AI Agent 과정

외부 시스템 연동 아키텍처 API

목차 외부 시스템 연동 아키텍처 API

1. API 기반 연동의 중요성 이해
2. REST API 개념 및 기초 설명

1. API 기반 연동의 중요성 이해

LLM의 근본적인 한계

Knowledge Cut-off:

- LLM의 지식은 특정 훈련 시점(2025년 3월)에 멈춰있음

  오늘의 날씨나 현재 주가를 알지 못함

Data Accessibility:

- 회사 내부 데이터베이스, 개인 이메일, 웹사이트 실시간 정보 등

  비공개적이거나 실시간으로 변하는 데이터 접근 불가

Action Limitation:

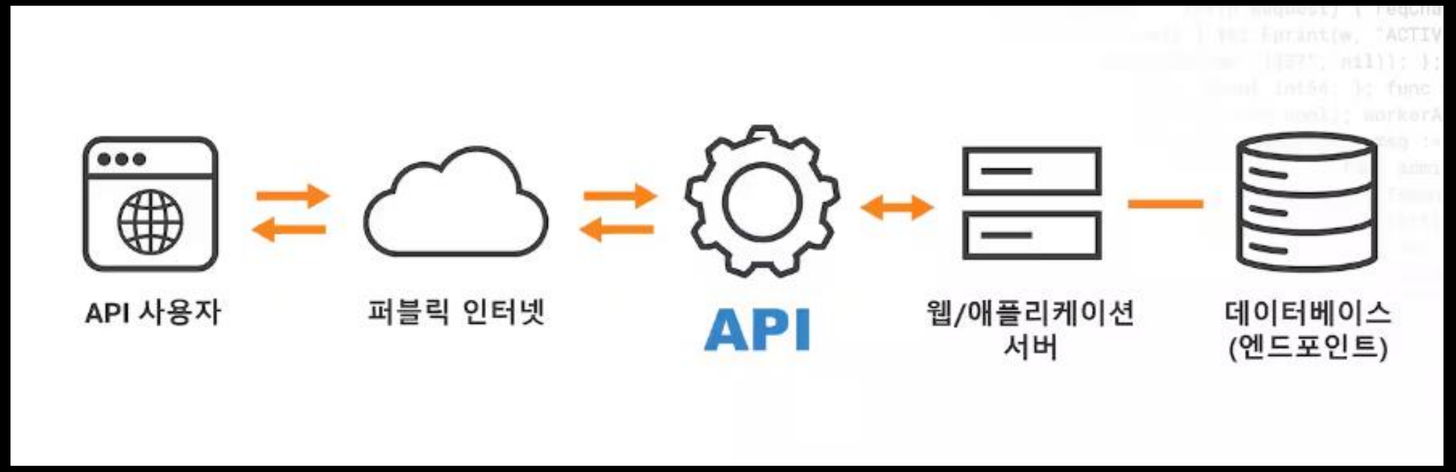
이메일 전송, 파일 저장, 소프트웨어 조작 등의 행동 수행 불가

LLM의 지능을 실제 행동으로 연결하기 위해서는 외부 세계와 소통할 수 있는 창구 필요

API를 통한 외부 세계와의 연결

API (Application Programming Interface):

- 소프트웨어의 기능, 데이터에 접근할 수 있도록 미리 정해놓은 통신 규칙(Protocol)
- Agent가 API를 통해 실시간 날씨, 주가, 뉴스 등 외부 정보 접근 가능
- Agent가 API를 통해 이메일 전송, 데이터베이스 기록, IoT 장치 제어 등 외부 시스템 행동
- 'Tool'은 본질적으로 이 API를 Python 함수 형태로 편리하게 감싼 것(Wrapping)

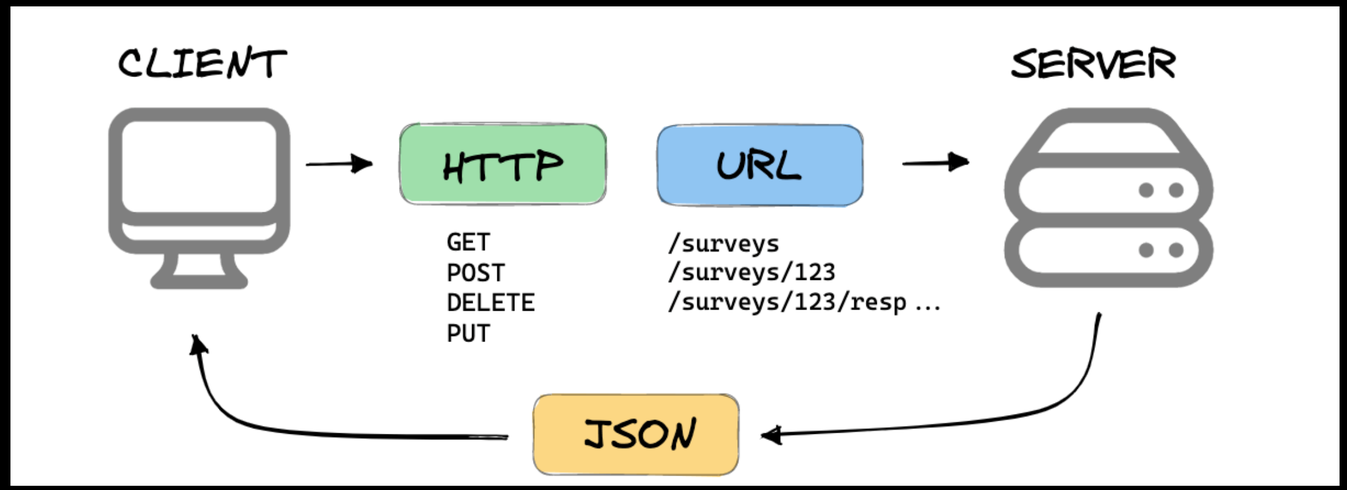


2. REST API 개념 및 기초 설명

현대 웹의 표준: REST API

REST (REpresentational State Transfer):

- 웹에서 사용되는 아키텍처 스타일, 설계 원칙
- 현대 웹 기반 서비스의 사실상 표준(De facto standard)
- HTTP 프로토콜을 그대로 활용
- 웹 환경에 통합 용이
- 대부분의 외부 서비스(Google, Slack, Notion, 금융 API 등)들이 REST API 형식



REST API의 핵심 구성요소

Endpoint (URI):

- 명령어의 대상
- API가 접근할 수 있는 특정 자원(Resource)의 주소
- 웹사이트의 URL과 유사한 개념
- https://api.openweathermap.org/data/2.5/weather
  - Base URL -> https://api.openweathermap.org
  - Path -> /data/2.5/weather
- Agent에게 "날씨 알려줘"라고 말하면, Agent는 이 Endpoint 주소로 요청을 보내야 함

REST API의 핵심 구성요소

Method(HTTP):

- 명령어의 행동
- Endpoint 리소스에 대해 수행할 수 있는 행동의 종류
  - GET: 데이터 조회 (Read). Agent가 가장 많이 사용하는 메소드 (현재 날씨 정보 조회)
  - POST: 데이터 생성 (Create). (데이터베이스에 새로운 고객 정보 추가)
  - PUT / PATCH: 데이터 수정 (Update). (기존 고객 정보 변경)
  - DELETE: 데이터 삭제 (Delete). (게시글 삭제)

- GET https://api.openweathermap.org/data/2.5/weather -> "현재 날씨 정보를 조회"

REST API의 핵심 구성요소

Query Parameters:

- 명령어의 세부 조건 인자
- API 요청에 필요한 구체적인 정보를 전달
- Endpoint의 자원을 필터링하거나 정렬하는 등 세부 조건을 명시
- Endpoint 주소 뒤에 ?로 시작하며, key=value 쌍을 &로 연결
- https://api.openweathermap.org/data/2.5/weather?q=Seoul&appid=YOUR_API_KEY
  - q=Seoul: 도시(q)가 Seoul
  - appid=YOUR_API_KEY: API 키(appid)

- LLM이 사용자의 말("서울 날씨 알려줘")에서 파라미터(q=Seoul)를 정확히 추출해내는 것이 핵

REST API의 핵심 구성요소

Headers & Body:

- 명령어의 부가 정보와 내용물
- Headers:
  - 요청에 대한 부가 정보(Metadata)
  - Authorization: 인증. API 키나 토큰을 담아 요청자 증명(Authorization: Bearer YOUR_API_KEY)
  - Content-Type: 요청 내용물의 형식. (Content-Type: application/json)

REST API의 핵심 구성요소

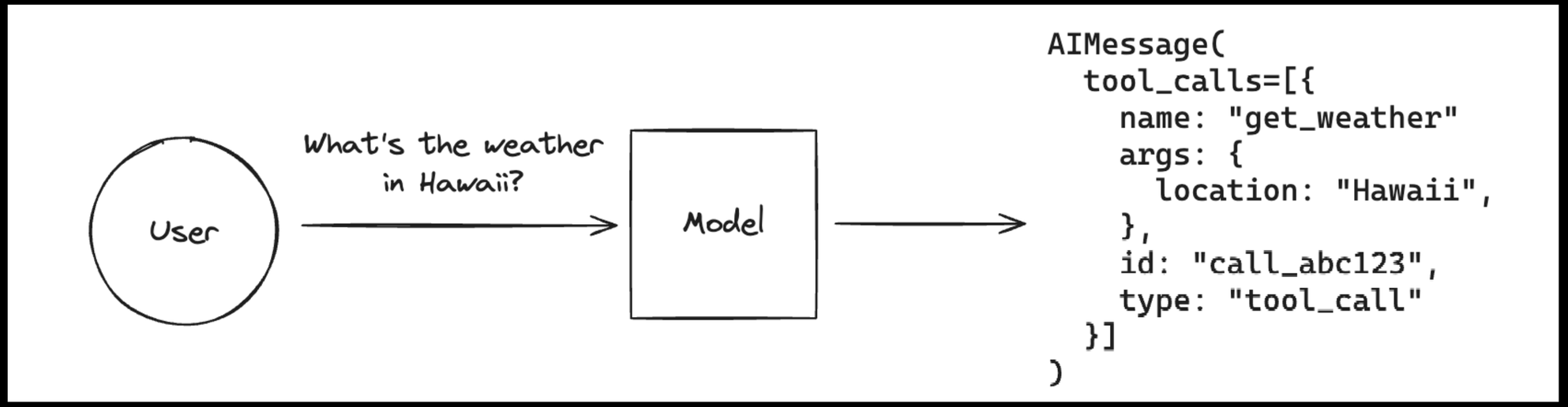
Headers & Body:

- 명령어의 부가 정보와 내용물
- Body:
  - POST나 PUT 요청 시, 서버로 보낼 실제 데이터를 담음
  - 주로 JSON 형식
  - {"username": "gildong", "email": "hong@example.com"}
- 대부분의 API는 Header를 통한 인증을 요구
- Tool을 만들 때 API 키를 Header에 담아 보내는 로직이 포함

API 요청의 전체 흐름

Agent의 Tool이 API를 호출하는 과정:

- 사용자 프롬프트를 LLM이 분석 후 Function Call(JSON)
- JSON 문자열을 Python Tool 함수에 전달 후 Requests 라이브러리
- HTTP 요청(Method, Endpoint, Headers, Params) -> 외부 API 서버



https://python.langchain.com/docs/how_to/tool_calling/

Status Code

HTTP Status Codes:

- 2xx Success
  - 200 OK: GET, PUT 요청 성공
  - 201 Created: POST 요청 성공
- 4xx Client Error
  - 400 Bad Request: 문법 오류 등 요청이 잘못 구성됨
  - 401 Unauthorized: 인증 실패. (API 키가 틀렸을 가능성)
  - 403 Forbidden: 인증은 되었으나, 해당 자원 접근 권한 없음
  - 404 Not Found: 요청한 주소(Endpoint)가 존재하지 않음
- 5xx Server Error
  - 500 Internal Server Error: API 서버 자체에 문제가 발생.
- 견고한 Tool을 만들기 위해서는 API 성공/실패 응답(Status Code) 처리 로직 반드시 포함

JSON Body

Response Body (JSON):

- API 요청이 성공했을 때, 서버가 보내주는 실제 데이터
- Agent는 이 JSON 데이터를 파싱하여 최종 사용자 답변을 생성


```
{
  "weather": [{"description": "clear sky"}],
  "main": {"temp": 28.5},
  "name": "Seoul"
}
```

이 수업에서 사용할 API

OpenWeatherMap & Finnhub:

- 1일차 실습에서 두 개의 실제 외부 API를 사용
- 두 서비스 모두 무료 플랜으로 충분한 사용량을 제공
- 실습에서 두 웹사이트에 가입하여 API 키를 발급받고 적용

- Openweathermap.org/api:
  - 전 세계 도시의 실시간 날씨 및 예보 데이터 제공.

- finnhub.io:
  - 실시간 주식 시세, 기업 정보 등 금융 데이터 제공