Node.js P240~

express 웹서버 만들기

const express = require('express');

const morgan = require('morgan');

const cookieParser = require('cookie-parser');

const session = require('express-session');

const dotenv = require('dotenv');

const path = require('path');

dotenv.config();//.env파일을 익어서 process.env로 만든다

//process.env.COOKIE\_SECRET에 cookiesecret 값이 할당된다 키=값 형식으로 추가하면됨

//.env를 별도의 파일로 관리하는 이유는 보안과 설정의 편의성 .env파일을 잘관리한다면 소스코드유치시에도 비밀키는 지킬수있다

const app = express();

app.set('port',process.env.PORT || 3000);

//설치한 패키지들을 불러와 app.use에 연결하는과정

app.use(morgan('dev'));

//인수로 dev외에 combined, common, short, tiny등을 넣을수있다

//개발환경에서는 dec를 배포환경에서는 combined를 사용

app.use('/', express.static(path.join(\_\_dirname,'public')));

//static미들웨어는 정적인 파일들을 제공하는 라우터 역활로 기본적으로 제공되어 express객체에서꺼내쓰면 된다

// app.use('요청경로', express.static('실제경로'));

app.use(express.json());

app.use(express.urlencoded({extended:false}));

/\*위의 두코드는 요청의 본문에 있는 데이터를 해석해서 req.body객체로 만들어주는 미들웨어

로 보통 폼데이터나 AJAX요청의 데이터를 처리

JSON과 URL-encoded형식의 데이터 외에소 Raw,Text형식의 데이터를 추가로 해석할수있으며

이때는 npm i body-parser로 추가로 설치하여

const bodyParesr = require('body-parser');

app.use(bodyParser.raw());

app.use(bodyParser.Text());

\*/

p252

app.use(cookieParser(process.env.COOKIE\_SECRET));

/\*cookiParser(비밀키)로사용하여 .env의 키값에해당하는값을 가져온다

쿠키를 생성/제거 하려면 res.cookie(키,값,옵션)/res.clearCooki(키,값,옵션)

이때 제거하기위해서 생성할때의 키,값,옵션이 일치해야 삭제할수있으며

expires나 maxAge옵션은 일치할 필요가 없다

\*/

app.use(session({

    resave: false,

    saveUninitialized:false,

    secret: process.env.COOKIE\_SECRET,

    cookie:{

        httpOnly:true,//클라이언트에서 쿠키를 확인하지 못하도록 설정

        secure:false,//https가 아닌 환경에서도 사용할수있게 설정

    },

    name: 'session-cookie',

}));

/\*세션 관리용 미들 웨어로 로그인등의 이유로 세션을 구현하거나 특정 사용자의 데이터

를 임시적으로 저장해둘때 유용

resave는 요청이 올때 세션에 수정사항이 생기지 않더라도 세션을 다시 저장할지 설정

saveUninitialized는 세션에 저장할 내역이 없더라도 처음부터 세션을 생성할지 설정

secret은 쿠키를 서명하는데 필요한 값을 설정하는것으로 cookie-paser과 같게 설정하는게 좋다

cookie는 세션 쿠기에대한 설정으로 maxAge, domain, path, expires, sameSite, httpOnly, secure등 설정가능

express-session으로 만들어진 req.session 객체에 값을 대입하거나 삭제해서 세션 변경 가능\*/

app.use((req,res,next)=>{

    console.log('모든 요청에 다 실행됩니다.');

    next()

})

app.get('/',(req,res,next)=>{

    res.sendFile(path.join(\_\_dirname,'/index.html'));

    console.log('GET / 요청에서만 실행됩니다');

    next();

}, (req,res)=>{

    throw new Error('에러는 에러 처리 미들웨어로 갑니다')

});

app.use((err, req, res, next)=>{

    console.error(err);

    res.status(500).send(err.message);

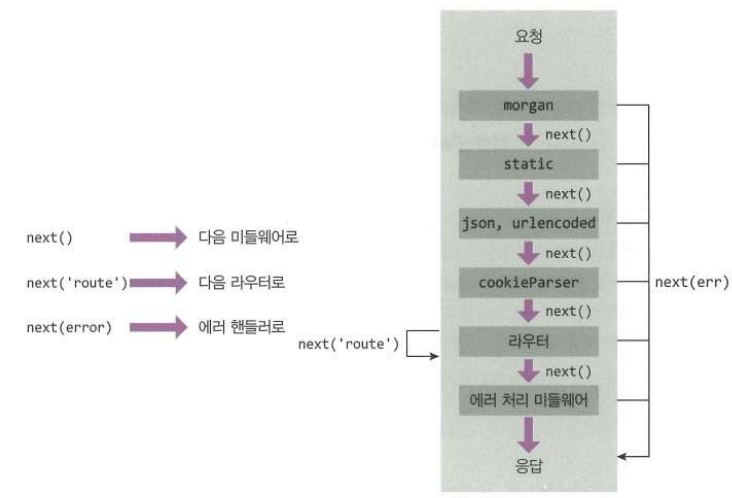
})

app.listen(app.get('port'),()=>{

    console.log(app.get('port'),'번 포트에서 대기중')

})

-next()의 동작



-multer 미들웨어 사용법

-USERS테이블 만들기

CREATE TABLE nodejs.users(

id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(20) NOT NULL,

age INT UNSIGNED NOT NULL,

married TINYINT NOT NULL,

comment TEXT NULL,

created\_at DATETIME NOT NULL DEFAULT NOW(),

PRIMARY KEY(ID),

UNIQUE INDEX name\_UNIQUE (name ASC))

COMMENT = '사용자 정보'

ENGINE = InnoDB;

-COMMENT테이블 만들기

create table nodejs.comments(

id int not null auto\_increment,

commenter int not null,

comment varchar(100) not null,

created\_at datetime not null default now(),

primary key(id),

index commenter\_idx (commenter asc),

constraint commenter

foreign key(commenter)

references nodejs.users(id)

on delete cascade

on update cascade)

comment = '댓글'

engine=innoDB;

-Sequelize 사용하여 MySQL 데이터베이스에 접속하기

1. npm i express morgan nunjucks sequelize sequelize-cli mysql2

// sequelize-cli는 시퀄라이즈 명령어를 실행하기 위한 패키지 이고

mysql2는 MySQL과 Sequelize를 이어주는 드라이버

2. npm i -D nodemon

3.npx sequerlize init

//Sequelize폴더 config, models, migragions, sedders 폴더 생성

p332