**A typical HTTP session**

Em protocolos cliente-servidor, como HTTP, as sessões consistem em três fases:

1. O cliente estabelece uma conexão TCP (ou a conexão apropriada se a camada de transporte não for TCP).
2. O cliente envia sua solicitação e aguarda a resposta.
3. O servidor processa a solicitação, enviando de volta sua resposta, fornecendo um código de status e dados apropriados.

A partir do HTTP/1.1, a conexão não é mais fechada após a conclusão da terceira fase, e o cliente agora recebe uma nova solicitação: isso significa que a segunda e a terceira fases agora podem ser executadas quantas vezes quiser.

**Estabelecendo uma conexão**

Em protocolos cliente-servidor, é o cliente quem estabelece a conexão. Abrir uma conexão em HTTP significa iniciar uma conexão na camada de transporte subjacente, geralmente o TCP.

Com TCP, a porta padrão para um servidor HTTP em um computador é a porta 80. Outras portas também podem ser usadas, como 8000 ou 8080. A URL de uma página a ser buscada contém o nome de domínio e o número da porta, embora este último possa ser omitido se for 80. Consulte a referência de URL para obter mais detalhes.

Observação: o modelo cliente-servidor não permite que o servidor envie dados ao cliente sem uma solicitação explícita. No entanto, diversas APIs Web permitem esse uso, incluindo a API Push, eventos enviados pelo servidor e a API WebSockets.

**Sending a client request**

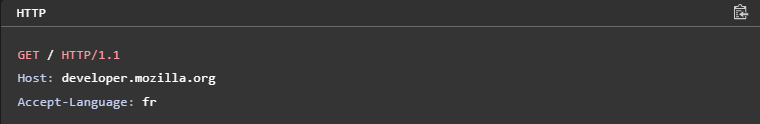
1. A primeira linha contém um método de solicitação seguido de seus parâmetros:

* o caminho do documento, como uma URL absoluta sem o protocolo ou nome de domínio
* a versão do protocolo HTTP

1. As linhas subsequentes representam um cabeçalho HTTP, fornecendo ao servidor informações sobre qual tipo de dado é apropriado (por exemplo, qual idioma, quais tipos MIME) ou outros dados que alteram seu comportamento (por exemplo, não enviar uma resposta se ela já estiver armazenada em cache). Esses cabeçalhos HTTP formam um bloco que termina com uma linha em branco.
2. O bloco final é um bloco de dados opcional, que pode conter dados adicionais usados ​​principalmente pelo método POST.

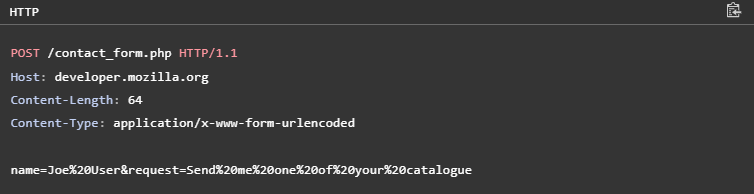
Example requests

Buscando a página raiz do developer.mozilla.org, (https://developer.mozilla.org/), e informando ao servidor que o agente do usuário prefere a página em francês, se possível:



Observe a última linha vazia, que separa o bloco de dados do bloco de cabeçalho. Como não há Content-Length fornecido em um cabeçalho HTTP, este bloco de dados é apresentado vazio, marcando o fim dos cabeçalhos, permitindo que o servidor processe a solicitação no momento em que recebe esta linha vazia.

Por exemplo, enviando o resultado de um formulário:



Request methods

O HTTP define um conjunto de métodos de solicitação que indicam a ação desejada a ser executada em um recurso. Embora também possam ser substantivos, esses métodos de solicitação são, às vezes, chamados de verbos HTTP. As solicitações mais comuns são GET e POST:

* O método GET solicita uma representação de dados do recurso especificado. Requisições que utilizam GET devem recuperar apenas dados.
* O método POST envia dados para um servidor para que este possa alterar seu estado. Este é o método frequentemente usado para formulários HTML.

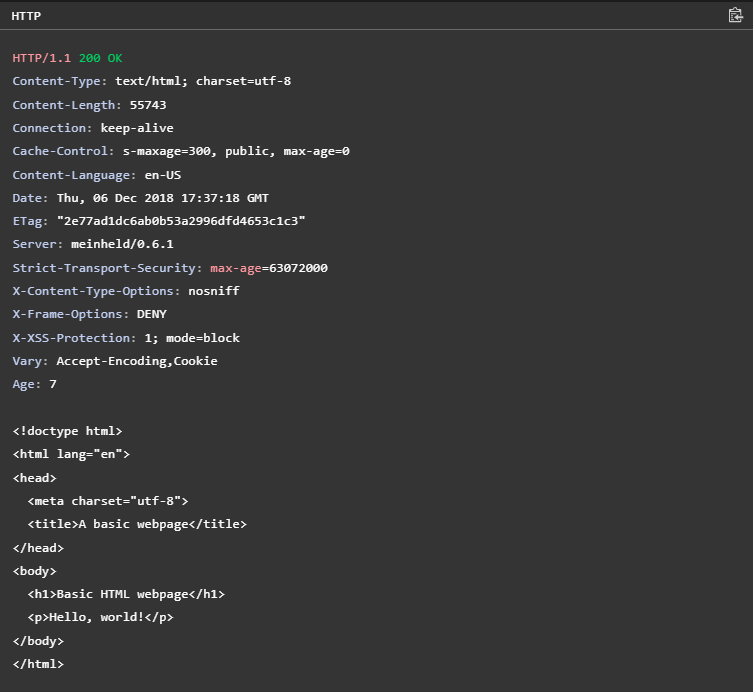
Estrutura de uma resposta do servidor

Após o agente conectado enviar sua solicitação, o servidor web a processa e, por fim, retorna uma resposta. Semelhante a uma solicitação de cliente, uma resposta do servidor é formada por diretivas de texto, separadas por CRLF, embora divididas em três blocos:

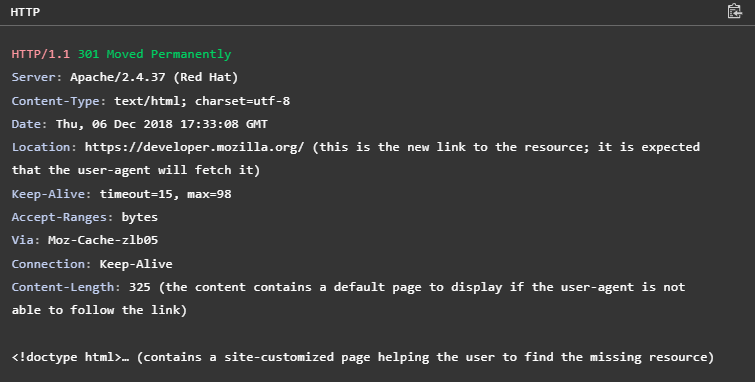
1. A primeira linha, a linha de status, consiste em uma confirmação da versão HTTP usada, seguida por um código de status de resposta (e seu breve significado em texto legível).
2. As linhas subsequentes representam cabeçalhos HTTP específicos, fornecendo ao cliente informações sobre os dados enviados (por exemplo, tipo, tamanho dos dados, algoritmo de compactação utilizado, dicas sobre cache). Semelhante ao bloco de cabeçalhos HTTP para uma solicitação do cliente, esses cabeçalhos HTTP formam um bloco que termina com uma linha em branco.
3. O bloco final é um bloco de dados, que contém os dados opcionais.

Respostas de exemplo

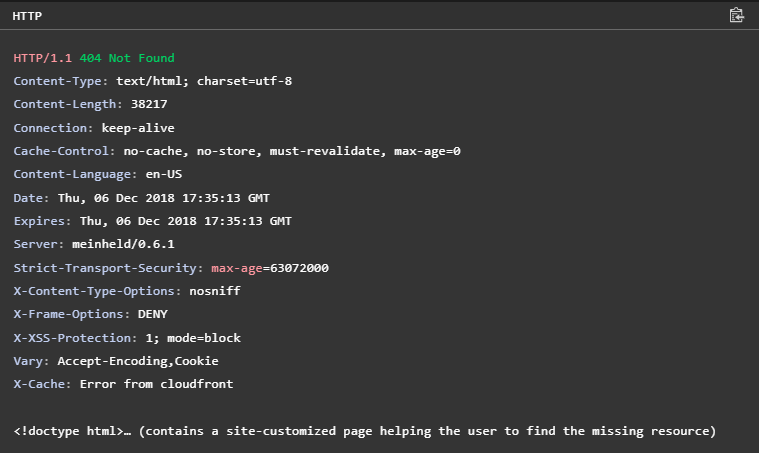
Resposta bem-sucedida da página da web:



Notificação de que o recurso solicitado foi movido permanentemente:



Notificação de que o recurso solicitado não existe:



Códigos de status de resposta

Os códigos de status de resposta HTTP indicam se uma solicitação HTTP específica foi concluída com sucesso. As respostas são agrupadas em cinco classes: respostas informativas, respostas bem-sucedidas, redirecionamentos, erros do cliente e erros do servidor.

200: OK. A solicitação foi bem-sucedida.

301: Movido permanentemente. Este código de resposta significa que o URI do recurso solicitado foi alterado.

404: Não encontrado. O servidor não consegue encontrar o recurso solicitado.