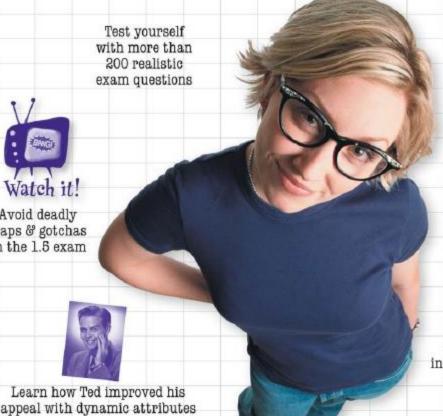
HTTP

HTTP

A Brain-Friendly Guide

Head First Servlets & JSP

Passing the Sun Certified Web Component Developer Exam



Watch it!

Avoid deadly traps & gotchas

on the 1.5 exam

Updated to cover the latest version of the SCWCD exam for J2EE 1.5

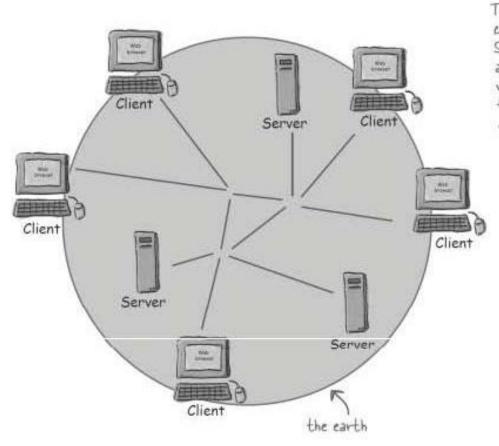


Use c:out to get your message to the world

Fool around in the Custom Tag Library

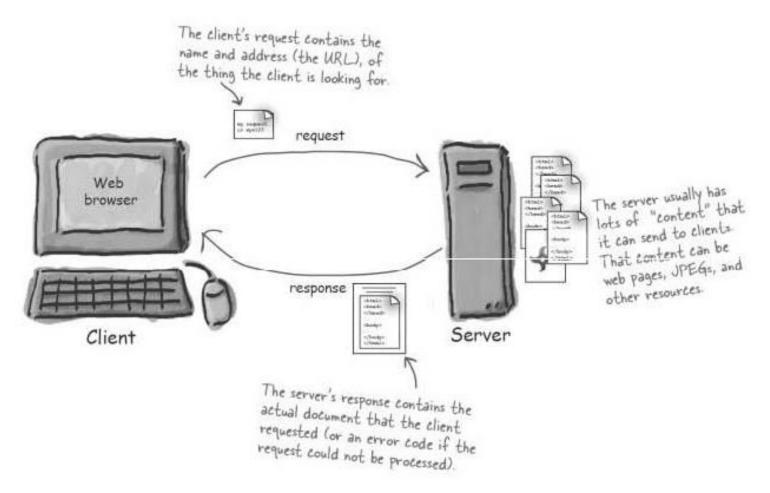


La web



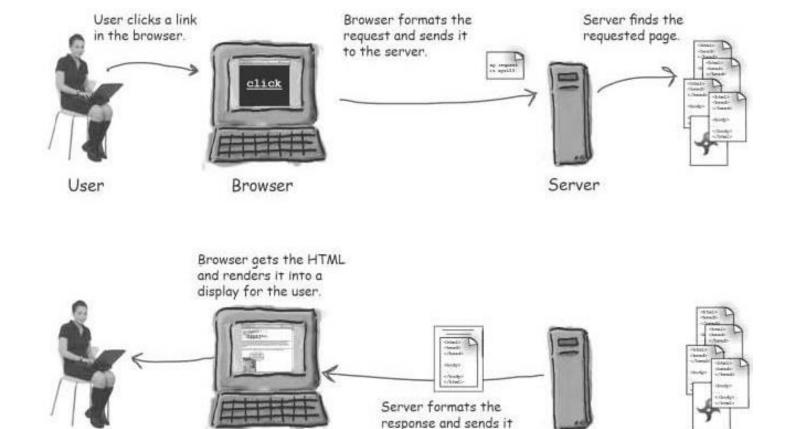
The web consists of gazillions of clients (using browsers like Mozilla or Safari) and servers (using web server apps like Apache) connected through wires and wireless networks. Our goal is to build a web application that clients around the globe can access. And to become obscenely rich.

¿Qué hace un servidor web?



Un servidor web toma una solicitud de un cliente y da algo de regreso al cliente

¿Qué hace un cliente web?



Browser

User

Un cliente web le permite a un usuario solicitar algo al servidor y muestra el usuario el resultado de la solicitud

to the client (browser).

Server

HTML y HTTP

 Los clientes y servidores conocen HTML y HTTP

HTML

 Cuando un servidor responde una solicitud, el servidor usualmente envía algún tipo de contenido al navegador de tal forma que el navegador pueda mostrarlo

HTML

 Los servidores envían normalmente al navegador un conjunto de instrucciones escritas en HTML (HyperText Markup Language)

HTML

- Los servidores envían normalmente al navegador un conjunto de instrucciones escritas en HTML (HyperText Markup Language)
 - EL HTML de dice al navegador como presentar el contenido al usuario

HTTP

- La mayoría de las conversaciones en la web entre los clientes y los servidores son hechas utilizando el protocolo HTTP (HyperText Transfer Protocol) el cual permite conversaciones simples de solicitud-respuesta
 - Cuando el servidor web envía al cliente una página HTML, la envía utilizando HTTP

En resumen hasta este momento ...



HTTP is the protocol clients and servers use on the web to communicate.

The server uses HTTP to send HTML to the client.

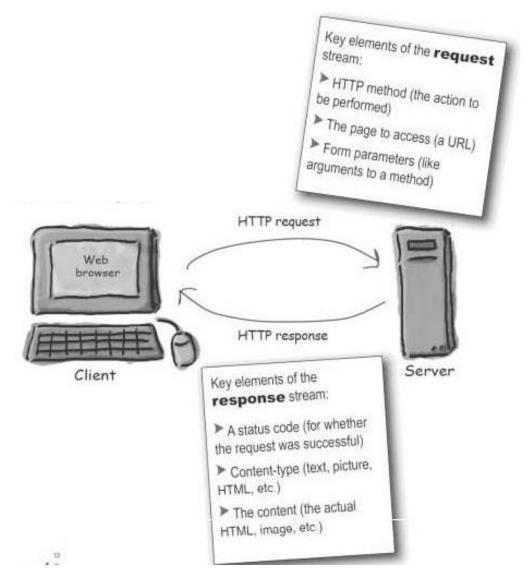
Más del protocolo HTTP

- El protocolo HTTP corre encima de TCP/IP
- TCP es el responsable de asegurarse que un archivo enviado de un nodo de red llega al nodo destino como un archivo completo, aun aunque el archivo sea dividido en fragmentos (paquetes) cuando es enviado
- IP es el protocolo básico que mueve/enruta los paquetes de un host a otro en su camino al destino

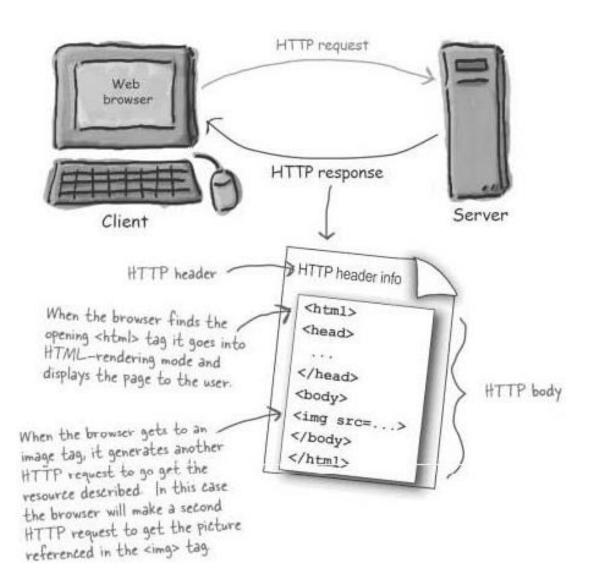
Más del protocolo HTTP

- El protocolo HTTP, es entonces otro protocolo de red el que tiene característica específicas de la web, pero depende de TCP/IP para completar la solicitud y respuesta de un lugar a otro
- La estructura de una conversación HTTP es una secuencia simple Solicitud/Respuesta
 - Un navegador solicita y un servidor responde

Más del protocolo HTTP



HTML es parte de la respuesta HTTP



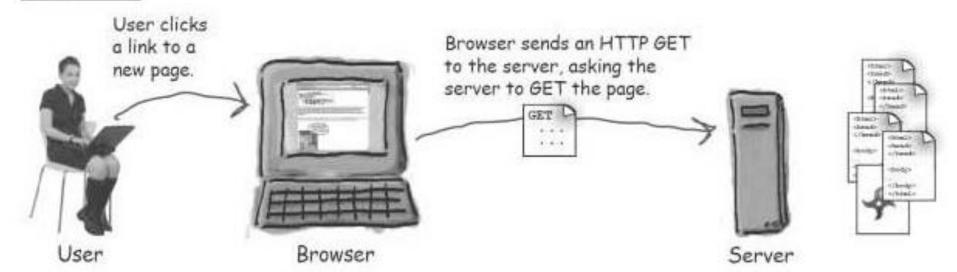
Solicitud

- La primera cosa que se encontrará en la solicitud es el nombre del método HTPP
- El nombre del método le indica al servidor la clase de solicitud que se está haciendo, y como se formateará el resto del mensaje
- El protocolo HTTP tiene varios métodos, pero los que normalmente se utilizan son los métodos GET y POST

Método GET

- El método GET es el más simple método HTPP
 - Su principal función es pedirle al servidor que obtenga un recurso y lo envíe de regreso
 - Este recurso puede ser una página HTML, un archivo JPEG, PDF, etc

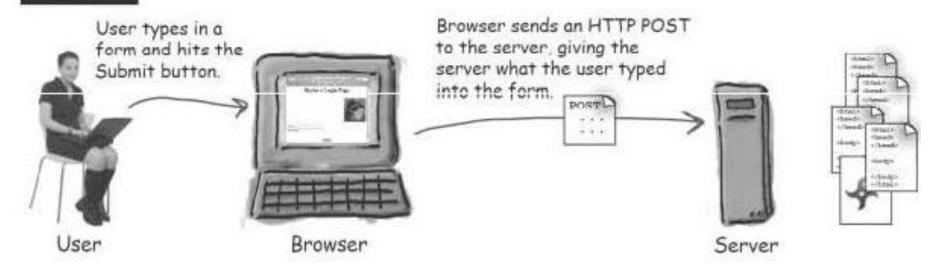
GET



POST

- El método POST es una solicitud más poderosa
 - Con éste método se puede solicitar algo y al mismo tiempo enviar datos de un formulario al servidor
 - Nota: Si los datos son limitados se puede utilizar GET. Ver siguiente lámina

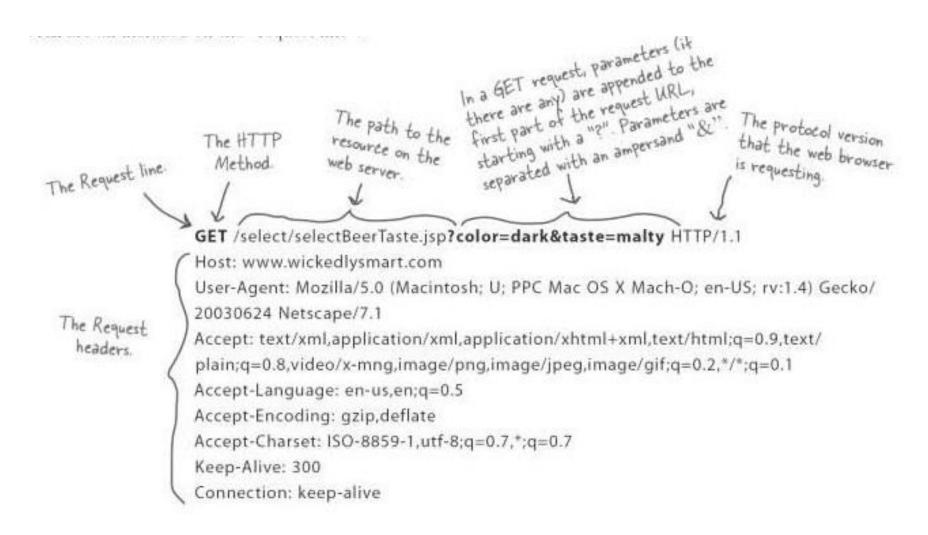
POST



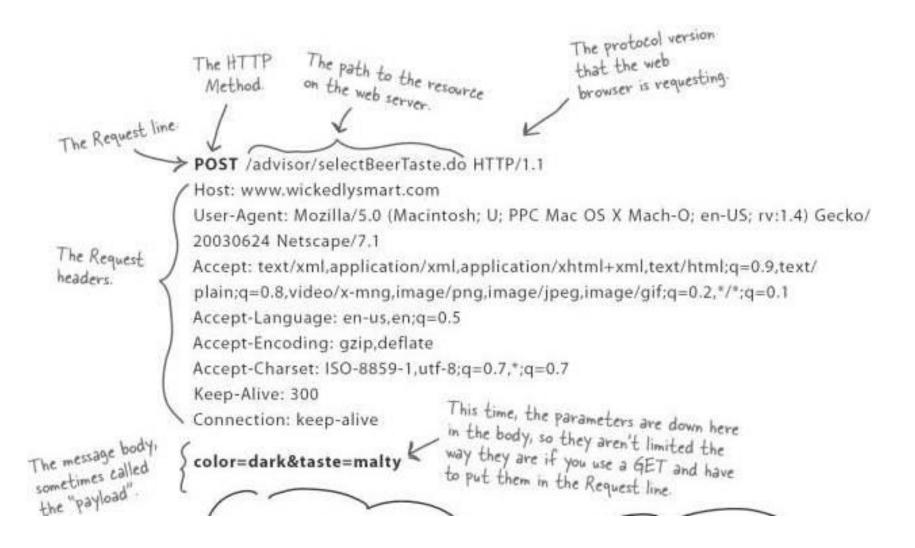
Uso de POST vs GET

- La cantidad total de caracteres en un método GET es muy limitado (dependiendo del servidor)
- El dato enviado con el método GET es insertado al final del URL en la barra del navegador, de tal forma que los datos son mostrados
 - Esto no puede ser adecuado en ciertos casos,
 como el uso de password u otros datos sensibles

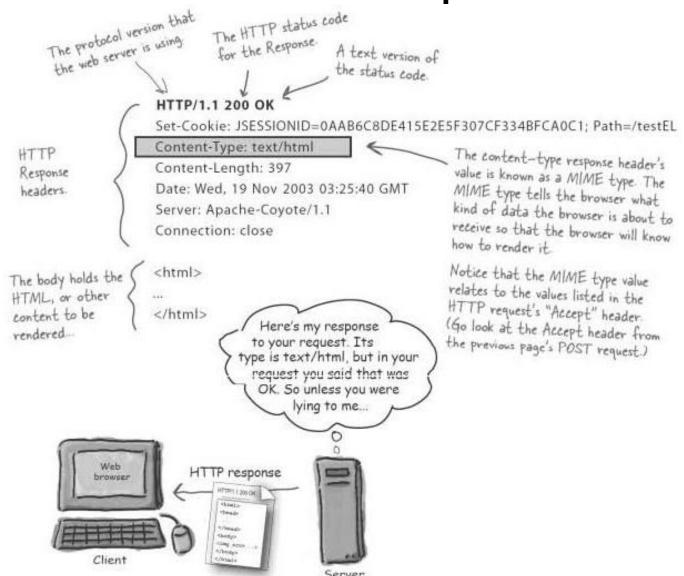
Anatomía de una solicitud HTTP GET

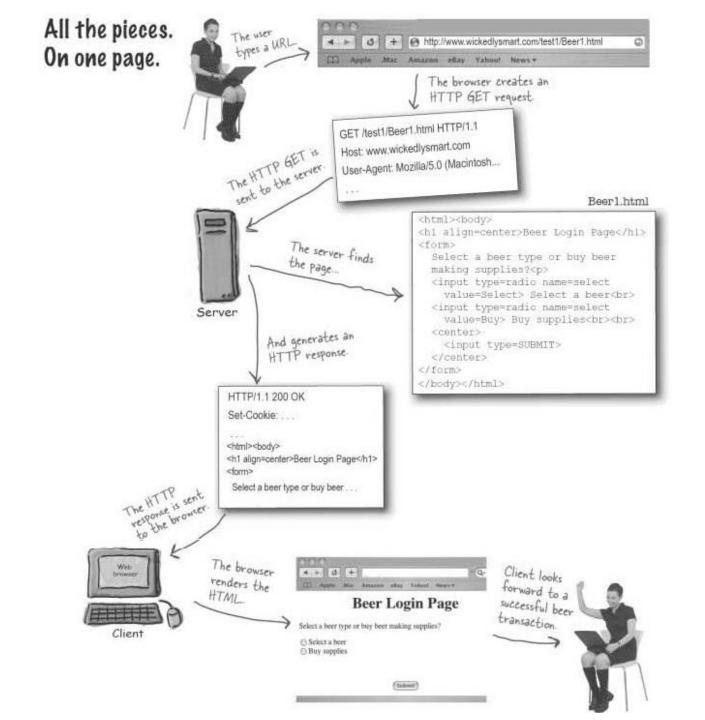


Anatomía de una solicitud HTTP POST



Anatomía de una respuesta HTTP





URL (Uniform Resource Locator)

Protocol: Tells the server which communications protocol (in this case HTTP) will be used. Port: This part of the URL is optional. A single server supports many ports. A server application is identified by a port. If you don't identified by a port in your URL, then specify a port in your URL, and as luck port 80 is the default, and as luck would have it, that's the default port for web servers.

Resource: The name of the content being requested. This could be an HTML page, a servlet, an image, PDF, music, video, or anything else the server feels like serving. If this optional part of the URL is left out, most web servers will look for index.html by default.

http://www.wickedlysmart.com:80/beeradvice/select/beer1.html

Not shown:

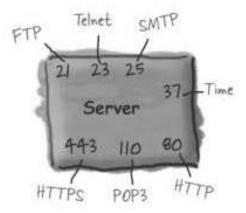
Server: The unique name of the physical server you're looking for. This name maps to a unique IP address. IP addresses are numeric and take the form "xxx. yyy.zzzaaa". You can specify an IP address here instead of a server name, but a server name is a lot easier to remember.

Path: The path to the location, on the server, of the resource being requested. Because most of the early servers on the web ran Unix, Unix syntax is still used to describe the directory hierarchies on the web server.

Cualquier recurso en la web tiene una dirección única, en el formato URL Optional Query String.
Remember, if this was a
GET request, the extra info
(parameters) would be appended
to the end of this URL, starting
with a question mark "?", and with
each parameter (name/value pair)
separated by an ampersand "&".

Puerto TCP

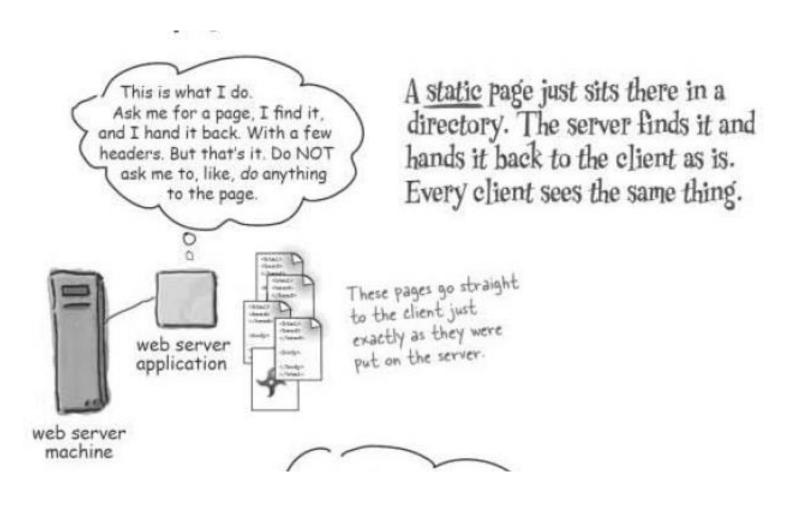
 Un puerto TCP es un número de 16 bits que identifica un programa de software en el hardware del servidor Well-known TCP port numbers for common server applications



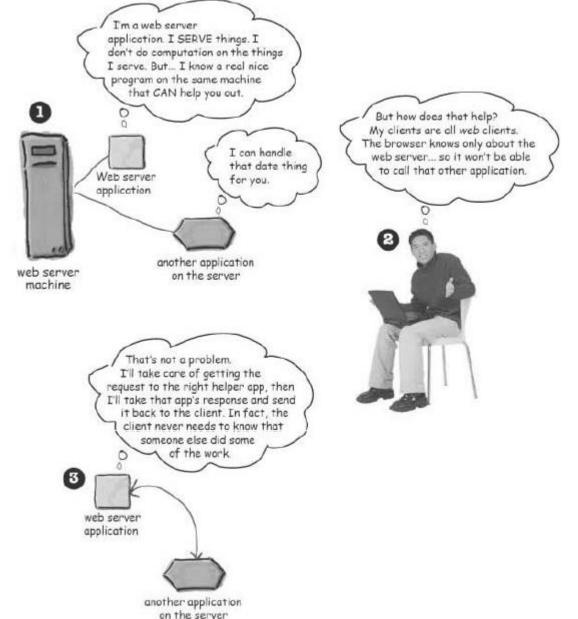
Using one server app per port, a server can have up to 65536 different server apps running.

The TCP port numbers from 0 to 1023 are reserved for well-known services (including the Big One we care aboutport 80). Don't use these ports for your own custom server programs!

Los servidores web adoran servir páginas web estáticas

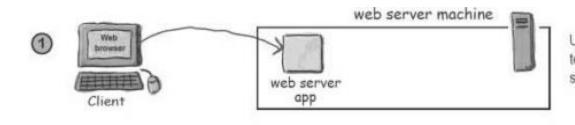


Pero algunas veces es necesario algo más...

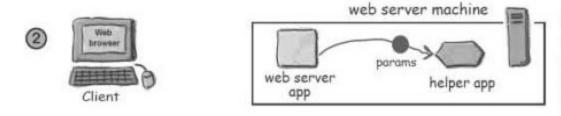


CGI

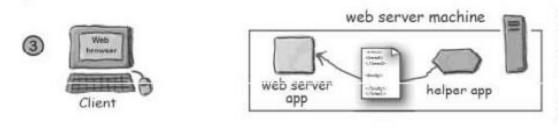
- CGI es el acrónimo de 'Common Gateway Interface'
- La mayoría de los programas CGI son escritos en scripts Perl, pero algunos otros lenguajes pueden trabajar, por ejemplo, C, Phyton y PHP
- En la siguiente lámina se muestra como un programa CGI trabaja para una página web dinámica



User clicks a link that has a URL to a CGI program instead of a static page.

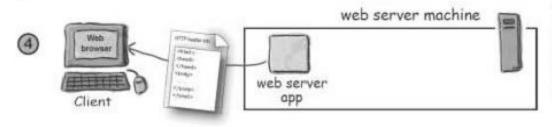


Web server application "sees" that the request is for a helper program, so the web server launches and runs the program. The web server app sends along any parameters from a GET or POST.



The helper app constructs the brand new page (that has the current date inserted) and sends the HTML back to the server.

As far as the web server is concerned, the HTML from the helper app is a static page.



The helper application is shut down, and the client gets back an HTML page that has the current date as part of its now-static content.