

```
python-dotenv==0.20.0
python-jose==3.3.0
sqlmodel==0.0.6
uvicorn==0.17.6
```

이렇게 하면 애플리케이션이 직접적으로 사용하는 의존 라이브러리만 관리할 수 있다.

환경 변수 설정

〈CHAPTER 6 데이터베이스 연결〉에서 환경 변수를 사용해봤다. 환경 변수를 배포 시점에 적용할 수도 있는데 이 내용은 다음 절에서 살펴보자.

환경 변수는 적절한 방법으로 관리하는 것이 중요하다. 특히 GitHub 같은 버전 관리 시스템에서는 제외시키는 것이 좋다.⁴⁶

배포를 위한 준비가 끝났다. 이제 도커를 사용해 애플리케이션을 로컬에 배포해보자.

9.2 도커를 사용한 배포

〈CHAPTER 1 FastAPI 소개〉에서 도커와 도커파일의 기본적인 내용을 살펴봤으므로 이벤트 플레너 API를 위한 도커파일을 직접 작성해보자.

도커는 컨테이너화(containerization)에 사용되는 가장 인기 있는 기술이다. 컨테이너는 패키지, 코드, 의존 라이브러리로 구성된 하나의 시스템으로, 실행 환경에 의존하지 않는다. 따라서 다른 환경에서도 쉽게 애플리케이션을 배포할 수 있다. 도커는 도커파일을 사용해서 컨테이너화한다.

도커는 로컬 개발 환경뿐만 아니라 프로덕션 애플리케이션을 배포할 때도 사용된다. 여기서는 로컬 개발 환경만 다루지만 클라우드 서비스에 배포하는 방법도 공유할 것이니 참고하도록 하자.

애플리케이션 컨테이너와 데이터베이스 컨테이너처럼 여러 컨테이너를 사용해서 애플리케이션을 구성하는 경우에는 도커 구성 도구를 사용한다. 도커 구성 도구는 여러 컨테이너로 구성

⁴⁶ 옮긴이_연결 정보 등 민감한 정보가 포함되기 때문이다. 하지만 환경 변수를 포함시키는 경우도 종종 있다. 이때 중요한 정보는 암호화하거나 CI/CD 파이프라인과 연동해서 사용한다.