

라우트에 보안을 적용했으니 각 라우트에 다음과 같은 처리를 적용해보자.

- **POST 라우트:** 생성된 이벤트를 사용자가 소유한 이벤트 목록에 추가한다.
- **UPDATE 라우트:** 사용자가 만든 이벤트만 변경할 수 있게 한다.
- **DELETE 라우트:** 사용자가 만든 이벤트만 삭제할 수 있게 한다.

라우트에 인증 의존성을 성공적으로 주입했다. 다음으로 다른 사용자가 이벤트를 삭제하거나 변경할 수 없도록 이벤트 문서 클래스와 라우트를 변경해보자.

## 이벤트 문서 클래스와 라우트 변경

다음과 같이 `creator` 필드를 `models/events.py` 파일의 `Event` 문서 클래스에 추가하자.

---

```
class Event(Document):
    creator: Optional[str]
```

---

`creator` 필드는 해당 이벤트를 소유한 사용자만 처리를 할 수 있게 한다.

다음과 같이 `routes/events.py`의 `POST` 라우트를 변경해서 새로운 이벤트가 생성될 때 `creator` 필드가 함께 저장되도록 해보자.

---

```
@event_router.post("/new")
async def create_event(body: Event, user: str = Depends(authenticate)) -> dict:
    body.creator = user
    await event_database.save(body)
    return {
        "message": "Event created successfully."
    }
```

---

`POST` 라우트를 변경해서 현재 사용자의 이메일을 이벤트 생성자(`creator`)로 추가했다. 따라서 다음 명령을 사용해 신규 이벤트를 만들면 해당 이벤트에 생성자의 이메일이 함께 저장된다.

---

```
$ curl -X 'POST' \
    'http://127.0.0.1:8000/event/new' \
    -H 'accept: application/json' \
```

---