```
    name: str
    location: str

    class Settings:
        name = "events"
```

여기서 Settings 서브 클래스는 몽고DB 데이터베이스 내에 설정한 이름으로 컬렉션을 생성한다.

문서 생성 방법을 알았으니 CRUD 처리를 위한 메서드를 살펴보자.

- **insert(), create()**: 문서 인스턴스에 의해 호출되며 데이터베이스 내에 새로운 레코드를 생성한다. 단일 데이터는 insert_one() 메서드를 사용해 추가하고 여러 개의 데이터는 insert_many() 메서드를 사용해 추가한다.

```
event1 = Event(name="Packt office launch", location="Hybrid")
event2 = Event(name="Hanbit office launch", location="Hybrid")
await event1.create()
await event2.create()
await Event.insert_many([event1, event2])
```

- **find(), get()**: find() 메서드는 인수로 지정한 문서를 문서 목록에서 찾는다. get() 메서드는 지정한 ID와 일치하는 단일 문서를 반환한다. find_one() 메서드는 다음과 같이 지정한 조건과 일치하는 단일 문서를 반환한다.[35]

```
event = await Event.get("74478287284ff")
# 일치하는 아이템의 리스트를 반환한다.
event = await Event.find(Event.location == "Hybrid").to_list()
# 단일 이벤트를 반환한다.
event = await.find_one(Event.location == "Hybrid")
```

- **save(), update(), upsert()**: save() 메서드는 데이터를 신규 문서로 저장할 때 사용된다. update() 메서드는 기존 문서를 변경할 때 사용되고, upsert() 메서드는 조건에 부합하는 문서가 없으면 신규로 추가할 때 사용된다.[36] 여기서는 update() 메서드를 사용하며 다음과 같이 변경용 쿼리를 지정한다.

35 옮긴이_"74478287284ff"는 몽고DB의 고유한 문서 ID를 의미한다.

36 옮긴이_즉, 변경하려는 문서가 없으면 새롭게 추가한다. update와 insert를 합친 용어다.