



# api 명세

■ 담당자	영준 이
■ 진행 상태	진행 중
◎ 프로젝트	대회 계 획
◆ 태그	



## E-do Backend API 명세서

Base URL:

<https://e-do.onrender.com>

## 파일 업로드 & 관리

### 1. 파일 업로드

- **Endpoint:** POST /api/upload
- **설명:** 클라이언트에서 선택한 파일을 서버에 업로드합니다.
- **요청 형식:** multipart/form-data

### 요청 파라미터

이름	타입	설명	필수 여부
file	File	업로드할 파일	✓

### 응답 예시 (성공)

```
{  
  "ok": true,  
}
```

```
"fileId": "a1b2c3d4",
"filename": "document.png",
"size": 445753,
"contentType": "image/png"
}
```

## 응답 예시 (실패)

```
{ "ok": false, "error": "file is empty" }
```

## 2. 파일 다운로드

- **Endpoint:** GET /api/files/{fileId}
- **설명:** 업로드한 파일을 다운로드합니다.

### 요청 파라미터

- fileId: 업로드 시 발급받은 파일 ID

### 응답

- 파일 바이너리 데이터 (Content-Disposition 헤더 포함 → 브라우저 다운로드 동작)

## 3. 파일 메타데이터 조회

- **Endpoint:** GET /api/files/{fileId}/meta
- **설명:** 파일 이름, 크기, 타입 등 메타 정보를 확인합니다.

### 요청 파라미터

- fileId: 업로드 시 발급받은 파일 ID

### 응답 예시

```
{
  "id": "a1b2c3d4",
  "originalName": "document.png",
  "size": 445753,
  "contentType": "image/png",
}
```

```
"createdAt": "2025-09-23T12:34:56Z"
}
```

## OCR (문자인식)

### 4. OCR 요청

- **Endpoint:** POST /api/ocr
- **설명:** 업로드된 이미지에서 텍스트를 추출합니다.
- **요청 형식:** application/json

### 요청 바디

```
{
  "fileId": "a1b2c3d4"
}
```

### 응답 예시 (성공)

```
{
  "ok": true,
  "fileId": "a1b2c3d4",
  "len": 1234,
  "preview": "인식된 텍스트 앞부분 최대 300자..."
}
```

### 응답 예시 (실패)

```
{
  "ok": false,
  "error": "Google Vision API error message"
}
```



## Quiz(GPT 4o mini api)

- **Endpoint:** POST /api/quiz
- **설명:** 업로드된 파일을 OCR로 읽은 뒤, GPT를 활용해 **5문항 5지선다 퀴즈**를 생성합니다.
- **요청 형식:** application/json

## 요청 바디

```
{
  "fileId": "a1b2c3d4",
  "level": "초급",
  "style": "지문 이해"
}
```

필드명	타입	필수	설명
fileId	string	✓	OCR을 수행할 파일 ID (업로드 후 반환된 값)
level	string	✗	난이도 설정 (예: "초급", "중급", "고급")
style	string	✗	문제 스타일 (예: "어휘", "지문 이해", "추론")

👉 level과 style은 GPT에게 힌트를 주는 값으로, 아무 문자열을 넣어도 됩니다. 비워두면 기본값(초급, 지문 이해)이 적용됩니다.

## 응답 예시(성공)

```
{
  "ok": true,
  "sourceFileId": "a1b2c3d4",
  "questions": [
    {
      "id": "Q1",
      "question": "'봄'이라는 단어가 뜻하는 계절은 무엇일까요?",
      "choices": ["봄", "여름", "가을", "겨울", "모르겠다"],
    }
  ]
}
```

```

    "answerIndex": 0,
    "explanation": "'봄'은 따뜻해지는 계절을 말해요."
  },
  {
    "id": "Q2",
    "question": "지문에서 주인공이 간 곳은 어디였나요?",
    "choices": ["학교", "도서관", "시장", "공원", "집"],
    "answerIndex": 3,
    "explanation": "지문에 '공원에 갔다'고 나와 있어요."
  }
  // ... 총 5문항
],
"error": null
}

```

## 응답 예시(실패)

```

{
  "ok": false,
  "sourceFileId": "a1b2c3d4",
  "questions": [],
  "error": "Invalid quiz format"
}

```

## 테스트 방법

- <https://e-do.onrender.com/swagger-ui/index.html>
- 저기 링크로 들어가면 swagger내에서 내가 만든 api 각각 body에 값 넣어보면서 테스트 가능함

## 협업 시 유의사항

1. **Base URL 고정:** 모든 요청 앞에는 <https://e-do.onrender.com> 붙여야 함.
  - 예) 파일 업로드 → POST <https://e-do.onrender.com/api/upload>

## 2. fileId 중요(파일 관리 보안 차원에서 fileId 사용함):

- 업로드 → fileId 발급
- OCR / 다운로드 / 메타 조회는 이 fileId로 진행

## 3. 데이터 초기화 주의:

- 현재 서버는 **테스트용 H2 메모리 DB** 사용 중
- 서버 재시작/재배포 시 기존 데이터는 사라질 수 있음 → 개발 중엔 "매번 업로드 후 테스트" 권장

## 4. 에러 응답 처리:

- ok: false 또는 status: 500 응답이 오면 프론트에서 사용자에게 안내 메시지를 보여주세요.

## 주의점

- 개발 가능 기간이 짧아 이 서비스는 정말 상남자식 DB 구성으로 이루어짐.
- h2 DB로 했기에 사용자가 꺾다키면 데이터 사라지는 구조