

FastAPI 기초 핵심 개념

1. FastAPI 객체 만들기

```
from fastapi import FastAPI
```

```
app = FastAPI() # 서버 앱 생성
```

- `app` 객체는 우리가 만드는 모든 **API**의 중심
- 이 객체에 기능을 등록하면서 **API** 서버를 완성해갑니다

2. 라우팅 (경로 연결)

python

복사편집

```
@app.get("/")
```

```
def root():
```

```
    return {"message": "Hello FastAPI"}
```

- `@app.get("/")`: GET 요청 처리
- `@app.post("/")`: POST 요청 처리
- `@app.put("/")`, `@app.delete("/")` 등도 있음

- 경로(/)에 요청이 오면 아래 함수가 실행됨

3. 요청(Request) 데이터 받기

① 쿼리 스트링 받기

python

복사편집

```
@app.get("/search")
def search(q: str):
    return {"검색어": q}
```

→ `/search?q=apple` 요청 → `{"검색어": "apple"}` 응답

② 경로 파라미터 받기

python

복사편집

```
@app.get("/user/{user_id}")
def get_user(user_id: int):
    return {"user_id": user_id}
```

→ `/user/3` 요청 → `{"user_id": 3}`

③ JSON 본문 받기 (POST 요청)

python

복사편집

```
from pydantic import BaseModel
```

```
class Item(BaseModel):
    name: str
    price: int
```

```
@app.post("/items")
def create_item(item: Item):
```

```
return {"name": item.name, "price": item.price}
```

→ JSON으로 { "name": "사과", "price": 3000 } 전송 시

→ 자동으로 `item`으로 파싱되고 응답

4. 자동 문서화 (Swagger UI)

- FastAPI는 `/docs` 경로에서 **Swagger** 문서 자동 생성
 - `/redoc` 도 함께 제공
-

5. 응답 형태 지정

python

복사편집

```
from fastapi.responses import JSONResponse
```

```
@app.get("/custom")
```

```
def custom():
```

```
    return JSONResponse(content={"ok": True},  
status_code=201)
```

6. 파일 업로드 (조별과제에 필요)

python

복사편집

```
from fastapi import File, UploadFile
```

```
@app.post("/upload")
```

```
def upload_file(file: UploadFile = File(...)):
```

```
    return {"filename": file.filename}
```

✅ 실전에서 꼭 쓰이는 기능 요약표

기능	키워드
서버 시작	<code>uvicorn</code> <code>main:app</code> <code>--reload</code>
GET 요청	<code>@app.get("/경로")</code>
POST 요청	<code>@app.post("/경로")</code>
경로 파라미터	<code>@app.get("/user/{id}")</code>
쿼리 파라미터	<code>def search(q: str)</code>
JSON 받기	<code>BaseModel,</code> <code>@app.post()</code>
파일 업로드	<code>UploadFile,</code> <code>File(...)</code>
응답 커스터마이징	<code>JSONResponse,</code> <code>status_code</code>
Swagger 문서	<code>http://localhost:8000/docs</code>

✅ 정리

FastAPI는 다음 3가지만 익숙해지면 바로 실전 API 작성이 가능합니다:

1. 라우팅 (경로 연결)
2. 요청 데이터 받기 (경로, 쿼리, **JSON**, 파일)
3. 자동 문서화와 응답 포맷 조절