

Introdução ao Desenvolvimento Back-end

No desenvolvimento de software, a parte "back-end" refere-se à camada responsável por processar e armazenar dados. Enquanto o front-end lida com a interface que os usuários interagem diretamente, o back-end cuida das funcionalidades internas que tornam essas interações possíveis. Isso inclui operações como processamento de formulários, autenticação de usuários, manipulação de dados e comunicação com o banco de dados.

Conceitos de Servidor, Roteamento e APIs

Um servidor é um computador ou sistema que fornece recursos ou serviços para outros computadores, conhecidos como clientes, por meio de uma rede. No contexto da web, um servidor web é responsável por fornecer conteúdo, como páginas da web, para os navegadores dos clientes.

O roteamento é o processo de direcionar solicitações de um cliente para o recurso correto no servidor. Isso é essencial para garantir que as solicitações sejam processadas corretamente e que as respostas sejam enviadas de volta ao cliente de forma eficiente.

As APIs (Interfaces de Programação de Aplicativos) são conjuntos de regras e protocolos que permitem que diferentes softwares se comuniquem entre si. No contexto do desenvolvimento web, as APIs são frequentemente usadas para permitir a comunicação entre o front-end e o back-end de uma aplicação.

Protocolo HTTP

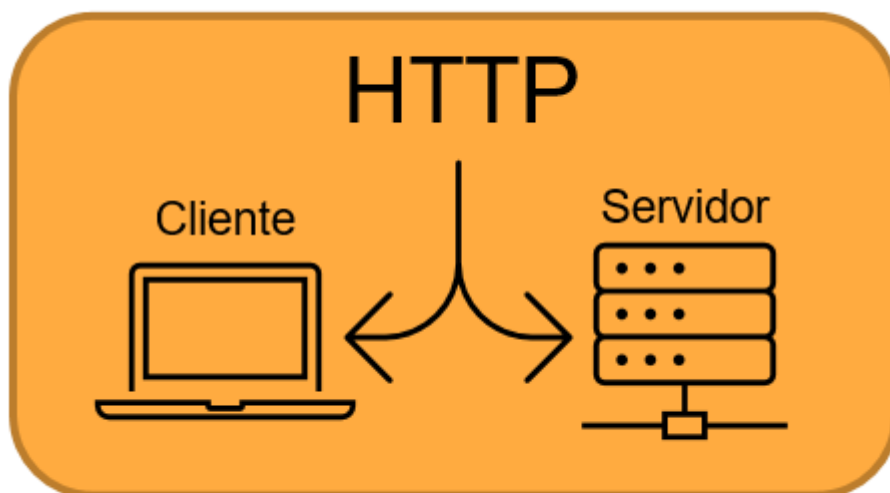
O Protocolo de Transferência de Hipertexto (HTTP), é o protocolo fundamental da World Wide Web (WWW) que define a maneira como as mensagens são formatadas e transmitidas pela internet. Ele permite a comunicação entre clientes, como navegadores da web, e servidores web, facilitando o acesso e a transferência de recursos, como páginas da web, imagens, vídeos e outros arquivos.

O HTTP opera em um modelo cliente-servidor, onde o cliente (geralmente um navegador da web) faz uma solicitação para um servidor web e espera uma resposta. Esta comunicação é realizada através de uma série de mensagens, chamadas de requisições e respostas, que são trocadas entre o cliente e o servidor.

Componentes Fundamentais do HTTP

1. **Métodos HTTP:** O HTTP define uma variedade de métodos que especificam a ação a ser realizada em um recurso identificado. Alguns dos métodos mais comuns incluem:
 - **GET:** utilizado para solicitar dados de um recurso.
 - **POST:** utilizado para enviar dados para um servidor para serem processados.
 - **PUT:** utilizado para atualizar dados de um recurso existente.
 - **DELETE:** utilizado para excluir um recurso.
2. **URLs (Uniform Resource Locators):** As URLs são endereços que identificam recursos na web. Elas consistem em um protocolo (como "http://" ou "https://"), um nome de domínio (como "example.com") e um caminho para o recurso desejado (como "/pagina.html").
3. **Cabeçalhos HTTP:** Os cabeçalhos são informações adicionais incluídas nas requisições e respostas HTTP. Eles fornecem metadados sobre a mensagem, como o tipo de conteúdo, a codificação de caracteres, as cookies, entre outros.
4. **Corpo da Mensagem:** O corpo da mensagem é a parte opcional de uma requisição ou resposta HTTP que contém os dados sendo enviados entre o cliente e o servidor. Ele é usado para transmitir informações adicionais, como parâmetros de formulário, payloads JSON ou conteúdo de arquivos.

Funcionamento do HTTP



Quando um cliente faz uma solicitação HTTP para um servidor, ele envia uma mensagem contendo o método HTTP desejado, a URL do recurso solicitado e outros cabeçalhos opcionais. O servidor então processa a solicitação e retorna uma resposta HTTP, que inclui um código de status indicando se a solicitação foi

bem-sucedida, redirecionada ou encontrou um erro, juntamente com o corpo da mensagem, se aplicável.

O HTTP é um protocolo sem estado, o que significa que cada solicitação é tratada de forma independente, sem conhecimento das solicitações anteriores. Isso permite que o HTTP seja altamente escalável e eficiente para a transferência de recursos na web.

Em resumo, o Protocolo HTTP é a base da comunicação na web, permitindo que clientes e servidores troquem informações de forma eficiente e confiável. É graças ao HTTP que podemos acessar e interagir com uma infinidade de recursos na internet, tornando-a uma parte indispensável de nossa experiência digital cotidiana.

Indicações:

- <https://www.hostinger.com.br/tutoriais/http#:~:text=HTTP%20%C3%A9%20a%20sigla%20para,um%20cliente%20e%20um%20servidor.>
- <https://www.webfx.com/web-development/glossary/http-status-codes>
- <https://httpstatusdogs.com/>
- <https://http.cat/>