通用视图

APIView是较为底层的封装，给我们的感觉，主要是对请求对象的封装，为查询字符串、提交的Json数据提供方便的属性等。

首先，EmpsView、EmpView的代码具有普遍性，其它对象增删改查也差不多。

其次，能否实现更加灵活的功能？有些需要列表页，有些需要详情页，有些则需要修改，有些需要新增，有些需要删除的功能，而能灵活做到这一点的可以是Mixin技术。

能否进一步封装，且做到自由组合？这就是GenericAPIView，它派生自APIView。

参考 https://www.django-rest-framework.org/api-guide/generic-views/#genericapiview

它只提供了

- **serializer_class**，类属性，表示序列化类型，必须指定或覆盖get_serializer_class()
- get_serializer_class()返回值才是真正的被使用的序列化类，默认是serializer_class。如有必要，覆盖
- **queryset**，类属性，默认为None，必须指定或者覆盖get_queryset()
- 搜索单个对象是按照某些字段搜索，需要提供某些参数，也就是在URL conf中定义的使用关键字传参注入到方法中的参数
  - lookup_field，其默认值为'pk'，该类属性一般不需要修改
  - lookup_url_kwarg默认None，如果为None就使用lookup_field，如果不是'pk'，建议一定要设置
  - `rest_framework.generics.GenericAPIView.get_object`中有`lookup_url_kwarg = self.lookup_url_kwarg or self.lookup_field`

```python
path('<int:pk>', EmpView.as_view()) # lookup_url_kwarg

class EmpView(GenericAPIView):
    def get(self, request:Request, pk:int): # lookup_url_kwarg
        pass
```

- get_queryset()，其内部使用queryset类属性。Mixin里面查数据集，调用都是get_queryset方法。如有必要，覆盖
- **get_object()**，详情页使用，其内部使用了get_queryset方法。Mixin里面获取单个对象调用它。如有必要，覆盖

简单讲，它定义了序列化、查询、分页功能接口。

注意：GenericAPIView不能增删改查，增删改查需要Mixin其他类。

Mixins

参考 https://www.django-rest-framework.org/api-guide/generic-views/#mixins

以下是孤立的功能Mixin类，需要和GenericAPIView类组合

1. ListModelMixin，list方法，列出所有数据，成功返回200。可以分页
2. CreateModelMixin，create方法，创建并保存一个对象，成功返回201，失败返回400
3. RetrieveModelMixin，retrieve方法，取回一个对象，成功返回200，失败返回404
4. UpdateModelMixin，update方法，更新并保存一个对象，成功返回200，失败返回400
5. DestroyModelMixin，destroy方法，对已存在对象进行删除，成功返回204，失败返回404

仅仅使用GenericAPIView不能简化什么代码，因为在EmpsView、EmpView中，写的最多的就是get、put、post、delete方法，省略方法也是目标。当然，派生自GenericAPIView后，不要忘了指定queryset和serializer_class。

下面就利用GenericAPIView和Mixins重构View代码

```
from .models import Employee
from .serializers import EmpSerializer

from rest_framework.generics import GenericAPIView
from rest_framework.mixins import (
    ListModelMixin,     # list -> get
    CreateModelMixin,   # create -> post
    RetrieveModelMixin, # retrieve -> get
    UpdateModelMixin,   # update -> put
    DestroyModelMixin   # destroy -> delete
)

class EmpsView(ListModelMixin, CreateModelMixin, GenericAPIView):
    """
    实现列表页get、新增post
    http://127.0.0.1:8000/emp/
    """
    queryset = Employee.objects.all()
    serializer_class = EmpSerializer

    get = ListModelMixin.list
    post = CreateModelMixin.create
    # def get(self, request):
    #     emps = Employee.objects.all()
    #     return Response(EmpSerializer(emps, many=True).data)

    # def post(self, request):
    #     serializer = EmpSerializer(data=request.data)
    #     serializer.is_valid(True)
    #     serializer.save()
    #     return Response(serializer.data)

class EmpView(RetrieveModelMixin, UpdateModelMixin,
              DestroyModelMixin, GenericAPIView):
    """
    实现详情页get、修改put、删除delete
    http://127.0.0.1:8000/emp/10021
    """
    queryset = Employee.objects.all()
    serializer_class = EmpSerializer

    get = RetrieveModelMixin.retrieve
    put = UpdateModelMixin.update
```

```
    delete = DestroyModelMixin.destroy
    patch = UpdateModelMixin.partial_update
    # def get(self, request, pk:int):
    #     obj = Employee.objects.get(pk=pk)
    #     return Response(EmpSerializer(obj).data)
    #
    # def put(self, request, pk:int):
    #     # 必须先查后改
    #     obj = Employee.objects.get(pk=pk)
    #     serializer = EmpSerializer(obj, data=request.data)
    #     serializer.is_valid(True)
    #     serializer.save()
    #     return Response(serializer.data, 201)
    #
    # def delete(self, request, pk:int):
    #     Employee.objects.get(pk=pk).delete()
    #     return Response(status=204)
```

利用了写好的Mixin省掉了大量的代码。

能把上面代码中的get、post、put、delete方法也能简化掉吗？可以，DRF替你写好了

Concrete视图类

Concrete View类是由GenericAPIView和诸多Mixin类构成的子类。

> Concrete具体的，也指混凝土，可以认为就是用泛型类GenericAPIView和各种Mixin类混合浇筑成的现成的、具体的，直接可以用的类。
>
> Concrete View Class，就简称CVS

如果不需要自定义的话，可以考虑使用这些提前定义好的通用的View类。

列表页View	**继承**	**方法**	**功能**
CreateAPIView	GenericAPIView, CreateModelMixin	post	列表页新增对象功能
ListAPIView	GenericAPIView, ListModelMixin	get	获得列表页内容
ListCreateAPIView	GenericAPIView, ListModelMixin, CreateModelMixin	get、post	**完整列表页功能**

详情页View	继承	方法	功能
RetrieveAPIView	GenericAPIView, RetrieveModelMixin	get	获取单个对象
UpdateAPIView	GenericAPIView, UpdateModelMixin	put、patch	修改单个对象
DestroyAPIView	GenericAPIView, DestroyModelMixin	delete	删除单个对象
RetrieveUpdateDestroyAPIView	GenericAPIView, RetrieveModelMixin, UpdateModelMixin, DestroyModelMixin	get、put、patch、delete	**详情页查、改、删功能**
RetrieveUpdateAPIView	GenericAPIView, RetrieveModelMixin, UpdateModelMixin	get、put、patch	
RetrieveDestroyAPIView	GenericAPIView, RetrieveModelMixin, DestroyModelMixin	get、delete	

```python
from .models import Employee
from .serializers import EmpSerializer
from rest_framework.generics import ListCreateAPIView, RetrieveUpdateDestroyAPIView

class EmpsView(ListCreateAPIView):
    """
    实现列表页get、新增post
    http://127.0.0.1:8000/emp/
    """
    queryset = Employee.objects.all()
    serializer_class = EmpSerializer

class EmpView(RetrieveUpdateDestroyAPIView):
    """
    实现详情页get、修改put、删除delete
    http://127.0.0.1:8000/emp/10021
    """
    queryset = Employee.objects.all()
    serializer_class = EmpSerializer
    #lookup_url_kwarg = 'id' # 路由配置中不是pk可以设置
```

分页

参考 https://www.django-rest-framework.org/api-guide/pagination/#setting-the-pagination-style

全局配置

```
REST_FRAMEWORK = {
    'DEFAULT_PAGINATION_CLASS':
'rest_framework.pagination.PageNumberPagination',
    'PAGE_SIZE': 10
}
```

测试http://127.0.0.1:8000/emp/，分页返回数据了。

测试http://127.0.0.1:8000/emp/?page=3，返回了第3页。

自定义分页类

测试http://127.0.0.1:8000/emp/?page=3&page_size=2，page_size似乎没有用，为什么？因为安全，默认不接受客户端发来的参数。通过page_size_query_param设置类属性来接收浏览器发来的参数的。

工具包下面创建类`utils.paginations.PageNumberPagination`，代码如下

```
from rest_framework import pagination

class PageNumberPagination(pagination.PageNumberPagination):
    page_size_query_param = 'size' # size=20
    page_size = 4 # 每页显示4个
    max_page_size = 8 # 对page_size进行限制
```

```
from .models import Employee
from .serializers import EmpSerializer
from rest_framework.generics import ListCreateAPIView,
RetrieveUpdateDestroyAPIView
from utils.paginations import PageNumberPagination

class EmpsView(ListCreateAPIView):
    """
    实现列表页get、新增post
    http://127.0.0.1:8000/emp/
    """
    queryset = Employee.objects.all()
    serializer_class = EmpSerializer
    pagination_class = PageNumberPagination # 指定自定义分页类，不使用全局

class EmpView(RetrieveUpdateDestroyAPIView):
    """
    实现详情页get、修改put、删除delete
    http://127.0.0.1:8000/emp/10021
    """
    queryset = Employee.objects.all()
    serializer_class = EmpSerializer
    #lookup_url_kwarg = 'id' # 路由配置中不是pk可以设置
```

测试http://127.0.0.1:8000/emp/、http://127.0.0.1:8000/emp/?size=2

CVS已经很方便了，已经简化到用列表页、详情页2个类配置一下就行了。但是，可以看到它们也有重复的地方，是否还可以简化？

这就需要使用视图集ViewSet。