routes/events.py를 클릭하면 [그림 8-19]와 같이 해당 파일의 테스트 커버리지 보고서를 볼 수 있다.

Coverage for **routes/events.py**: 90%

41 statements | 37 run | 4 missing | 0 excluded

```
1  from typing import List
2
3  from auth.authenticate import authenticate
4  from beanie import PydanticObjectId
5  from database.connection import Database
6  from fastapi import APIRouter, Depends, HTTPException, status
7  from models.events import Event, EventUpdate
8
9  event_router = APIRouter(
10     tags=["Events"]
11 )
12
13 event_database = Database(Event)
14
15
16 @event_router.get("/", response_model=List[Event])
17 async def retrieve_all_events() -> List[Event]:
18     events = await event_database.get_all()
19     return events
20
21
22 @event_router.get("/{id}", response_model=Event)
23 async def retrieve_event(id: PydanticObjectId) -> Event:
24     event = await event_database.get(id)
25     if not event:
26         raise HTTPException(
27             status_code=status.HTTP_404_NOT_FOUND,
28             detail="Event with supplied ID does not exist"
29         )
30     return event
31
32
33 @event_router.post("/new")
```

그림 8-19 이 그림에는 색상이 표현되지 않았지만 실행된(사용된) 코드는 초록색, 그렇지 않은 코드는 빨간색으로 표시된다.

정리하기

CHAPTER 8에서는 인증 및 CRUD 라우트 테스트를 작성해서 API의 전체 기능을 테스트했다. 구체적으로는 테스트가 무엇인지, pytest를 사용하여 테스트를 어떻게 작성하는지 살펴봤다. 또한 pytest의 픽스처에 관해 배우고 이를 사용해서 재사용할 수 있는 접속 토큰과 데이터