

이번에는 데이터베이스에 저장된 이벤트 개수(지금까지 2개)를 확인하기 위한 테스트를 작성해보자. 다음과 같이 코드를 추가한다.

```
@pytest.mark.asyncio
async def test_get_events_count(default_client: httpx.AsyncClient) -> None:
    response = await default_client.get("/event/")


    events = response.json()

    assert response.status_code == 200
    assert len(events) == 2
```

이 코드는 JSON 응답을 `events`라는 변수에 저장하고 `events`의 길이가 예상한 값과 일치하는지 확인한다. 다음 명령을 사용해 테스트 파일을 다시 실행해보자.

```
(venv)$ pytest tests/test_routes.py
```

실행 결과는 [그림 8-9]와 같다.



```
pytest
(venv) → planner git:(main) × pytest tests/test_routes.py
test session starts
platform darwin -- Python 3.8.2, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/youngestdev/Work/Building-Web-APIs-with-FastAPI-and-Python/ch08/planner, configfile: pytest.ini
plugins: asyncio-0.18.3, anyio-3.5.0
asyncio: mode=auto
collected 4 items

tests/test_routes.py .... [100%]

===== 4 passed in 0.04s =====
(venv) → planner git:(main) ×
```

그림 8-9 이벤트 개수를 확인하는 테스트 결과