Step 1: Load fields

# 1. Delete in class members

# 2. Aggregate to BBSerializer.\_declared\_fields = {"yy":object}

class BBSerializer(serializers.Serializer):

yy = serializers.CharField(source='name')

name = 123

# 1. Delete in class members

# 2. Aggregate to BBSerializer.\_declared\_fields = {"xx": object, "yy": object}

class UserSerializer(serializers.ModelSerializer,BBSerializer):

xx = serializers.CharField(source='name')

class Meta:

model = models.UserInfo

fields = ["name", "age","xx"]

Step 2: Serialization

queryset = models.UserInfo.objects.all()

ser = UserSerializer(instance=queryset, many=True) # ListSerializer object (UserSerializer object)

db->queryset = [{id:xxx,name:xx,age},]-> Loop each object in the queryset and call UserSerializer to instantiate it.

instance = models.UserInfo.objects.all().first()

ser = UserSerializer(instance=instance, many=False) # UserSerializer object

db->instance = {id:xxx,name:xx,age} => UserSerializer

=====> Object->UserSerializer->Serialization,

Serialization process:

db\_instance = models.UserInfo.objects.all().first() instance.title order count models.IntegerField(verbose\_name=" order")

ser = UserSerializer(instance=db\_instance,many=False) titte order count serializers.CharField(source='name')

ser.data #trigger

-Search for corresponding relationships internally

-Serialize one by one