

# RESTful Notes API Guide

:

# Table of Contents

1. Auth (Auth)	1
1.1. Auth Auth Auth	1
1.2. Auth Auth Auth	2
2. Sign (Sign)	4
2.1. Auth	4
2.2. Auth Auth Auth	5
2.3. Auth	7
2.4. Auth Auth Auth	8
3. User (User)	10
3.1. Auth Auth Auth	10
3.2. Auth Auth	12
3.3. Auth Auth	13
3.4. Auth Auth	14
3.5. Auth Auth	16
3.6. Auth	17
4. StudyClass (StudyClass)	19
4.1. Auth Auth (Auth)	19
4.2. Auth Auth (Auth)	20
4.3. Auth	22
4.4. Auth	24
4.5. Auth	26
4.6. Auth	27
4.7. Auth Auth	28
4.8. Auth	29

# Chapter 1. 认证 (Auth)

## 1.1. 认证 接口 列表

- POST /auth/send/email

请求 示例

```
POST /api/auth/send/email?email=test%40test.com HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 33
Host: localhost:8080

{
  "email" : "test@test.com"
}
```

响应 示例

Name	Description
Accept	application/json

请求 参数

Name	Type	Required	Description
email	String	true	邮箱

响应 参数

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 41

{
  "output" : 1,
  "msg" : ""
}
```

返回 数据

Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□

## 1.2. □□□ □□□□ □□

- GET /auth/check/email

□□ □□

```
GET /api/auth/check/email?authNum=999999&email=test%40test.com HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 58
Host: localhost:8080

{
  "authNum" : "999999",
  "email" : "test@test.com"
}
```

□□ □□

Name	Description
Accept	application/json

□□ □□

Name	Type	Required	Description
authNum	String	true	□□□□
email	String	true	□□□

□□ □□

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 41
```

```
{
  "output" : 1,
  "msg" : ""
}
```

□□ □□

Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□

# Chapter 2. 注册 (Sign)

## 2.1. 注册接口

- POST /sign/signup

请求

```
POST
/api/sign/signup?uid=userId001&type=none&password=pwd1234&name=%ED%9A%8C%EC%9B%90%EB%A
A%85&email=user%40mail.com&phone=01012345678&address=%EC%84%9C%EC%9A%B8%ED%8A%B9%EB%B3
%84%EC%8B%9C+%EA%B8%88%EC%B2%9C%EA%B5%AC+%EA%B0%80%EC%82%B0%EB%94%94%EC%A7%80%ED%84%B8
1%EB%A1%9C+168&addressDetail=%EC%9A%B0%EB%A6%BC%EB%9D%BC%EC%9D%B4%EC%98%A8%EC%8A%A4%EB
%B0%B8%EB%A6%AC HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 274
Host: localhost:8080

{
  "uid" : "userId001",
  "type" : "none",
  "password" : "pwd1234",
  "name" : "用户",
  "email" : "user@mail.com",
  "phone" : "01012345678",
  "address" : "北京市海淀区 168",
  "addressDetail" : "北京市海淀区"
}
```

响应

Name	Description
Accept	application/json

参数

Name	Type	Required	Description
uid	String	true	uid (sns:uid)
type	String	true	注册类型 (none, sns)
password	String	true	密码
name	String	true	姓名
email	String	true	邮箱

Name	Type	Required	Description
phone	String		手机号码('-' 号 忽略)
address	String		地址
addressDetail	String		详细地址

返回 数据

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 81

{
  "output" : 0,
  "msg" : "用户 uid 已经 注册过。"
}
```

返回 数据

Path	Type	Description
output	Number	是否成功
msg	String	返回 消息
data.id	Number	注册成功 用户 ID

## 2.2. 注册 或 登录 接口

- POST /sign/regProfile

返回 数据

```
POST /api/sign/regProfile?id=1&nickname=nick01&age=24&gender=F&img= HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 93
Host: localhost:8080
```

```
{
  "id" : 1,
  "nickname" : "nick01",
  "age" : 24,
  "gender" : "F",
  "img" : null
}
```

□□ □□

Name	Description
Accept	application/json

□□ □□

Name	Type	Required	Description
id	Number	true	□□ □□□
nickname	String	true	□□□
age	Number		□□
gender	String	true	□□(M, F, ")
img	Object		□□□

□□ □□

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 41
```

```
{
  "output" : 1,
  "msg" : "□□"
}
```

□□ □□



Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□

## 2.3. □□□

- GET /sign/login

□□ □□

```
GET
/api/sign/login?uid=userId001&password=pwd1234&token=ebbWwQ22T_OMpoJYq36vjc%3AAPA91bFs
AjbvRRiiE63qZURN83Hsn8nKTXZLYRD8ztrtYRkQ17_gT5S86P6qrrEnsI-
o_57w6SrybTb_OudCt6rFTvXQJVdfv5r0f47ZzMAxZSH_I1knJ8kR5gO_bNtUkLzm8L1-
Yh7sU&deviceId=a5zz3d1b-c26a-4111-ae03-d1e92336c14f&type=none HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: application/json
Content-Length: 311
Host: localhost:8080

{
  "uid" : "userId001",
  "password" : "pwd1234",
  "token" :
"ebbWwQ22T_OMpoJYq36vjc:APA91bFsAjbvRRiiE63qZURN83Hsn8nKTXZLYRD8ztrtYRkQ17_gT5S86P6qrrE
nsI-o_57w6SrybTb_OudCt6rFTvXQJVdfv5r0f47ZzMAxZSH_I1knJ8kR5gO_bNtUkLzm8L1-Yh7sU",
  "deviceId" : "a5zz3d1b-c26a-4111-ae03-d1e92336c14f",
  "type" : "none"
}
```

□□ □□

Name	Description
Accept	application/json

□□ □□

Name	Type	Required	Description
uid	String	true	uid (□□□□:□□□, sns□□□:uid□)
password	String	true	□□□□
token	String		□□ □□□ □□ fcm □□□
deviceId	String		□□□□ □□ □□□
type	String	true	□□□ □□ (none, sns)

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 270

{
  "output" : 1,
  "msg" : "",
  "data" : {
    "id" : 54,
    "token" :
"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NCIsInJvbmVzIjpbIlJPTeVfVVNFUiJdLCJ1c2VyUGsiOiI1NCIsIm1ldCI6MTYzOTk2NTE4MywiZXhwIjoxNjQyNTU3MTgzfQ.oEStP6RsheI90vsyqk2l8guQas_LnVtUh-SCxrXx0HA"
  }
}

```

Path	Type	Description
output	Number	0000
msg	String	00 000
data.id	Number	00 000
data.token	String	000 00000 0000 JWT 000

## 2.4. 00 00000 00 00

- GET /sign/exists/social

```
GET /api/sign/exists/social?uid=userId001&type=sns HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 46
Host: localhost:8080
```

```
{
  "uid" : "userId001",
  "type" : "sns"
}
```

요청 파라미터

Name	Description
Accept	application/json

요청 본문

Name	Type	Required	Description
uid	String	true	uid (sns: sns, uid)
type	String	true	sns (none, sns)

응답 헤더

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 114
```

```
{
  "output" : 0,
  "msg" : "sns sns. sns sns."
}
```

응답 본문

Path	Type	Description
output	Number	0
msg	String	sns sns.

# Chapter 3. 사용자 (User)

## 3.1. 사용자 관련 API

- GET /user/myClass

요청

```
GET /api/user/myClass HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiUk9MRV9VU0VSIl0sInVzZXJQayI6IjEiLCJpYXQiOiJlMzk0NTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Host: localhost:8080
```

응답

Name	Description
Accept	application/json
X-AUTH-TOKEN	사용자 인증 토큰 (JWT 형식)

참고

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 477

```

```

{
  "output" : 1,
  "msg" : "",
  "data" : {
    "hostClassList" : [ {
      "id" : 73,
      "name" : "JAVA 클래스",
      "info" : "이 클래스는 JAVA 클래스입니다.",
      "img" : ""
    } ],
    "participantClassList" : [ {
      "id" : 3,
      "name" : "클래스 이름",
      "info" : "이 클래스는 클래스입니다.",
      "img" : null
    } ]
  }
}

```

클래스

Path	Type	Description
output	Number	클래스
msg	String	클래스
data.hostClassList	Array	클래스 목록
data.hostClassList{}	Object	클래스 목록의 객체
data.hostClassList[].id	Number	클래스 목록의 ID > 클래스
data.hostClassList[].name	String	클래스 목록의 이름 > 클래스
data.hostClassList[].info	String	클래스 목록의 정보 > 클래스
data.hostClassList[].img	String	클래스 목록의 이미지 > 클래스
data.participantClassList	Array	참여자 클래스 목록
data.participantClassList{}	Object	참여자 클래스 목록의 객체
data.participantClassList[].id	Number	참여자 클래스 목록의 ID > 클래스
data.participantClassList[].name	String	참여자 클래스 목록의 이름 > 클래스



```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 158
```

```
{
  "output" : 1,
  "msg" : "",
  "data" : {
    "id" : 1,
    "nickname" : "nick01",
    "gender" : "F",
    "age" : 24,
    "img" : ""
  }
}
```

□□ □□

Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□
data.id	Number	□□□ □□
data.nickname	String	□□□ □□□
data.gender	String	□□□ □□
data.age	Number	□□□ □□
data.img	String	□□□ □□□ □□□

### 3.3. □□ □□□ □□

- GET /user/myPush

□□ □□

```
GET /api/user/myPush HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiI0IiwiaWF0IjE2MzkyMzY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Host: localhost:8080
```

응답

Name	Description
Accept	application/json
X-AUTH-TOKEN	JWT 토큰 (JWT 토큰)

응답

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 58

{
  "output" : 1,
  "msg" : "완료",
  "data" : "Y"
}
```

응답

Path	Type	Description
output	Number	결과
msg	String	메시지
data	String	데이터 (Y:on, N:off)

### 3.4. PUT /user/profile

- PUT /user/profile

응답



```
PUT /api/user/profile?nickname=newNick&age=25&gender=M&img= HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiI0IiwiaWF0Ij0iUk9MRV9VU0VSIl0sInVzZXJQayI6IjEiLCJpYXQiOiJlE2Mzk0NTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Content-Length: 81
Host: localhost:8080

{
  "nickname" : "newNick",
  "age" : 25,
  "gender" : "M",
  "img" : null
}
```

API 명세

Name	Description
Accept	application/json
X-AUTH-TOKEN	JWT 토큰 (API 호출 시 필수)

요청 파라미터

Name	Type	Required	Description
nickname	String	true	닉네임
age	Number	true	나이
gender	String	true	성별(M, F, ")
img	Object		프로필 이미지 (null이면 기본 이미지 사용)

응답

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 41

```

```

{
  "output" : 1,
  "msg" : ""
}

```

□□ □□

Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□

### 3.5. □□ □□□ □□

- PUT /user/push

□□ □□

```

PUT /api/user/push/Y HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiUk9MRV9VU0VSIl0sInVzZXJQayI6IjEiLCJpYXQiOiJlMzkyNTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Host: localhost:8080

```

□□ □□

Name	Description
Accept	application/json
X-AUTH-TOKEN	□□□ □□□□□ □□□□ □□ □□□ (□□ □□ □□)

□□ □□□□ ./api/user/push/{pushYN}

Path Parameter	Description
pushYN	□□ □□□. (Y:on, N:off)

□□ □□

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 41

{
  "output" : 1,
  "msg" : "□□"
}
```

□□ □□

Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□

## 3.6. □□□□

- DELETE /user/

□□ □□

```
DELETE /api/user/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiI0siUk9MRV9VU0VSIl0sInVzZXJQayI6IjEiLCJpYXQiOiJlMzk0NTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Host: localhost:8080
```

□□ □□

Name	Description
Accept	application/json
X-AUTH-TOKEN	□□□ □□□□□ □□□□ □□ □□□ (□□ □□ □□)

□□ □□

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 41
```

```
{
  "output" : 1,
  "msg" : ""
}
```

□□ □□

Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□

# Chapter 4. 클래스 (StudyClass)

## 4.1. API (API)

- GET /studyclass/list

API

```
GET /api/studyclass/list?sort=latest&page=1 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiUk9MRV9VU0VSIl0sInVzZXJQayI6IjEiLCJpYXQiOiJlMzk0NTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Content-Length: 40
Host: localhost:8080

{
  "sort" : "latest",
  "page" : 1
}
```

API

Name	Description
Accept	application/json
X-AUTH-TOKEN	JWT 토큰 (JWT 토큰)

API

Name	Type	Required	Description
sort	String	true	정렬 (latest:최신, old:가장 오래)
page	Number	true	페이지

API

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 250

```

```

{
  "output" : 1,
  "msg" : "",
  "data" : [ {
    "id" : 3,
    "name" : "한글 2020 00 00",
    "info" : "한글 2020 00. 000000 00 00 00000.",
    "img" : null
  } ]
}

```

한글 2020

Path	Type	Description
output	Number	한글 2020
msg	String	한글 2020
data	Array	한글 2020 00 00 0000
data{}	Object	한글 2020 00 00 0000 0 0000
data[].id	Number	한글 2020
data[].name	String	한글 2020
data[].info	String	한글 2020
data[].img	String	한글 2020

## 4.2. 한글 2020 (한글)

- GET /studyclass/search

한글 2020

```

GET /api/studyclass/search?search=%EC%8A%A4%ED%84%B0%EB%94%94&page=1 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiI0IiwiaWF0IjE2Mzk0NTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Content-Length: 45
Host: localhost:8080

{
  "search" : "",
  "page" : 1
}

```

API 명세

Name	Description
Accept	application/json
X-AUTH-TOKEN	JWT 토큰 (로그인 후 발급)

요청 파라미터

Name	Type	Required	Description
search	String		검색어
page	Number	true	페이지 번호

응답 데이터

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 250

```

```

{
  "output" : 1,
  "msg" : "",
  "data" : [ {
    "id" : 3,
    "name" : "한글 2020 00 00",
    "info" : "한글 2020 00. 202000 00 00 202000.",
    "img" : null
  } ]
}

```

한글 2020

Path	Type	Description
output	Number	한글 2020
msg	String	한글 2020
data	Array	한글 2020 00 00 00
data{}	Object	한글 2020 00 00 00 0 00
data[].id	Number	한글 2020
data[].name	String	한글
data[].info	String	한글 2020
data[].img	String	한글 2020

## 4.3. 한글 2020

- GET /studyclass/detail

한글 2020



```
GET /api/studyclass/detail/1 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMlOiI0IiwiaWF0Ij0iYXZlZXJQayI6IjEiLCJpYXQiOiJlZ2MzNTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Host: localhost:8080
```

응답

Name	Description
Accept	application/json
X-AUTH-TOKEN	인증 토큰 (JWT 형식)

응답 본문: /api/studyclass/detail/{classId}

Path Parameter	Description
classId	클래스 ID

응답

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 107
```

```
{
  "output" : 0,
  "msg" : "클래스 정보 조회 성공."
}
```

응답

Path	Type	Description
output	Number	결과 코드
msg	String	메시지
data	Object	데이터
data.id	Number	클래스 ID
data.name	String	클래스명



API 명세

Name	Type	Required	Description
name	String	true	이름
info	String	true	이름에 대한 설명
img	Object		이름에 대한 이미지

API 응답

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 393
```

```
{
  "output" : 1,
  "msg" : "성공",
  "data" : [ {
    "id" : 73,
    "name" : "JAVA 이름",
    "info" : "이름에 대한 설명",
    "img" : ""
  }, {
    "id" : 3,
    "name" : "이름에 대한 설명",
    "info" : "이름에 대한 설명",
    "img" : null
  } ]
}
```

API 명세

Path	Type	Description
output	Number	결과
msg	String	메시지
data	Array	데이터
data{}	Object	데이터 객체
data[].id	Number	아이디
data[].name	String	이름
data[].info	String	이름에 대한 설명



```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 107

```

```

{
  "output" : 0,
  "msg" : "0000 00 00000000. 00 000000."
}

```

00 00

Path	Type	Description
output	Number	0000
msg	String	00 0000

## 4.6. 0000

- POST /studyclass/participate

00 00

```

POST /api/studyclass/participate/3 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiI0IiwiaWF0Ij0iInVzZXJQayI6IjEiLCJpYXQiOiJlMzk0NTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Host: localhost:8080

```

00 00

Name	Description
Accept	application/json
X-AUTH-TOKEN	000 000000 0000 00 0000 (00 00 00)

00 0000 ./api/studyclass/participate/{classId}

Path Parameter	Description
classId	000000 0000 00

□□ □□

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 77
```

```
{
  "output" : 0,
  "msg" : "□□ □□ □□□□□□."
}
```

□□ □□

Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□

## 4.7. □□□ □□

- DELETE /studyclass/user

□□ □□

```
DELETE /api/studyclass/user?classId=1&deleteUser=9 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
X-AUTH-TOKEN:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxIiwicm9sZXMiOiI0IiwiaWF0Ij0iInVzZXJQayI6IjEiLCJpYXQiOiJlMzZk0NTc0NTIsImV4cCI6MTY0MjA0OTQ1Mn0.afQNA4zfLFMzSXuLG75SPSopDPtkg97Kh9TBptaJDYI
Content-Length: 42
Host: localhost:8080

{
  "classId" : 1,
  "deleteUser" : 9
}
```

□□ □□

Name	Description
Accept	application/json



□□ □□

Name	Description
Accept	application/json
X-AUTH-TOKEN	□□□ □□□□□ □□□□ □□ □□□ (□□ □□ □□)

□□ □□□□ ./api/studyclass/{classId}

Path Parameter	Description
classId	□□□ □□□ □□

□□ □□

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 41

{
  "output" : 1,
  "msg" : "□□"
}
```

□□ □□

Path	Type	Description
output	Number	□□□□
msg	String	□□ □□□