
```

@event_router.delete("/{id}")
async def delete_event(id: PydanticObjectId) -> dict:
    event = await event_database.delete(id)
    if not event:
        raise HTTPException(
            status_code=status.HTTP_404_NOT_FOUND,
            detail="Event with supplied ID does not exist"
        )
    return {
        "message": "Event deleted successfully."
    }

```

이벤트 라우트용 CRUD 처리를 모두 구현했다. 이어서 사용자 등록 및 로그인 라우트를 구현해보자.

routes/users.py

import문과 데이터베이스 인스턴스 생성 코드를 다음과 같이 변경한다.

```

from fastapi import APIRouter, HTTPException, status
from database.connection import Database

from models.users import User, UserSignIn

user_router = APIRouter(
    tags=["User"],
)

user_database = Database(User)

```

사용자 등록용 POST 라우트를 다음과 같이 변경한다.

```

@user_router.post("/signup")
async def sign_new_user(user: User) -> dict:
    user_exist = await User.find_one(User.email == user.email)
    if user_exist:
        raise HTTPException(
            status_code=status.HTTP_409_CONFLICT,

```
