

Desenvolvimento de Sistemas Web

Arquitetura da Aplicação Web (Comunicação Cliente/Servidor)

Prof. Mauro Lopes

Objetivos

- Iniciaremos aqui o estudo sobre o desenvolvimento de sistemas web usando o Java. Apresentaremos nesta aula os conceitos básicos sobre a Arquitetura de uma Aplicação Web. Ao final o aluno terá absorvido os conceitos básicos da arquitetura que irá permitir um melhor entendimento do funcionamento de *Servlets* e *JSF*. Bom estudo a todos!

Plano de Aula

- Protocolo HTTP
 - Requisição/Resposta HTTP
 - Quem são os user agents?
 - Quem são os Servidores?
 - Servidor Web Apache
 - Servidor Web IIS
- Comunicação com o Servidor
 - Requisição/Resposta HTTP – Parte 2
 - Métodos HTTP
 - Status HTTP
 - Resumindo
 - Cliente-Servidor e Métodos HTTP

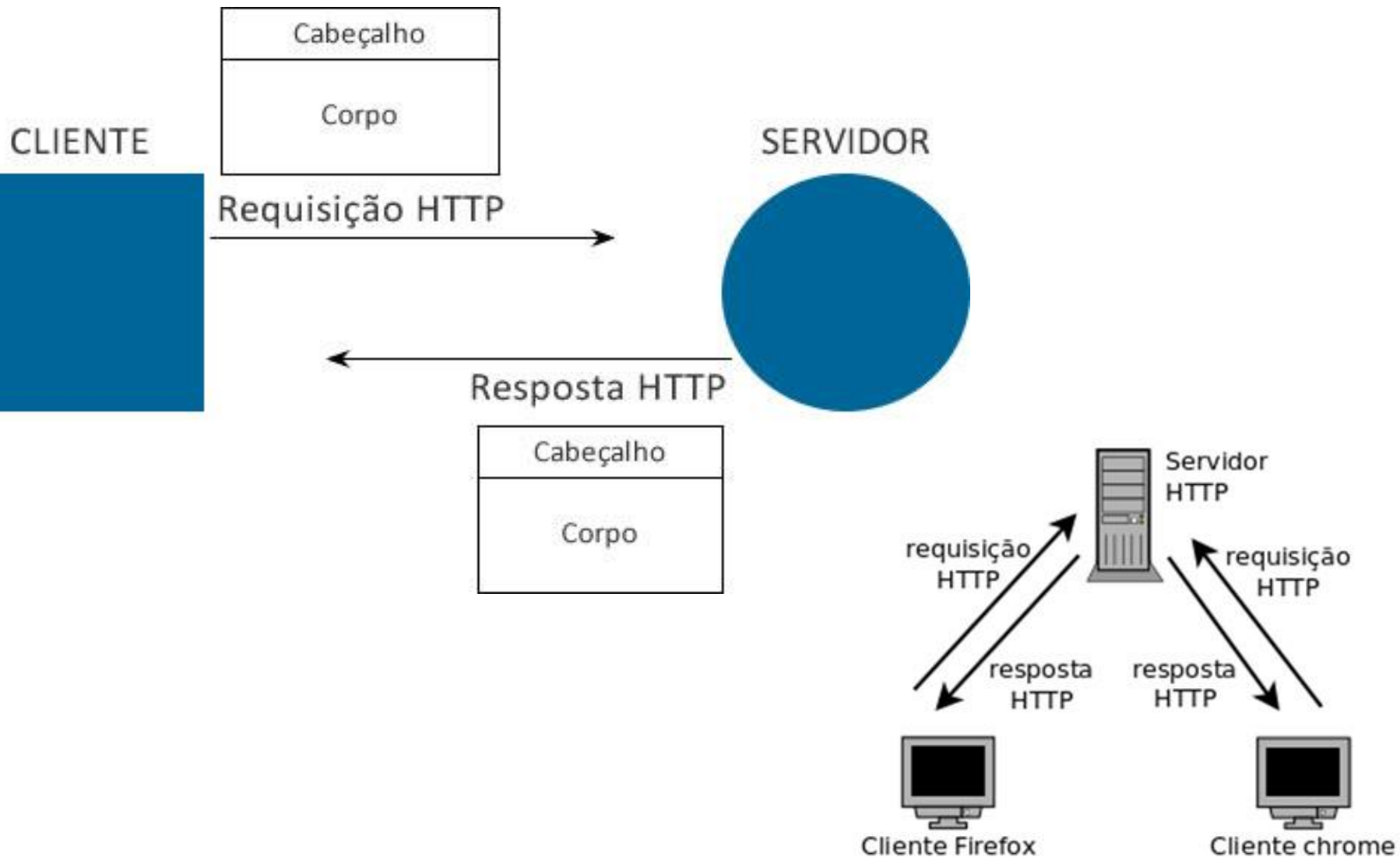
Protocolo HTTP

- *Hypertext Transfer Protocol* (HTTP) é o método utilizado para enviar e receber informações na web;
- A versão mais utilizada atualmente é a 1.1, definida pela especificação RFC 2616;
- Embora esta especificação devesse ser leitura obrigatória para todo desenvolvedor web, muitos nem sabem como ela se parece!!!

Protocolo HTTP

- O protocolo HTTP é baseado em **requisições e respostas** entre **clientes e servidores**;
- O **cliente** — navegador ou dispositivo que fará a **requisição**; também é conhecido como *user agent* — solicita um determinado **recurso** (resource), enviando um pacote de informações contendo alguns cabeçalhos (*headers*) a um **URI** ou, mais especificamente, **URL**;
- O **servidor** recebe estas informações e envia uma **resposta**, que pode ser um **recurso** ou simplesmente um outro cabeçalho.

Requisição/Resposta HTTP



Quem são os *user agents*?



Quem são os Servidores?

- Um servidor web é um computador que tem **software** e **hardware** em que vai permitir que os usuários solicitem páginas web;
- O computador servidor vai **entregar** essas páginas da web para os computadores que o solicitem;
- Ele também pode fazer outras transformações com as páginas da web, como **cálculos** e **acessos a bancos de dados**.

Servidor Web Apache

- Apache é um software de servidor web que é muito popular e projetado para rodar em computadores com o sistema operacional Linux (mas que também pode “rodar” no Windows);



Servidor Web IIS

- **IIS** (*Internet Information Services*) é um servidor web criado pela Microsoft;
- Sua primeira versão foi introduzida com o Windows NT Server versão 4, e passou por várias atualizações;
- A versão mais recente é o **IIS 8.5** (disponível apenas no Windows Server 2012 R2 e Windows 8.1);
- A função do IIS nos servidores Windows é **oferecer uma plataforma para a hospedagem de sites, serviços e aplicativos**;
- O IIS é capaz de integrar as seguintes tecnologias: ASP.NET, FTP, PHP, WCF etc.

Comunicação com o Servidor

- Vamos supor que o usuário queira acessar o site:
 - <http://www.spesa.com.br>
- Com um programa tipo *sniffer* poderíamos ver a solicitação e a resposta desta ação do usuário;

The screenshot shows the HttpDetect (EffeTech HTTP Sniffer) application window. The main window displays a list of captured HTTP requests. The columns are: No., Time, Client (IP:PORT), Server (IP:PORT), URL, File Size, and Status. The status column shows 'FIN, 200' for most requests, indicating successful completion.

No.	Time	Client (IP:PORT)	Server (IP:PORT)	URL	File...	Status
69	Ju...	192.168.1.3 :3057	www.iffetech.com :80	/	35660	FIN, 200
70	Ju...	192.168.1.3 :3058	www.iffetech.com :80	/images/style.css		FIN, 304
71	Ju...	192.168.1.3 :3058	www.iffetech.com :80	/images/logo_main.jpg		FIN, 304
72	Ju...	192.168.1.3 :3058	www.iffetech.com :80	/images/chinese_edition.gif		FIN, 304
73	Ju...	192.168.1.3 :3058	www.iffetech.com :80	/images/space.gif		FIN, 304
74	Ju...	192.168.1.3 :3059	www.iffetech.com :80	/images/arrow_small.gif		FIN, 304
75	Ju...	192.168.1.3 :3058	www.iffetech.com :80	/images/award_tucows_4ratel...		FIN, 304
76	Ju...	192.168.1.3 :3060	www.etherdetect.com...	/images/logo_ms.gif	628	FIN, 200
77	Ju...	192.168.1.3 :3061	www.etherdetect.com...	/images/logo_ibm.gif	1217	FIN, 200
78	Ju...	192.168.1.3 :3059	www.iffetech.com :80	/images/award_FileHungry_5s...		FIN, 304
79	Ju...	192.168.1.3 :3058	www.iffetech.com :80	/images/award_softwareseeke...		FIN, 304
80	Ju...	192.168.1.3 :3061	www.etherdetect.com...	/images/logo_mit.gif	259	FIN, 200
81	Ju...	192.168.1.3 :3060	www.etherdetect.com...	/images/logo_ms.gif		Requested
82	Ju...	192.168.1.3 :3059	www.iffetech.com :80	/images/award_webaward2002e...		FIN, 304
83	Ju...	192.168.1.3 :3058	www.iffetech.com :80	/images/ed_small.gif	24269	FIN, 200
84	Ju...	192.168.1.3 :3061	www.etherdetect.com...	/images/logo_cornell.gif	2027	FIN, 200
85	Ju...	192.168.1.3 :3059	www.iffetech.com :80	/images/flag_detail.gif	1026	FIN, 200
86	Ju...	192.168.1.3 :3061	www.etherdetect.com...	/images/logo_reuters.gif	1822	FIN, 200
87	Ju...	192.168.1.3 :3059	www.iffetech.com :80	/images/flag_demo.gif	1013	FIN, 200
88	Ju...	192.168.1.3 :3060	www.etherdetect.com...	/images/logo_mit.gif	259	FIN, 200

The bottom section of the window shows the details of the selected request (No. 77). The HTTP Request Header is as follows:

```
GET /images/logo_ibm.gif HTTP/1.1
Accept: */*
Referer: http://www.iffetech.com/
Accept-Language: zh-cn
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible;
MSIE 6.0; Windows NT 5.1)
Host: www.etherdetect.com
Connection: Keep-Alive
```

The HTTP Response Header is as follows:

```
HTTP/1.1 200 OK
Date: Sat, 07 Jun 2003 13:32:07 GMT
Server: Apache/1.3.27
Last-Modified: Mon, 14 Apr 2003 14:11:33 GMT
ETag: "bdae-4c1-3e9ac195"
Accept-Ranges: bytes
Content-Length: 1217
Keep-Alive: timeout=5, max=100
```

The status bar at the bottom indicates: Ready, Buffer: 3%, URLs: 95, Packets: 393.

Requisição/Resposta HTTP - Parte 2

- Solicitação

```
GET / HTTP/1.1
Host: spesa.com.br
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US) Gecko/20061201 Firefox/2.0.0.3 (Ubuntu-feisty)
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

- Resposta

```
HTTP/1.x 200 OK
Date: Fri, 04 May 2007 16:05:43 GMT
Server: Apache/2.0.59 (Unix) mod_ssl/2.0.59 OpenSSL/0.9.7a DAV/2 PHP/4.4.4 mod_bwlimited/1.4
Cache-Control: no-cache
Keep-Alive: timeout=3, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
```

Métodos HTTP

- Quando você vai fazer uma requisição, é preciso que você especifique qual o **método** será utilizado;
- Os **métodos HTTP**, também conhecidos como **verbos**, identificam qual a ação que deve ser executada em um determinado recurso;
- Existem **8** métodos HTTP, mas apenas **5** são mais utilizados.

Métodos HTTP

— GET

- Solicita a representação de um determinado recurso. É definido como um método seguro e não deve ser usado para disparar uma ação.

— POST

- As informações enviadas no corpo (*body*) da requisição são utilizadas para criar um novo recurso. Também é responsável por fazer processamentos que não são diretamente relacionados a um recurso.

— DELETE

- Remove um recurso. Deve retornar o status 204 caso não exista nenhum recurso para a URI especificada.

— PUT

- Atualiza um recurso na URI especificada. Caso o recurso não exista, ele pode criar um. A principal diferença entre POST e PUT é que o primeiro pode lidar não somente com recursos, mas pode fazer processamento de informações, por exemplo.

— HEAD

- Retorna informações sobre um recurso. Na prática, funciona semelhante ao método GET, mas sem retornar o recurso no corpo da requisição.

Status HTTP

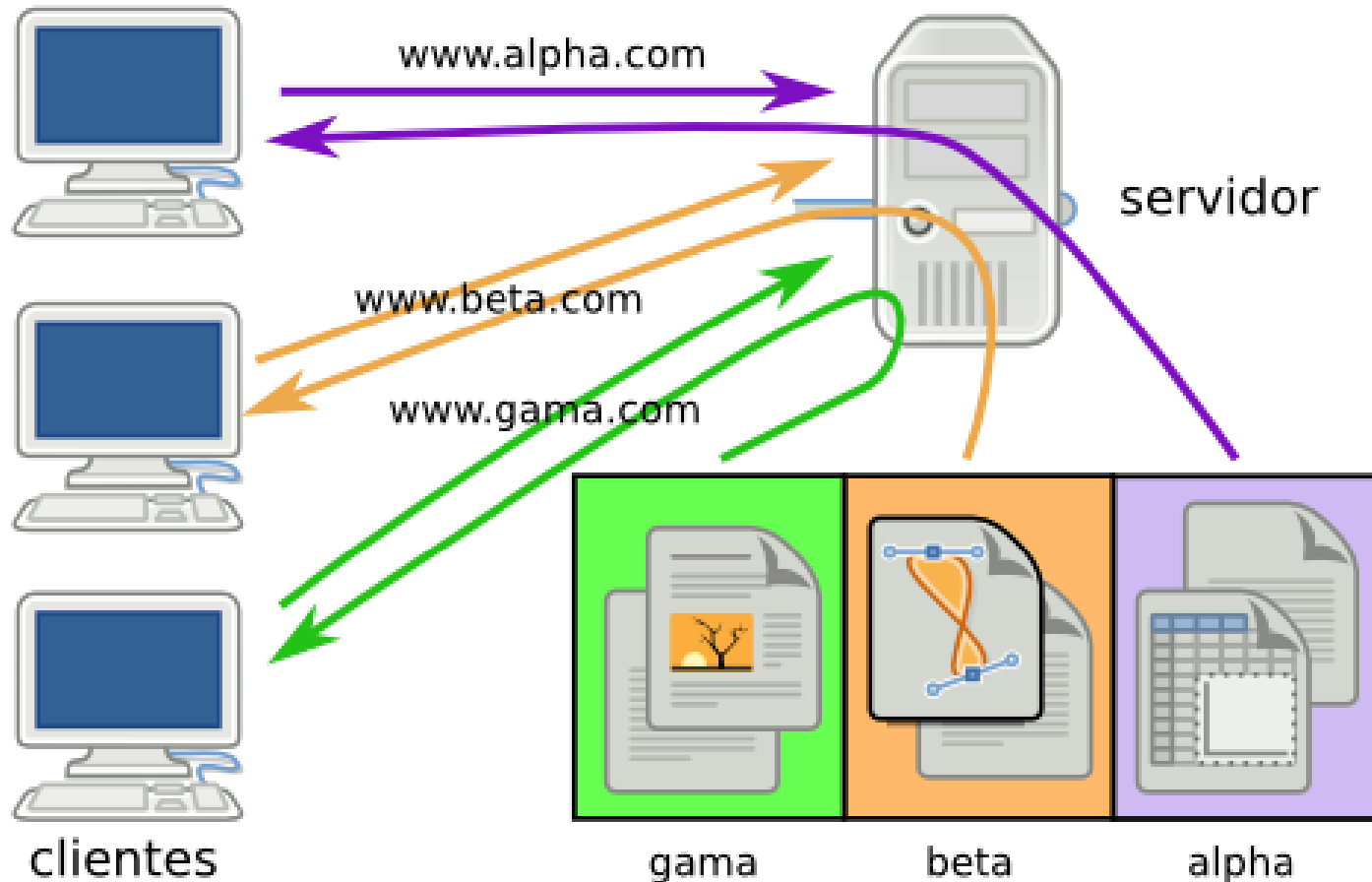
- Toda requisição recebe um código de resposta conhecido como **status**;
- Com o **status** é possível saber se uma operação foi realizada com sucesso (**200**), se ele foi movida e agora existe em outro lugar (**301**) ou se não existe mais (**404**);
- Existem muitos **status** divididos em diversas categorias;
- Na especificação você pode ver cada um deles com uma descrição bastante detalhada.

Status HTTP

- Abaixo, apresentamos alguns códigos que são mais frequentes:
 - **200 OK**
 - A requisição foi bem sucedida.
 - **301 Moved Permanently**
 - O recurso foi movido permanentemente para outra URI.
 - **401 Unauthorized**
 - A URI especificada exige autenticação do cliente.
 - **403 Forbidden**
 - O servidor entende a requisição, mas se recusa em atendê-la.
 - **404 Not Found**
 - O servidor não encontrou nenhuma URI correspondente.
 - **500 Internal Server Error**
 - O servidor não foi capaz de concluir a requisição devido a um erro inesperado.
 - **503 Service Unavailable**
 - O servidor não é capaz de processar a requisição pois está temporariamente indisponível.

Resumindo

- Servidores Web - Hospedagem



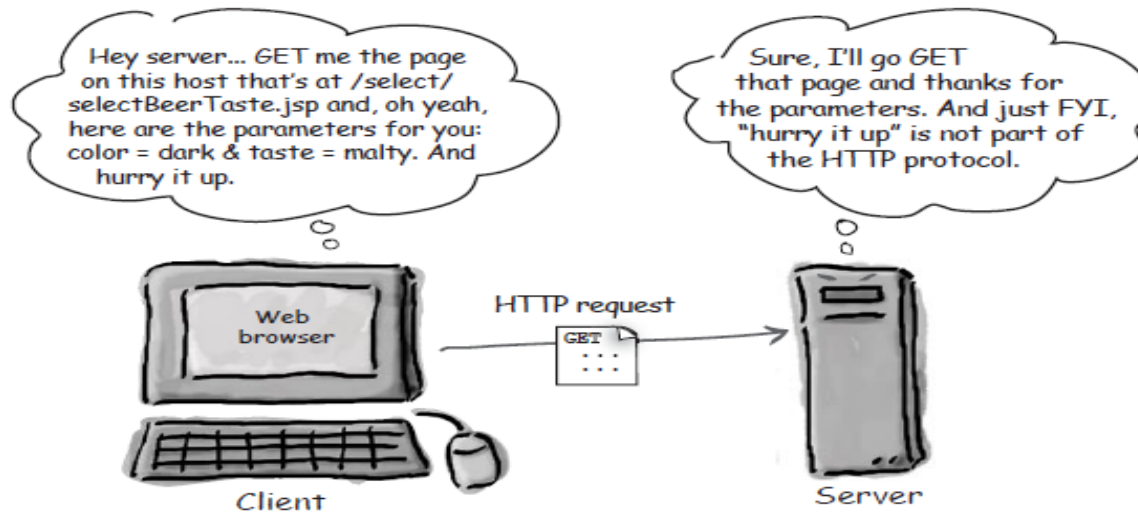
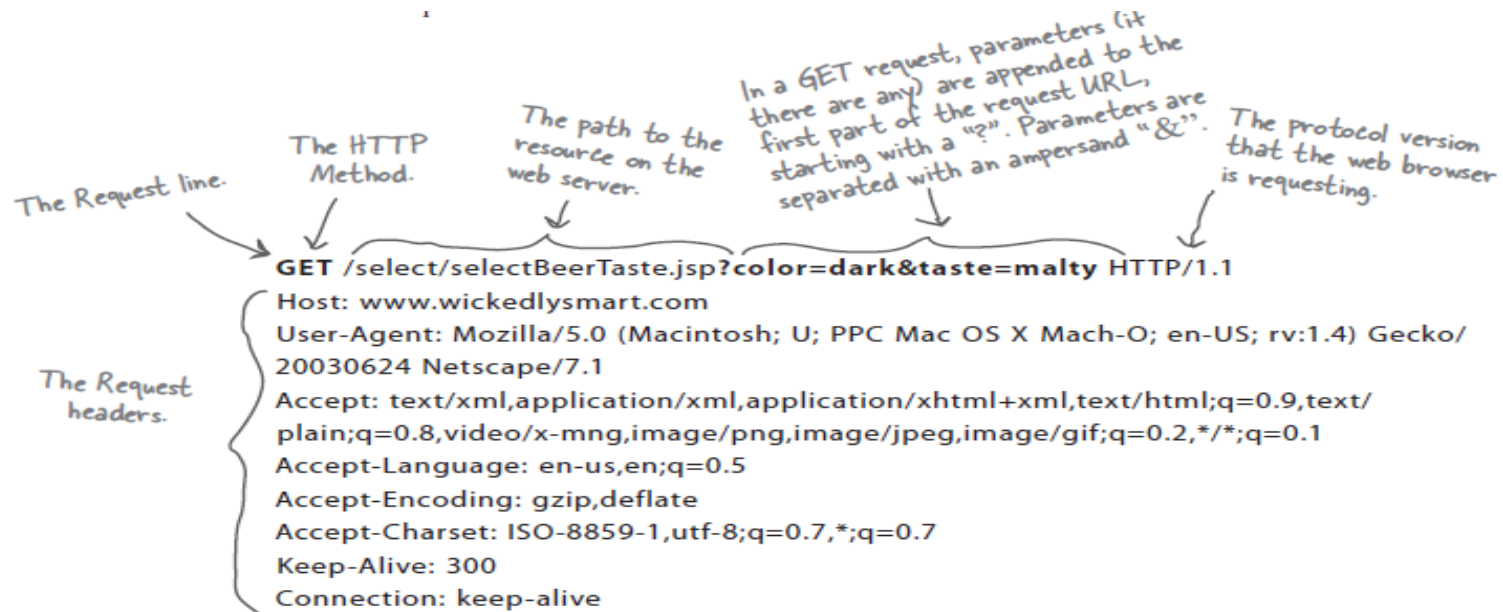
Resumindo

- Os dois métodos mais utilizados pelo Protocolo HTTP para comunicação Cliente e Servidor são: GET e POST.

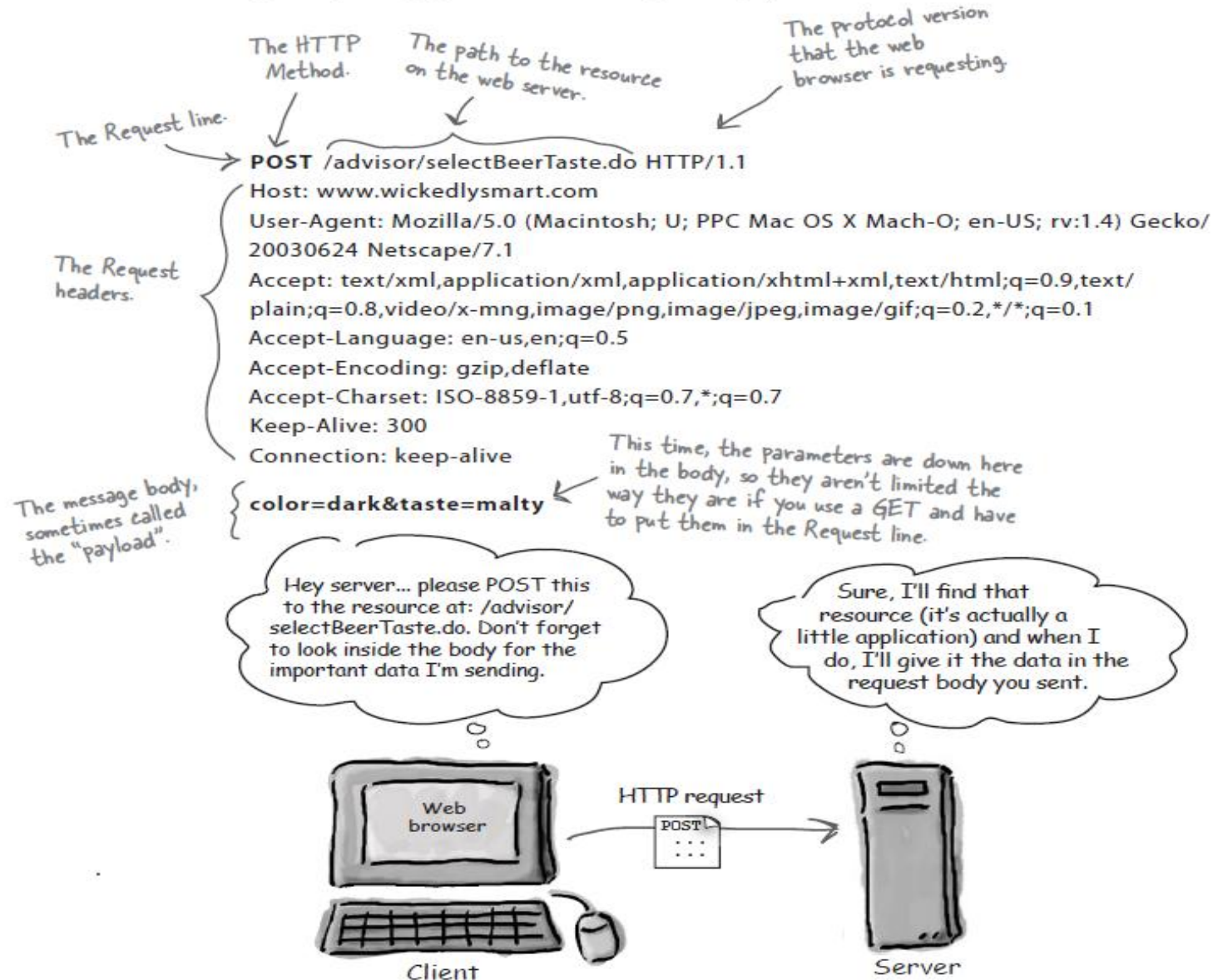
Requisição e Resposta HTTP



Métodos HTTP - GET



Métodos HTTP - POST



Métodos HTTP - REQUISIÇÃO-RESPOSTA

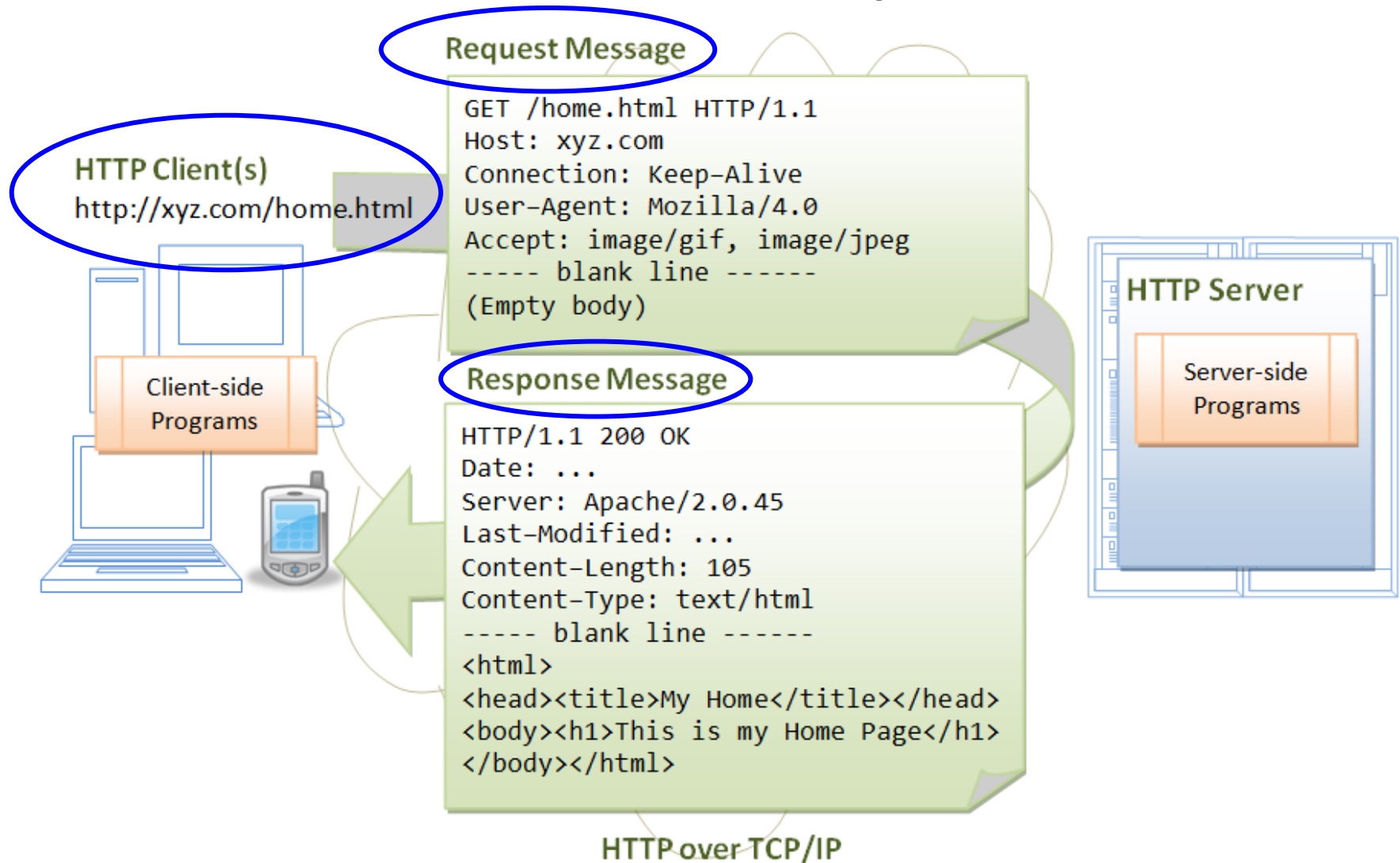


Imagem retirada de www3.ntu.edu.sg

Cliente-Servidor e Métodos HTTP

- Vamos codificar um pequeno exemplo para que possamos entender melhor todo esse processo;
- Criaremos um formulário simples de usuário e senha e enviaremos estes dados para o Servidor;
- Abordaremos nesta codificação:
 - Comunicação Cliente x Servidor
 - Métodos HTTP
- Usaremos as ferramentas abaixo:



Apache Tomcat



**The Apache
Software Foundation**
<http://www.apache.org/>

Perguntas



Página do Professor Mauro:
<http://www.dai.ifma.edu.br/~mlcsilva>

Próxima Aula...

- **Introdução a Servlets**



Referências

- Materiais avulsos da Internet e o Livro Use a Cabeça! Servlets e JSP, Bryan Basham, Kathy Sierra e Bert Bates - Editora Alta Books, 879 páginas.