

이번에는 패스워드를 해싱해서 데이터베이스에 저장하도록 routes/users.py의 사용자 등록 라우트를 수정해보자.

---

```
from auth.hash_password import HashPassword
from database.connection import Database

user_database = Database(User)
hash_password = HashPassword()

@user_router.post("/signup")
async def sign_user_up(user: User) -> dict:
    user_exist = await User.find_one(User.email == user.email)

    if user_exist:
        raise HTTPException(
            status_code=status.HTTP_409_CONFLICT,
            detail="User with email provided exists already"
        )
    hashed_password = hash_password.create_hash(user.password)
    user.password = hashed_password
    await user_database.save(user)
    return {
        "message": "User created successfully."
    }
```

---

이렇게 하면 사용자 등록 라우트가 사용자를 등록할 때 패스워드를 해싱한 후 저장한다.

실제 사용자를 등록해보자. 터미널(또는 명령 프롬프트) 창에서 다음 명령을 사용해 애플리케이션을 실행한다.

---

```
(venv)$ python main.py
INFO:      Uvicorn running on http://127.0.0.1:8000 (Press Ctrl+C to quit)
INFO:      Started reloader process [8144] using statreload
INFO:      Started server process [8147]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
```

---

새로운 터미널 창을 열어서 몽고DB 인스턴스를 실행한다.<sup>40</sup>

---

<sup>40</sup> 올긴이 만약 --dbpath가 실행되지 않는다면 다른 버전의 몽고DB를 설치해보자.