**1. 背景**

最近在断断续续学习Python，并将Python投入至实际生产中。团队研发的模式是前后端分离，之前有尝试直接使用Django写API，没问题可以运行，但相比其他语言感觉还是蹩脚，所以尝试看看有没有更好的方式，今天就体验Django REST framework的一些感受分享一下。

**1.1. 前后端分离的好处：**

前端现在有越来越强大的框架，后端也有很多语言可以实现接口，将前后端分离有助于技术栈的选择和配合，也能更好的让技术人员发挥长板。

对于一个系统存在多个使用端的时候，比如微信、App、网站，前后端分离后大部分接口是可以直接复用的。

**2. Django REST framework 简介**

Django REST framework 是一套基于Django框架编写RESTful风格API的组件。

**2.1. Django REST framework 的特点：**

a) Serializers（序列化）：简单来说就是将复杂的数据结构变成JSON或XML格式，支持ORM和非ORM数据源的序列化。

Serializers allow complex data such as querysets and model instances to be converted to native Python datatypes that can then be easily rendered into JSON, XML or other content types. Serializers also provide deserialization, allowing parsed data to be converted back into complex types, after first validating the incoming data.

b) Authentication（身份认证）：快速实现支持Auth1、Auth2方式的权限验证。

c) Request and Response（请求与响应）：扩展了常规的HttpResquest和TemplateResponse。

d) ViewSet and Routers（视图集和路由）：类似与Java框架的Controllers  
……

**3. 小试牛刀**

**3.1. 配置环境**

Python 版本： 3.6

pip install django==2.0.6

pip install djangorestframework==3.8.2

**3.2. 创建项目**

django-admin.py startproject study20180721

cd study20180721

**3.3. 创建App**

python manage.py startapp product

**3.4. 在/study20180721/settings.py添加App**

INSTALLED\_APPS = (

...

'rest\_framework',

'product',

)

**3.5. 配置 /study20180721/urls.py**

from django.conf.urls import url, include

urlpatterns = [

url(r'^product/', include('product.urls')),

]

**3.6. 编写model层 /product/model.py**

from django.db import models

class Product(models.Model):

name = models.CharField(max\_length=50)

price = models.DecimalField(max\_digits=8,decimal\_places=2

def \_\_str\_\_(self):

return self.name

**3.7. 配置数据库**

python manage.py makemigratons

python manage.py migrate

**3.8. 编写序列化模块 /product/serializer.py**

from rest\_framework import serializers

from .models import Product

class ProductSerializer(serializers.ModelSerializer):

class Meta:

model = Product

fields = ("id","name","price")

**3.9. 编写视图 /product/views.py**

from rest\_framework import status

from rest\_framework.response import Response

from rest\_framework.views import APIView

from rest\_framework import generics

from .models import Product

from .serializers import ProductSerializer

# 使用APIView

class ProductView(APIView):

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

product = Product.objects.all()

serializer = ProductSerializer(product, many=True)

return Response(serializer.data)

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

serializer = ProductSerializer(data=request.data)

if serializer.is\_valid():

serializer.save()

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

else:

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

**3.10. 编写路由 /product/urls.py**

from django.conf.urls import url

from . import views

urlpatterns = [

url(r'^api/$', views.ProductView.as\_view(), name='product'),

]

**3.11. 测试**

curl http://localhost:8000/product/api/

curl -X POST http://localhost:8000/product/api/ -d "name=iPhone8&price=168.88"

curl http://localhost:8000/product/api/

**4. 参考资料**

**4.1. 官方网站**

<http://www.django-rest-framework.org/>  
<https://github.com/encode/django-rest-framework/>

**4.2. 参考文档**

Django Rest Framework--oauth实验笔记--参考官方文档  
<https://blog.csdn.net/yu359697395/article/details/78918840>