

# API Documentation

시대마켓

소공24

2024-12-15

Revision history

버전	일자	변경 내역	작 성 자
v 1.0	2024-12-15	초안 작성	최승우

# Table of contents

인증.....	4
사용자(User) API.....	4
회원가입.....	4
로그인.....	4
사용자 정보 수정.....	5
내 정보 조회.....	6
강좌(Course) API.....	7
강좌 목록 조회.....	7
강좌 상세 정보 조회.....	7
노트(Note) API.....	8
전체 노트 조회.....	8
강좌별 노트 목록 조회.....	9
노트 상세 정보 조회.....	10
노트 업로드.....	11
노트 좋아요.....	12
노트 싫어요.....	13
노트 구매.....	14
노트 다운로드.....	16
노트 삭제.....	17

## 인증

대부분의 API 엔드포인트는 인증이 필요합니다. 인증은 JWT(JSON Web Token)를 사용하여 수행되며, 로그인 시 발급된 토큰을 **Authorization** 헤더에 포함시켜 요청해야 합니다.

---

## 사용자(User) API

### 회원가입

- URL: `/api/users/signup`
- 메서드: **POST**
- 설명: 새로운 사용자를 회원가입합니다.
- 요청 본문:  
json

```
{  
  "email": "user@example.com",  
  "password": "securePassword123",  
  "username": "사용자이름"  
}
```

- 응답:
  - 성공 (**201 Created**):  
json  

```
{  
  "message": "User created",  
  "userId": 1  
}
```
  - 오류 (**400 Bad Request**): 이메일이 이미 사용 중인 경우  
json  

```
{  
  "error": "Email already in use"  
}
```

### 로그인

- URL: `/api/users/login`
- 메서드: `POST`
- 설명: 사용자 로그인을 수행하고 JWT 토큰을 발급받습니다.
- 요청 본문:

json

```
{
  "email": "user@example.com",
  "password": "securePassword123"
}
```

- 응답:
  - 성공 (200 OK):

json

```
{
  "message": "Successful login",
  "token": "JWT_TOKEN"
}
```

- 오류 (400 Bad Request): 잘못된 자격 증명

json

```
{
  "error": "Invalid credentials"
}
```

## 사용자 정보 수정

- URL: `/api/users`
- 메서드: `PUT`
- 설명: 인증된 사용자의 정보를 수정합니다.
- 헤더: `Authorization: Bearer <JWT_TOKEN>`
- 요청 본문

json

```
{
  "password": "newSecurePassword123",
  "username": "새로운이름"
}
```

- 응답:

- 성공 (200 OK):

json

```
{
  "message": "User information updated"
}
```

- 오류 (404 Not Found): 사용자를 찾을 수 없는 경우

json

```
{
  "error": "User not found"
}
```

## 내 정보 조회

- URL: `/api/users/me`
- 메서드: `GET`
- 설명: 인증된 사용자의 정보를 조회합니다.
- 헤더: `Authorization: Bearer <JWT_TOKEN>`
- 응답:

- 성공 (200 OK):

json

```
{
  "id": 1,
  "email": "user@example.com",
  "username": "사용자이름",
  "point": 100,
  "accessibleNoteIDs": [1, 2, 3]
}
```

- 오류 (404 Not Found): 사용자를 찾을 수 없는 경우

json

```
{
  "error": "User not found"
}
```

---

## 강좌(Course) API

### 강좌 목록 조회

- URL: `/api/courses`
- 메서드: `GET`
- 설명: 특정 학년과 학기에 해당하는 강좌 목록을 조회합니다.
- 헤더: `Authorization: Bearer <JWT_TOKEN>`
- 쿼리 파라미터:
  - `grade` : 학년 (예: 1)
  - `semester` : 학기 (예: 2)
- 응답:
  - 성공 (**200 OK**):  
json

```
[
  {
    "id": 1,
    "name": "수학",
    "grade": 1,
    "semester": 2
  },
  {
    "id": 2,
    "name": "과학",
    "grade": 1,
    "semester": 2
  }
]
```

- 오류 (**500 Internal Server Error**):  
json

```
{
  "error": "오류 메시지"
}
```

### 강좌 상세 정보 조회

- URL: `/api/courses/detail/:courseID`
- 메서드: `GET`

- 설명: 특정 강좌의 상세 정보를 조회합니다.
  - 헤더: `Authorization: Bearer <JWT_TOKEN>`
  - URL 파라미터:
    - `courseID`: 강좌 ID
  - 응답:
    - 성공 (**200 OK**):  
json

```
{  "id": 1,  "name": "수학",  "grade": 1,  "semester": 2,  "description": "수학 과목에 대한 상세 설명."}
```
    - 오류 (**500 Internal Server Error**):  
json

```
{  "error": "오류 메시지"}
```
- 

## 노트(Note) API

### 전체 노트 조회

- URL: `/api/notes`
- 메서드: `GET`
- 설명: 모든 노트를 조회합니다.
- 헤더: `Authorization: Bearer <JWT_TOKEN>`
- 응답:
  - 성공 (**200 OK**):  
json

```
[  {    "id": 1,    "uploaderID": 1,    "courseID": 1,
```



```

        "title": "미적분 노트",
        "description": "미적분 관련 노트입니다.",
        "fileURL":
"http://example.com/uploads/1630425600000-note.pdf",
        "previewURL":
"http://example.com/uploads/previews/1630425600000-note-preview.png",
        "likesCount": 10,
        "dislikesCount": 2
    },
    ...
]

```

- 오류 (500 Internal Server Error):  
json

```

{
    "error": "오류 메시지"
}

```

## 강좌별 노트 목록 조회

- URL: `/api/notes/:courseID`
- 메서드: GET
- 설명: 특정 강좌에 속한 노트 목록을 조회합니다.
- 헤더: `Authorization: Bearer <JWT_TOKEN>`
- URL 파라미터:
  - `courseID`: 강좌 ID

- 응답:
  - 성공 (200 OK):  
json

```

[
    {
        "id": 1,
        "uploaderID": 1,
        "courseID": 1,
        "title": "미적분 노트",
        "description": "미적분 관련 노트입니다.",
        "fileURL":
"http://example.com/uploads/1630425600000-note.pdf",

```

```

        "previewURL":
"http://example.com/uploads/previews/1630425600000-note-preview.png",
        "likesCount": 10,
        "dislikesCount": 2
    },
    ...
]

```

- 오류 (**500 Internal Server Error**):  
json

```

{
    "error": "오류 메시지"
}

```

## 노트 상세 정보 조회

- URL: `/api/notes/detail/:noteID`
- 메서드: GET
- 설명: 특정 노트의 상세 정보를 조회합니다.
- 헤더: `Authorization: Bearer <JWT_TOKEN>`
- URL 파라미터:
  - `noteID`: 노트 ID

- 응답:

- 성공 (**200 OK**):  
json

```

{
    "id": 1,
    "uploaderID": 1,
    "courseID": 1,
    "title": "미적분 노트",
    "description": "미적분 관련 노트입니다.",
    "fileURL":
"http://example.com/uploads/1630425600000-note.pdf",
    "previewURL":
"http://example.com/uploads/previews/1630425600000-note-preview.png",
    "likesCount": 10,
    "dislikesCount": 2
}

```

```
}
```

- 오류 (**500 Internal Server Error**):  
json

```
{
```

```
    "error": "오류 메시지"
```

```
}
```

## 노트 업로드

- URL: `/api/notes`
- 메서드: `POST`
- 설명: 새로운 노트를 업로드합니다. PDF 파일만 업로드 가능합니다.
- 헤더:

- `Authorization: Bearer <JWT_TOKEN>`
- `Content-Type: multipart/form-data`

- 폼 데이터:
  - `uploaderID`: 업로더 사용자 ID
  - `courseID`: 강좌 ID
  - `title`: 노트 제목
  - `description` (선택): 노트 설명
  - `file`: PDF 파일

- 응답:
  - 성공 (**201 Created**):  
json

```
{
```

```
    "note": {  
        "id": 1,  
        "uploaderID": 1,  
        "courseID": 1,  
        "title": "미적분 노트",  
        "description": "미적분 관련 노트입니다.",  
        "fileURL":
```

```
        "http://example.com/uploads/1630425600000-note.pdf",
```

```
        "previewURL":
```

```
        "http://example.com/uploads/previews/1630425600000-note-preview.png",
```

```
        "likesCount": 0,
```

```
        "dislikesCount": 0
```

```
    },
    "previewImage":
"uploads/previews/1630425600000-note-preview.png"
  }
}
```

- 오류 (**400 Bad Request**): 파일이 없거나 PDF가 아닌 경우  
json

```
{
  "error": "File is required"
}
json
```

```
{
  "error": "Only PDF files are allowed"
}
```

- 오류 (**404 Not Found**): 업로더 사용자 또는 강좌를 찾을 수 없는 경우  
json

```
{
  "error": "User not found"
}
```

## 노트 좋아요

- URL: `/api/notes/:noteID/like`
- 메서드: `POST`
- 설명: 특정 노트에 좋아요를 추가합니다.
- 헤더: `Authorization: Bearer <JWT_TOKEN>`
- URL 파라미터:
  - `noteID`: 노트 ID
- 응답:
  - 성공 (**200 OK**): 업데이트된 노트 정보 반환  
json

```
{
  "id": 1,
  "uploaderID": 1,
  "courseID": 1,
```

```

        "title": "미적분 노트",
        "description": "미적분 관련 노트입니다.",
        "fileURL":
"http://example.com/uploads/1630425600000-note.pdf",
        "previewURL":
"http://example.com/uploads/previews/1630425600000-note-preview.png",
        "likesCount": 11,
        "dislikesCount": 2
    }

```

- 오류 (**404 Not Found**): 노트를 찾을 수 없는 경우  
json

```

{
    "error": "Note not found"
}

```

- 오류 (**500 Internal Server Error**):  
json

```

{
    "error": "오류 메시지"
}

```

## 노트 싫어요

- URL: `/api/notes/:noteID/dislike`
- 메서드: **POST**
- 설명: 특정 노트에 싫어요를 추가합니다.
- 헤더: **Authorization: Bearer <JWT\_TOKEN>**
- URL 파라미터:
  - **noteID**: 노트 ID
- 응답:
  - 성공 (**200 OK**): 업데이트된 노트 정보 반환  
json

```

{
    "id": 1,
    "uploaderID": 1,
    "courseID": 1,

```

```

        "title": "미적분 노트",
        "description": "미적분 관련 노트입니다.",
        "fileURL":
"http://example.com/uploads/1630425600000-note.pdf",
        "previewURL":
"http://example.com/uploads/previews/1630425600000-note-preview.png",
        "likesCount": 10,
        "dislikesCount": 3
    }

```

- 오류 (**404 Not Found**): 노트를 찾을 수 없는 경우  
json

```

{
    "error": "Note not found"
}

```

- 오류 (**500 Internal Server Error**):  
json

```

{
    "error": "오류 메시지"
}

```

## 노트 구매

- URL: [/api/notes/:noteID/purchase](#)
- 메서드: POST
- 설명: 특정 노트를 구매하여 접근 권한을 얻습니다.
- 헤더: **Authorization: Bearer <JWT\_TOKEN>**
- URL 파라미터:
  - **noteID**: 노트 ID
- 응답:
- 성공 (**200 OK**): 사용자 정보 및 노트 정보 반환  
json

```

{
    "user": {
        "id": 1,
        "email": "user@example.com",

```

```

        "username": "사용자이름",
        "point": 99,
        "accessibleNoteIDs": [1, 2, 3]
    },
    "note": {
        "id": 1,
        "uploaderID": 1,
        "courseID": 1,
        "title": "미적분 노트",
        "description": "미적분 관련 노트입니다.",
        "fileURL":
"http://example.com/uploads/1630425600000-note.pdf",
        "previewURL":
"http://example.com/uploads/previews/1630425600000-note-preview.png"
    },
    "likesCount": 10,
    "dislikesCount": 2
}

```

- 오류 **(403 Forbidden)**: 포인트 부족  
json

```

{
    "error": "Not enough points"
}

```

- 오류 **(404 Not Found)**: 사용자 또는 노트를 찾을 수 없는 경우  
json

```

{
    "error": "User not found"
}

```

```

{
    "error": "Note not found"
}

```

- 오류 **(500 Internal Server Error)**:  
json

```
{  
  
  "error": "오류 메시지"  
}
```

## 노트 다운로드

- **URL:** `/api/notes/:noteID/download`
- 메서드: **GET**
- 설명: 구매한 노트를 다운로드합니다.
- 헤더: **Authorization: Bearer <JWT\_TOKEN>**
- **URL** 파라미터:
  - **noteID**: 노트 ID
- 응답:
  - 성공 (**200 OK**): 파일 다운로드 시작
  - 오류 (**400 Bad Request**): 다운로드 권한이 없는 경우  
json

```
{  
  
  "error": "No download permissions"  
}
```

- 오류 (**404 Not Found**): 사용자 또는 노트를 찾을 수 없는 경우  
json

```
{  
  
  "error": "Note not found"  
}  
json
```

```
{  
  
  "error": "User not found"  
}
```

- 오류 (**500 Internal Server Error**): 파일 다운로드 실패  
json

```
{  
  
  "error": "Failed to download file"  
}
```



json

```
{
  "error": "오류 메시지"
}
```

## 노트 삭제

- **URL:** `/api/notes/:noteID`
- 메서드: **DELETE**
- 설명: 특정 노트를 삭제합니다.
- 헤더: **Authorization: Bearer <JWT\_TOKEN>**
- **URL** 파라미터:
  - **noteID**: 노트 ID
- 응답:
- 성공 (**200 OK**): 삭제된 노트 정보 반환

json

```
{
  "id": 1,
  "uploaderID": 1,
  "courseID": 1,
  "title": "미적분 노트",
  "description": "미적분 관련 노트입니다.",
  "fileURL":
"http://example.com/uploads/1630425600000-note.pdf",
  "previewURL":
"http://example.com/uploads/previews/1630425600000-note-preview.png",
  "likesCount": 10,
  "dislikesCount": 2
}
```

- 오류 (**404 Not Found**): 노트를 찾을 수 없는 경우

json

```
{
  "error": "Note not found"
}
```

- 오류 (**500 Internal Server Error**):

json

```
{  
  "error": "오류 메시지"  
}
```