

视图集

参考 <https://www.django-rest-framework.org/api-guide/viewsets/>

视图集类关系

类	父类	
ViewSet	ViewSetMixin APIView	不提供操作。一般不用
ViewSetMixin		重写了as_view, 使用actions字典建立method到函数的映射 method => handler
GenericViewSet	ViewSetMixin GenericAPIView	具有get_object, get_queryset, 但是没有提供增删改查方法
ModelViewSet	ListModelMixin CreateModelMixin RetrieveModelMixin UpdateModelMixin DestroyModelMixin GenericViewSet	在GenericViewSet和Mixins基础上, 提供了list()、create()、retrieve()、update()、destroy()、partial_update()
ReadOnlyModelViewSet	ListModelMixin RetrieveModelMixin GenericAPIView	list()、retrieve()

`ViewSetMixin.as_view(cls, actions=None, **kwargs)` 通过actions字典来解决method和handler映射, 例如 `ViewSet.as_view({'get': 'list', 'post': 'create'})`。2个get方法可以通过路径配置不一样解决。

ModelViewSet

- 5个Mixin提供了方法, 例如list、retrieve、create等
- ViewSetMixin解决了类二合一路由问题
 - 要指定action, 指明request method和handler映射关系
- GenericAPIView
 - 要指定queryset、serializer_class

视图集应用

视图层代码

注意, 一定要在路由中配置一下actions

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

```

3 # from rest_framework.generics import ListCreateAPIView,
  RetrieveUpdateDestroyAPIView
4 from rest_framework.viewsets import ModelViewSet
5
6 class EmpViewSet(ModelViewSet):
7     queryset = Employee.objects.all()
8     serializer_class = EmpSerializer
9     #lookup_url_kwarg = 'id' # 路由配置中不是pk可以设置
10
11 # class EmpsView(ListCreateAPIView):
12 #     """
13 #     实现列表页get、新增post
14 #     http://127.0.0.1:8000/emp/
15 #     """
16 #     queryset = Employee.objects.all()
17 #     serializer_class = EmpSerializer
18 #
19 # class EmpView(RetrieveUpdateDestroyAPIView):
20 #     """
21 #     实现详情页get、修改put、删除delete
22 #     http://127.0.0.1:8000/emp/10021
23 #     """
24 #     queryset = Employee.objects.all()
25 #     serializer_class = EmpSerializer
26 #     #lookup_url_kwarg = 'id' # 路由配置中不是pk可以设置

```

路由配置

employee/urls.py 指定列表页、详情页的action，建立到handler的映射

```

1 urlpatterns = [
2     path('', EmpViewSet.as_view({'get': 'list', 'post': 'create'})), # /emp/
3     path('<int:pk>/', EmpViewSet.as_view({
4         'get': 'retrieve', 'put': 'update', 'delete': 'destroy'
5     })), # /emp/10021/
6 ]

```

路由器

视图集的action配置很固定，能否也简化？

下面就以简单路由器类实现路由的简化。

SimpleRouter注册需要2个参数

- prefix: 前缀，可以为空串
- viewset: 提供视图集

最终返回路由配置的**列表**

employee/urls.py

```

1 from django.urls import path
2 from .views import EmpViewSet
3 from rest_framework.routers import SimpleRouter
4

```

```
5 router = SimpleRouter()
6 router.register('', EmpViewSet) # 在一级路由中配置过了前缀，这里省略
7
8 urlpatterns = router.urls
9 # urlpatterns = [
10 #     path('', EmpViewSet.as_view({'get': 'list', 'post': 'create'})), # /emp/
11 #     path('<int:pk>/', EmpViewSet.as_view({
12 #         'get': 'retrieve', 'put': 'update', 'delete': 'destroy'
13 #     })), # /emp/10021/
14 # ]
15
16 print('+ ' * 30)
17 print(urlpatterns)
18 print('+ ' * 30)
```

```
1 <URLPattern '^$' [name='employee-list']>
2 <URLPattern '^(?P<pk>[^/.]+)/$' [name='employee-detail']>
```

