

7 방금 생성한 계정으로 로그인해보자.

```
(venv)$ curl -X 'POST' \  
  'http://127.0.0.1:8000/user/signin' \  
  -H 'accept: application/json' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "email": "fastapi@packt.com",  
    "password": "strong!!!"  
  }'
```

윈도우에서는 다음과 같이 실행한다.

```
curl -X POST "http://127.0.0.1:8000/user/signin" -H "accept: application/  
json" -H "Content-Type: application/json" -d "{\"email\": \"fastapi@packt.  
com\", \"password\": \"strong!!!\"}"
```

명령을 실행하면 HTTP 200 응답이 반환된다. 정상적으로 요청이 처리됐다는 의미다.

```
{  
  "message": "User signed in successfully."  
}
```

정리하기

CHAPTER 6에서는 SQLAlchemy과 beanie를 사용해 애플리케이션을 SQL 및 NoSQL 데이터베이스와 연동하고 각각의 이벤트가 의도한 대로 실행되는지 테스트했다.

CHAPTER 7에서는 웹 애플리케이션 보안에 관해 소개한다. 기본 인증 방법을 포함하여 FastAPI 개발자가 사용할 수 있는 다양한 인증 방법을 다룬 다음 JWT^{JSON Web Token}를 기반으로 한 인증 시스템을 구현하고 각 이벤트 라우트에 인증을 적용한다. 마지막으로 라우트를 변경하여 이벤트와 사용자를 연결한다.