

date: 0331

# m:n PROFILE

---

- 선행할 것 m:n like

## 1. Model

---

- 기본 유저 모델 사용,
- 아래는 코드이해의 편의를 돕기 위해 작성, 추가 작성x

```
class User(AbstractUser):  
    pass
```

```
class Article(models.Model):  
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)  
    like_users = models.ManyToManyField(settings.AUTH_USER_MODEL,  
related_name='like_articles')  
    #.....  
  
class Comment(models.Model):  
    article = models.ForeignKey(Article, on_delete=models.CASCADE)  
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)  
    #.....
```

## 2. URL

---

```
urlpatterns = [  
    path('<username>/', views.profile, name='profile'),  
]
```

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

## 3. views.py

---

```
def profile(request, username):
    # 유저모델을 가져오고, 그 중, username이 동일한 객체를 가져옴
    person = get_object_or_404(get_user_model(), username=username)
    context = {
        'person': person,
    }
    return render(request, 'accounts/profile.html', context)
```

- `user` 라는 변수명은 사용하지 말 것 (기본값이 있어서 아쉬워지기 때문에,)
- `person` 객체의 모든 정보를 전달

## 4. profile.html

- 편의상 `##내용##` 을 주석으로 사용

```
{% extends 'base.html' %}

{% block content %}
<h1>{{ person.username }}님의 프로필</h1>

## 1: n 관계, 필스 객체에 article정보를 역참조 ##
<h2>{{ person.username }}'s 게시글</h2>
{% for article in person.article_set.all %}
    <div>{{ article.title }}</div>
{% endfor %}

<hr>

## 1: n 관계, 필스 객체에 comment정보를 역참조 ##
<h2>{{ person.username }}'s 댓글</h2>
{% for comment in person.comment_set.all %}
    <div>{{ comment.content }}</div>
{% endfor %}

<hr>

## m: n 관계, 필스 객체에 like_users정보를 역참조 ##
## related_name에 따라 `like_articles` 사용 ##
<h2>{{ person.username }}'s likes</h2>
{% for article in person.like_articles.all %}
    <div>{{ article.title }}</div>
{% endfor %}

<hr>

<a href="{% url 'articles:index' %}">[back]</a>
{% endblock %}
```

- `user` 객체는 ForeignKey 없이 모두 참조 당하고 있으므로 역참조로 해당정보들을 가져옴

