SK네트웍스 **Family AI**과정 **5**기

모델배포 API 정의서

☐ 개요

- 산출물 단계 : API 정의서
- 평가 산출물 : API 정의서
- 제출 일자 : 2025. 02. 10
- 깃허브 경로 : https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN05-final-3Team
- 작성 팀원 : 허상호, 서장호, 최영민

개요	• API 엔드포인트 개요
정의	• 프론트엔드 ↔ 백엔드 엔드포인트 정의 • 백엔드 ↔ 모델 엔드포인트 정의
기타	• 공통 사항

API 정의서

프로젝트: 채팅 기반 구글 독스 문서 어시스턴트 크롬 확장 프로그램

버전: 1.0

목적:

프론트엔드에서 발생하는 요청(질문 입력, 파일 업로드/삭제, 로그인, 문서 연동 및 채팅 히스토리 조회 등)을 백엔드에서 처리하기 위한 REST API 명세를 제공

API 엔드포인트 개요

엔드포인트	HTTP 메서드	설명
/qnas	POST	QnA(질문/답변) 등록 (질문 전송 시 호출)
/qnas	GET	채팅 히스토리 조회 (사이드바 열릴 때)
/pdfs	POST	PDF 문서 업로드
/pdfs	DELETE	PDF 문서 삭제
/csvs	POST	CSV 파일 업로드
/csvs	DELETE	CSV 파일 삭제
/users	POST	사용자 로그인 (회원 정보 등록)
/sessions	POST	문서 연동(세션 생성)

프론트엔드 ↔ 백엔드 엔드포인트 정의

1. 사용자 로그인
- URL: /users
- HTTP 메서드: POST
- 설명:
사용자가 로그인하면, 해당 사용자 이메일 정보를 데이터베이스의 member_tbl에 등록.
Request Body (JSON):

```
json
{
  "user_email": "사용자 이메일"
}
```

Response:
로그인 성공 시 HTTP 200/201 상태 코드와 성공 메시지를 반환.

2. 문서 연동 **(**세션 생성**)**
- URL: /sessions
- HTTP 메서드: POST
- 설명:

문서와 사용자를 연동하여 세션을 생성. 사용자가 작업할 문서에서 FinPilot의 사이트 패널을 열 때 호출.
Request Body (JSON):

```
json
{
  "user_email": "사용자 이메일",
  "docs_id": "문서 ID"
}
```

Response:
세션 생성에 성공하면 생성된 session_id 등 세션 정보를 포함하여 HTTP 200/201 상태 코드를 반환.

3. QnA 입력 **/** 질문 전송
3.1 질문 및 답변 등록
- URL: /qnas
- HTTP 메서드: POST
- 설명:
사용자가 질문을 입력하고 전송하면, 세션을 확인(또는 생성)한 후 chat_option에 따라

 - 일반 텍스트 질문일 경우 모델 호출(query_non_image) 후 QnA 데이터베이스에 저장 및 응답(JSONResponse) 반환

 - “데이터 시각화” 등이 포함된 요청이면 query_image를 호출하여 그래프 응답(JSONResponse) 반환

Request Body (JSON):

```
json
{
  "user_email": "사용자 이메일",
  "docs_id": "문서 고유 ID",
  "question": "사용자 질문 내용",
  "chat_option": "요청 옵션 (예: '일반' 또는 '데이터 시각화')"
}
```

Response 예시:
[텍스트 응답]

```
json
{
  "user_email": "user@example.com",
  "docs_id": "문서ID",
  "qna_id": 0,
  "session_id": "세션ID",
  "question": "질문 내용",
```

```
  "answer": "모델 답변 텍스트",
  "chat_option": "일반",
  "ask_time": "2025-01-21T05:07:09.185Z",
  "source": ["출처1", "출처2"]
}
```

[그래프 응답]

```
json
{
  "images": [
    {
      "file_name": "그래프파일.png",
      "image_data": "base64 인코딩된 이미지 문자열",
      "source": [
        "출처 URL1",
        "출처 URL2",
        "출처 URL3"
      ]
    },
    ...
  ]
}
```

3.2 채팅 히스토리 조회
- URL: /qnas
- HTTP 메서드: GET
- 설명:
사이드바가 열릴 때, 특정 사용자와 문서에 대해 저장된 QnA 기록(채팅 히스토리)을 조회.
Query Parameters:

```
json
{
  "user_email": "사용자 이메일",
  "docs_id": "문서 ID",
}
```

Response 예시:

```
json
[
  {
    "user_email": "user@example.com",
    "docs_id": "string",
    "qna_id": 0,
```

```
    "session_id": "string",
    "question": "string",
    "answer": "string",
    "chat_option": "",
    "ask_time": "2025-01-21T05:10:20.430Z",
    “source”: “List”
  },
  {
    "user_email": "user@example.com",
    "docs_id": "string",
    "qna_id": 1,
    "session_id": "string",
    "question": "string",
    "answer": "string",
    "chat_option": "",
    "ask_time": "2025-01-21T05:11:05.123Z",
    “source”: “List”
  },
  …
]
```

4. PDF 파일 관련 엔드포인트
4.1 PDF 파일 업로드
- URL: /pdfs
- HTTP 메서드: POST
- 설명:
사용자가 PDF 파일을 업로드하면,
 - 사용자 및 문서에 대한 세션을 생성 또는 확인
 - 업로드된 파일을 임시 디렉토리에 저장
 - 데이터베이스에 PDF 파일 메타 정보(파일 이름 등)를 저장
 - 벡터 스토어(vector_store)에 PDF 내용을 추가하여 검색 등을 지원

Request Body (JSON):

```
json
{
  "user_email": "사용자 이메일",
  "docs_id": "문서 ID",
  "file": "파일 데이터 (예: base64 인코딩 문자열)"
}
```

Response:
업로드 성공 시 HTTP 200/201 상태 코드와 성공 메시지를 반환.

4.2 PDF 파일 목록 조회
- URL: /pdfs
- HTTP 메서드: GET
- 설명:

Query Parameters:
- skip (기본값: 0)
- limit (기본값: 10)

Response:
- PDF 파일 정보 목록 배열

4.3 PDF 파일 삭제
- URL: /pdfs
- HTTP 메서드: DELETE
- 설명:

사용자가 PDF 파일 삭제 요청 시,
- 세션을 확인(또는 생성)하고, 벡터 스토어에서 해당 파일을 삭제
- 데이터베이스에서 PDF 파일 메타 정보를 조회 후 삭제 수행

Request Body (JSON):

```
json
{
  "user_email": "사용자 이메일",
  "docs_id": "문서 ID",
  "file_name": "삭제할 파일 이름"
}
```

Response:
삭제 성공 시

```
json
{
  "message": "PDF file deleted successfully"
}
```

5. CSV 파일 엔드포인트
5.1 CSV 파일 업로드
- URL: /csvs
- HTTP 메서드: POST
- 설명:

사용자가 CSV 파일을 업로드하면,
- 파일 확장자가 “.csv”인지 확인하고,
- 사용자 및 문서에 대한 세션을 생성 또는 확인 후
- 파일을 지정된 데이터 디렉토리에 저장하여 모델에 전달.

Request Body (JSON):

```
json
{
  "user_email": "사용자 이메일",
  "docs_id": "문서 ID",
  "file": "파일 데이터 (예: base64 인코딩 문자열)"
```

	}
	
	Response:
	업로드 성공 시
	
	json
	{
	"message": "CSV file processed successfully",
	"session_id": "세션ID"
	}
	
	5.2 CSV 파일 삭제
	- URL: /api/csvs
	- HTTP 메서드: DELETE
	- 설명:
	사용자가 CSV 파일 삭제 요청 시,
	- 세션을 생성 또는 확인한 후
	- 해당 세션의 CSV 파일들을 지정된 데이터 디렉토리에서 삭제.
	
	Request Body (JSON):
	
	json
	{
	"user_email": "사용자 이메일",
	"docs_id": "문서 ID",
	"file_name": "삭제할 파일 이름"
	}
	
	Response:
	삭제 성공 시
	
	json
	{
	"message": "CSV file deleted successfully"
	}

백엔드 ↔ 모델 엔드포인트 정의	내부 모델 호출 및 유틸리티 함수 **1. query_non_image** 용도: 텍스트 기반 일반 질문에 대해, pilot.ainvoke를 호출하여 답변(답변 텍스트 및 출처)을 받아 반환. 입력 값: - question: 질문 텍스트 - session_id: 세션 식별자

- chat_option: 요청 옵션
- pilot: 외부 모델 호출 인스턴스

출력 예시:

```
json
{
  "answer": "생성된 답변 텍스트",
  "source": ["출처 정보1", "출처 정보2"]
}
```

2. query_image
용도:
“데이터 시각화” 요청 시,
pilot.ainvoke를 호출하고 지정된 폴더에서 PNG 이미지 파일을 읽어 base64 인코딩 후 이미지를 JSON 배열로 반환.

입력 값:
- question: 질문 텍스트
- session_id: 세션 식별자
- chat_option: 요청 옵션
- pilot: 외부 모델 호출 인스턴스

출력 예시:

```
json
{
  "images": [
    {
      "file_name": "차트1.png",
      "image_data": "base64 문자열",
      "source": ["출처 URL1", "출처 URL2"]
    },
    ...
  ]
}
```

3. PDF 및 **CSV** 업로드/삭제 함수
upload_pdfs:
PDF 파일을 파싱(parse_pdf)한 후 문서 객체를 생성, 벡터 스토어에 추가하는 함수

delete_pdfs:
벡터 스토어에서 특정 session_id와 파일명을 기준으로 PDF 데이터를 삭제하는 함수

upload_csvs:
CSV 파일을 지정된 데이터 디렉토리에 저장하고, 기존 파일이 있으면 삭제한 후 파일을 처리하는 함수

	delete_csvs: 세션 관련 데이터 디렉토리 내의 CSV 파일들을 삭제하는 함수
공통 사항	- HTTP 상태 코드: 200: 정상 처리 201: 자원 생성 성공 400/404/500 등: 오류 발생 시 적절한 오류 코드와 에러 메시지 반환 - 응답 형식: 모든 응답은 JSON 형식으로 반환. 에러 처리: 기타 정의되지 않은 에러 사유는 "error" 필드를 포함하여 반환.