

Server API 호출 가이드

v1.1

목차

1. 회원 가입

A. OCR

B. 회원 가입

2. 로그인

A. 로그인

3. 강의 평가

A. 작성

B. 삭제

C. 수정

1. 회원 가입

A. OCR 서비스 호출 - 요청 URL (POST, http://서버주소/signUp/ocr)

파라미터 이름	형식
img	Multipart

HTTP Request 예시

```
1  POST /signUp/ocr HTTP/1.1
2  Host: localhost:8080
3  Content-Length: 229
4  Content-Type: multipart/form-data;
    boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW
5
6  ----WebKitFormBoundary7MA4YWxkTrZu0gW
7  Content-Disposition: form-data;
    name="img"; filename="/C:/Users/
    Hyeonho/Documents/카카오톡 받은 파일/
    studCard_card_resized.jpg"
8  Content-Type: image/jpeg
9
10 (data)
11 ----WebKitFormBoundary7MA4YWxkTrZu0gW
```

HTTP Response 예시

Response Body	HTTP Response code	비고
{ "name": String, "department": String, "studNo": number }	200	OCR 서비스 호출 성공
"service error"	500	OCR 서비스 호출 실패

예시

```
1  {
2    "name": "고일석",
3    "department": "컴퓨터과학과",
4    "studNo": 201611771
5  }
```

A. 회원 가입 - 요청 URL (POST, http://서버주소/signUp)

파라미터 이름	형식
studNo	JSON : number
password	JSON : String
name	JSON : String
department	JSON : String
email	JSON : String

HTTP Request 예시

```
1  POST /signUp HTTP/1.1
2  Host: localhost:8080
3  Content-Type: application/json
4  Content-Length: 130
5
6  {
7    "studNo":201713883,
8    "password":"qlalfqjsgh",
9    "name":"신현호",
10   "department":"컴퓨터공학부",
11   "email":"ruldarm00@kyonggi.ac.kr"
12 }
```

HTTP Response 예시

Response Body	HTTP Response code	비고
true	200	회원 가입 정상 처리
false	400	회원 가입 실패

2. 로그인

A. 로그인 - 요청 URL (POST, http://서버주소/signIn)

파라미터 이름	형식
email	JSON : String
password	JSON : String

HTTP Request 예시

```
1  POST /signIn HTTP/1.1
2  Host: localhost:8080
3  Content-Type: application/json
4  Content-Length: 73
5
6  {
7      "email": "ruldarm00@kyonggi.ac.kr",
8      "password": "qlalfqjsgh"
9  }
10
```

HTTP Response 예시

Response Body	HTTP Response code	비고
"SignIn Success."	200	로그인 성공
"Server in Error."	500	로그인 실패, 서버 에러
"Password not matching."	400	로그인 실패, 비밀번호 오류

3. 강의 평가

A. 작성- 요청 URL (POST, http://서버주소/critic/apply)

파라미터 이름	형식
email	JSON : String
content	JSON : String
profName (교수명)	JSON : String
lectureName (강의명)	JSON : String
grade (성적)	JSON : String (AP, A, BP, B, CP ... F)

HTTP Request 예시

```
1  POST /critic/apply HTTP/1.1
2  Host: localhost:8080
3  Content-Type: application/json
4  Content-Length: 148
5
6  {
7      "email": "ruldarm00@kyonggi.ac.kr",
8      "content": "zzzzzzzzzz test",
9      "profName": "권기현",
10     "lectureName": "소프트웨어 공학",
11     "grade": "A"
12 }
13
```

HTTP Response 예시

Response Body	HTTP Response code	비고
{ "lectureName": String, "profName": String, "content": String, "grade": String, "writer": null, "cid": number }	200	작성 성공 Response Body는 DB에 저장한 강의 평가 정보를 JSON 형식으로 표현한 데 이터
false	400	서버 오류, 작성 실패

B. 삭제 - 요청 URL (POST, http://서버주소/critic/rm)

파라미터 이름	형식
criticId	JSON : number

HTTP Request 예시

```

1  POST /critic/rm HTTP/1.1
2  Host: localhost:8080
3  Content-Type: application/json
4  Content-Length: 23
5
6  {
7    "criticId": 5
8  }
```

HTTP Response 예시

Response Body	HTTP Response code	비고
{ "criticId":number }	200	삭제 성공, 삭제한 강의 평가의 번호를 반환
{ "criticId":number }	500	삭제 실패, 삭제에 실패한 강의 평가의 번호를 반환

C. 수정 - 요청 URL (POST, http://서버주소/critic/update)

파라미터 이름	형식
criticId	JSON : String
content	JSON : String, 옵션
grade	JSON : String, 옵션
lectureName	JSON : String, 옵션
profName	JSON : String, 옵션

HTTP Request 예시

```
1  POST /critic/update HTTP/1.1
2  Host: localhost:8080
3  Content-Type: application/json
4  Content-Length: 53
5
6  {
7    "criticId":2,
8    "content":"강의 평가 수정합니다."
9  }
```

HTTP Response 예시

Response Body	HTTP Response code	비고
{ "criticId":number }	200	수정 성공, 수정한 강의 평가의 번호를 반환
{ "criticId":number }	500	수정 실패, 수정에 실패한 강의 평가의 번호를 반환