

Module Serializer in DRF

What is Serializer?

Manual way to define serialization logic. You define each field explicitly. Use when you're not using Django models or want full control.

```
from rest_framework import serializers

class BookSerializer(serializers.Serializer):
    title = serializers.CharField(max_length=100)
    author = serializers.CharField(max_length=100)
    published = serializers.DateField()
```

What is ModelSerializer?

A shortcut for creating serializers using Django models Automatically generates fields based on model. Less boilerplate code.

```
from rest_framework import serializers
from .models import Book

class BookSerializer(serializers.ModelSerializer):
    class Meta:
        model = Book
        fields = '__all__'
```

Diff b/w ModelSerializer and Serializer

Feature	Serializer	ModelSerializer
Field Definition	Manual	Auto from model
Boilerplate	More	Less
Model Integration	Not tied to model	Tightly integrated with model
Flexibility	More control	Less control, but easy to customize

What is Meta class?

Nested class used in ModelSerializer.

Tells serializer:

- Which model to use
- Which fields to include/exclude

```
class Meta:
    model = Book          # your Django model
    fields = '__all__'    # or list of field names
```

How to include fields in ModelSerializer?

```
class BookSerializer(serializers.ModelSerializer):
    class Meta:
        model = Book
        fields = ['id', 'title', 'author']
```

How to exclude fields?

```
class BookSerializer(serializers.ModelSerializer):
    class Meta:
        model = Book
        exclude = ['created_at', 'updated_at']
```

Types of validation in Django