为什么要有通知模块，而不是和新鲜事模块合并？

定义不同，应用场景不同。

不要自己新造轮子，通知模块是一个非常常用的模块，可以在github上去搜索。

站内通知：Twitter手机端，网站上有类似一个铃铛标志的通知提醒。

，你的tweet，comment被谁点赞了，你的tweet被谁评论了，这个会存储在数据库中。

django-notifications和django-push-notifications

手机上的通知叫push notifications，推送通知，这个通知没有存储在数据库，推送通知就是手机屏幕向下滑动时显示的通知。

<https://blog.csdn.net/weixin_43217710/article/details/90319309>

django-notifications中的send方法

from notifications.signals import notify

notify.send(actor, recipient, verb, action\_object, target, level, description, public, timestamp, \*\*kwargs)

参数：

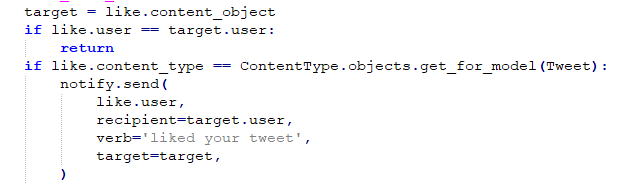
actor：发送通知的对象[必需]，注意：如果使用关键字参数，使用sender，即send(sender:xxx)，而不是actor

recipient：接收通知对象[必需]

verb：动词短语[必需]，告知recipient对象actor干了什么

action\_object：链接到动作的对象[可选]，相当于下图中的like对象

target：被发送通知的对象[可选]，例如当点赞了某条tweet时，actor就是点赞的user，recipient就是tweet的user，而target就是点赞的tweet。



level：通知等级[可选]，有4种(success', 'info', 'warning', 'error')，默认为info

description：描述字符串[可选]

public：是否公开[可选]，默认为True

timestamp：时间戳[可选]，默认为当前时间



verb动词短语有一些小技巧，例如：

给你的帖子点了赞，前端进行渲染时，希望把tweet对应的超链接渲染到“给你的帖子”后面，“点了赞”的前面，这时可以和前端约定：



前端拿到verb后，将其中的{target}替换成想渲染的超链接，并且能支持英文：



后端一般不处理具体渲染的对象，因为前端有可能是Web端，也有可能是手机端，跳转的方式不一致，因此需要足够的灵活性，后端之负责提供好数据。

工程中inbox是notifications的别名，notifications已经被第三方的库占用了。

django中的Signal机制和Qt的信号与槽机制类似，二者之间应该有区别，否则不会有在Python中调用Qt的信号与槽的方法。

django-notifications中自带的url不太符合restful规范，需要自己实现。

django中有自带的用户模块，但它有些属性没有包含，例如昵称，头像等，所以需要添加这些属性，有两种方式，一种方式是直接继承django中的AbstractUser类，然后再修改配置文件，定义AUTH\_USER\_MODEL字段，这种方式不建议，一是因为用户自定义的Model一般都直接继承models.Model就行了，避免过深的继承。二是之前使用了User类，如果重新自定义User类，有可能会产生数据不兼容的问题。三是根据“劝分不劝和”的原则，User类中原本的username，password，email通常是登陆时使用，而昵称，头像则是展示用户时，二者的应用场景不同，而且前者不太经常改，而后者则有可能经常改。另一种方式是增加一个UserProfile模型，存放新增的字段。这样的好处是二者可以分开存储，尤其是如果使用缓存cache时，二者放在不同的地方，由于UserProfile会经常更改，这样cache中会经常失效，但不会影响User的cache。

上传头像时，默认情况下会放到工程文件夹下，显然不符合我们的要求，需要进行设置上传文件的存放路径。