

REST Framework and API Integration – Theory Questions and Answers

1. Introduction to APIs

An API (Application Programming Interface) is a set of rules and protocols that allows different software applications to communicate with each other. APIs act as intermediaries between clients and servers.

Types of APIs:

REST APIs use HTTP methods and are lightweight, scalable, and widely used in web and mobile applications.

SOAP APIs are protocol-based, rely on XML, and are more rigid and secure.

Importance of APIs:

APIs enable integration, scalability, reusability, and interoperability in modern web development. They allow frontend and backend systems to work independently.

2. Requirements for Web Development Projects

Project requirements define what the system should do. This includes functional and non-functional requirements.

Environment Setup:

Includes installing Python, Django, virtual environments, databases, and third-party libraries.

Dependency Management:

requirements.txt ensures consistency across development and deployment environments.

3. Serialization in Django REST Framework

Serialization is the process of converting complex data types like Django QuerySets into JSON or XML formats.

Purpose:

Allows data transfer between server and client.

DRF Serializers:

They validate, serialize, and deserialize data efficiently, acting as a bridge between models and APIs.

4. Requests and Responses in DRF

HTTP requests define how clients interact with servers.

Common Methods:

GET retrieves data, POST creates data, PUT updates data, DELETE removes data.

Responses:

DRF sends structured responses with status codes, headers, and serialized data.

5. Views in Django REST Framework

Views handle request logic.

Function-Based Views:

Simple and easy to understand.

Class-Based Views:

Reusable, scalable, and organized, ideal for CRUD operations.

6. URL Routing in DRF

URL routing maps API endpoints to views.

Importance:

Provides structured navigation and RESTful endpoints.

Dynamic URLs:

Allow resource-based access using identifiers.

7. Pagination in DRF

Pagination divides large datasets into smaller chunks.

Benefits:

Improves performance and user experience.

DRF Pagination:

Supports page-based, limit-offset, and cursor pagination.

8. Settings Configuration in Django

Django settings control project behavior.

Configurations:

Database, static files, security, and API keys.

Best Practices:

Use environment variables for sensitive data.

9. Project Setup in DRF

Setting up a DRF project involves configuring Django, creating apps, and enabling REST Framework.

Benefits:

Structured API development and modular design.

10. Social Authentication, Email, and OTP APIs

Social Authentication:

Uses OAuth2 to authenticate users via third-party providers.

Email and OTP APIs:

Enable notifications, verification, and secure user onboarding.

11. RESTful API Design

REST principles include statelessness, resource-oriented URLs, and standardized HTTP methods.

Benefits:

Scalability, simplicity, and flexibility.

12. CRUD API

CRUD stands for Create, Read, Update, Delete.

Importance:

Foundation of backend systems enabling data manipulation.

13. Authentication and Authorization API

Authentication verifies user identity, while authorization controls access.

Token-Based Authentication:

Provides secure, stateless access control.

14. OpenWeatherMap API Integration

Weather APIs provide real-time and forecast data.

Use Cases:

Location-based services and analytics dashboards.

15. Google Maps Geocoding API

Geocoding converts addresses into coordinates.

Applications:

Location tracking, mapping, and navigation.

16. GitHub API Integration

GitHub API enables automation and interaction with repositories.

Use Cases:

CI/CD, analytics, and developer tools.

17. Twitter API Integration

Twitter API allows interaction with tweets and user data.

Applications:

Social media analytics and content management.

18. REST Countries API Integration

REST Countries API provides country-specific data.

Use Cases:

Educational apps, dashboards, and travel services.

19. Email Sending APIs

Email APIs enable transactional and marketing emails.

Advantages:

Reliability, scalability, and analytics.

20. SMS Sending APIs (Twilio)

SMS APIs allow sending messages and OTPs.

Applications:

Authentication, alerts, and notifications.

21. Payment Integration (PayPal, Stripe)

Payment gateways enable secure online transactions.

Features:

Encryption, fraud detection, and global payments.

22. Google Maps API Integration

Google Maps API provides mapping, distance calculation, and visualization tools.

Use Cases:

Location-based services and logistics.