< index >

1. 서버
2. 요청/응답 기본 정보
3. 상태 코드
4. 요청/응답 object
5. DB 정보

2018-05-27 이미지 최대 크기 4MB 변경

1. 서버

\*레디스 설치

\*key.json => host, port, user, password, database, secret, env(development/production)

\*pm2 logrotate

<https://gist.github.com/kilnamkim/0febea79e30a9d2422453f5a68a0f84e>

pm2 set pm2-logrotate:rotateInterval '\* \* \* /1 \* \*'

\*시작(재시작) : pm2 start(restart) ecosystem.config.js

\*파일

├─ecosystem.js, index.js, worker.js //pm2설정, 마스터, 워커

├─config

│ ├─db.js //db

│ ├─express.js //세팅

│ ├─localUpload.js //파일 업로드, 삭제

│ ├─log.js //~~로그 설정~~(사용안함)

│ └─util.js //로그인, 권한 체크

├─logs

├─public

│ ├─club\_img //동아리 사진 정적 폴더

│ └─main\_img //메인 사진 정적 폴더

├─routes

│ ├─club\_controller //동아리

│ │ ├─getInfo.js //해당 카테고리 목록, 검색, 동아리 info, 해당 동아리 일정

│ │ └─postInfo.js //info수정, 사진수정

│ ├─event\_controller //일정

│ │ ├─getEvent.js //해당 날짜 일정

│ │ └─postEvent.js //일정 등록, 수정, 삭제

│ ├─club.js //동아리 라우터

│ ├─event.js //일정 라우터

│ ├─main.js //main

│ └─user.js //login, logout

2. 요청/응답 기본 정보

※ 사용할 카테고리 : 문화, 스포츠, 교양학술, 취미전시, 종교, 봉사

※ 요청/응답 object는 json 형식이며 자세한 내용은 문서의 4. 요청/응답 object 참조

|  |  |  |  |
| --- | --- | --- | --- |
| 메서드 | 설명 & 요청 url | request | 성공 response |
| get | main화면  /main |  | 200  + 사진url |
| post | 동아리 관리자 로그인 요청  /user/login | id, pw | 201  + 동아리번호(string) |
| get | 로그아웃  /user/logout |  | 200 |
| get | 해당 카테고리 동아리 목록  /club/category/카테고리명 |  | 200  + 리스트 object |
| get(쿼리) | 해당 단어와 관련된 동아리 목록  /club/search?keyword=검색단어 |  | 200  + 리스트 object |
| get | 1. 해당 동아리 페이지  2. 동아리 내용 수정 시 기존 내용 불러올 때에도 이용 가능  /club/info/동아리번호 |  | 200  + 페이지 object |
| post | 동아리 내용 수정  /club/info/동아리번호 | 동아리수정 object | 201 |
| post | 동아리 사진 수정  /club/image/동아리번호/사진번호  (예시) 1번동아리 1번사진 수정  /club/image/1/1 | 파일 하나 | 201 |
| get | 해당 날짜 일정목록  (예시) /event/list/2018-02-14 |  | 200  + 일정 목록 object |
| get | 해당 동아리 일정 목록  /club/event/동아리번호 |  | 200  + 일정 목록 object |
| post | 일정 등록  /event/new | 일정 등록 object | 201 |
| post | 일정 수정  /event/일정번호/edit | 일정 등록 object | 201 |
| post | 일정 삭제  /event/일정번호/delete |  | 201 |

3. 상태코드

----------------요청 성공----------------

200 get성공

201 post성공

----------------요청 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(4MB) 초과

463 파일(사진)을 업로드 하지 않음

464 이미지가 아닌 파일 업로드

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

500 서버에러

503 응답 시간 초과. **Service** Unavailable

4. 요청/응답 object

**4-1. 메인페이지**

4-1-1. 요청 url : (get) /main

4-1-2. 정상응답

|  |  |  |
| --- | --- | --- |
| 필드 | 타입 | 설명 |
| Img | String | 사진url |

4-1-3. 정상 응답 예시

* (사진 경로 하나가 포함된)객체들이 배열 형태로 전송되며 객체 개수는 가변

[

{

“img” : "main\_img/untitleds.png"

},

{

“img” : "main\_img/vdWfo23dfs.png"

}

]

**4-2. 로그인 요청**

4-2-1. 요청 url : /user/login

4-2-2. 요청 변수

|  |  |  |
| --- | --- | --- |
| 필드 | 타입 | 설명 |
| id(필수) | String | 관리자 id |
| pw(필수) | String | 관리자 pw |

4-2-3. 정상 응답 : 19 동아리번호(string) 하나 전송됨

**4-3. 리스트 object**

4-3-1. 요청 url : (get) /club/category/카테고리명

/club/search?keyword=검색단어

4-3-2. 정상 응답

|  |  |  |
| --- | --- | --- |
| 필드 | 타입 | 설명 |
| num | Number | 동아리 번호  (page접근시 사용 요망) |
| clubname | String | 동아리 이름 |
| location | String | 동아리 위치 |
| Image1 | string | 이미지 url |

4-3-3. 정상 응답 예시

* 객체들이 배열 형태로 전송되며 객체 개수는 가변

[

{

"num": 19,

"clubname": "TIME",

"location": "17호관 304호",

"image1": "club\_img/club\_19\_1521289676391.png"

},

{

"num": 20,

"clubname": "unexpecTED",

"location": "17호관 305호",

"image1": null

}

]

**4-4. 페이지 object**

4-4-1. 요청 url : /club/info/동아리번호

4-4-2. 정상응답

|  |  |  |
| --- | --- | --- |
| 필드 | 타입 | 설명 |
| num | Number | 동아리 번호 |
| clubname | String | 동아리 이름 |
| location | String | 동아리 위치 |
| image1 | String | 이미지1 url |
| image2 | String | 이미지2 url |
| image3 | String | 이미지3 url |
| image4 | String | 이미지4 url |
| representative | String | 대표자 |
| phone | String | 연락처 |
| application | String | 지원서 주소 |
| contents | String | 내용 |

4-4-3. 정상 응답 예시

* 객체 하나가 배열에 포함되어 전송됨

[

{

"num": 19,

"clubname": "TIME",

"location": "17호관 304호",

"image1": "club\_img/club\_19\_1521289676391.png",

"image2": "club\_img/club\_19\_1521289676367.png",

"image3": "club\_img/club\_19\_1521289676392.png",

"image4": "club\_img/club\_19\_1521289676366.png",

"representative": "김선일",

"phone": "010-3949-3955",

"application": "http://www.naver.com",

"contents": ""

}

]

**4-5. 동아리 수정 object**

4-5-1. 요청 url : (post) /club/info/동아리번호

4-5-2. 요청 변수

|  |  |  |
| --- | --- | --- |
| 필드 | 타입 | 설명 |
| representative | String 최대 5자 | 대표자 |
| phone | String 최대 13자 | 연락처 |
| application | String 최대 250자 | 지원서 주소 |
| contents | String 최대 1000자 | 내용 |

4-5-3. 요청 실패는 상태코드 참조

**4-6. 파일전송**

4-6-1. 요청 url : (post) /club/image/동아리번호/사진순서번호

4-6-2. 요청 변수

|  |  |  |
| --- | --- | --- |
| 필드 | 타입 | 설명 |
| userfile | 파일 (image 확장자) | 동아리 사진 |

4-6-3. 경로 예시) /club/image/17/3 17번동아리 3번사진 수정

4-6-4. 요청 실패는 상태코드 참조

**4-7. 일정 등록 object**

4-7-1. 요청 url : 등록할 때 🡺 (post) /event/new

수정할 때 🡺 (post) /event/일정번호/edit

4-7-2. 요청 변수

|  |  |  |
| --- | --- | --- |
| 필드 | 타입 | 설명 |
| eventname(필수) | String (최대 15자) | 일정이름 |
| date(필수) | String  (형식 : yyyy-MM-dd) | 날짜 |
| time(필수) | 시간  (형식 : hh:mm) | 시간 |
| Location | 장소 (최대 15자) | 장소 |

4-7-3.

date형식 예시) 2018-02-14

time형식 예시) 05:30

17:05

4-7-4. 요청 실패는 상태코드 참조

**4-8 행사 목록 object**

4-8-1. 요청 url : 해당 날짜 행사 목록 🡺 (get) /event/list/yyyy-MM-dd

해당 동아리 일정 목록 🡺 (get) /club/event/동아리번호

4-8-2. 정상 응답

|  |  |  |
| --- | --- | --- |
| 필드 | 타입 | 설명 |
| eventnum | Number | 행사 번호 |
| clubname | String | 동아리 이름 |
| eventname | String | 일정 이름 |
| date | String | 날짜 |
| time | String | 시간 |
| location | String | 장소 |

**5. DB 정보**

DB생성 / 테이블 정보 / 신규 동아리 추가 쿼리 (수동입력) / 동아리 제거 (수동입력)

<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(필수)*

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

*2. 생성된 번호 확인*

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

*3. info 생성(필수)*

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

<동아리 제거>

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