

---

```

@pytest.fixture(scope="session")
async def default_client():
    await init_db()
    async with httpx.AsyncClient(app=app, base_url="http://app") as client:
        yield client
    # 리소스 정리
    await Event.find_all().delete()
    await User.find_all().delete()

```

---

여기서는 먼저 데이터베이스를 초기화한 후에 애플리케이션을 AsyncClient로 호출한다. AsyncClient는 테스트 세션이 끝날 때까지 유지된다. 테스트 세션이 끝나면 이벤트(Event)와 사용자(User) 컬렉션의 데이터를 모두 삭제하여 테스트를 실행할 때마다 데이터베이스가 비어 있도록 한다.

지금까지 테스트 환경 구축 방법을 살펴봤다. 이어서 애플리케이션의 각 라우트를 테스트하는 코드를 작성해보자.

## 8.3 REST API 라우트 테스트 작성

이제 모든 것이 준비됐다. test\_login.py 파일을 작성해서 인증 로직을 테스트해보자.

---

```

(venv)$ touch tests/test_login.py

```

---

먼저 필요한 의존 라이브러리를 임포트한다.

---

```

import httpx
import pytest

```

---

### 사용자 등록 라우트 테스트

첫 번째 테스트 대상은 사용자 등록 라우트다. pytest.mark.asyncio 데코레이터를 추가해서 비동기 테스트라는 것을 명시한다. 다음과 같이 테스트 함수와 요청 페이로드를 정의해보자.