

픽스처 데코레이터는 인수를 선택적으로 받을 수 있다. 예를 들어 `scope` 인수는 픽스처 함수의 유효 범위를 지정할 때 사용된다. 여기서는 두 가지 `scope`를 사용한다.

- **session**: 테스트 전체 세션 동안 해당 함수가 유효하다.
- **module**: 테스트 파일이 실행된 후 특정 함수에서만 유효하다.

픽스처를 살펴봤으니 테스트 환경을 구축해보자.

8.2 테스트 환경 구축

앞서 테스트의 기본 개념과 픽스처에 관해 살펴보았다. 이번에는 CRUD 처리용 라우트와 사용자 인증을 테스트해보자. 우리가 만든 비동기 API를 테스트하려면 `httpx`와 `pytest-asyncio` 라이브러리를 설치해야 한다.

```
(venv)$ pip install httpx pytest-asyncio
```

설치가 완료됐으면 `pytest.ini`라는 설정 파일을 만들어야 한다. 파일을 루트 폴더(`main.py`가 있는 폴더)에 생성한 후 다음과 같이 코드를 추가한다.

```
[pytest]
asyncio_mode = auto
```

`pytest`가 실행될 때 이 파일의 내용을 불러온다. 이 설정은 `pytest`가 모든 테스트를 비동기식으로 실행한다는 의미다.

설정 파일이 준비됐으니 `tests` 폴더 아래에 테스트 시작점이 될 `conftest.py` 파일을 만든다. `conftest.py` 파일은 테스트 파일이 필요로 하는 애플리케이션의 인스턴스를 만든다.

```
(venv)$ touch tests/conftest.py
```

`conftest.py` 파일에 의존 라이브러리를 임포트한다.