

1. What is client-side and server-side in web development, and what is the main difference between the two?

Client-side: Code that runs in the user's web browser, responsible for the user interface and interactivity of the web application.

Server-side: Code that runs on the server, handles requests, executes business logic, and generates dynamic content.

2. What is an HTTP request and what are the different types of HTTP requests?

HTTP request: A message sent by a client (typically a web browser) to a server to request data or perform an action. It includes information such as the request method, URL, headers, and sometimes a body.

Types of HTTP requests:

- GET: Retrieves data from the server.
- POST: Submits data to the server to create a new resource.
- PUT: Updates an existing resource on the server.
- DELETE: Requests the removal of a resource on the server.
- PATCH: Partially updates an existing resource on the server.
- And more (HEAD, OPTIONS, etc.).

3. What is JSON and what is it commonly used for in web development?

JSON (JavaScript Object Notation) is a lightweight data interchange format. It is commonly used in web development to represent and transmit data between a server and a client. JSON is human-readable and easy to parse in various programming languages. It is often used in web APIs to send and receive structured data.

4. What is a middleware in web development, and give an example of how it can be used?

Middleware refers to functions or components that sit between the web application and the server. It intercepts and processes incoming requests and outgoing responses. Middleware can handle tasks such as authentication, authorization, request processing, logging, and error handling. An example is authentication middleware that verifies user credentials before allowing access to protected routes in a web application.

5. What is a controller in web development, and what is its role in the MVC architecture?

In web development, a controller is a component of a web application that is responsible for handling user requests. It receives requests from the client, processes them, and then returns a response.

Controllers are typically used in the MVC (Model-View-Controller) architecture. In MVC, controllers are responsible for mediating between the model and the view. The model contains the application's data, and the view displays the data to the user.