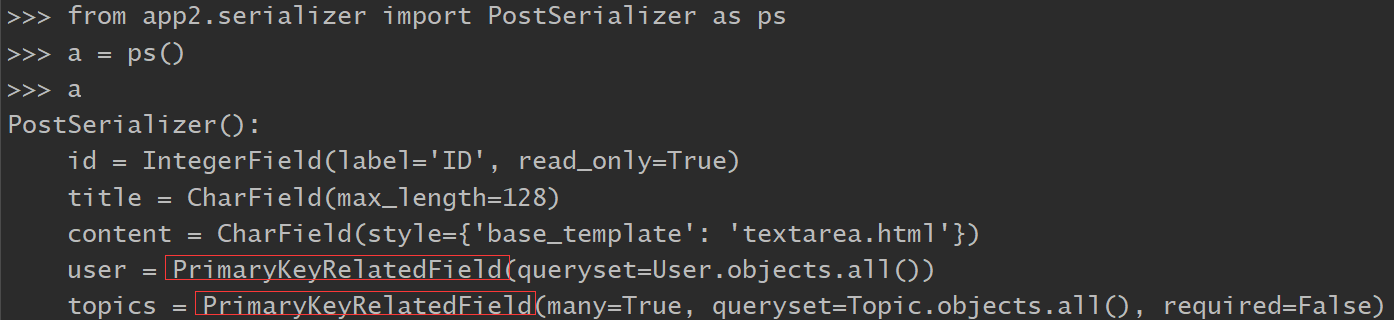
ModelSerializer类：

1. 根据模型的定义，自动生成默认字段
2. 自动生成相应的验证器
3. 实现了简单的create（）和update（）方法
4. 自动默认将关系字段映射成为PrimaryKeyRelatedField 下图为上一节课列子



字段序列化的三种方式：三存一

class PostSerializer(serializers.ModelSerializer):

class Meta:

model = Post

**# 1.显示指定需要序列化的字段（推荐使用）**

fields = ('id', 'title', 'content', 'user', 'topics')

**# 2.全部字段序列化（不推荐使用，会把密码等字段序列化）**

fields = "\_\_all\_\_"

**# 3.除了name字段不序列化，其余字段都序列化**

exclude = ("name",)

# 设置深度depth,depth可以把主键显示为对应的内容

depth = 1

# 设置只读字段，不允许用户修改

read\_only\_field = ('user',)

**HyperLinkModelSerializer：**

在fields序列化字段一栏，默认增加了url的序列化



需要注意的是：

1. 通过url序列化的字段默认是一个绝对路径如果想要修改为相对路径则需要如下设置：默认为：serializer = PostSerializer(queryset, context={‘request’:’request’})

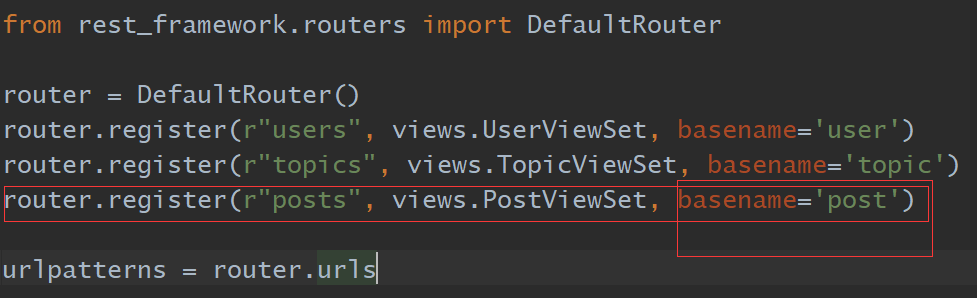
改为{‘request’:None}

1. 怎样生成的链接，因为一个链接需要一个对应的url或者说是一个视图来处理。方式如下：

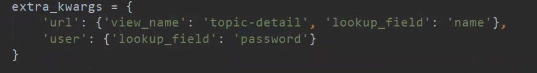
默认跳转到：{model\_name} - detail

如何实现：{model\_name} - detail name=‘post-detail’ /posts/<int:pk>

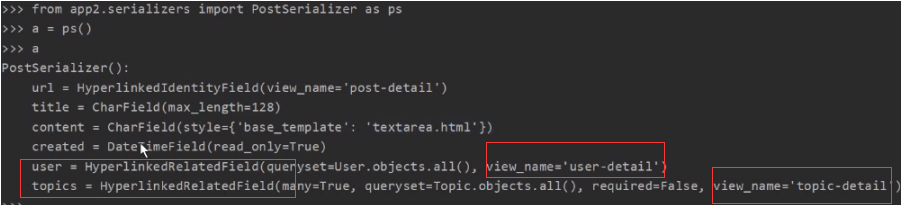
原理：



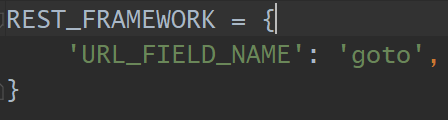
如果需要更改默认的view\_name(那个视图) 和lookup\_field（查询字段 ，默认为pk）。如下即为/topics/<str:title>



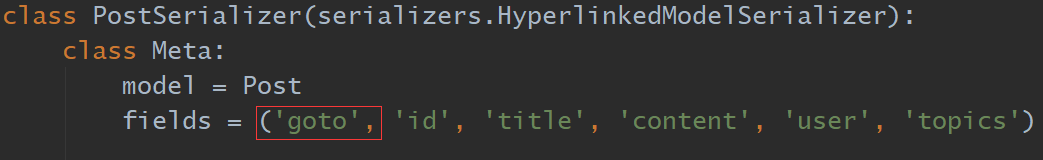
自动默认将关系字段映射成为HyperlinkedRelatedField



想更改url名称则需要在settings下配置



然后



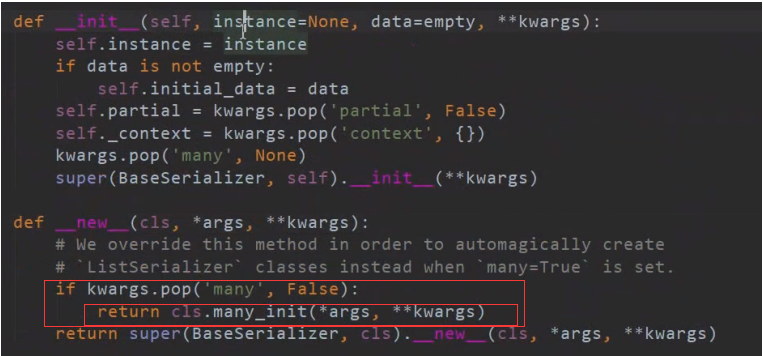
结果



**ListSerializer：**

1. 适用于极高的自定义或者是涉及源码的修改
2. 当我们在序列化包含many=true时，ListSrializer会自动调用

在BaseSrializer里有如下源代码：



在many.init如有下方法

