

## 1. 엔티티 - DB 구성

### a. 카테고리(Category)

- i. category\_id
  - 1. Long
- ii. parent\_id
  - 1. Long
- iii. name
  - 1. String
- iv. state
  - 1. Boolean
- v. reg\_user
  - 1. User
  - 2. @ManyToOne
  - 3. @JoinColumn(name = "reg\_user\_id", updatable = false)
- vi. reg\_date
  - 1. LocalDateTime
  - 2. @Column(updatable = false)
- vii. mod\_user
  - 1. User
  - 2. @ManyToOne
  - 3. @JoinColumn(name = "mod\_user\_id")
- viii. mod\_date
  - 1. LocalDateTime

### b. 공지사항(Notice)

- i. notice\_id
  - 1. Long
- ii. title
  - 1. String
  - 2. @Column(nullable = false, length = 40)
- iii. state
  - 1. enum
- iv. notice\_start
  - 1. LocalDateTime
  - 2. @Column(nullable = false)
- v. reg\_user
  - 1. User
  - 2. @ManyToOne
  - 3. @JoinColumn(name = "reg\_user\_id", updatable = false)
- vi. reg\_date
  - 1. LocalDateTime
  - 2. @Column(updatable = false)
- vii. mod\_user
  - 1. User
  - 2. @ManyToOne
  - 3. @JoinColumn(name = "mod\_user\_id", updatable = false)
- viii. mod\_date
  - 1. LocalDateTime
  - 2. @Column(updatable = false)

- ix. view
  - 1. int
- x. mod\_reason
  - 1. String
  - 2. @Column(nullable = false)
- xi. content
  - 1. String
  - 2. @Column(nullable = false, length = 5000)
- c. 사용자(User)
  - i. user\_id
    - 1. Long
  - ii. name
    - 1. String
- d. FAQ
  - i. faq\_no
    - 1. Long
  - ii. faq\_cate
    - 1. String
  - iii. faq\_title
    - 1. String
    - 2. @Column(nullable = false, length = 40)
  - iv. faq\_rdate
    - 1. LocalDateTime
    - 2. @Column(updatable = false)
  - v. reg\_user
    - 1. User
    - 2. @ManyToOne
    - 3. @JoinColumn(name = "reg\_user\_id", updatable = false)
  - vi. mod\_user
    - 1. User
    - 2. @ManyToOne
    - 3. @JoinColumn(name = "mod\_user\_id", updatable = false)
  - vii. faq\_start
    - 1. LocalDateTime
  - viii. faq\_sdate
    - 1. LocalDateTime
  - ix. faq\_mdate
    - 1. LocalDateTime
    - 2. @Column(updatable = false)
  - x. faq\_state
    - 1. enum
  - xi. faq\_reason
    - 1. String
  - xii. view
    - 1. int
  - xiii. faq\_content
    - 1. String
    - 2. @Column(nullable = false, length = 5000)

## 2. API("/api")

### a. 카테고리("/categories")

#### i. GET

("/api/categories/")	모든 카테고리를 조회하는 API
Request	/api/categories/
Response	<pre>{   "success": true,   "code": 0,   "data": [     {       "id": {{category_id}},       "name": {{name}},       "children": [...]     }, ...   ] }</pre>

#### ii. POST

("/api/categories/")	카테고리를 생성하는 API
Request	<pre>{   "name": String,   "parent_id": Long }</pre>
Response	<pre>{   "success": true,   "code": 0,   "data": [     {       "id": {{category_id}},       "name": {{category_name}},       "parent_id": {{parent_id}}     }   ] }</pre>

### iii. PUT

/api/categories/{category_id}	카테고리를 수정하는 API
Request	<pre>{   "name": String,   "parent_id": Long }</pre>
Response	<pre>{   "success": true,   "code": 0,   "data": [     {       "id": {{category_id}},       "name": {{category_name}},       "parent_id": {{parent_id}}     }   ] }</pre>

### iv. DELETE

/api/categories/{category_id}	카테고리를 삭제하는 API
Request	/api/categories/{category_id}
Response	<pre>{   "success": true,   "code": 0 }</pre>

b. 공지사항("/notices")

i. GET

/api/notices/	모든 공지사항을 조회하는 API
Request	/api/notices/
Response	<pre>{   "success": true,   "code": 0   "data": [     {       "id": {{notice_id}},       "title": {{title}},       "reg_user": {         "id": {{user_id}},         "name": {{name}}       },       "reg_date": {{reg_date}},       "notice_start": {{notice_start}},       "state": {{state}},       "view": {{view}},     },...   ] }</pre>

/api/notices/search	조건에 맞게 공지사항을 조회하는 API @RequestParam
Request	/api/notices/search?state={state}&reg_start={reg_start}&reg_end={reg_end}&notice_start={notice_start}&notice_end={notice_end}
Response	<pre>{   "success": true,   "code": 0   "data": [     {       "id": {{notice_id}},       "title": {{title}},       "reg_user": {         "id": {{user_id}},         "name": {{name}}       },       "reg_date": {{reg_date}},       "notice_start": {{notice_start}},       "state": {{state}},       "view": {{view}},     },...   ] }</pre>

	<pre>         ]     } </pre>
--	------------------------------

## ii. POST

/api/notices/	공지사항을 생성하는 API
Request	<pre> {     "title": String,     "notice_start": LocalDateTime,     "content": String } </pre>
Response	<pre> {     "success": true,     "code": 0,     "data": [         {             "title": {{title}},             "notice_start": {{notice_start}},             "writer": {                 "id": {{user_id}},                 "name": {{name}}             },             "reg_date": {{reg_date}},             "content": {{content}}         }     ] } </pre>

### iii. PUT

/api/notices/{notice_id}	공지사항을 수정하는 API
Request	<pre>{   "title": String,   "state": enum,   "notice_start": LocalDateTime,   "mod_reason": String,   "content": String }</pre>
Response	<pre>{   "success": true,   "code": 0,   "data": [     {       "title": {{title}},       "state": {{state}},       "notice_start": {{notice_start}},       "reg_user": {         "id": {{user_id}},         "name": {{name}}       },       "mod_user": {         "id": {{user_id}},         "name": {{name}}       },       "reg_date": {{reg_date}},       "mod_date": {{mod_date}},       "view": {{view}},       "mod_reason": {{mod_reason}},       "content": {{content}}     }   ] }</pre>

### iv. DELETE

/api/notices/{notice_id}	공지사항을 삭제하는 API
Request	/api/notices/{notice_id}
Response	<pre>{   "success": true,   "code": 0 }</pre>

- c. FAQ("/faq")  
i. GET

/api/faq/search/	모든 faq를 조회하는 API
Request	/api/faq/search/
Response	<pre>{   "success": true,   "code": 0,   "result": {     "data": {       "faq_no": {faq_no},       "faq_cate": {faq_cate},       "faq_title": {faq_title},       "faq_rdate": {faq_rdate},       "reg_user": {reg_user},       "faq_sdate": {faq_sdate},       "faq_state": {faq_state}     }   } }</pre>

/api/faq/search/	조건을 포함한 faq를 조회하는 API
Request	<pre>{   "faq_no": Long,   "faq_rdate": LocalDateTime,   "faq_sdate": LocalDateTime,   "reg_user": User,   "faq_title": String }</pre>
Response	<pre>{   "success": true,   "code": 0,   "result": {     "data": {       "faq_no": {faq_no},       "faq_cate": {faq_cate},       "faq_title": {faq_title},       "faq_rdate": {faq_rdate},       "reg_user": {reg_user},       "faq_sdate": {faq_sdate}, </pre>



	<pre>         "faq_state" : {faq_state}       }     }   } } </pre>
--	--

## ii. POST

/api/faq/register	faq를 등록하는 API
Request	<pre> {   "faq_cate" : enum,   "faq_title" : String,   "reg_user" : User,   "faq_rdate" : LocalDateTime,   "faq_sdate" : LocalDateTime,   "faq_content" : String } </pre>
Response	<pre> {   "success": true,   "code": 0,   "result" : {     "data" : {       "faq_cate" : {faq_cate},       "faq_title" : {faq_title},       "faq_rdate" : {faq_rdate},       "reg_user" : {reg_user},       "faq_sdate" : {faq_sdate}     }   } } </pre>

## iii. PUT

/api/faq/modify	faq를 수정하는 API
Request	<pre> {   "faq_cate" : enum,   "faq_title" : String,   "reg_user" : User,   "mod_user" : User,   "faq_mdate" : LocalDateTime,   "faq_rdate" : LocalDateTime,   "faq_sdate" : LocalDateTime, </pre>

	<pre>"faq_content" : String, "faq_state" : enum, "faq_reason" : String }</pre>
Response	<pre>{   "success": true,   "code": 0,   "result": {     "data": {       "faq_cate" : {faq_cate},       "faq_title" : {faq_title},       "faq_rdate" : {faq_rdate},       "reg_user" : {reg_user},       "mod_user" : {mod_user},       "faq_mdate" : {faq_mdate},       "faq_sdate" : {faq_sdate},       "faq_state" : {faq_state},       "faq_reason" : {faq_reason},       "faq_content" : {faq_content}     }   } }</pre>