

TRAINING DAY22 REPORT

18 JULY 2025

What is Django REST Framework?

Django REST Framework (DRF) is a **powerful and flexible toolkit** built on top of **Django** that helps developers create **Web APIs** easily.

In simple words - Django is used to build **websites**, while DRF is used to build **APIs** that allow your frontend (React, Android, etc.) to communicate with your backend.

A **REST API (Representational State Transfer)** is a way for two systems to communicate over the internet using standard HTTP methods (GET, POST, PUT, DELETE).

Why Use DRF?

While Django is great for rendering HTML pages, modern applications (like mobile apps or React frontends) need **data in JSON format**.

DRF provides tools to easily build and manage such APIs.

Key Advantages:

1. **Easy Serialization:** Converts Django models → JSON (and vice versa).
2. **Authentication & Permissions:** Built-in user authentication, token-based login, etc.
3. **Browsable API:** Auto-generated web interface to test APIs directly in browser.

4. **Viewsets & Routers:** Automatically create routes for CRUD operations.
5. **Validation:** Built-in and custom data validation.
6. **Pagination & Filtering:** Easily manage large datasets with page-wise results.
7. **Flexibility:** Works with Django ORM and even non-relational data sources.

Understanding RESTful API Concepts:

Before understanding DRF, let's recall what **REST (Representational State Transfer)** means.

A REST API follows certain principles:

HTTP Method	Purpose	Example
GET	Retrieve data	/api/tasks/
POST	Create new data	/api/tasks/
PUT / PATCH	Update existing data	/api/tasks/1/
DELETE	Delete data	/api/tasks/1/

DRF provides a simple way to handle all these operations.

Basic Components of DRF:

DRF has five main building blocks:

Component	Description	Example
Serializer	Converts model data ↔ JSON	TaskSerializer
APIView / ViewSet	Controls logic for API endpoints	TaskViewSet
Router / URLconf	Maps URLs to views automatically	/api/tasks/
Permissions	Controls who can access the API	IsAuthenticated
Authentication	Validates users	Token / Session Auth

Advantages of DRF over Normal Django:

Feature	Django	Django REST Framework
API support	Not built-in	Yes
JSON Response	Manual handling	Built-in Serializers
Auth for APIs	No	Token, JWT, Session
Browsable API	No	Yes
Frontend connection	Harder	Very Easy (for React, Flutter, etc.)