

# TRAINING DAY23 REPORT

19 JULY 2025

## Setting up Django REST Framework

### Step 1: Install Django and DRF:

Django is already installed so now install Django REST Frameworks:

```
bash

pip install djangorestframework
```

### Step 2: Add DRF to INSTALLED\_APPS:

In your **project/settings.py**, add:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    # 3rd party
    'rest_framework',

    # your app
    'diary',
]
```

### Step 3: Basic REST\_FRAMEWORK Configuration:

Add this in settings.py :

```
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': [
        'rest_framework.authentication.SessionAuthentication',
        'rest_framework.authentication.BasicAuthentication',
    ],
    'DEFAULT_PERMISSION_CLASSES': [
        'rest_framework.permissions.AllowAny',
    ]
}
```

This means:

- You can use **login sessions** or **basic auth**.
- Anyone can access the API for now (no restriction).

## What Are Serializers in Django REST Framework?

A **Serializer** is a component that **converts complex data** (like Django model objects or querysets) into **Python datatypes** (and then JSON) — so it can be easily sent through an **API response**.

It also does the reverse — **takes JSON input** and converts it into a **Django model instance** (used for saving data in the database).

Serializer = Bridge Between Django and JSON

Think of a serializer as a **translator** between your database and the external world.

Action	Serializer's Role	Example
Serialize (Output)	Model → JSON	Show diary entries in API
Deserialize (Input)	JSON → Model	Save user's new diary entry

### Why Serializers Are Needed?

Without DRF, you'd need to manually convert data to JSON like this:

```
from django.http import JsonResponse
data = {'title': 'My Day', 'content': 'It was awesome!'}
return JsonResponse(data)
```

With DRF, **serializers** do this automatically — and they also handle:

- Validation (checking input data)
- Saving data
- Updating existing records

### Serializer Fields

Common field types you can use:

Field	Description
CharField()	Text data
IntegerField()	Whole numbers
BooleanField()	True/False
DateTimeField()	Date & time
EmailField()	Validated email
URLField()	URL links
ChoiceField()	Predefined options
SlugField()	For slugs like my-first-post
PrimaryKeyRelatedField()	Reference to other models