```
import asyncio
import httpx
import pytest

from main import app
from database.connection import Settings
from models.events import Event
from models.users import User
```

이 코드는 asyncio, httpx, pytest를 임포트한다. asyncio 모듈은 활성 루프 세션을 만들어서 테스트가 단일 스레드로 실행되도록 한다. httpx 테스트는 HTTP CRUD 처리를 실행하기 위한 비동기 클라이언트 역할을 한다. pytest 라이브러리는 픽스처 정의를 위해 사용된다. 또한 애플리케이션 인스턴스(app), Settings 클래스, 모델도 임포트한다.

다음과 같이 루프 세션 픽스처를 정의해보자.[45]

```
@pytest.fixture(scope="session")
def event_loop():
    loop = asyncio.get_event_loop()
    yield loop
    loop.close()
```

Settings 클래스에서 새로운 데이터베이스 인스턴스를 만든다.

```
async def init_db():
    test_settings = Settings()
    test_settings.DATABASE_URL = "mongodb://localhost:27017/testdb"

    await test_settings.initialize_database()
```

이 코드는 DATABASE_URL과 〈CHAPTER 6 데이터베이스 연결〉에서 정의한 초기화 함수를 호출한다. 그리고 testdb라는 새로운 데이터베이스를 사용한다.

마지막으로 기본 클라이언트 픽스처를 정의한다. 이 픽스처는 httpx를 통해 비동기로 실행되는 애플리케이션 인스턴스를 반환한다.

45 옮긴이_최근 버전에서는 경고(warning)가 표시될 수 있으나 무시해도 상관없다. 이전 버전의 작성 방법에 관한 경고 메시지다.