

2주차: HTTP 메서드 실습 (1) — GET / POST

# FastAPI로 쉽고 빠르게 만드는 마이크로서비스

# 오늘의 목표

- HTTP 요청/응답 흐름 이해
- FastAPI GET / POST API 실습
- Swagger(/docs) 테스트 실습

# 1주차 체크 (환경 준비)

uv 설치 확인: uv --version

프로젝트 생성: uv init

의존성 설치: uv add fastapi --extra standard

---

서버 실행: uv run uvicorn app.main:app --reload

문서 확인: <http://127.0.0.1:8000/docs>

# HTTP Method

Request → Response

클라이언트(브라우저/앱) → 서버(API)

서버는 보통 JSON으로 응답

URL 구성

/users/3 (Path Parameter)

/search?keyword=fastapi (Query Parameter)

GET vs POST

GET: 조회

POST: 생성 (Body에 JSON 포함)

# FastAPI 기본 구조

app = FastAPI()

@app.get('/')로 라우팅 추가

서버 실행: uv run uvicorn app.main:app --reload

Swagger 자동 생성: /docs

# FastAPI 기본 구조

`router = APIRouter(prefix="...", tags=[...])`로 라우터 생성

`@router.get(...)`로 라우터에 엔드포인트 추가

`app.include_router(router)`로 FastAPI 앱에 라우터 등록

`prefix`가 붙어서 URL이 `/week2/practice1/...` 형태로 자동 구성됨

# 실습 1: GET

```
# app/week2/practice1.py
from fastapi import APIRouter

router = APIRouter(prefix="/practice1", tags=["practice1"])

@router.get("/")
def root():
    return {"message": "Hello FastAPI"}

@router.get("/users/{user_id}")
def read_user(user_id: int):
    return {"user_id": user_id}

@router.get("/search")
def search(keyword: str):
    return {"keyword": keyword}
```

# Body(JSON)와 Pydantic

POST는 보통 JSON Body를 받는다

Pydantic 모델로 입력값 검증(Validation)을 자동 수행

입력이 틀리면 422 Unprocessable Entity 반환

# 실습 2: POST /memos – 메모 생성

```
# app/week2/practice2.py
from pydantic import BaseModel
from fastapi import APIRouter

router = APIRouter(prefix="/practice2", tags=[ "practice2" ])

class MemoCreate(BaseModel):
    title: str
    content: str

MEMOS = []
NEXT_ID = 1

@router.post('/memos', status_code=201)
def create_memo(payload: MemoCreate):
    global NEXT_ID
    memo = { 'id': NEXT_ID, **payload.model_dump() }
    MEMOS.append(memo)
    NEXT_ID += 1
    return memo
```

# 실습 3: GET /memos, GET /memos/{memo\_id}

```
# app/week2/practice3.py
from fastapi import HTTPException
from fastapi import APIRouter
from app.week2.practice2 import MEMOS

router = APIRouter(prefix="/practice3", tags=["practice3"])

@router.get('/memos')
def list_memos():
    return MEMOS

@router.get('/memos/{memo_id}')
def read_memo(memo_id: int):
    for memo in MEMOS:
        if memo['id'] == memo_id:
            return memo
    raise HTTPException(status_code=404, detail='Memo not found')
```

# 상태코드 기초

200 OK: 정상 응답

201 Created: 생성 성공 (POST에 자주 사용)

404 Not Found: 리소스 없음

422 Unprocessable Entity: 입력 검증 실패(FastAPI 자동 처리)

# 2주차 과제

필수

Memo API 기초 구현(실습)

Memo API 심화 구현

- Memo에 title, content 외 정보 추가해보기
- 쿼리 파라미터를 이용하여 추가적인 메모 검색 구현

제출

구글 폼으로 제출