



그림 3-2 404 응답 코드 반환

마지막으로 데코레이터 함수에 `status_code` 인수를 추가해서 기본 응답 코드(200)를 다른 코드로 변경해보자.

```
@todo_router.post("/todo", status_code=201)
async def add_todo(todo: Todo) -> dict:
    todo_list.append(todo)
    return {"message": "Todo added successfully."}
```

지금까지 클라이언트에게 정확한 응답 코드를 반환하고 기본 상태 코드를 변경하는 방법을 살펴봤다. 참고로 요청 성공을 의미하는 기본 상태 코드는 200이니 기억하도록 하자.

정리하기

CHAPTER 3에서는 응답 모델과 오류 처리가 무엇인지 배우고 HTTP 상태 코드와 그 중요성도 다뤘다. 또한 <CHAPTER 2 라우팅>에서 배운 모델 작성법을 활용해 응답 모델을 생성하고 `todo_list`가 ID 없이 아이템만 반환하도록 정의했다. 마지막으로 오류 처리에 대해 설명하고 기존 라우트를 변경해서 기본 응답 코드인 200 대신 적절한 응답 코드를 반환하게 했다.

CHAPTER 4에서는 Jinja를 사용한 FastAPI 애플리케이션 템플릿 생성 방법을 배운다. Jinja 템플릿의 실행 및 설정 방법을 익힌 다음 템플릿 지식을 활용해 UI를 만드는 방법을 알아보자.