API-Related Interview Questions

1. What is an API, and why is it used?

   - An API (Application Programming Interface) is a set of protocols and tools that allows different software applications to communicate with each other. It's used to enable the integration of different systems, allowing one software component to interact with others.

2. What are the different types of APIs?

   - Open APIs (Public APIs): Available to external developers.
   - Internal APIs (Private APIs): Used within an organization.
   - Partner APIs: Shared with specific partners or external parties.
   - Composite APIs: Combine multiple APIs into a single interface.

3. What is the difference between REST and SOAP?

   - REST (Representational State Transfer): Lightweight, uses HTTP methods (GET, POST, PUT, DELETE), stateless, and returns data in formats like JSON or XML.
   - SOAP (Simple Object Access Protocol): Heavier, relies on XML, supports strict security and messaging protocols, and is more rigid in design.

4. What are HTTP methods, and how are they used in APIs?

   - GET: Retrieves data from the server.
   - POST: Sends data to the server.
   - PUT: Updates existing data on the server.
   - DELETE: Deletes data from the server.
   - PATCH: Partially updates data on the server.

5. What is the difference between a public API, private API, and partner API?

   - Public API: Accessible to anyone, typically intended for third-party developers.
   - Private API: Internal to an organization, not accessible to external developers.
   - Partner API: Shared between specific business partners or third parties under agreed terms.

6. What is the difference between RESTful & RESTless API?

   - RESTful API: Follows REST principles, such as stateless communication, use of HTTP methods, and resource-based URIs.
   - RESTless API: Does not follow REST principles, often uses proprietary methods or stateful interactions.

7. What is RESTful API design?

   - RESTful API design involves designing APIs based on REST principles, where endpoints represent resources, and standard HTTP methods (GET, POST, PUT, DELETE) are used for operations on those resources.

8. What is the difference between REST and GraphQL?

- REST: Uses multiple endpoints for different resources and relies on HTTP methods to perform actions.
- GraphQL: Uses a single endpoint and allows clients to request only the data they need, providing more flexibility in querying.

9. How do you secure an API?

- Use authentication (e.g., OAuth, JWT).
- Implement authorization mechanisms (e.g., role-based access).
- Encrypt data (e.g., HTTPS).
- Validate input to prevent injections.
- Rate limit and log activity.

10. What is API rate limiting, and why is it important?

- Rate limiting controls the number of requests a user or system can make to an API within a certain time frame. It prevents abuse, ensures fair usage, and protects against DDoS attacks.

11. What is the difference between synchronous and asynchronous APIs?

- Synchronous: The client waits for the server's response before continuing.
- Asynchronous: The client sends a request and continues processing, with the response arriving later.

12. What is a 401 error? How does it differ from a 403 error?

- 401 Unauthorized: The client must authenticate to access the resource.
- 403 Forbidden: The client is authenticated but does not have permission to access the resource.

13. What is a 500 error, and how would you debug it?

- A 500 Internal Server Error indicates a server-side problem. To debug, check the server logs for errors, misconfigurations, or issues in code execution.

14. How do you design error responses in an API?

- Return meaningful HTTP status codes (e.g., 400 for bad requests).
- Include error messages in the response body with details about the error.
- Provide error codes that make it easier for the client to understand and resolve issues.

15. What is OAuth, and how does it work?

- OAuth is an authorization framework that allows third-party applications to access user data without exposing credentials. It uses tokens to grant temporary access.

16. What is the difference between API keys and JWT?

- API Key: A simple identifier used to authenticate a client.

- JWT (JSON Web Token): A more secure, self-contained token used for authorization, containing user and session information.

17. What tools or libraries have you used for working with APIs?

- Tools: Postman, Swagger, Insomnia.
- Libraries: Axios, Fetch (JavaScript), Requests (Python), Retrofit (Android), Alamofire (iOS).

18. How do you document an API?

- Use tools like Swagger/OpenAPI to generate interactive, detailed API documentation.
- Include clear descriptions of endpoints, parameters, response formats, status codes, and examples.

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

- CORS (Cross-Origin Resource Sharing) is a security feature that restricts how web pages from one domain can request resources from another domain. It's important for preventing unauthorized access to resources.

20. How would you improve the performance of an API?

- Optimize database queries.
- Cache responses to reduce load.
- Use pagination for large datasets.
- Minimize payload size by selecting only necessary data.
- Compress responses (e.g., GZIP).