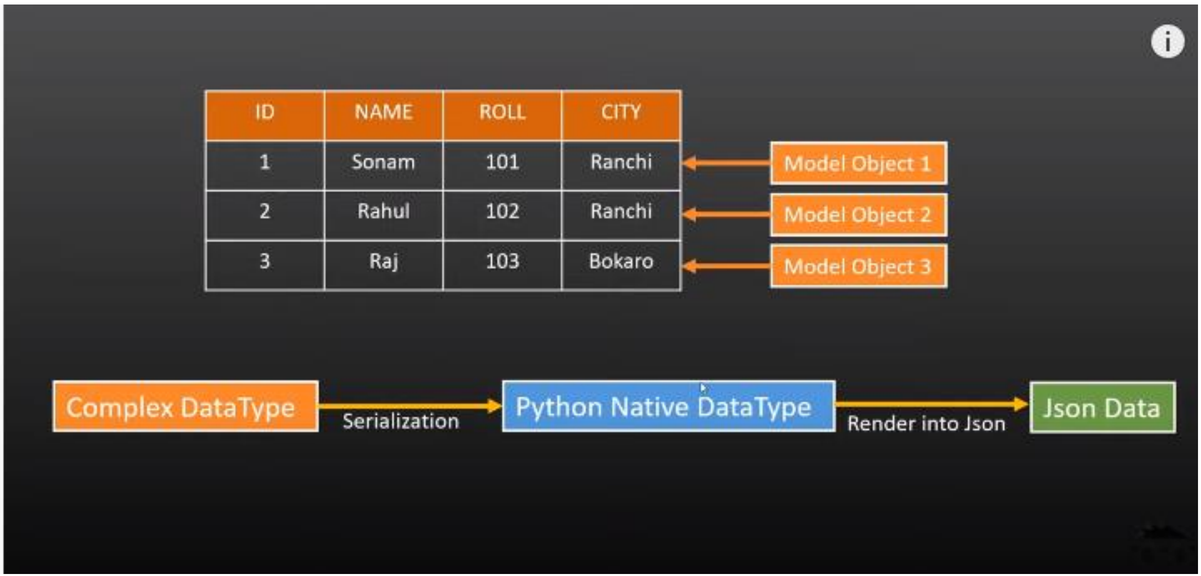


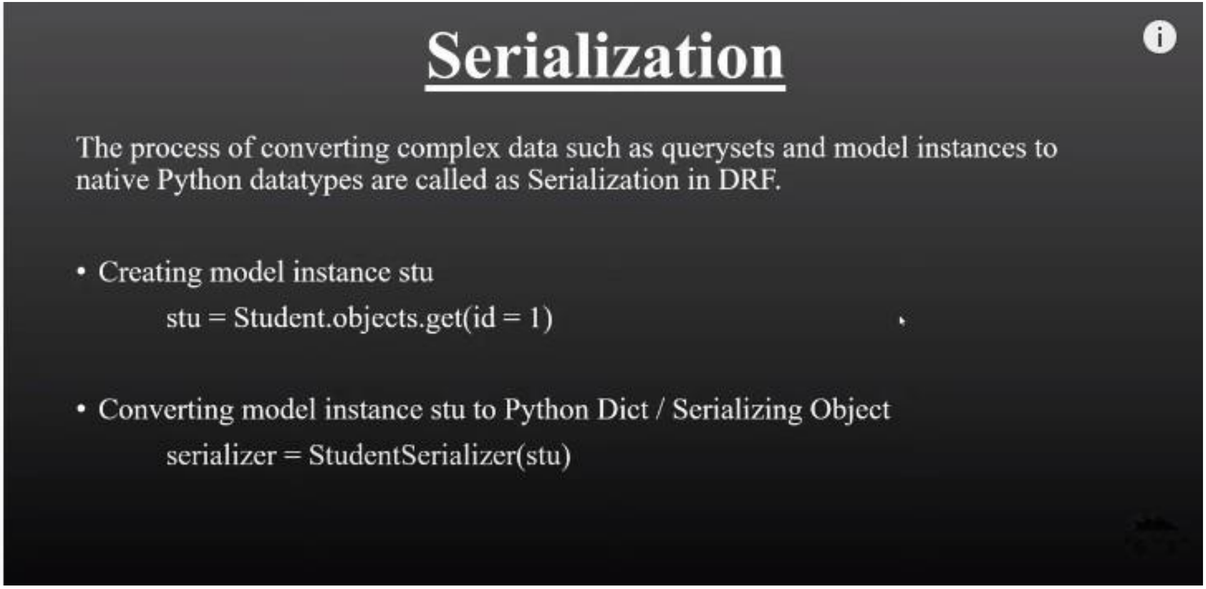
ID	NAME	ROLL	CITY
1	Sonam	101	Ranchi
2	Rahul	102	Ranchi
3	Raj	103	Bokaro

Model Object 1

Model Object 2

Model Object 3

Complex DataType → Serialization → Python Native DataType → Render into Json → Json Data



Serialization

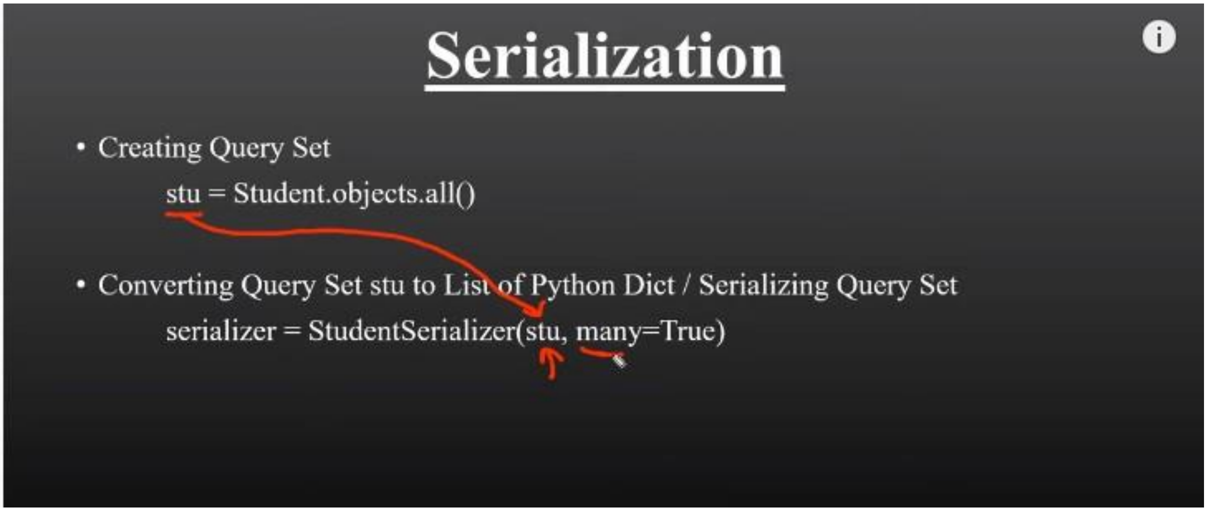
The process of converting complex data such as querysets and model instances to native Python datatypes are called as Serialization in DRF.

- Creating model instance stu

```
stu = Student.objects.get(id = 1)
```

- Converting model instance stu to Python Dict / Serializing Object

```
serializer = StudentSerializer(stu)
```



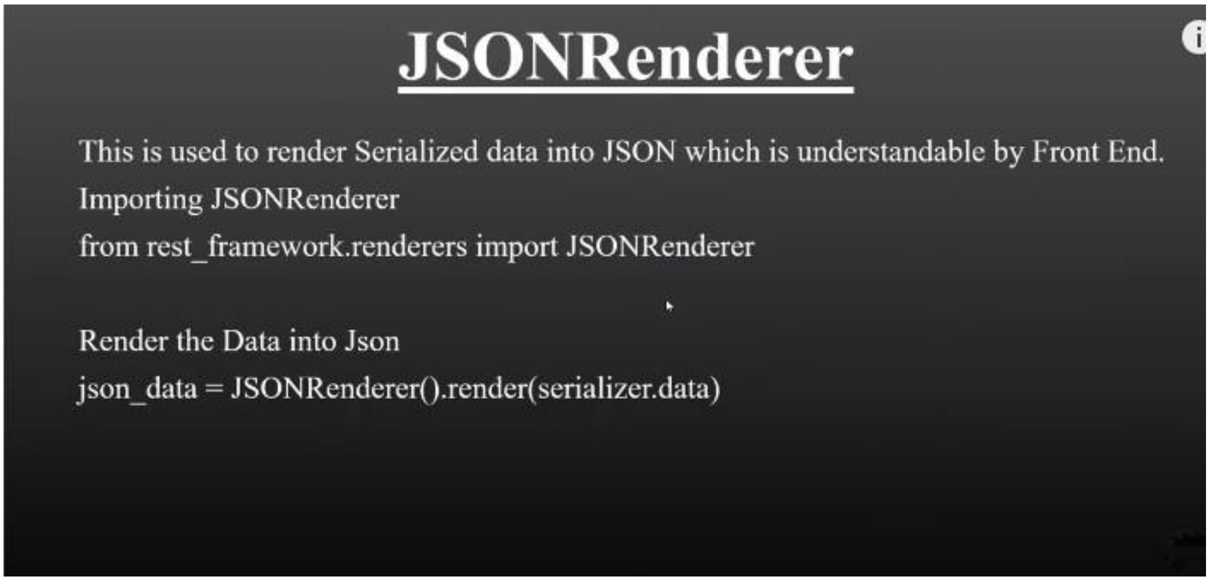
Serialization

- Creating Query Set

```
stu = Student.objects.all()
```

- Converting Query Set stu to List of Python Dict / Serializing Query Set

```
serializer = StudentSerializer(stu, many=True)
```



JSONRenderer

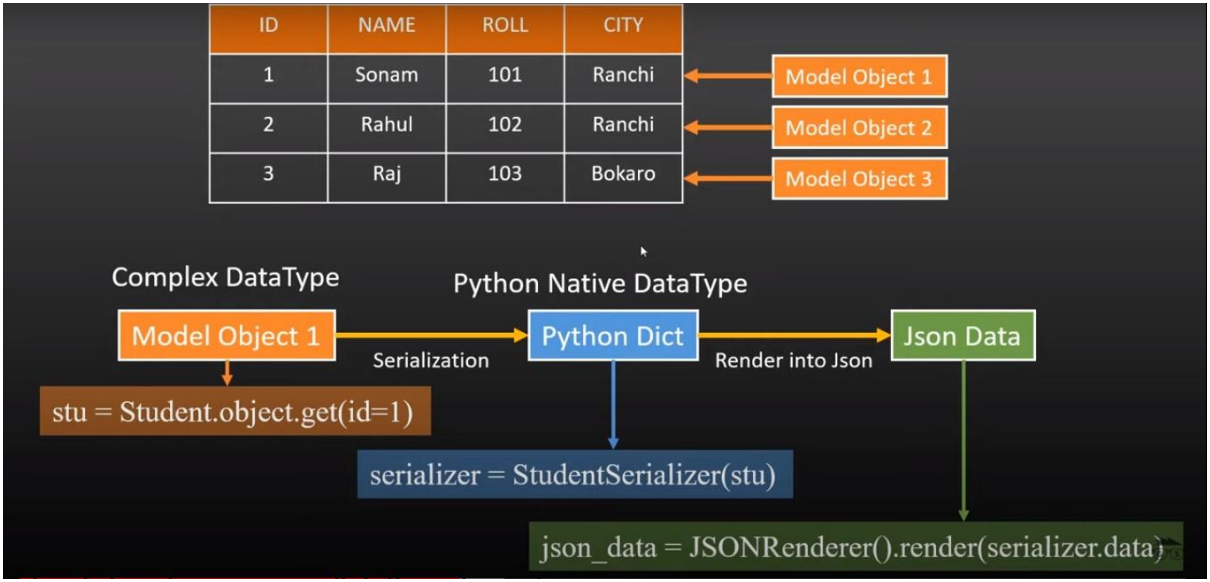
This is used to render Serialized data into JSON which is understandable by Front End.

Importing JSONRenderer

from rest_framework.renderers import JSONRenderer

Render the Data into Json

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

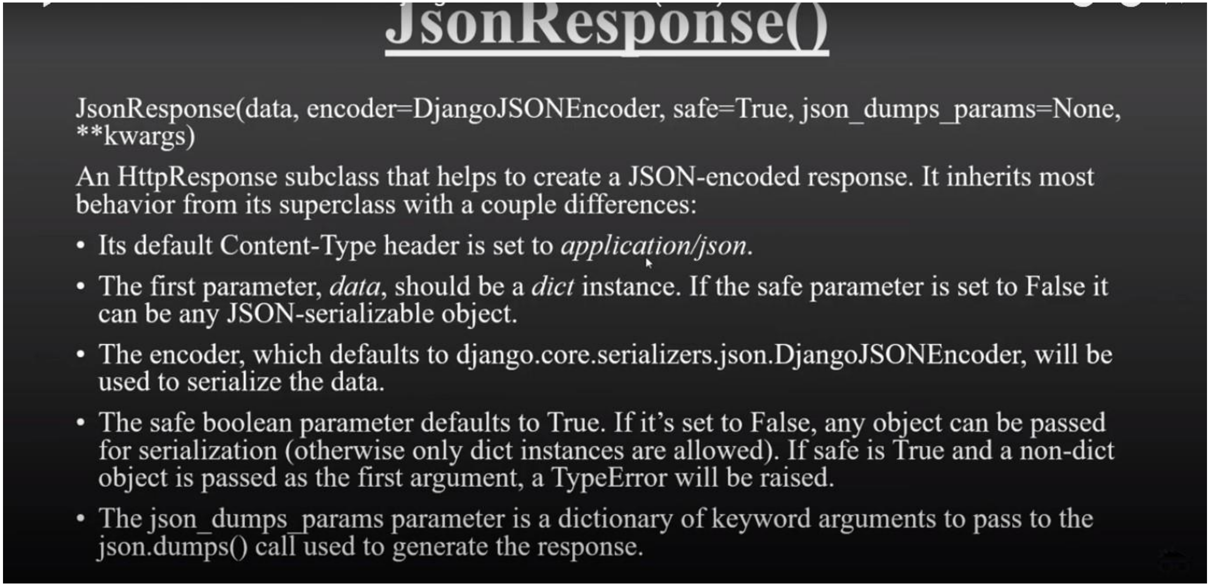


ID	NAME	ROLL	CITY
1	Sonam	101	Ranchi
2	Rahul	102	Ranchi
3	Raj	103	Bokaro

stu = Student.object.get(id=1)

serializer = StudentSerializer(stu)

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



JsonResponse()

JsonResponse(data, encoder=DjangoJSONEncoder, safe=True, json_dumps_params=None, **kwargs)

An HttpResponse subclass that helps to create a JSON-encoded response. It inherits most behavior from its superclass with a couple differences:

- Its default Content-Type header is set to *application/json*.
- The first parameter, *data*, should be a *dict* instance. If the safe parameter is set to False it can be any JSON-serializable object.
- The encoder, which defaults to django.core.serializers.json.DjangoJSONEncoder, will be used to serialize the data.
- The safe boolean parameter defaults to True. If it's set to False, any object can be passed for serialization (otherwise only dict instances are allowed). If safe is True and a non-dict object is passed as the first argument, a TypeError will be raised.
- The json_dumps_params parameter is a dictionary of keyword arguments to pass to the json.dumps() call used to generate the response.