```
(venv)$ uvicorn api:app --port 8000 --reload
```

이렇게 FastAPI() 인스턴스를 라우팅 작업에 사용할 수 있다. 그러나 이 방식은 라우팅 중에 단일 경로만 고려하는 애플리케이션에서 일반적으로 사용된다. 고유한 함수를 처리하는 각각의 라우트가 FastAPI() 인스턴스를 사용하는 경우 애플리케이션은 한 번에 여러 라우트를 처리할 수 없다. uvicorn이 하나의 엔트리 포인트만 실행할 수 있기 때문이다.

그렇다면 여러 함수를 사용하는 연속적인 라우트 처리는 어떻게 할 수 있을까? APIRouter 클래스를 사용하면 다중 라우팅이 허용되므로 이 문제를 해결할 수 있다. 다음 절에서 살펴보자.

2.2 APIRouter 클래스를 사용한 라우팅

APIRouter 클래스는 다중 라우팅을 위한 경로 처리 클래스로, fastapi 패키지에 포함되어 있다. APIRouter 클래스를 통해 애플리케이션 라우팅과 로직을 독립적으로 구성하고 모듈화할 수 있다.

fastapi 패키지에서 APIRouter 클래스를 임포트import한 후 APIRouter() 인스턴스를 생성할 수 있다. 라우팅 메서드는 다음과 같이 APIRouter() 인스턴스를 사용해 생성한다.

```
from fastapi import APIRouter

router = APIRouter()

@router.get("/hello")
async def say_hello() -> dict:
    return {
        "message": "Hello!"
    }
```

APIRouter 클래스를 사용해서 새로운 라우트와 라우트 처리기를 만들어보자. 여기서 todos를 생성 및 추출하는 함수를 만든다. 먼저 CHAPTER 1에서 만든 todos 폴더에 todo.py라는 새로운 파일을 만들자.

```
(venv)$ touch todo.py
```