**\*\* << API 배포는 어떻게 하는가 >> \*\***

새로운 모듈 설치

**pip install djangorestframework**

앱 폴더에 serializers.py 생성

# 직렬화 serializer

from  rest\_framework import serializers

from .models import Item

class ItemSerializer(serializers.ModelSerializer):

    class Meta:

        model = Item

        fields = ("no", "name", "price", "regdate")

**fields**에 **model의 attibute 순서에 맞게** 나열해주어야 한다.

**models.py** > **obj**  > **serializers.py** > **json or xml**

from .serializers import ItemSerializer

from rest\_framework.renderers import JSONRenderer

import json

**- views.py**에 필요한 import

obj = Item.objects.get(no=no)

serializer = ItemSerializer(obj)

data = JSONRenderer().render(serializer.data)

return HttpResponse(data)

- 1개를 가져올 때

- **obj를 직렬화 > JSON 변경 > html에 전송**

obj = Item.objects.all()

serializer = ItemSerializer(obj, many=True)

data = JSONRenderer().render(serializer.data)

- 여러 개를 가져올 때

- **many 옵션만 추가! >** **1 or many일 경우 many로 설정 잡아주어야한다.**

===============================================

**\*\* << Serialize를 하는 이유 >> \*\***

디비에서 가져올 때는 obj로 가져온다 ex) A, 홍길동, 3

**외부 디바이스 ( 스마트폰, 크롤링 ... ) ( 내부의 templates를 제외한 나머지 )에서**

**정보를 받을 때는?**

> 그냥 obj 형태가 아니라 그냥 string처럼 받아들이게 된다.

> 하나가 없어져도 문제가 있는지 모른다.

> **검증**이 필요하다!

그래서 xml, json을 쓴다.

xml 태그형태

<xml>

<id>”A”</id>

<name>”홍길동”</name>

<age>3</age>

</xml>

json 형태

{“id”:”A”, “name”:”홍길동, “age”:3}

===============================================

**\*\* << CORS error( 크로스 도메인 에러) >> \*\***

- **외부에서 데이터를 볼 수 없는 상태**

- 추가 라이브러리 필요 >>**새로운 모듈** 설치

**pip install django-cors-headers**

– settings.py에 추가 설정이 필요하다.

INSTALLED\_APPS = [

    'corsheaders', ... ]

MIDDLEWARE = [

    'corsheaders.middleware.CorsMiddleware', ... ]

CORS\_ORIGIN\_ALLOW\_ALL = True

마지막 줄은 들여쓰기 없이 밖에서

===============================================

**\*\* << 최종 views.py의 코드 >> \*\***

- **basic**

def select1(request):

    if request.method == "GET":

        key = request.GET.get("key", "")

        no = request.GET.get("no", 1)

        if key == "abc":

            obj = Item.objects.get(no=no)

            serializer = ItemSerializer(obj)

            data = JSONRenderer().render(serializer.data)

            return HttpResponse(data)

        else:

            data = json.dumps({"ret":"key error"})

            return HttpResponse(data)

- **many**

def select2(request):

    if request.method == "GET":

        key = request.GET.get("key", "")

        data = json.dumps({"ret":"key error"})

        search = 1

        if key == "all":

            obj = Item.objects.all()

            serializer = ItemSerializer(obj, many=True)

            data = JSONRenderer().render(serializer.data)

        return HttpResponse(data)

- **search 추가** > name에 search가 있는 것들을 num개만큼 출력

def select3(request):

    if request.method == "GET":

        key = request.GET.get("key", "")

        num = int(request.GET.get("num", 1))

        search = request.GET.get("search", "")

        data = json.dumps({"ret":"key error"})

        if key == "abc":

            obj = Item.objects.filter(name\_\_contains=str(search))[0:num]

            serializer = ItemSerializer(obj, many=True)

            data = JSONRenderer().render(serializer.data)

        return HttpResponse(data)

끝