**一、权限**Permissions

DRF的权限类位于permissions模块中。

permission\_class = [1,2,3]。**不满足其中任何一个权限都会导致权限失败。**从而抛出exceptions.PermissionDenied或者exceptions.NotAuthenticated即

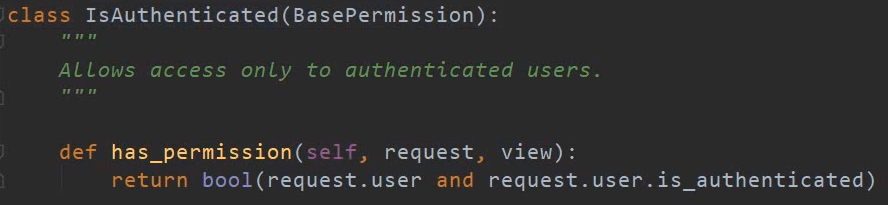
403 Forbidden 或者401 Unatho rized

**AllowAny**

AllowAny 权限类允许不受限制的访问，并且**不管该请求是否通过身份验证或未经身份验证**。

**IsAuthenticated**

IsAuthenticated 权限类将拒绝任何未经身份验证的用户的访问，通过身份验证的用户则拥有所有权限。 如果你希望你的API仅供注册用户访问，则此权限比较合适。继承BasePermission类，需要重写方法has\_permission(self, request, view)

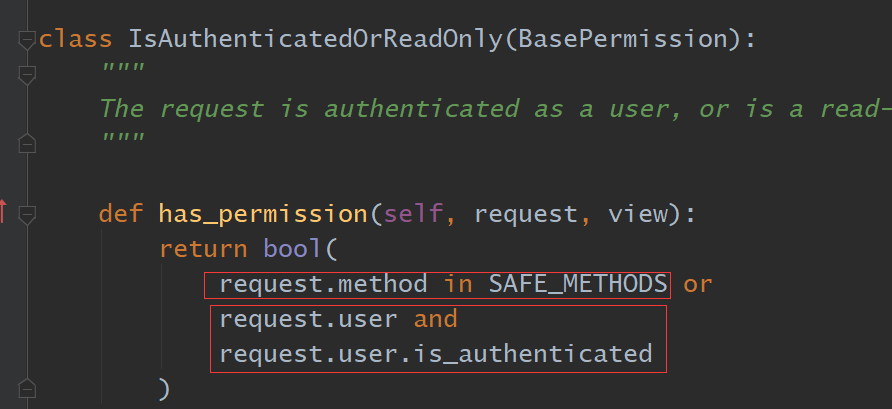


**IsAdminUser**

除非 user.is\_staff 属性的值为 True ，否则 IsAdminUser 权限类将拒绝来者访问API。 user.is\_staff 属性是Django的auth框架中，管理员的类别，可以在Admin后台中查看和指 定，也可以通过createsuperuser命令创建。如果你希望你的API只能被部分受信任的管理员访问，则此权限比较适合。



**IsAuthenticatedOrReadOnly** IsAuthenticatedOrReadOnly 权限类允许经过身份验证的用户执行任何请求。未经授权的 用户请求，只有当请求方法是“安全”的（即 GET , HEAD 或 OPTIONS 之一时），才允许访问。如果你希望你的API允许匿名用户读取，并且只允许对通过身份验证的用户可读写，则此权限比较适合。这也是大多数公开服务的选择。



**三、自定义权限**

1. 需要继承 BasePermission 类

2. 实现has\_permission(self, request, view)或者has\_object\_permission(self, request, view, obj)。并且返回一个bool值，(授予权限)True或者False

如果是安全的访问方法如：GET/HEAD/OPTIONS则执行权限一，如果是不安全的方法执行权限二。

SAFE\_METHODS=（“GET”,”HEAD”,”OPTIONS”）

3. if request.method in permissions.SAFE\_METHODS:

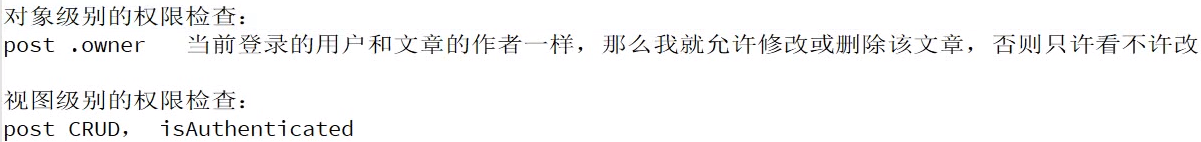
# 检查只读请求的权限

else:

# 检查写入请求的权限

has\_permission(self, request, view) 是视图级别的权限检测方法（普通）

has\_object\_permission(self, request, view, obj)是对象级别的权限检测方法



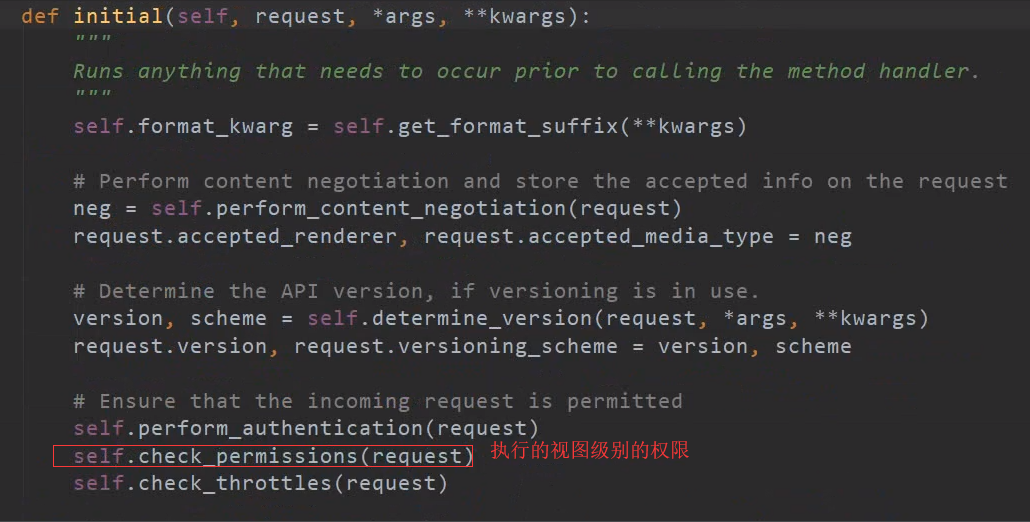
注意：在调用对象级别的权限时。如果你的视图不是继承了通用视图比如ListCreateGernericAPIview，而是继承了更为底层的APIView时，你需要在视图中显示的调用如下方法：

**check\_object\_permissions(request, obj)**

**在APIView有源码如下：**



但是在APIView的initial方法中只执行了试图级别的权限，因此需要显示的调用has\_object\_permission(self, request, view, obj)。



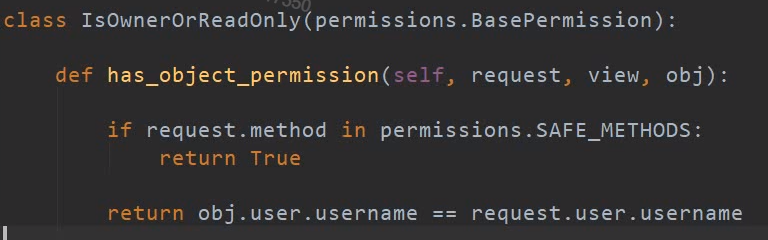
如果你继承了通用试图，但是在视图中重写了（GenericAPIView）get\_object()方法你依然要示的调用has\_object\_permission(self, request, view, obj)。

1. 自定义一个简单的黑名单权限(试图级别)





1. 如果是自己写的就有全部权限，否则只读(对象级别)

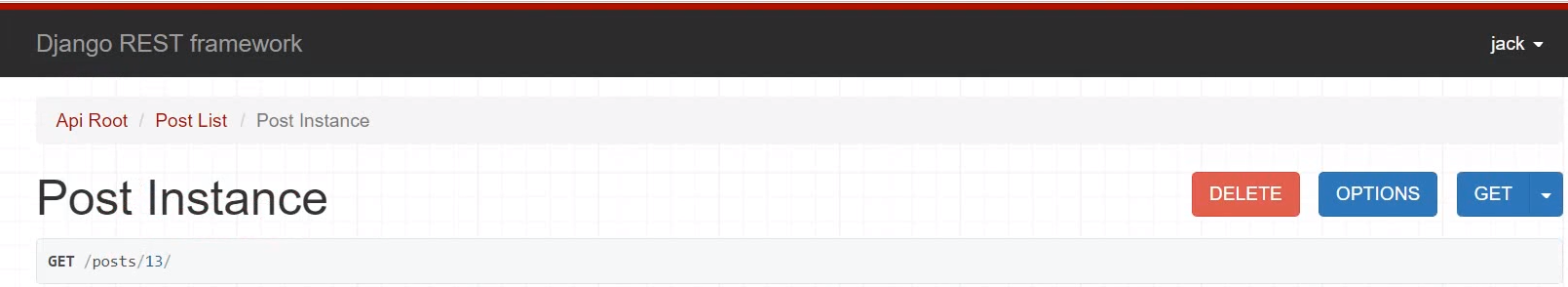


obj.user.username与request.user.username

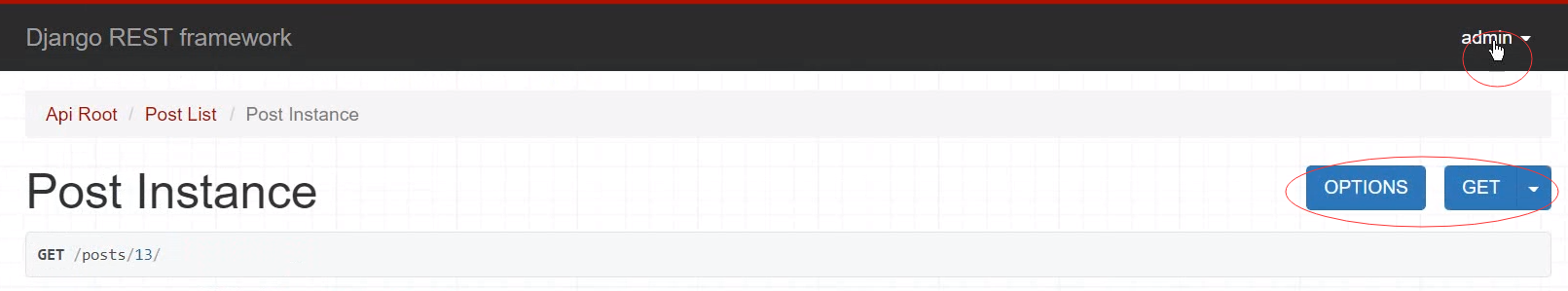
一个是文章关联的用户，一个是登录用户

下面两种情况访问jack所写的一个文章

Jack:

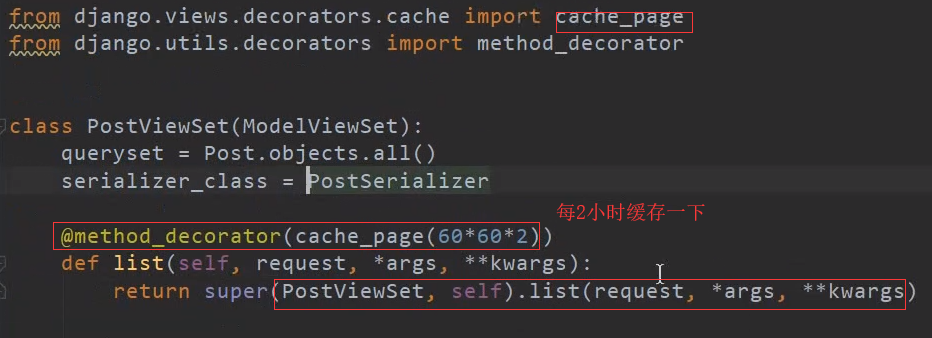


Admin:



**缓存：**

用Django自带的缓存



cache\_page只能缓存返回200的GET和HEAD响应。其他不行