Node.js

* Javascript 언어가 구동되는 환경
* 서버 만들기도 가능

설치 방법

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash

. ~/.nvm/nvm.sh

nvm install node

npm init - package.json 파일 생성

npm install express –save – express 모듈을 dependency로서 설치

npm start – 서버 실행

REST API

* 자원을 이름으로 구분하여 해당 자원의 상태(정보)를 주고 받는 모든 것
* JSON 혹은 XML을 통해 데이터를 주고 받는 것이 일반적이다.
* 웹의 기존 기술과 HTTP프로토콜을 그대로 활용하기 때문에 웹의 장점을 최대한 활용할 수 있는 아키텍처 스타일이다.

HTTP Method

* GET: 조회

Ex) CommunityMember 테이블에서 멤버 정보 가져오기

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

connection.query('SELECT \* from CommunityMember', (error, rows) => {

if (error) throw error;

console.log('Community member info is: ', rows);

res.send(rows);

});

});

* POST: 생성

EX) TravelClub테이블에 클럽 이름, 클럽 리더, 클럽 활동 지역, 클럽 소개 입력

app.post('/add\_club/post\_clubName\_owner\_area\_content', (req, res) => {

var clubName = req.body.clubName;

var owner = req.body.owner;

var area = req.body.area;

var content = req. body.content;

connection.query("INSERT INTO TravelClub (clubName,owner,area,content) VALUES(?,?,?,?)",

[clubName, owner, area, content], (error, resutls, fields) => {

if (error) throw error’

res.json({ok:"true"});

});

});

* PUT: 갱신

EX) CommunityMemeber 테이블의 사용자 id를 검색하여 클럽 리더 갱신

app.put('/add\_club/put\_leader/:id\_', (req,res)=>{

var leader=req.body.leader;

connection.query("UPDATE CommunityMember SET `leader`=? WHERE id\_="+"'"+req.params.id\_+"'",[leader],(error, resutls, fields)=>{

if(error) throw error;

res.json({ok:"true"});

});

});

* DELETE: 삭제