

4차 기획 회의

날짜	@2026/01/12
상태	완료

흐름

- 프론트가 이미지 파일 업로드 (`multipart/form-data`)
- 서버가 파일을 **S3/Cloud Storage/서버 디스크**에 저장
- 서버가 DB에 **이미지 URL**(또는 파일 키) 저장
- 상품 조회 API 응답에 **imageUrl** 내려줌
- 프론트는 그대로 ``

예시 응답

```
json 코드 복사  
  
{  
  "productId": 1,  
  "name": "67W 충전기",  
  "imageUrl": "https://cdn.example.com/products/1/main.jpg"  
}
```

피그마에 로그인 추가하기

각 페이지 별 필요한 데이터

- 로그인 페이지

path

o `/auth/login`

GET

Body

userId	string	사용자 아이디
password	string	비밀번호

```
{
  "userId" : "abc123",
  "password" : "pw1234"
}
```

response form

userID	사용자 PK
accessToken	JWT Access Token
refreshToken	JWT Refresh Token (자동로그인용)
tokenType	"Bearer"
accessTokenExpiresIn	만료까지 초
refreshTokenExpiresIn	만료까지 초

```
{
  "isSuccess": true,
  "code": 1000,
  "message": "요청에 성공하였습니다.",
  "timestamp": "2026-01-02T01:05:46.861610",
  "result": {
    "accessToken": "eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9...",
    "refreshToken": "eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9...",
    "tokenType": "Bearer",
    "accessTokenExpiresIn": 3600,
    "refreshTokenExpiresIn": 1209600
  }
}
```

- 회원가입

희망 ID
희망 PW
ID 중복 여부
이름

학번
전화번호
단위

1-1. 아이디 중복 체크

Endpoint

```
GET /api/auth/check-username?username={username}
```

Request (Query)

필드	타입	필수	설명
username	string	O	사용자가 입력한 아이디

Success (200)

```
{
  " isSuccess":true,
  " code":200,
  " message":"사용 가능한 아이디입니다.",
  " result":{
    " available":true
  }
}
```

Fail 예시

- 이미 존재 (200으로 내려도 되고 409로 내려도 됨 / 둘 중 하나만 택하면 됨)

옵션 A: 200 + available=false

```
{
  " isSuccess":true,
  " code":200,
  " message":"이미 사용 중인 아이디입니다.",
```

```
"result": {"available": false}  
}
```

옵션 B: 409 Conflict

```
{  
    "isSuccess": false,  
    "code": 409,  
    "message": "이미 사용 중인 아이디입니다."  
}
```

1-2. 회원가입

Endpoint

```
POST /api/auth/signup
```

Headers

Key	Value
Content-Type	application/json

Body

```
{  
    "username": "doeun05",  
    "password": "Password123!",  
    "name": "김도은",  
    "unit": "컴퓨터공학부",  
    "studentId": "202411257",  
    "phone": "010-1234-5678"  
}
```

Body 필드 설명

필드	타입	필수	설명
username	string	O	아이디
password	string	O	비밀번호
name	string	O	이름
unit	string	O	단위(학부/부서 등)
studentId	string	O	학번
phone	string	O	전화번호

Success (201 Created)

```
{
  " isSuccess":true,
  " code":201,
  " message":"회원가입 성공",
  " result":{
    " userId":1,
    " username":"doeun05"
  }
}
```

Error Responses

(400) 유효성 검증 실패

```
{
  " isSuccess":false,
  " code":400,
  " message":"입력값이 올바르지 않습니다."
}
```

(409) 아이디 중복

```
{
  " isSuccess":false,
  " code":409,
```

```
    "message":"이미 사용 중인 아이디입니다."  
}
```

(409) 학번 중복(원하면 추가)

```
{  
  "isSuccess":false,  
  "code":409,  
  "message":"이미 등록된 학번입니다."  
}
```

- 아이디 찾기

이름
학번
전화번호

- 비밀번호 찾기 - <비밀번호 찾기 방식에 따라 추가 논의 >

이름
학번
전화번호
아이디

- 학복위 소개
 - 필요한 정보는 없음.
- 중앙대여사업
 - ▼ 물품 목록 조회

- 페이지 로드 시 : 대여 물품 목록 배열 (물품ID, 사진, 이름, 전체 수량)
- 상품 담기를 눌렀을 때 : 장바구니 배열에 물품 ID 추가
- 상품 카드를 눌렀을 때 : 세부 설명 내용 (이미지), 영상 링크 Load
- 예약하기 버튼을 눌렀을 때 : 로그인 여부(ID, 토큰), 패널티 여부

▼ 물품 상세 조회

- 페이지 로드 시 : 물품 설명 이미지, 대여 물품 목록 배열 (물품ID, 사진, 이름, 전체 수량)
- 물품 이름, 물품 사용 설명 영상 URL, 재고 현황 달

- 플로터인쇄사업

◦

- 홍보게시판
 - 필요한 정보는 없음.

사용자 key
토큰

로그인(Auth)

◁ Request, Response	▶ Path	◎ Method
<u>로그인</u>	/auth/login	POST

</> Request, Response	█ Path	ⓧ Method
<u>회원가입</u>	/auth/register	
<u>로그아웃</u>	/auth/logout	
<u>아이디 찾기</u>	/auth/find_id	
<u>비밀번호 찾기</u>	/auth/find_pw	

물품 대여()

</> Request, Response	█ Path	ⓧ Method
<u>물품 목록 조회</u>		GET
<u>물품 담기</u>		
<u>물품 상세 조회</u>		
<u>예약하기</u>		
<u>장바구니 캘린더 조회</u>		
<u>장바구니 대여 확정</u>		
<u>물품 대여 상태 수정</u>		

플로터 예약()

</> Request, Response	█ Path	ⓧ Method
<u>예약 신청</u>		
<u>예약 상태 변경</u>		
<u>예약 취소</u>		
<u>예약 정보 확인</u>		
<u>예약 상태 조회</u>		
<u>제목 없음</u>		

< 다음 회의까지 할 일 >

- 물품 대여 명세 작성 - 세현
- 플로터 인쇄 명세 작성 - 도은
- 로그인 명세 작성 - 건희