REST APIs

1.Could you explain how REST APIs typically operate?

- REST APIs work by exposing resources (data entities) over HTTP. Clientsaccess these resources through URL endpoints and perform actions using HTTP methods like GET, POST, PUT, and DELETE.

2.Would you mind sharing what the main HTTP methods signify in a RESTful context**?**

- **GET** retrieves data,
- **POST** creates new data,
- **PUT** updates data,
- **DELETE** removes data.

3.How would you describe a REST resource in simple terms?

- In REST, a resource is any object or data entity accessible via a unique URL, representing things like users, products, or orders.

4.What is CORS, and why is it important in REST APIs?

- CORS (Cross-Origin Resource Sharing) is a security feature that controls resource sharing across different origins. It's essential for secure API access, especially in web applications.

5.How do you secure a REST API?

- **Authentication** (e.g., OAuth, API keys, JWT).
- **HTTPS** for encrypted communication.
- **Rate limiting** to prevent abuse.
- **CORS policies** for cross-origin security.

6.What is the difference between REST and SOAP?

- REST is an architectural style using HTTP methods, and is more flexible with data formats (commonly JSON). SOAP (Simple Object Access Protocol) is a protocol with strict standards and relies on XML, commonly used for enterprise applications.

7.Could you explain the difference between PUT and PATCH operations?

- Absolutely! **PUT** replaces an entire resource, while **PATCH** modifies only the specified fields.

8.How do you handle errors in REST APIs?

- Errors are usually handled with HTTP status codes and a response body with details on the error. A standardized error format (like an "error code" and "message") helps clients understand and address issues.

9.What is the purpose of using JSON in REST APIs?

- JSON (JavaScript Object Notation) is commonly used in REST APIs for its simplicity, readability, and compatibility across different programming languages.

10.What is HATEOAS?

- HATEOAS (Hypermedia as the Engine of Application State) is a REST constraint where clients interact with the application through hypermedia links provided dynamically by the server.

11.What is a RESTful API?

- A RESTful API is an API that adheres to REST principles, allowing for stateless, scalable interactions and resource-oriented operations using standard HTTP methods.

12.What is an endpoint in an API?

- An endpoint is a specific URL where an API resource is made available to clients. It represents an accessible location where interactions occur.

13.What is API throttling, and why is it used?

- Throttling limits the number of API requests from a client within a specified time to prevent abuse, improve performance, and control resource usage.

14.Could you elaborate on HATEOAS and its role in REST?

- Certainly! HATEOAS is a REST principle where the server provides links in responses to guide clients on available actions, enhancing discoverability and allowing more dynamic client-server interactions.

15.What is an Idempotent operation?

- An operation is idempotent if multiple identical requests have the same effect as a single request. **GET**, **PUT**, and **DELETE** are idempotent, but **POST** is not.

16.What are the features of RESTful Web Services?

- The service is based on the Client-Server model.
- The service uses HTTP Protocol for fetching data/resources, query execution, or any other functions.
- The medium of communication between the client and server is called “Messaging”.
- Resources are accessible to the service by means of URIs.
- It follows the statelessness concept where the client request and response are not dependent on others and thereby provides total assurance of getting the required data.
- These services also use the concept of caching to minimize the server calls for the same type of repeated requests.
- These services can also use SOAP services as implementation protocol to REST architectural pattern.

17.What is a REST Resource?

Every content in the REST architecture is considered a resource. The resource is analogous to the object in the object-oriented programming world. They can either be represented as text files, HTML pages, images, or any other dynamic data.

- The REST Server provides access to these resources whereas the REST client consumes (accesses and modifies) these resources. Every resource is identified globally by means of a URI.

18.Can you tell what constitutes the core components of HTTP Request?

- Method/Verb − This part tells what methods the request operation represents. Methods like GET, PUT, POST, DELETE, etc are some examples.
- URI − This part is used for uniquely identifying the resources on the server.
- HTTP Version − This part indicates what version of HTTP protocol you are using. An example can be HTTP v1.1.
- Request Header − This part has the details of the request metadata such as client type, the content format supported, message format, cache settings, etc.
- Request Body − This part represents the actual message content to be sent to the server.

19.What makes REST services to be easily scalable?

- REST services follow the concept of statelessness which essentially means no storing of any data across the requests on the server. This makes it easier to scale horizontally because the servers need not communicate much with each other while serving requests.

20.Is it possible to send payload in the GET and DELETE methods?

- Payload refers to the data passes in the request body. It is not the same as the request parameters. The payload can be sent only in POST methods as part of the request body.