
```

@event_router.delete("/{id}")
async def delete_event(id: PydanticObjectId, user: str = Depends(authenticate)) ->
dict:
...
    event = await event_database.get(id)
    if event.creator != user:
        raise HTTPException(
            status_code=status.HTTP_404_NOT_FOUND,
            detail="Operation not allowed"
        )
...

```

먼저 현재 사용자가 생성한 이벤트인지 확인하고 아니면 예외를 발생시킨다.

다음 예제에서는 다른 사용자 계정을 사용해 이벤트를 삭제한다.

```

$ curl -X 'DELETE' \
  'http://127.0.0.1:8000/event/6265a83fc823a3c912830074' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoziMmFzdGFwaUBwYWNrdCj5jb20iLCJleHBpcmVzIjoxNjUwODMzOTc2LjI2NzgzMjE0LmMRT6pwEDBVHTU5C1a6MV8j9wCfWhqbza9NBpZz08xE'

```

명령을 실행하면 다음과 같이 해당 이벤트의 소유자가 아니라는 응답이 반환된다.

```

{
  "detail": "Operation not allowed"
}

```

해당 이벤트를 소유한 사용자라면 이벤트를 삭제할 수 있다.

```

$ curl -X 'DELETE' \
  'http://127.0.0.1:8000/event/6265a83fc823a3c912830074' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoziMmFzdGFwaUBwYWNrdCj5jb20iLCJleHBpcmVzIjoxNjUwODMzOTc2LjI2NzgzMjE0LmMRT6pwEDBVHTU5C1a6MV8j9wCfWhqbza9NBpZz08xE'

```

명령을 실행하면 다음과 같이 이벤트를 삭제했다는 메시지를 볼 수 있다.