Rest Framework

**Introduction to APIs**

1. What is an API (Application Programming Interface)?

* An **API (Application Programming Interface)** is a set of rules and protocols that allows different software applications to communicate with each other.

1. Types of APIs: REST, SOAP.

* APIs mainly come in two common types: **REST** and **SOAP**. Both enable communication between systems, but they follow different rules and formats.

REST is a modern, lightweight, and widely-used API architecture that uses HTTP for communication.

SOAP is a protocol-based API that is more rigid and standardized. It is XML-based and often used in enterprise-level applications.

1. Why are APIs important in web development?

* APIs are **crucial** in web development because they enable seamless communication between different software components, services, and platforms. Here's why they're so important:

Connect Frontend and Backend

Enable Third-Party Integrations

**Promote Reusability and Scalability**

Speed Up Development

Enhance Security

Microservices Architecture

1. Understanding project requirements

* Understanding project requirements is the **first and most critical step** in successful software or web development. It ensures that the final product meets the client's expectations and functions correctly.

1. Setting up the environment and installing necessary packages

* Before starting a web development project, it's important to set up a clean development environment and install all the necessary tools and packages. Here's a step-by-step guide using **Python (Django)** as an example.

1. What is Serialization?

* **Serialization** is the process of converting complex data types like **Python objects** (e.g., querysets, models) into **native data formats** such as **JSON**, **XML**, or **plain text** — so they can be easily sent over the network or stored.

1. Converting Django QuerySets to JSON

* you often need to convert a **QuerySet** (which is a collection of database objects) into **JSON format** — especially when building APIs or sending data to frontend apps like React or Vue.

1. Using serializers in Django REST Framework (DRF).

* **Serializers** in Django REST Framework (DRF) are used to convert complex data types like Django models or QuerySets into native Python data types (which can be easily rendered as JSON, XML, etc.) — and vice versa.

1. HTTP request methods (GET, POST, PUT, DELETE)

* The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP works as a request-response protocol between a client and server.

HTTP Methods

GET,POST,PUT,HEAD,DELETE,PATCH,OPTIONS,CONNECT,TRACE

1. Sending and receiving responsesin DRF.

* Handling requests and responses is a fundamental part of creating APIs with Django Rest Framework (DRF). This tutorial will guide you through how to effectively handle incoming requests and send appropriate responses using DRF.

Handling Requests

Accessing Request Data

DRF provides several ways to access data from the request object. Commonly used attributes include request.data, request.query\_params, and request.FILES.

request.data: Contains parsed content of the request body.

request.query\_params: Contains query parameters in the URL.

request.FILES: Contains uploaded files.

1. Understanding views in DRF: Function-based views vs Class-based views.

* Django Rest Framework (DRF) provides two main ways to define views: Function-based views (FBVs) and Class-based views (CBVs). Each approach has its own advantages and use cases.

Prerequisites

Before you start, ensure you have a DRF project set up. If not, please refer to our previous tutorial on setting up a DRF project.

Function-based Views (FBVs)

Function-based views are simple functions that take a web request and return a web response. They provide a straightforward way to define views in Django and DRF.

Class-based Views (CBVs)

Class-based views use Python classes to encapsulate the view logic. They provide a more structured and reusable way to define views in Django and DRF.

1. Defining URLs and linking them to views.

* Define URLs to route incoming HTTP requests to the appropriate views. This step connects your API endpoints to the logic that handles requests

1. Adding pagination to APIs to handle large data sets.

* **Pagination** in APIs refers to the process of breaking up large sets of data into smaller, manageable "pages" or chunks. Instead of returning **all records at once**, which can slow down performance and overload clients, pagination sends data in segments based on a specified size.

1. Configuring Django settings for database,static files, and API keys.

* Properly configuring Django settings is essential for building scalable, secure, and maintainable web applications. Here's what each configuration area involves:

1. Database Configuration

Django needs to connect to a database to store models and application data.

2. Static Files Configuration

Static files (CSS, JS, images) are used in frontend templates. Django needs to know where to find and serve them.

3. API Key Configuration

API keys (like for third-party services: Stripe, SendGrid, Google Maps) must be stored securely, not hardcoded.

1. Setting up a Django REST Framework project.

* **Setting up a Django REST Framework (DRF) project** means configuring Django to build a **RESTful API** backend that can handle HTTP requests and return data in formats like JSON. DRF extends Django to easily create APIs for web and mobile apps.

1. Implementing social authentication (e.g., Google, Facebook) in Django.

* Implementing social authentication in Django means enabling users to log in or register using their existing accounts from platforms like Google, Facebook, GitHub, etc., instead of creating a new username and password.

This is commonly done using third-party packages like:

social-auth-app-django

django-allauth

1. Sending emails and OTPs using third-party APIslike Twilio, SendGrid.

* Sending emails and OTPs (One-Time Passwords) using third-party APIs means using external services like SendGrid (for emails) and Twilio (for SMS) to reliably deliver messages and authentication codes to users.

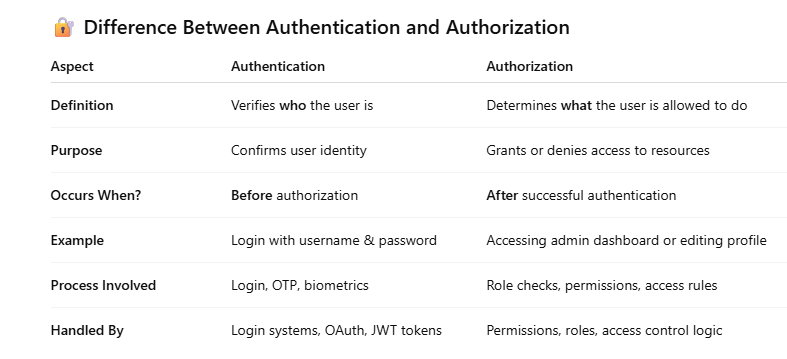
Email delivery: Send account activation, password reset, welcome emails, etc.

OTP delivery: Send verification codes for two-factor authentication (2FA) or user sign-up/login validation.

1. What is CRUD, and why is it fundamental to backend development?

* CRUD is the backbone of backend systems. It allows applications to perform essential operations on data and ensures seamless interaction between users, servers, and databases

1. Difference between authentication and authorization.



1. Implementing authentication using Django REST Framework’s token-based system.

* Token-based authentication in DRF is a method where each authenticated user is given a unique token (a string of characters) that must be included in the header of HTTP requests to access protected resources. This eliminates the need to repeatedly send the user's username and password.

1. Introduction to OpenWeatherMap API and how to retrieve weather data.

* OpenWeatherMap API is a cloud-based service that provides real-time and forecast weather data for any location in the world. It allows developers to access weather information such as temperature, humidity, wind speed, air pressure, and more using simple HTTP requests.

1. Using Google Maps Geocoding API to convert addresses into coordinates

* The Google Maps Geocoding API is a web service that allows developers to convert addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (latitude and longitude), and vice versa (called reverse geocoding).

1. Introduction to GitHub API and how to interact with repositories, pull requests, and issues.

* The GitHub API is a RESTful web service provided by GitHub that allows developers to programmatically access and manage GitHub resources such as repositories, pull requests, issues, commits, branches, users, and more.

1. Using Twitter API to fetch and post tweets, and retrieve user data.

* The Twitter API is a set of RESTful web services provided by Twitter that allows developers to programmatically access, manage, and interact with Twitter data, including tweets, users, timelines, and more. You can fetch tweets, post new tweets, and retrieve information about Twitter users using the API.

1. Introduction to REST Countries API and how to retrieve country-specific data.

* The **REST Countries API** is a free, public API that provides detailed information about countries around the world in a **RESTful format**. You can use it to **retrieve country-specific data** like name, capital, population, currencies, flags, region, borders, languages, and more.

1. Using email sending APIs like SendGrid and Mailchimp to send transactional emails.

* SendGrid and Mailchimp Transactional (formerly Mandrill) are cloud-based email delivery services that provide APIs to send transactional emails—like password resets, order confirmations, OTPs, and notifications—from web or mobile applications.

Transactional emails are automated, real-time emails sent to individuals based on specific actions or events.

1. Introduction to Twilio API forsending SMS and OTPs.

* Twilio API is a cloud communications platform that allows developers to send and receive SMS, voice calls, WhatsApp messages, and more. It is widely used for sending SMS alerts and One-Time Passwords (OTPs) in applications for authentication, notifications, and user verification.

1. Introduction to integrating payment gateways like PayPal and Stripe.

* Payment gateways like PayPal and Stripe are services that allow websites and applications to securely process online payments using credit/debit cards, digital wallets, or direct bank transfers. Integration enables businesses to accept payments through their platforms.

1. Using Google Maps API to display maps and calculate distances between locations.

* The Google Maps API allows developers to embed interactive maps, calculate distances, and perform geolocation-related tasks in their applications. Using the Google Maps JavaScript API, Distance Matrix API, or Geocoding API, you can display maps and calculate distances between two or more locations (e.g., driving, walking, or cycling routes).