Social 模块开发

> **Organization**: 千锋教育 Python 教学部
>
> **Date**: 2019-02-12
>
> **Author**: 张旭

功能概述

1. 交友模块
    - 获取推荐列表
    - 喜欢 / 超级喜欢 / 不喜欢
    - 反悔最后一次的滑动 (每天允许反悔 3 次，反悔的记录只能是五分钟之内的)
    - 查看都有谁喜欢过我的人
2. 好友模块
    - 查看好友列表

开发中的难点

1. 滑动需有大量用户，如何初始化大量用户以供测试?
2. 推荐算法
3. 如何从推荐列表中去除已经滑过的用户
4. 滑动操作，如何避免重复滑动同一人
5. 如果双方互相喜欢，需如何处理
6. 好友关系如何记录，数据库表结构如何设计?
7. 反悔接口

    1. “反悔”都应该执行哪些操作
    2. 每日只允许“反悔” 3 次应如何处理
    3. 后期运营时，如何方便的修改反悔次数
8. 内部很深的逻辑错误如何比较方便的将错误码返回给最外层接口

关系分析

1. 滑动者与被滑动者
    - 一个人可以滑动很多人
    - 一个人可以被多人滑动
    - 结论: 同表之内构建起来的逻辑上的多对多关系
2. 用户与好友
    - 一个用户由多个好友
    - 一个用户也可以被多人加为好友
    - 结论: 同表之内构建起来的逻辑上的多对多关系, Friend 表实际上就是一个关系表

模型设计参考

1. Swiped (划过的记录)

Field	Description
uid	用户自身 id
sid	被滑的陌生人 id
mark	滑动类型
time	滑动的时间

2. Friend (匹配到的好友)

Field	Description
uid1	好友 ID
uid2	好友 ID

类方法与静态方法

- `method`
  - 通过实例调用
  - 可以引用类内部的 **任何属性和方法**
- `classmethod`
  - 无需实例化
  - 可以调用类属性和类方法
  - 无法取到普通的成员属性和方法
- `staticmethod`
  - 无需实例化
  - **无法**取到类内部的任何属性和方法, 完全独立的一个方法

利用 Q 对象进行复杂查询

```
from django.db.models import Q

# AND
Model.objects.filter(Q(x=1) & Q(y=2))

# OR
Model.objects.filter(Q(x=1) | Q(y=2))

# NOT
Model.objects.filter(~Q(name='kitty'))
```