

```
pytest
(venv) + planner git:(main) x pytest

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 14 items

tests/test_arithmetic_operations.py .... [ 28%]
tests/test_fixture.py . [ 35%]
tests/test_login.py .. [ 50%]
tests/test_routes.py ..... [100%]

===== 14 passed in 0.57s =====
(venv) + planner git:(main) x
```

그림 8-14 0.57초만에 완료된 전체 테스트

이벤트 플래너 API의 모든 라우트를 테스트했다. 이제 커버리지 테스트를 실행하여 테스트 대상이 되는 코드의 비율을 파악해보자.

8.4 테스트 커버리지

테스트 커버리지 보고서는 테스트가 전체 애플리케이션 코드 중 어느 정도 비율의 코드를 테스트하는지 정량화해서 보여준다. coverage 모듈을 설치해서 우리가 만든 API가 적절하게 테스트되고 있는지 확인해보자.

```
(venv)$ pip install coverage
```

다음 명령을 실행해서 테스트 커버리지 보고서를 생성해보자.

```
(venv)$ coverage run -m pytest
```

명령을 실행하면 [그림 8-15]와 같이 테스트 커버리지 보고서가 생성된다.