# 1 使用Django原生返回序列化json

**1）配置url**

url(r'goods/$', GoodsListView.as\_view(), name="goods-list")

**2）写view**

class GoodsListView(View):

def get(self, request):

"""

商品列表页

:param request:

:return:

"""

json\_list = []

goods = Goods.objects.all()[:10]

# 使用遍历方法填充json\_dict

# for good in goods:

# json\_dict = {}

# json\_dict['name'] = good.name

# json\_dict['category'] = good.category.name

# json\_dict['market\_price'] = good.market\_price

# json\_list.append(json\_dict)

# 使用model\_to\_dict方法

# for good in goods:

# json\_dict = model\_to\_dict(good)

# json\_list.append(json\_dict)

json\_list = serializers.serialize('json', goods)

json\_list = json.loads(json\_list)

return JsonResponse(json\_list, safe=False)

# 2 django REST framework方式返回json

**1）使用APIView的方式**

class GoodsListView(APIView):

"""

List all goods.

"""

def get(self, request, format=None):

goods = Goods.objects.all()[:10] # QuerySet instance

goods\_serializer = GoodsSerializer(goods, many=True) # Json(Serializer) instance

return Response(goods\_serializer.data)

def post(self, request, format=None):

goods\_serializer = GoodsSerializer(data=request.data)

if goods\_serializer.is\_valid():

goods\_serializer.save() # 此处会调用GoodsSerializer中的create方法

return Response(goods\_serializer.data, status=status.HTTP\_201\_CREATED)

return Response(goods\_serializer.errors, status=status.HTTP\_400\_BAD\_REQUEST)

**2）继承GenericAPIView实现的方式**

class GoodsListView(mixins.ListModelMixin,

mixins.CreateModelMixin,

generics.GenericAPIView):

"""

商品列表页

"""

queryset = Goods.objects.all()[:10]

serializer\_class = GoodsSerializer

def get(self, request, \*args, \*\*kwargs):

return self.list(request, \*args, \*\*kwargs)

def post(self, request, \*args, \*\*kwargs):

return self.create(request, \*args, \*\*kwargs)

**3）继承ListAPIView实现的方式**

class GoodsListView(generics.ListAPIView):

"""

商品列表页

"""

# queryset = Goods.objects.all()[:10]

queryset = Goods.objects.all()

serializer\_class = GoodsSerializer

pagination\_class = GoodsPagination