

요청에 대한 예상 응답을 정의한다.

```
response = await default_client.post("/user/signin", data=payload,
headers=headers)
```


응답을 비교해서 요청이 성공했는지 확인하는 코드를 작성한다.

```
assert response.status_code == 200
assert response.json()["token_type"] == "Bearer"
```

다시 테스트를 실행해보자.

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

로그인 라우트 테스트가 성공하면 [그림 8-4]와 같은 화면을 볼 수 있다.



```
pytest
(venv) + planner git:(main) x pytest tests/test_login.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 2 items

tests/test_login.py ..                                     [100%]

----- 2 passed in 0.52s -----
(venv) + planner git:(main) x
```

그림 8-4 성공한 로그인 라우트 테스트

로그인할 때 사용하는 사용자명(username)을 다른 것으로 변경해서 테스트가 실패하는지 확인해보자.