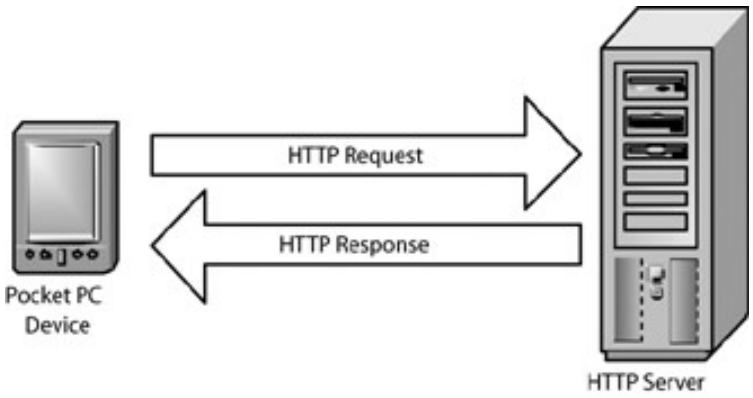
Redes de Computadores

Prof. Robson de Souza

Aulas 17 e 18

Conteúdo: Camada de Aplicação: HTTP (HyperText Transfer Protocol).

HTTP (HyperText Transfer Protocol)

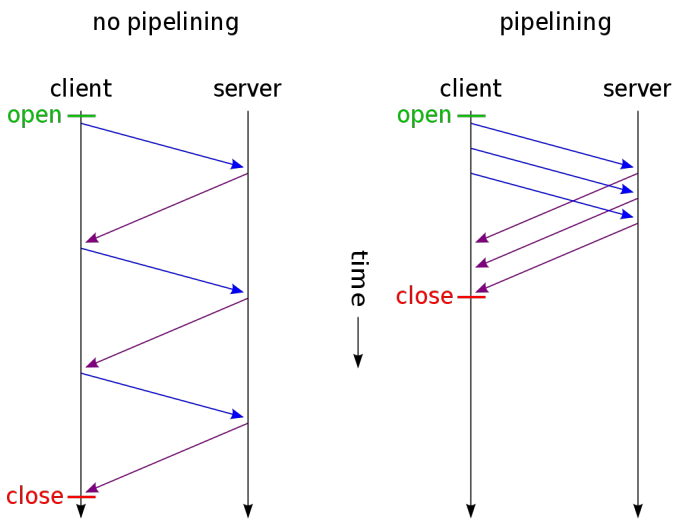


http://www.writeopinions.com/hypertext-transfer-protocol

O HTTP é o protocolo de transferência utilizado em toda World Wide Web.
Ele especifica as mensagens que os clientes podem enviar aos servidores e que respostas eles receberão.
O modo habitual de um navegador entrar em contato com um servidor é estabelecer uma conexão TCP com a porta 80 da máquina servidora.

A vantagem de se utilizar o TCP é que nem os navegadores, nem os servidores tem de se preocupar com mensagens perdidas, duplicadas, longas ou confirmações. Tudo isso é tratado pelo TCP.
No HTTP 1.0, uma única solicitação era enviada e uma única resposta era recebida, então, a conexão TCP era encerrada. Atualmente, as páginas contém muitos ícones, imagens, etc. Com isso, estabelecer uma conexão TCP para transportar cada elemento é algo que exige muito.

No HTTP 1.1, existem conexões persistentes, com isso, é possível estabelecer uma conexão TCP, enviar uma solicitação e obter uma resposta e enviar solicitações adicionais e receber respostas adicionais. Também é possível transportar as solicitações por pipeline, ou seja, enviar a solicitação 2 antes de chegar a resposta da solicitação 1.



https://en.wikipedia.org/wiki/HTTP

*Métodos HTTP

O HTTP aceita operações chamadas métodos, diferentes da simples solicitação de uma página Web. O método **GET** solicita ao servidor que envie a página (arquivo). A grande maioria das solicitações a servidores da Web tem a forma de métodos GET.

Ex:

GET nomeArquivo HTTP/1.1

nomeArquivo → Arquivo a ser buscado.

1.1 → Versão do protocolo que está sendo usado.

O método **HEAD** solicita apenas o cabeçalho da mensagem, sem a página propriamente dita. Esse método pode ser usado para se obter informações da página, tais como a ultima modificação, codificação, etc.

O método **PUT** é o inverso do GET. Ao invés de ler, ele grava a página, isto possibilita a criação de um conjunto de páginas da Web em um servidor remoto.

O método **POST** é semelhante ao PUT, mas ao invés de substituir os dados existentes, os novos dados são “anexados” a ele.

O método **DELETE** serve para excluir uma página.

O método **TRACE** serve para verificar as solicitações que foram recebidas, esse método instrui o servidor a enviar de volta a solicitação. Isso é útil quando as solicitações não estão sendo processadas de forma correta e o cliente deseja saber qual solicitação o servidor recebeu de fato.

O método **OPTIONS** serve para que o cliente consulte o servidor sobre suas propriedades ou sobre as de um arquivo específico.

Toda solicitação obtém uma resposta que consiste em uma linha de status e, possivelmente, informações adicionais. A linha de status tem um código de três dígitos informando se a solicitação foi atendida e, se não foi, porque não.

O primeiro dígito é usado para dividir as respostas em cinco grupos:

Código	**Significado**
1XX	Informação
2XX	Sucesso
3XX	Redirecionamento
4XX	Erro do cliente
5XX	Erro do servidor

Exemplos:

301 → Page moved.

404 → Page not found.

503 → Try again later.

A linha de solicitação (por exemplo, a linha com o comando GET) pode ser seguida por linhas adicionais com mais informações. Elas são chamadas **cabeçalhos de solicitação**.

Exemplos:

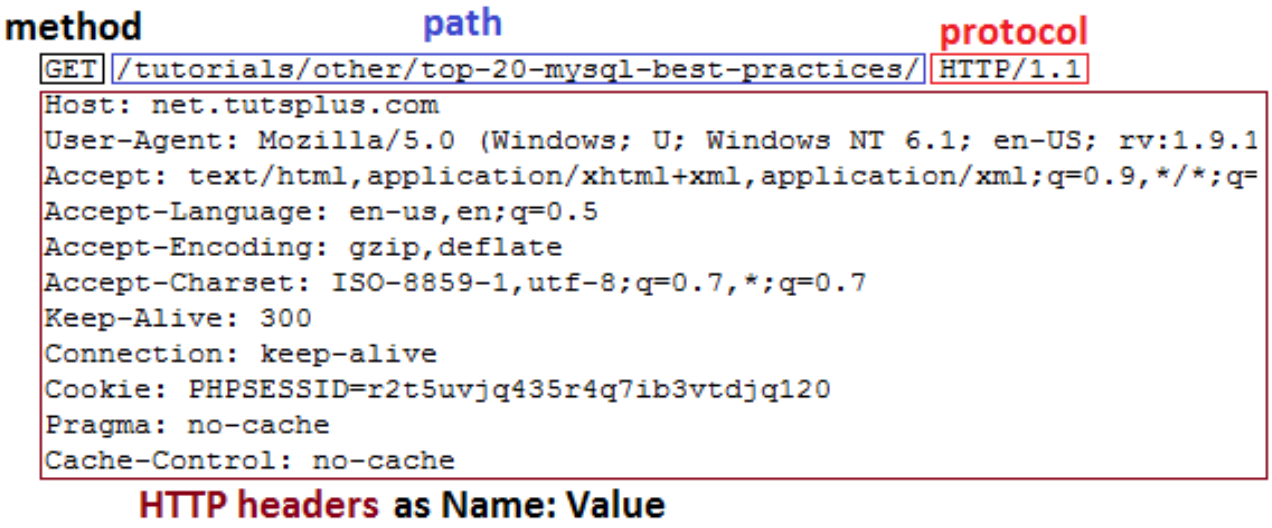
User-Agent → Permite ao cliente informar o servidor sobre seu navegador, Sistema Operacional e outras propriedades.

Accept → Tipo de páginas que o cliente pode manipular.
Accept-Charset → caracteres aceitáveis.
Accept-Language → Idiomas.
Etc.

Host → O nome DNS do servidor.

Authorization → Serve para o cliente provar que tem o direito de ver a página solicitada.

Existem vários outros cabeçalhos … Tanto de solicitação, quanto resposta.



https://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039

Para utilizar o HTTP, basta uma conexão TCP para a porta 80 no servidor.

Exemplo de comandos:

telnet www.ietf.org 80 > log

GET /rfc.html HTTP/1.1

Host: www.ietf.org

close

Referências bibliográficas:

TANENBAUM, Andrew. S. Redes de Computadores. São Paulo: *Pearson*, 5ª Ed. 2011.

KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet – Uma Abordagem Top-Down. *São Paulo: Pearson*, 6ª Ed. 2013.