2022 부트캠프 033

- REST, RESTful API Set-up

!Tips & Links

※ 강의 https://www.udemy.com/course/the-complete-web-development-bootcamp/

※ Robo 3T https://robomongo.org/ (구글로 가입함, Studio 3T로 변경됨)

REST : REpersentational State Transfer

an architectural style for designing API's, ex) SOAP(전에 사용), GraphQL, FALCOR

RESTful API : 1. use HTTP Request Verbs 2. use Specific Pattern of Routes/Endpoint URLs

※ HTTP Request Verbs:

CRUD	VERBS	설명
READ	GET	읽기, returning the data for searching result
CREATE	POST	쓰기, request'll contain the data
UPDATE	갱신, update pieces of data	
	PUT	sending an entire entry to replace
	PATCH	updating the thing that needs to be
DELETE	DELETE	deletes a particular piece of data in DB

Robo 3T / Studio 3T : DB관리, GUL, https://studio3t.com/download-studio3t-free

Connection Manager > New connection에서 이름이랑 DB서버 지정해서 save

서버 선택해서 connect하기 ex) localhost:27017

add Database... / add Collection...(lowercase plural form) / add Document...

※ **Set-up a new env ~new server for new app~**

1) create new dir : wikiAPI
2) npm init –y, npm i body-parser mongoose ejs express < 한꺼번에 써도 됨!
3) touch app.js / add server code = nodemon app.js => log Sever stared on port 3000.

Sever starting code : jshint / require / set / use / (get / post) / listen

Jshint require

```
//jshint esversion:6
const express=require("express");
const bodyParser=require("body-parser");
const ejs=require("ejs");
const mongoose=require("mongoose");
const app=express();
```

set use

```
app.set('view engine', 'ejs');  // view engine은 ejs다.
app.use(bodyParser.urlencoded({
                    // bodyParser의 urlencoded메소드
  extended: true          // extended를 true로 설정
}));
app.use(express.static("public"))
    // storing static file 디렉토리(파일, URI 관리)
```

listen

```
app.listen(3000, (err)=>{ console.log("Server Started on port 3000") }) // 연결확인
```

4) set-up MongoDB // use **mongosh**! < up to date

Create+Read	DB만들기 / DB확인 / Collection만들기 / Collection확인 / Doc입력 / Doc확인
use 데이터베이스명 / db / db.createCollection("~") / show collections / db.~.insert([{}, {}]) 배열 / db.~.find();	

5) mongoose.connect("주소") "mongodb://localhost:27017/wikiDB"

"mongodb+srv://user~:pass~@cluster0.cw4wk.mongodb.net/wikiDB"

Connect	mongoose.connet("mongodb+src://유저명:비번@클러스터.mongodb.net/데이터베이스명");

6) 스키마 세팅: const articleSchema={title:String} / const Article = mongoose.model("Article", articleSchema)

/* -----<lang: en>---------- */

quite a mouthful / HTTP: Hypertext Transfer Protocol / FTP: File Transfer Protocol // DB architectural style

be tap into / your req can be tapped in(by lots of people) => HTTP secure, cryptography, encrypt