

FastAPI 라우터 등록 가이드

본 문서는 새로운 API 라우터 파일을 생성하고 `main.py`에 등록하는 방법을 설명합니다.

작성일: 2026-01-15

참고 예시: `customer.py` 라우터 등록 방법

라우터 등록 단계

1단계: API 라우터 파일 생성

`fastapi/app/api/` 폴더에 새로운 Python 파일을 생성합니다.

파일명 규칙: 테이블명과 동일하게 작성 (예: `customer.py`, `store.py`, `menu.py`)

예시: `fastapi/app/api/store.py` 파일 생성

```
"""
Store API - 식당 정보 CRUD
"""

from fastapi import APIRouter, Form
from pydantic import BaseModel
from typing import Optional
from datetime import datetime
from ..database.connection import connect_db

router = APIRouter()

# =====
# 모델 정의
# =====
class Store(BaseModel):
    store_seq: Optional[int] = None
    store_address: str
    store_lat: float
    store_lng: float
    # ... 기타 필드

# =====
# 전체 식당 조회
# =====
@router.get("")
async def select_stores():
    """전체 식당 목록을 조회합니다."""
    try:
        conn = connect_db()
```

```
curs = conn.cursor()
curs.execute("""
    SELECT store_seq, store_address, store_lat, store_lng
    FROM store
    ORDER BY store_seq
""")
rows = curs.fetchall()
conn.close()

result = []
for row in rows:
    result.append({
        'store_seq': row[0],
        'store_address': row[1],
        'store_lat': row[2],
        'store_lng': row[3],
    })

return {"results": result}
except Exception as e:
    import traceback
    error_msg = str(e)
    traceback.print_exc()
    return {"result": "Error", "errorMsg": error_msg, "traceback": traceback.format_exc()}

# =====
# ID로 식당 조회
# =====
@router.get("/{store_seq}")
async def select_store(store_seq: int):
    """특정 식당을 ID로 조회합니다."""
    try:
        conn = connect_db()
        curs = conn.cursor()
        curs.execute("""
            SELECT store_seq, store_address, store_lat, store_lng
            FROM store
            WHERE store_seq = %s
        """, (store_seq,))
        row = curs.fetchone()
        conn.close()

        if row is None:
            return {"result": "Error", "message": "Store not found"}

        result = {
            'store_seq': row[0],
            'store_address': row[1],
            'store_lat': row[2],
            'store_lng': row[3],
        }
        return {"result": result}
```

```

except Exception as e:
    import traceback
    error_msg = str(e)
    traceback.print_exc()
    return {"result": "Error", "errorMsg": error_msg, "traceback": traceback.format_exc()}

```

중요 사항:

- 파일 상단에 모듈 설명 주석 추가
 - `router = APIRouter()` 반드시 생성
 - 데이터베이스 사용 시 `from ..database.connection import connect_db` import (상대 경로 사용)
 - 각 엔드포인트에 docstring 추가
-

2단계: main.py에 라우터 등록

fastapi/app/main.py 파일을 열고 다음 두 가지를 추가합니다:

2-1. Import 문 추가

파일 상단의 import 섹션에 라우터 import를 추가합니다.

위치: main.py의 라우터 등록 섹션 (약 27-37번째 줄 근처)

예시:

```

# Customer API 라우터 등록
from app.api import customer
app.include_router(customer.router, prefix="/api/customer", tags=["customer"])

# Store API 라우터 등록 (새로 추가)
from app.api import store
app.include_router(store.router, prefix="/api/store", tags=["store"])

```

2-2. 라우터 등록 형식

```

from app.api import [파일명]
app.include_router([파일명].router, prefix="/api/[엔드포인트_경로]", tags=["[태그명]"])

```

파라미터 설명:

- [파일명]: 생성한 Python 파일명 (확장자 제외)
- `prefix`: API 엔드포인트의 기본 경로 (예: `/api/store`)
- `tags`: Swagger UI에서 그룹화할 태그명 (예: `["store"]`)

실제 예시 (customer.py 등록):

```
from app.api import customer
app.include_router(customer.router, prefix="/api/customer", tags=["customer"])
```

이렇게 등록하면:

- GET /api/customer → 전체 고객 조회
- GET /api/customer/{customer_seq} → 특정 고객 조회
- POST /api/customer/register → 회원가입
- POST /api/customer/login → 로그인

전체 예시: store.py 등록하기

1. 파일 생성: fastapi/app/api/store.py

```
#####
Store API – 식당 정보 CRUD
#####

from fastapi import APIRouter, Form
from typing import Optional
from ..database.connection import connect_db

router = APIRouter()

@router.get("")
async def select_stores():
    """전체 식당 목록을 조회합니다."""
    try:
        conn = connect_db()
        curs = conn.cursor()
        curs.execute("SELECT * FROM store ORDER BY store_seq")
        rows = curs.fetchall()
        conn.close()

        result = []
        for row in rows:
            result.append({
                'store_seq': row[0],
                'store_address': row[1],
                # ... 기타 필드
            })

        return {"results": result}
    except Exception as e:
        import traceback
```

```
        return {"result": "Error", "errorMsg": str(e), "traceback": traceback.format_exc()}
```

2. main.py에 등록

fastapi/app/main.py 파일의 라우터 등록 섹션에 추가:

```
# =====
# 라우터 등록
# =====

# Customer API 라우터 등록
from app.api import customer
app.include_router(customer.router, prefix="/api/customer", tags=["customer"])

# Store API 라우터 등록
from app.api import store
app.include_router(store.router, prefix="/api/store", tags=["store"])
```

3. 서버 재시작 및 확인

서버를 재시작하고 다음 URL에서 확인:

- Swagger UI: <http://localhost:8000/docs>
- [GET /api/store](#) 엔드포인트가 표시되는지 확인

주의사항

1. Import 경로

- 절대 경로 사용 금지: `from app.database.connection import connect_db` (X)
- 상대 경로 사용: `from ..database.connection import connect_db` (O)
- 이유: `app/api/` 폴더에서 `app/database/` 폴더로의 상대 경로이므로 `..` 사용

2. 파일명과 import명 일치

- 파일명: `store.py`
- import: `from app.api import store` (파일명과 동일)

3. router 변수명

- 반드시 `router = APIRouter()`로 생성
- 다른 이름 사용 시 `app.include_router()`에서 오류 발생

4. prefix 경로

- 일반적으로 `/api/[테이블명]` 형식 사용

- 예: /api/customer, /api/store, /api/menu

5. tags 이름

- Swagger UI에서 그룹화되는 이름
- 일반적으로 테이블명과 동일하게 사용
- 예: ["customer"], ["store"], ["menu"]

리턴 타입 규칙

이전 프로젝트와 동일한 리턴 타입 형식을 사용합니다:

성공 응답

전체 조회 (GET ""):

```
return {"results": result} # "result": "OK" 없이 바로 results
```

단일 조회 (GET "{id}"):

```
return {"result": result} # "data" 키 없이 바로 result에 객체
```

생성/수정/삭제 (POST/PUT/DELETE):

```
# 생성 (회원가입 등)
return {
    "result": {
        "id": ....,
        "name": ....,
        ...
    },
    "message": "성공 메시지"
}

# 수정/삭제
return {"result": "OK"}
```

에러 응답

```
return {
    "result": "Error",
    "errorMsg": error_msg,
    "traceback": traceback.format_exc()
}
```

체크리스트

새로운 라우터를 등록할 때 다음을 확인하세요:

- `fastapi/app/api/` 폴더에 파일 생성
- 파일명은 테이블명과 동일하게 작성
- `router = APIRouter()` 선언
- `from ..database.connection import connect_db` import (데이터베이스 사용 시)
- `main.py`에 import 문 추가
- `main.py`에 `app.include_router()` 추가
- prefix 경로는 `/api/[테이블명]` 형식
- tags는 `["테이블명"]` 형식
- 서버 재시작 후 Swagger UI에서 확인

참고 파일

- 예시 라우터: `fastapi/app/api/customer.py`
- 메인 파일: `fastapi/app/main.py`
- 데이터베이스 연결: `fastapi/app/database/connection.py`

변경 이력

- 2026-01-15: 초기 문서 작성
 - 라우터 등록 단계별 가이드 작성
 - customer.py 등록 방법 참고
 - 리턴 타입 규칙 정리
 - 체크리스트 추가