

응답 상태 코드

응답 모델과 같은 방법으로, 어떤 *경로* 작동이든 `status_code` 매개변수를 사용하여 응답에 대한 HTTP 상태 코드를 선언할 수 있습니다.

- `@app.get()`
- `@app.post()`
- `@app.put()`
- `@app.delete()`
- 기타

```
{!../../../../../docs_src/response_status_code/tutorial001.py!}
```

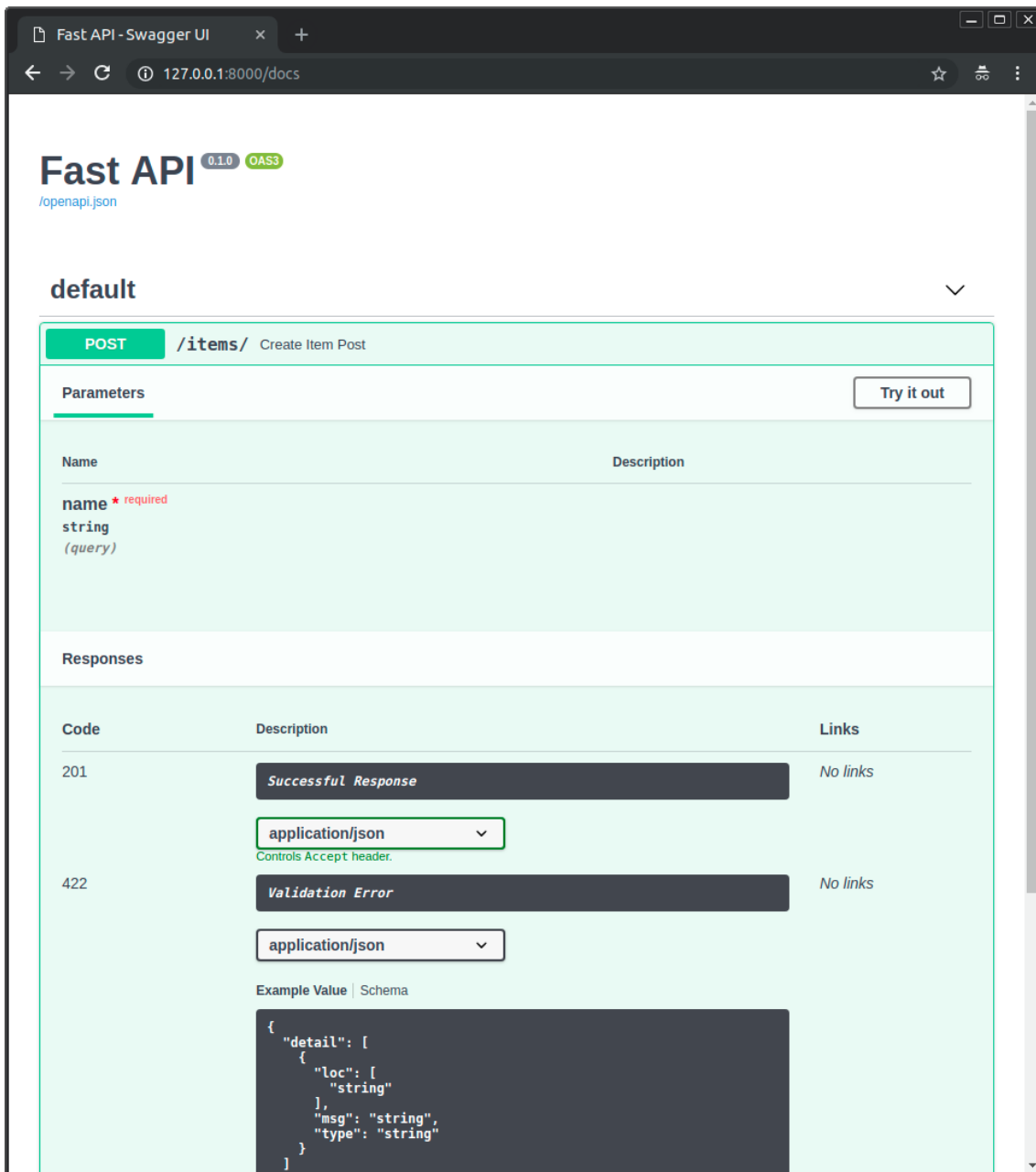
!!! note "참고" `status_code` 는 "데코레이터" 메소드(`get` , `post` 등)의 매개변수입니다. 모든 매개변수들과 본 문처럼 *경로* 작동 함수가 아닙니다.

`status_code` 매개변수는 HTTP 상태 코드를 숫자로 입력받습니다.

!!! info "정보" `status_code` 는 파이썬의 `http.HTTPStatus` 와 같은 `IntEnum` 을 입력받을 수도 있습니다.

`status_code` 매개변수는:

- 응답에서 해당 상태 코드를 반환합니다.
- 상태 코드를 OpenAPI 스키마(및 사용자 인터페이스)에 문서화 합니다.



!!! note "참고" 어떤 응답 코드들은 해당 응답에 본문이 없다는 것을 의미하기도 합니다 (다음 항목 참고).

이에 따라 FastAPI는 응답 본문이 없음을 명시하는 OpenAPI를 생성합니다.

HTTP 상태 코드에 대하여

!!! note "참고" 만약 HTTP 상태 코드에 대하여 이미 알고있다면, 다음 항목으로 넘어가십시오.

HTTP는 세자리의 숫자 상태를 응답의 일부로 전송합니다.

이 상태 코드들은 각자를 식별할 수 있도록 지정된 이름이 있으나, 중요한 것은 숫자 코드입니다.

요약하자면:

- ****1xx**** 상태 코드는 "정보"용입니다. 이들은 직접적으로는 잘 사용되지는 않습니다. 이 상태 코드를 갖는 응답들은 본문을 가질 수 없습니다.
- ****2xx**** 상태 코드는 "성공적인" 응답을 위해 사용됩니다. 가장 많이 사용되는 유형입니다.
 - 200 은 디폴트 상태 코드로, 모든 것이 "성공적임"을 의미합니다.
 - 다른 예로는 201 "생성됨"이 있습니다. 일반적으로 데이터베이스에 새로운 레코드를 생성한 후 사용합니다.
 - 단, 204 "내용 없음"은 특별한 경우입니다. 이것은 클라이언트에게 반환할 내용이 없는 경우 사용됩니다. 따라서 응답은 본문을 가질 수 없습니다.
- ****3xx**** 상태 코드는 "리다이렉션"용입니다. 본문을 가질 수 없는 304 "수정되지 않음"을 제외하고, 이 상태 코드를 갖는 응답에는 본문이 있을 수도, 없을 수도 있습니다.
- ****4xx**** 상태 코드는 "클라이언트 오류" 응답을 위해 사용됩니다. 이것은 아마 가장 많이 사용하게 될 두 번째 유형입니다.
 - 일례로 404 는 "찾을 수 없음" 응답을 위해 사용합니다.
 - 일반적인 클라이언트 오류의 경우 400 을 사용할 수 있습니다.
- ****5xx**** 상태 코드는 서버 오류에 사용됩니다. 이것들을 직접 사용할 일은 거의 없습니다. 응용 프로그램 코드나 서버의 일부에서 문제가 발생하면 자동으로 이들 상태 코드 중 하나를 반환합니다.

!!! tip "팁" 각각의 상태 코드와 이들이 의미하는 내용에 대해 더 알고 싶다면 [MDN HTTP 상태 코드에 관한 문서](#) 를 확인하십시오.

이름을 기억하는 쉬운 방법

상기 예시 참고:

```
{!../../../docs_src/response_status_code/tutorial001.py!}
```

201 은 "생성됨"을 의미하는 상태 코드입니다.

하지만 모든 상태 코드들이 무엇을 의미하는지 외울 필요는 없습니다.

`fastapi.status` 의 편의 변수를 사용할 수 있습니다.

```
{!../../../docs_src/response_status_code/tutorial002.py!}
```

이것은 단순히 작업을 편리하게 하기 위한 것으로, HTTP 상태 코드와 동일한 번호를 갖고 있지만, 이를 사용하면 편집기의 자동완성 기능을 사용할 수 있습니다:

```
from fastapi import FastAPI, status

app = FastAPI()

@app.post("/items/", status_code=status.CREATED)
async def create_item(name: str):
    return {"name": name}
```

HTTP 201 CREATED
 WS 1000 NORMAL CLOSURE
 WS 1005 ABNORMAL CLOSURE

HTTP 201 CREATED: int

!!! note "기술적 세부사항" `from starlette import status` 역시 사용할 수 있습니다.

****FastAPI****는 개발자인 당신의 편의를 위해 ``fastapi.status`` 와 동일한 ``starlette.status`` 도 제공합니다. 하지만 이것은 `Starlette`로부터 직접 제공됩니다.

기본값 변경

추후 여기서 선언하는 기본 상태 코드가 아닌 다른 상태 코드를 반환하는 방법을 [숙련된 사용자 지침서](#){internal-link target=_blank}에서 확인할 수 있습니다.