

`update()` 메서드는 하나의 ID와 pydantic 스키마(모델)를 인수로 받는다. 스키마에는 클라이언트가 보낸 PUT 요청에 의해 변경된 필드가 저장된다. 변경된 요청 바디는 딕셔너리에 저장된 다음 `None` 값을 제외하도록 필터링된다. 이 작업이 완료되면 변경 쿼리에 저장되고 beanie의 `update()` 메서드를 통해 실행된다.

삭제 처리

데이터베이스에서 레코드를 삭제하는 메서드를 작성해보자.

```
async def delete(self, id: PydanticObjectId) -> bool:
    doc = await self.get(id)
    if not doc:
        return False
    await doc.delete()
    return True
```

이 메서드는 먼저 해당 레코드가 있는지 확인하고 있으면 삭제한다.

CRUD 처리용 메서드가 추가된 데이터베이스 파일을 완성했다. 이제 라우트를 변경해보자.

routes/events.py

import문과 database 인스턴스를 다음과 같이 변경한다.

```
from beanie import PydanticObjectId
from fastapi import APIRouter, HTTPException, status
from database.connection import Database

from models.events import Event, EventUpdate
from typing import List
event_database = Database(Event)
```

모든 라우트를 변경해야 한다. GET 라우트부터 시작하자. GET 라우트는 앞서 데이터베이스 파일에 정의한 메서드를 호출한다.