

# m:n FOLLOW

- 선행할 것 m:n like, profile

## 1. Model

```
class User(AbstractUser):
    followings = models.ManyToManyField('self', symmetrical=False,
    related_name='followers')
```

- 기본 유저 모델을 수정해서 사용
  - 재귀 참조는 'self' : user가 user를 참조
  - symmetrical: 내가 팔로우하면 자동으로 상대도 날 팔로우 시키는 기능, sns에서는 보통 False
  - 재귀여서 바뀌어도 상관 없지만, 내가 팔로우한 사람(참조) `followings` 나를 팔로우한 사람 (역참조) `followers` 로 의미상 맞게 지어주었다.

## 2. URL

```
urlpatterns = [
    path('<int:user_pk>/follow/', views.follow, name='follow'),
]
```

- 인스타, 페이스북 등에서 유저네임이 URL로 흔하기 때문에, 여기서도 이를 사용

## 3. views.py

```
@require_POST
def follow(request, user_pk):
    if request.user.is_authenticated:
        # 팔로우 받는 사람
        you = get_object_or_404(get_user_model(), pk=user_pk)
        me = request.user

        # 나 자신은 팔로우 할 수 없다.
        if you != me:
            #내가 이미 팔로우를 하고 있는 지 확인
            if you.followers.filter(pk=me.pk).exists():
                # if request.user in person.followers.all():
                # 팔로우 끊음
                you.followers.remove(me)
            else:
                # 팔로우 신청
                you.followers.add(me)
```

```
return redirect('accounts:profile', you.username)
return redirect('accounts:login')
```

- 왜 팔로우확인 부분에서 get이 아니라 filter를 쓰는 걸까요?
  - get은 값이 없으면 오류를 발생시키기 때문에 오류가 없는 filter를 사용
- person으로 해도 되지만, 이해를 돕기 위해 you,me로 쉽게 구분

## 4. profile.html

- 편의상 `##내용##`을 주석으로 사용

```
# {% with %} 와 {%endwith%} 사이에서 변수명= 변수(following = person.following.all)
로 시각적으로 편리하게 표기할 수 있다.
```

```
{% with followings=person.followings.all followers=person.followers.all %}
<div>
  <div>
    팔로잉 : {{ followings|length }} / 팔로워 : {{ followers|length }}
  </div>
</div>
```

```
#####
# 내 프로필에서는 팔로우 버튼 자체가 보이지 않아야함
{% if request.user != person %}
  # 팔로우 중이면 언팔표시, 언팔중이면 팔로우 표시
  <div>
    <form action="{% url 'accounts:follow' person.pk %}" method="POST">
      {% csrf_token %}
      {% if request.user in followers %}
        <button>언팔로우</button>
      {% else %}
        <button>팔로우</button>
      {% endif %}
    </form>
  </div>
{% endif %}
</div>
{% endwith %}
```

- with tag를 이용하여, 변수를 만들어 문서를 정리해줄 수 있다.
  - 최적화가 아니다.
- 재귀이므로 참조, 역참조 구분에 주의할 것
- 팔로잉: `person.following.all|length` 팔로워(역참조): `person.followers.all|length`
- 3번 유저가 1번 유저를 팔로잉 함(from, to로 테이블이 구분 된다.)
  - 이게 (1,3), (3,1)로 대칭일 시 맞팔

SQL ▼

id	from_user_id	to_user_id
1	3	1