# TRAINING DAY24 REPORT

21 JULY 2025

## Types of Serializers in DRF:

There are **three main types** of serializers provided by Django REST Framework:

| Type | Description | When to Use |
|---|---|---|
| **1. Serializer class** | Manually define fields and behavior | When you need full control over the data or are not using Django models |
| **2. ModelSerializer** | Automatically generates fields from Django models | Most commonly used (quick and efficient) |
| **3. HyperlinkedModelSerializer** | Same as ModelSerializer, but uses hyperlinks instead of primary keys | When you want RESTful, hyperlink-based representations |

## ModelSerializer:

The **most commonly used serializer** in DRF.

It automatically creates fields based on a Django model and also provides create() and update() methods by default.

**Example:**

```
electronics_shop > accounts > 🐍 admin.py > ...
1    from rest_framework import serializers
2    from .models import Task
3
4    class TaskModelSerializer(serializers.ModelSerializer):
5        class Meta:
6            model = Task
7            fields = '_all_'
8
```

**Advantages:**

- Less code, auto-generates everything from model.

- Automatically validates data types.

- Easy to use for CRUD APIs.

**Example JSON Output:**

```
{
  "id": 1,
  "title": "Complete DRF notes",
  "description": "Finish writing the serializers topic",
  "completed": false,
  "created_at": "2025-10-14T10:00:00Z"
}
```

**Customization Example:**

You can specify fields manually if you don't want all:

```
electronics_shop > accounts > 🐍 admin.py > ...
  1    from rest_framework import serializers
  2    from .models import Task
  3
  4    class TaskModelSerializer(serializers.ModelSerializer):
  5        class Meta:
  6            model = Task
  7            fields = ['title', 'completed']
  8
```

**Difference Between Forms and Serializers:**

| Feature | Django Forms | DRF Serializers |
|---|---|---|
| Purpose | For HTML form input | For JSON-based APIs |
| Output | HTML | JSON |
| Validation | Client-side or server-side | Server-side validation |
| Data Storage | Model instances | Model instances or plain data |
| Example | ModelForm | ModelSerializer |

So basically, **Serializers are like Django Forms for APIs**.