

정상적으로 요청이 처리되면 다음과 같은 메시지가 표시된다.

```
{
  "message": "Event created successfully."
}
```

처리가 실패한 경우에는 예외가 반환된다.

이벤트 조회

전체 이벤트를 추출하는 GET 라우트를 변경해서 데이터베이스에서 데이터를 가져오도록 만들어보자. 다음과 같이 routes/events.py 파일을 변경하면 된다.

```
from sqlmodel import select

@event_router.get("/", response_model=List[Event])
async def retrieve_all_events(session=Depends(get_session)) -> List[Event]:
    statement = select(Event)
    events = session.exec(statement).all()
    return events
```

마찬가지로 지정한 ID의 이벤트 정보를 표시하는 라우트도 변경한다.

```
@event_router.get("/{id}", response_model=Event)
async def retrieve_event(id: int, session=Depends(get_session)) -> Event:
    event = session.get(Event, id)
    if event:
        return event

    raise HTTPException(
        status_code=status.HTTP_404_NOT_FOUND,
        detail="Event with supplied ID does not exist"
    )
```

두 라우트의 응답 모델이 모델 클래스를 사용하도록 설정했다.

이제 이벤트 목록을 반환하는 GET 요청을 테스트해보자.