!)

프로미스(Promise)를 반환한다!! (.then()을 붙여서 결과값 사용 가능)