

## HTTP/S Basic

O que é HTTP/S? (Hypertext Transfer Protocol)

HTTP é usado sempre que você visualiza um site, desenvolvido por Tim Berners-Lee e sua equipe entre 1989 e 1991. HTTP é o conjunto de regras usadas para comunicação com servidores para a transmissão de dados de páginas web, sejam eles HTML, imagens, vídeos, etc.

HTTPS é a versão segura do HTTP. Os dados HTTPS são criptografados, o que não só impede que as pessoas vejam os dados que você está recebendo e enviando, como também lhe dá garantias de que você está se comunicando com o servidor web correto e não com algum site supostamente falso.

O que é URL? (Uniform Resource Locator)

Se você já usou a internet, provavelmente já usou uma URL. Uma URL é uma instrução sobre como acessar um recurso na internet.

<http://example.com:8080/login.php?id=1#task2>

**Scheme** = [**http://**] Isso instrui sobre qual protocolo usar para acessar o recurso, como HTTP, HTTPS, FTP (File Transfer Protocol).

**Host** = [[example.com](http://example.com)] O nome de domínio ou endereço IP do servidor que você deseja acessar.

**Port** = [**:8080**] A porta à qual você vai se conectar, geralmente 80 para HTTP e 443 para HTTPS, mas pode ser hospedada em qualquer porta entre 1 e 65535.

**Path** = [**/login.php**] O nome do arquivo ou local do recurso que você está tentando acessar.

**Query String** = [**?id=1**] Informações extras que podem ser enviadas para o caminho solicitado. Por exemplo, `/login.php?id=1` poderia ser um identificador de usuário ou sessão, mas não é uma informação padrão e depende da implementação do sistema.

**Fragment** = [**#task2**] Esta é uma referência a um local na página solicitada. É comumente usada para páginas com conteúdo longo e pode ter uma determinada parte da página diretamente vinculada a ela, para que fique visível ao usuário assim que ele acessar a página.

## Cabeçalhos (Headers)

Cabeçalhos são bits de dados adicionais que você pode enviar ao servidor web ao fazer solicitações. Embora nenhum cabeçalho seja estritamente necessário ao fazer uma solicitação HTTP, você terá dificuldade para visualizar um site corretamente.

## HTTP Request

GET / HTTP/1.1

Host: [tryhackme.com](https://tryhackme.com/)

Accept-Encoding: gzip

Cookie: WmkojKIKHuyguykgYKGykGuyguy...

User-Agent: Mozilla/5.0 Firefox/87.0

Referer: <https://tryhackme.com/>

## HTTP Response

HTTP/1.1 200 OK

Server: nginx/1.15.8

Date: Fri, 09 Apr 2021 13:34:03 GMT

Content-Type: text/html

Content-Length: 98

Content-Encoding: gzip

Server: tryhackme.com

Set-Cookie: WsMkINklJKbjhBjhgyyjhgJHuyjh...

<html>

<head>

    <title>TryHackMe</title>

</head>

<body>

    Welcome To TryHackMe.com

</body>

</html>

## Métodos comuns HTTP

**GET Request** = usado para obter informações de um servidor web.

**POST Request** = usado para enviar dados ao servidor web e potencialmente criar novos registros.

**PUT Request** = usado para enviar dados a um servidor web para atualizar informações.

**DELETE Request** = usado para excluir informações/registros de um servidor web.

Realizar de modo manual uma requisição HTTP

\$nc [google.com](https://google.com/) 80

GET / HTTP/1.1

Host: www.google.com

## Códigos de Status HTTP

100-199 - Information Response = são enviados para informar ao cliente que a primeira parte da solicitação foi aceita e que ele deve continuar enviando o restante. Esses códigos não são muito comuns.

200-299 - Success = esse intervalo de códigos de status é usado para informar ao cliente que sua solicitação foi bem-sucedida.

300-399 - Redirection = são usados para redirecionar a solicitação do cliente para outro recurso, que pode ser para uma página da web diferente ou para um site completamente diferente.

400-499 - Client Errors = usado para informar o cliente que houve um erro em sua solicitação.

500-599 - Server Errors = reservado para erros que acontecem no lado do servidor e geralmente indicam um problema sério com o servidor que está manipulando a solicitação.

## Códigos Comuns de Status HTTP

**200 - OK** = A solicitação foi concluída com sucesso.

**201 - Created** = Um recurso foi criado (por exemplo, um novo usuário ou uma nova postagem de blog).

**301 - Moved Permanently** = Isso redireciona o navegador do cliente para uma nova página da web ou informa aos mecanismos de busca que a página foi movida para outro local e que devem procurá-la.

**302 - Found** = Semelhante ao redirecionamento permanente acima, mas, como o nome sugere, esta é apenas uma alteração temporária e pode mudar novamente em um futuro próximo.

**400 - Bad Request** = Isso informa ao navegador que algo estava errado ou ausente na solicitação. Às vezes, isso pode ser usado se o recurso do servidor web solicitado esperava um determinado parâmetro que o cliente não enviou.

**401 - Not Authorised** = Você não tem permissão para visualizar este recurso até que tenha sido autorizado pelo aplicativo web, geralmente com um nome de usuário e senha.

**403 - Forbidden** = Você não tem permissão para visualizar este recurso, esteja você logado ou não.

**405 - Method Not Allowed** = O recurso não permite esta solicitação de método. Por exemplo, você envia uma solicitação GET para o recurso /create-account quando ele esperava uma solicitação POST.

**404 - Page Not Found** = A página/recurso solicitado não existe.

**500 - Internal Service Error** = O servidor encontrou algum tipo de erro em sua solicitação que ele não sabe como tratar corretamente.

**503 - Service Unavailable** = Este servidor não pode processar sua solicitação, pois está sobrecarregado ou inativo para manutenção.

Fonte:

<https://http.cat>

[https://www.w3schools.com/whatis/whatis\\_http.asp](https://www.w3schools.com/whatis/whatis_http.asp)