<2018.02.20>

\* 사용할 카테고리 : 문화, 스포츠, 교양학술, 취미전시, 종교, 봉사

\* 동아리 내용/일정 수정할 때 기존내용 불러오기, contents 띄울 때 개행(\n이나 \r\n) 처리하기

\* 상태코드 465 추가

\* 일정 get요청 응답객체 통합

|  |  |  |  |
| --- | --- | --- | --- |
| http 메소드 | url | 요청 value | 성공시 응답 value |
| post | /user/login  동아리 관리자 로그인 요청 | id, pw | 201 동아리번호(int) |
| get | /user/logout  로그아웃 |  | 200 |
| get | /main  main화면 |  | 200 사진 접근 경로 |
| get | /club/category/카테고리명  해당 카테고리 동아리 목록 |  | 200 리스트 object |
| get(쿼리) | /club/search?keyword=검색단어  해당 단어와 관련된 동아리 목록 |  | 200 리스트 object |
| get | /club/info/동아리번호  해당 동아리 페이지 |  | 200 페이지 object |
| post | /club/info/동아리번호  동아리 내용 수정 | 수정 object | 201 |
| post | /club/image/동아리번호/사진번호  (예시)  /club/image/1/1  1번동아리 1번사진 수정 | 파일 하나 | 201 |
| get | /event/list/2018-02-14  해당 날짜 일정목록 |  | 200 행사목록 object |
| get | /club/event/동아리번호  해당 동아리 일정 목록 |  | 200 행사목록 object |
| post | /event/new  일정 등록 | 일정등록 object | 201 |
| post | /event/일정번호/edit  일정 수정 | 일정등록 object | 201 |
| post | /event/일정번호/delete  일정 삭제 |  | 201 |
| get | /club/ 동아리 전체 정보 | 필요하시면쓰세요 |  |

**<사용되는 상태코드>**

200 get성공

200 post성공

----------------실패----------------

400 잘못된 요청

401 비로그인

403 권한없음

404 Not found

412 입력한 데이터 제한길이 초과

460 로그인 실패

461 파일 업로드 실패

462 이미지 제한크기(2MB) 초과

463 파일(사진)을 업로드 하지 않음

464 이미지가 아닌 파일 업로드

500 서버에러

465 데이터의 형태가 올바르지 않음 (ex. 날짜나 시간, not null 등등)

**1. /main** < 서버 🡺 안드, ios object>

{

img

}

객체 개수는 가변적 이구요. 각 객체에 img라는 이름으로 string 하나만 보낼거에요.

받은 경로 앞에 baseUrl를 더해서 사용하셔야해요!

예를 들어 baseUrl이 <http://naver.com/> 라면

ArrayList<main> array = response.body();

for(int i = 0; i < array.length; i++)

array[i].img = “<http://naver.com/>” + array[i].img

**2. 동아리 목록(list) 띄울 때** < 서버 🡺 안드, ios object>

/club/category/카테고리명 또는 /club/search?keyword=검색단어 접근 시 받을(get) 객체

{

num (동아리 번호, int, 담아두었다가 page접근 시 이용)

만약int로 안 담긴다면 자료형 Number로 담거나 Integer.parseInt(num) 처럼 하셔야 할거에요…

clubname (동아리이름, string)

~~introduce (동아리 간단소개, string)~~

location (동아리위치, string)

image1 (이미지 경로, string)

}

**3. 동아리 페이지(page) 띄울 때** < 서버 🡺 안드, ios object>

/club/info/동아리번호 접근 시 받을(get) 객체

{

num (동아리 번호, int)

clubname (동아리이름, string)

location (동아리위치, string)

image1 (이미지 경로1, string)

image2 (이미지 경로2, string)

image3 (이미지 경로3, string)

image4 (이미지 경로4, string)

representative (대표자, string),

phone (연락처, string)

application (지원서 주소, string)

contents (내용, string)

}

**4. 동아리 수정페이지 기존 내용 불러올 때**

동아리 페이지를 get요청하시던지 아니면 직접 값을 이전 페이지에서 넘기시면 될 것 같아요

**5. 동아리 내용 수정 요청할 때** < 안드, ios 🡺 서버 전송 >

/club/info/동아리번호 로 보낼(post) object

{

~~introduce (소개, 최대 20자)~~

representative (대표자, 최대 5자)

phone (연락처, 최대 13자)

application (지원서주소, 최대 250자)

contents (내용, 최대 1000자)

}

**6. 이미지 수정은 한 칸에 하나씩 등록+전송**

예시) post /club/image/17/3

17번동아리 3번사진 수정

**7. 일정**

시작종료 날짜, 시간 관련해서 생각해보니 추가 하실게 많아질 것 같아서

간단하게 날짜, 시간 두 가지만으로 했어요.

등록할 때 🡺 post /event/new

수정할 때 🡺 post /event/일정번호/edit

{

eventname 일정이름(최대 15)

date 날짜(ex. 2018-02-14)

time 시간(ex. 15:00)

location 장소(최대 15자)

} 서버로 전송

해당 날짜 행사 목록 🡺 get /event/list/2018-02-14

해당 동아리 일정 목록 🡺 get /club/event/동아리번호

{

eventnum 행사번호(int)

clubname 동아리 이름

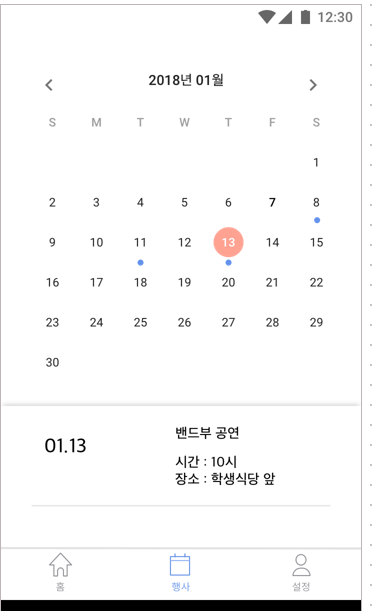
eventname 일정 이름

date 날짜(ex. 2018-02-14)

time 시간(ex. 14:00)

location 장소

} 서버로부터 수신



여기서 학생들이 일정목록을 볼 때 01.13 날짜부분을 동아리이름으로 바꾸는게 괜찮을 것 같아요.

<DB 생성>

CREATE DATABASE inuclub CHARACTER SET utf8 COLLATE utf8\_general\_ci;

<동아리 table1>

* 고유번호, 동아리이름, 아이디, 비번

CREATE TABLE `club\_authority` (

`num` int(11) NOT NULL AUTO\_INCREMENT,

`clubname` char(20) NOT NULL,

`authId` varchar(20) NOT NULL,

`password` varchar(20) NOT NULL,

PRIMARY KEY (`num`),

UNIQUE KEY `clubname\_UNIQUE` (`clubname`),

UNIQUE KEY `authId\_UNIQUE` (`authId`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8

<동아리 table2>

* 고유번호, 소개, 위치, 카테고리, 이미지1234, 회장이름, 번호, 지원서주소, 소개내용

CREATE TABLE `club\_info` (

`num` int(11) NOT NULL,

`location` char(20) DEFAULT NULL,

`category` char(10) NOT NULL,

`image1` varchar(30) DEFAULT NULL,

`image2` varchar(30) DEFAULT NULL,

`image3` varchar(30) DEFAULT NULL,

`image4` varchar(30) DEFAULT NULL,

`representative` char(5) DEFAULT NULL,

`phone` char(13) DEFAULT NULL,

`application` varchar(255) DEFAULT NULL,

`contents` varchar(1000) DEFAULT NULL,

PRIMARY KEY (`num`),

KEY `idx\_category` (`category`),

CONSTRAINT `fk\_info` FOREIGN KEY (`num`) REFERENCES `club\_authority` (`num`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8

<일정 table>

* 행사번호, 동아리 이름, 행사이름, 날짜, 시간, 위치

CREATE TABLE `club\_event` (

`eventnum` int(11) NOT NULL AUTO\_INCREMENT,

`clubname` char(20) NOT NULL,

`eventname` varchar(15) NOT NULL,

`date` char(10) NOT NULL,

`time` char(5) NOT NULL,

`location` varchar(15) DEFAULT NULL,

PRIMARY KEY (`eventnum`),

KEY `idx\_date` (`date`),

KEY `idx\_clubname` (`clubname`),

CONSTRAINT `fk\_event` FOREIGN KEY (`clubname`) REFERENCES `club\_authority` (`clubname`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=utf8

<동아리 추가>

*1. auth 생성*

insert into club\_authority (clubname, authId, password) values ("동아리명", "아이디", "비밀번호");

*2. 번호 확인*

select num from inuclub.club\_authority where clubname = "동아리명";

*3. info 생성*

insert into inuclub.club\_info (num, location, category) values (번호, "위치", "카테고리");

<동아리 제거>

delete from inuclub.club\_authority where clubname = "동아리명";

├─config

│ ├─db.js //db

│ ├─express.js //세팅

│ ├─localUpload.js //파일 업로드, 삭제

│ ├─log.js //로그 설정

│ └─util.js //로그인, 권한 체크

├─log //log폴더

├─public

│ ├─club\_img //동아리 사진 정적 폴더

│ └─main\_img //메인 사진 정적 폴더

├─routes

│ ├─club\_controller //동아리 컨트롤러

│ │ ├─getInfo.js

│ │ └─postInfo.js

│ ├─event\_controller //일정 컨트롤러

│ │ ├─getEvent.js

│ │ └─postEvent.js

│ ├─club.js //동아리 라우터

│ ├─event.js //일정 라우터

│ ├─main.js //main

│ └─user.js //login, logout