Head.First.Servlet.Jsp

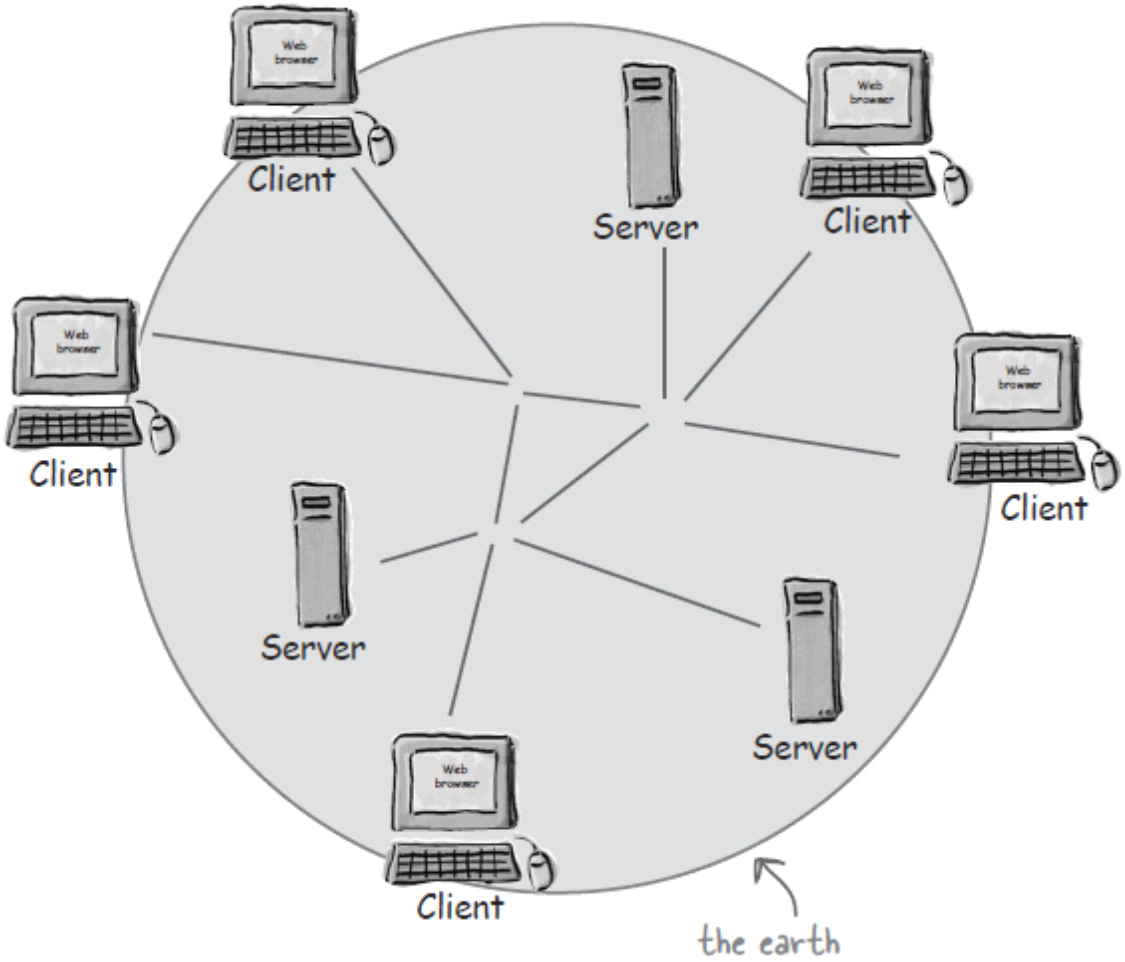
01Why Servlet and Jsp?

Table of Contents

Table of Contents

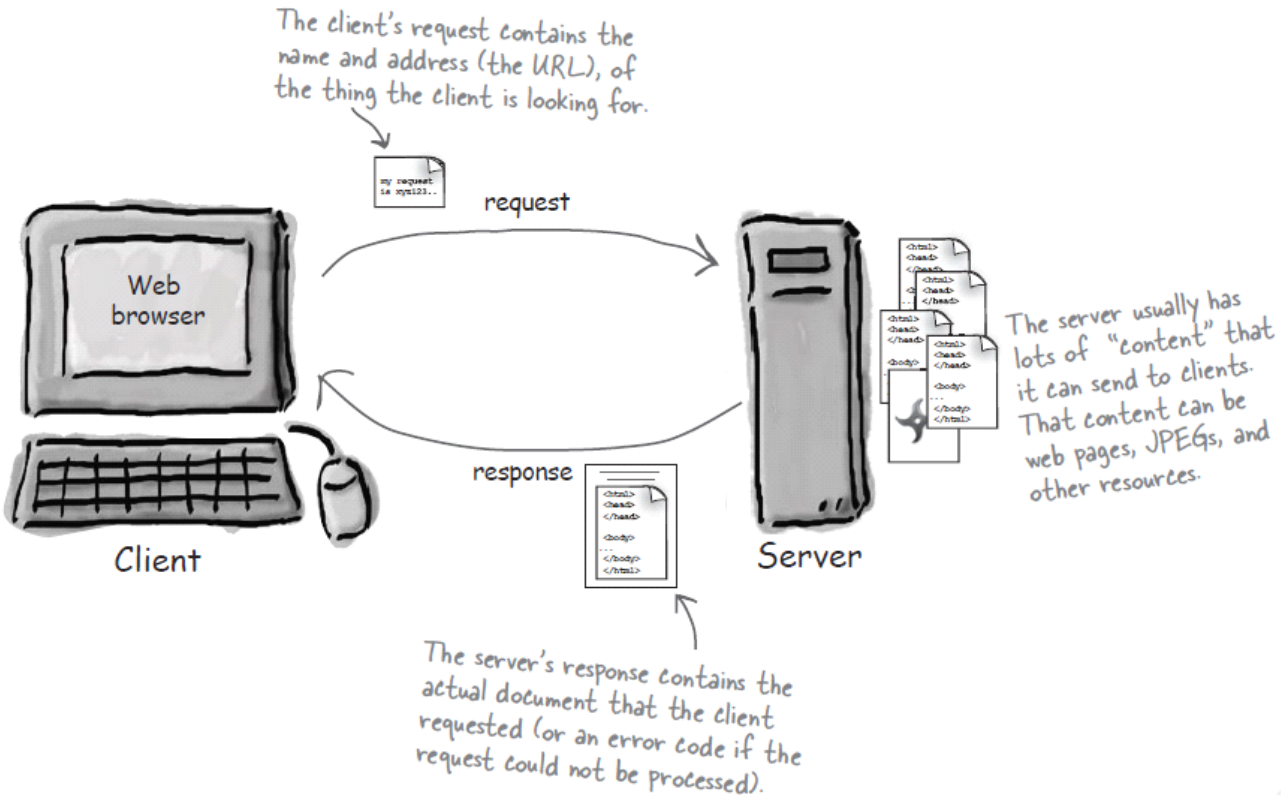
Why use Servlets & JSPs: *an introduction*

- What web servers and clients do, and how they talk?
- What is the HTTP protocol?
- Anatomy of HTTP GET and POST requests and HTTP responses
- Locating web pages using URLs
- Web servers, static web pages, and CGI
- Servlets Demystified: write, deploy, and run a servlet
- JSP is what happened when somebody introduced Java to HTML



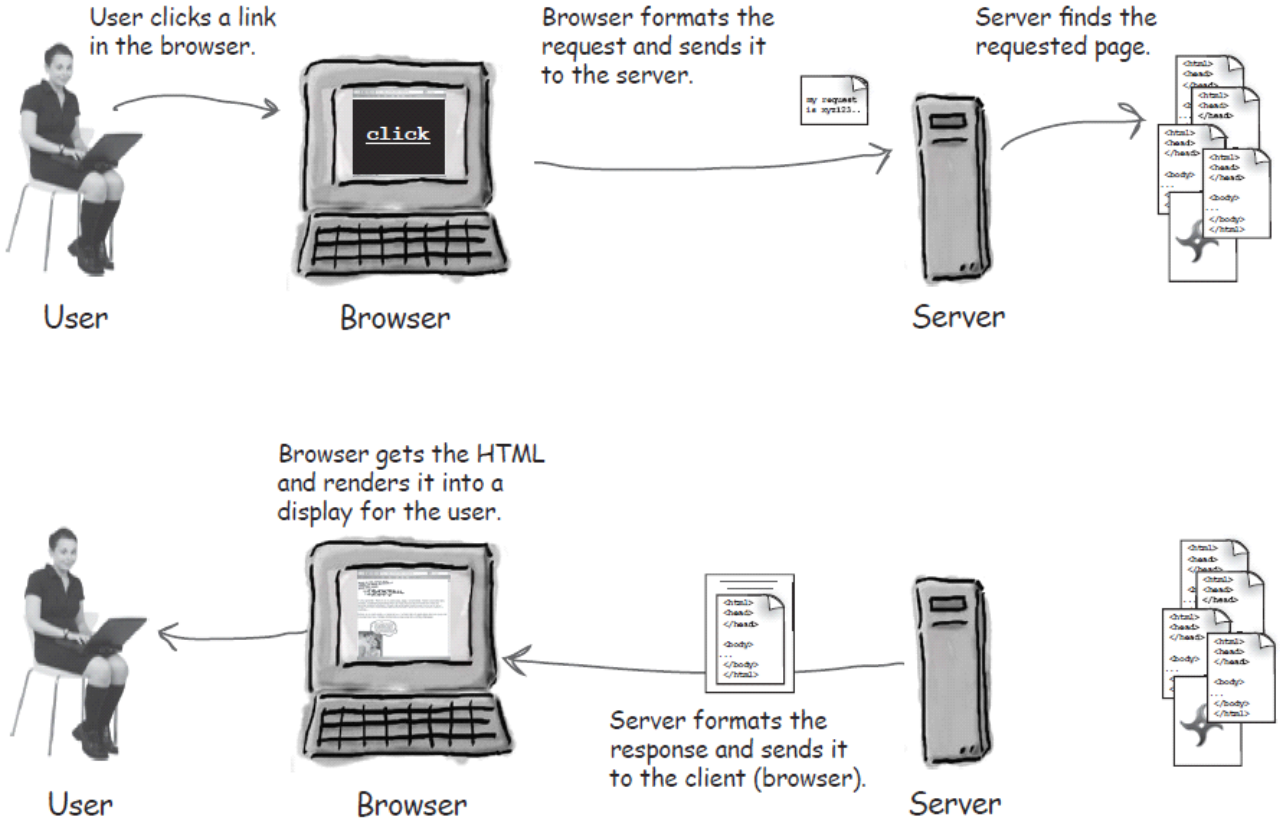
What does your web server do?

- **A web server takes a client request and gives something back to the client.**



What does a web client do?

- **A web client lets the user request something on the server, and shows the user the result of the request.**



Why use Servlets & JSPs: *an introduction*

- What web servers and clients do, and how they talk?
- What is the HTTP protocol?
- Anatomy of HTTP GET and POST requests and HTTP responses
- Locating web pages using URLs
- Web servers, static web pages, and CGI
- Servlets Demystified: write, deploy, and run a servlet
- JSP is what happened when somebody introduced Java to HTML

Clients and servers know HTML and HTTP

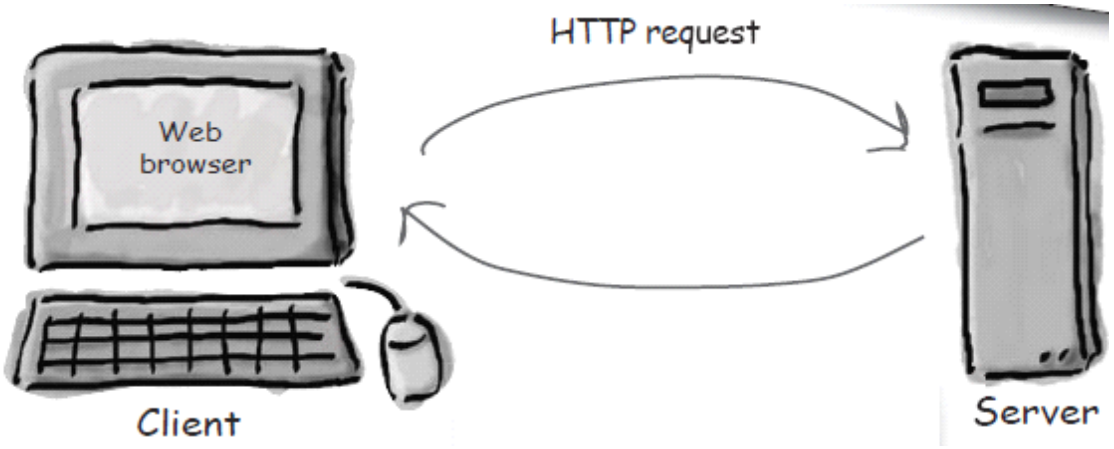
- **HTML tells the browser how to display the content to the user.**
- **HTTP is the protocol clients and servers use on the web to communicate.**
- **The server uses HTTP to send HTML to the client.**

What is the HTTP protocol?

- HTTP runs on top of TCP/IP.
- HTTP, then, is another network protocol that has Web-specific features, but it depends on TCP/IP to get the complete request and response from one place to another.
- The structure of an HTTP conversation is a simple **Request/ Response** sequence; a browser *requests*, and a server *responds*.

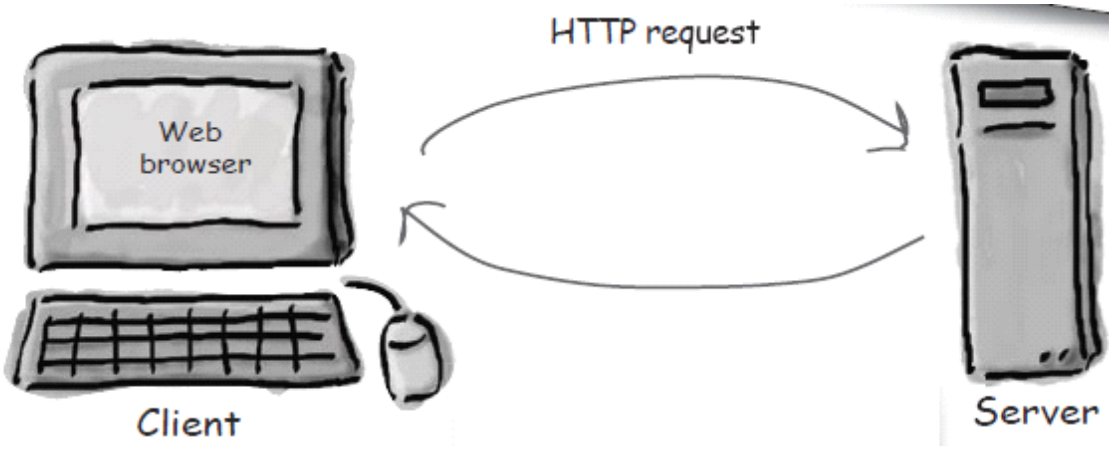
Key elements of the request

- HTTP method
- The page to access(URL)
- Form parameter(like arguments to a method)

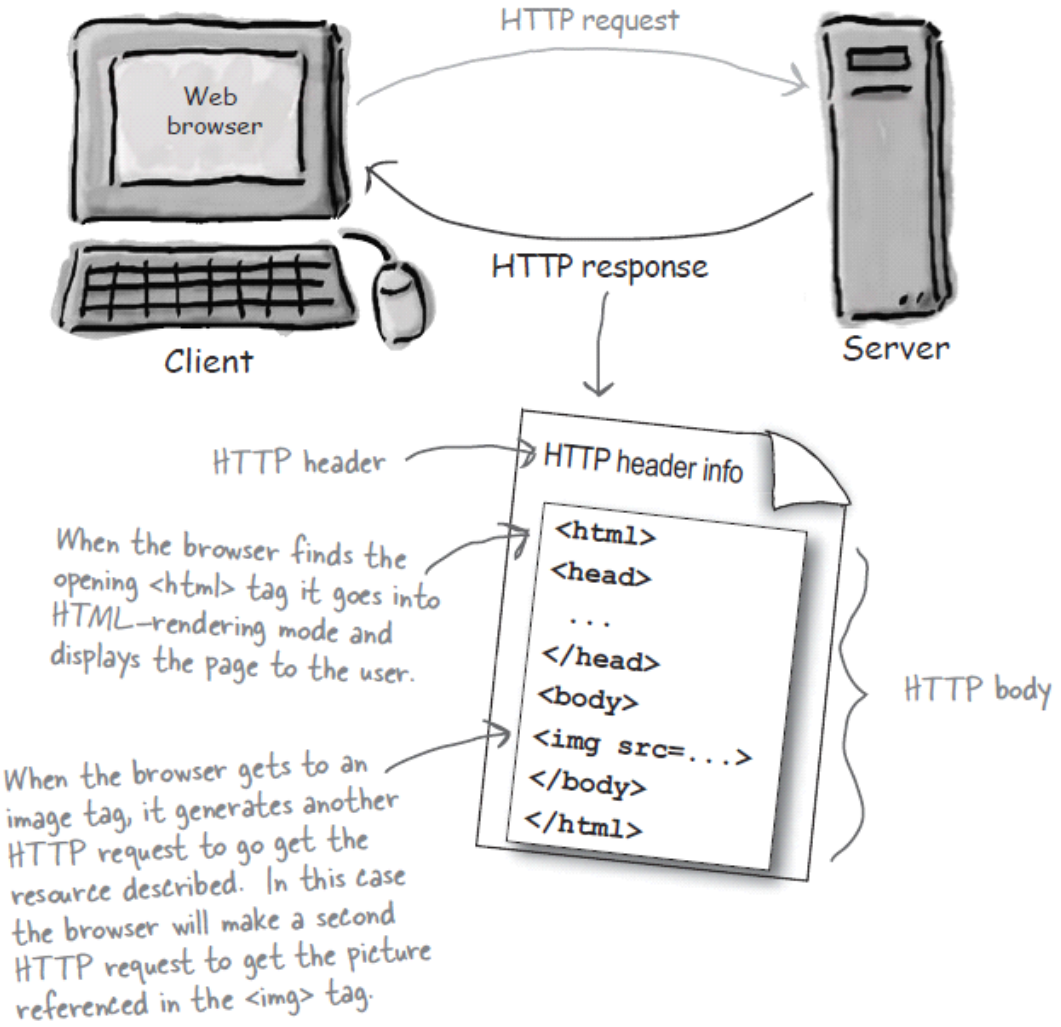


Key elements of the response

- A status code(for whether the request was successful)
- Content-type(text,pic,HTML,etc.)
- The content(the actual HTML,image,etc.)



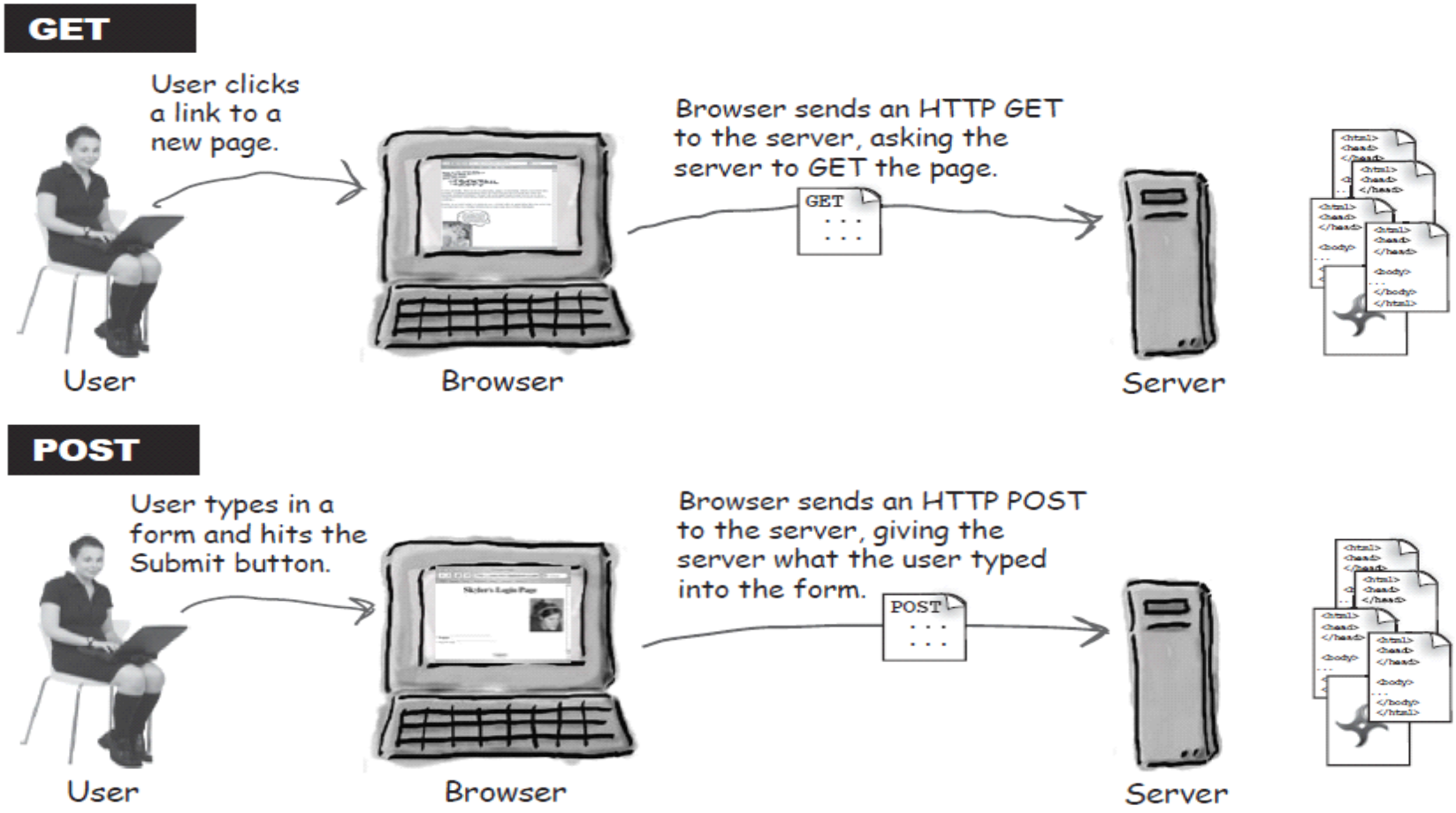
HTML is part of the HTTP response



Why use Servlets & JSPs: *an introduction*

- What web servers and clients do, and how they talk?
- What is the HTTP protocol?
- Anatomy of HTTP GET and POST requests and HTTP responses
- Locating web pages using URLs
- Web servers, static web pages, and CGI
- Servlets Demystified: write, deploy, and run a servlet
- JSP is what happened when somebody introduced Java to HTML

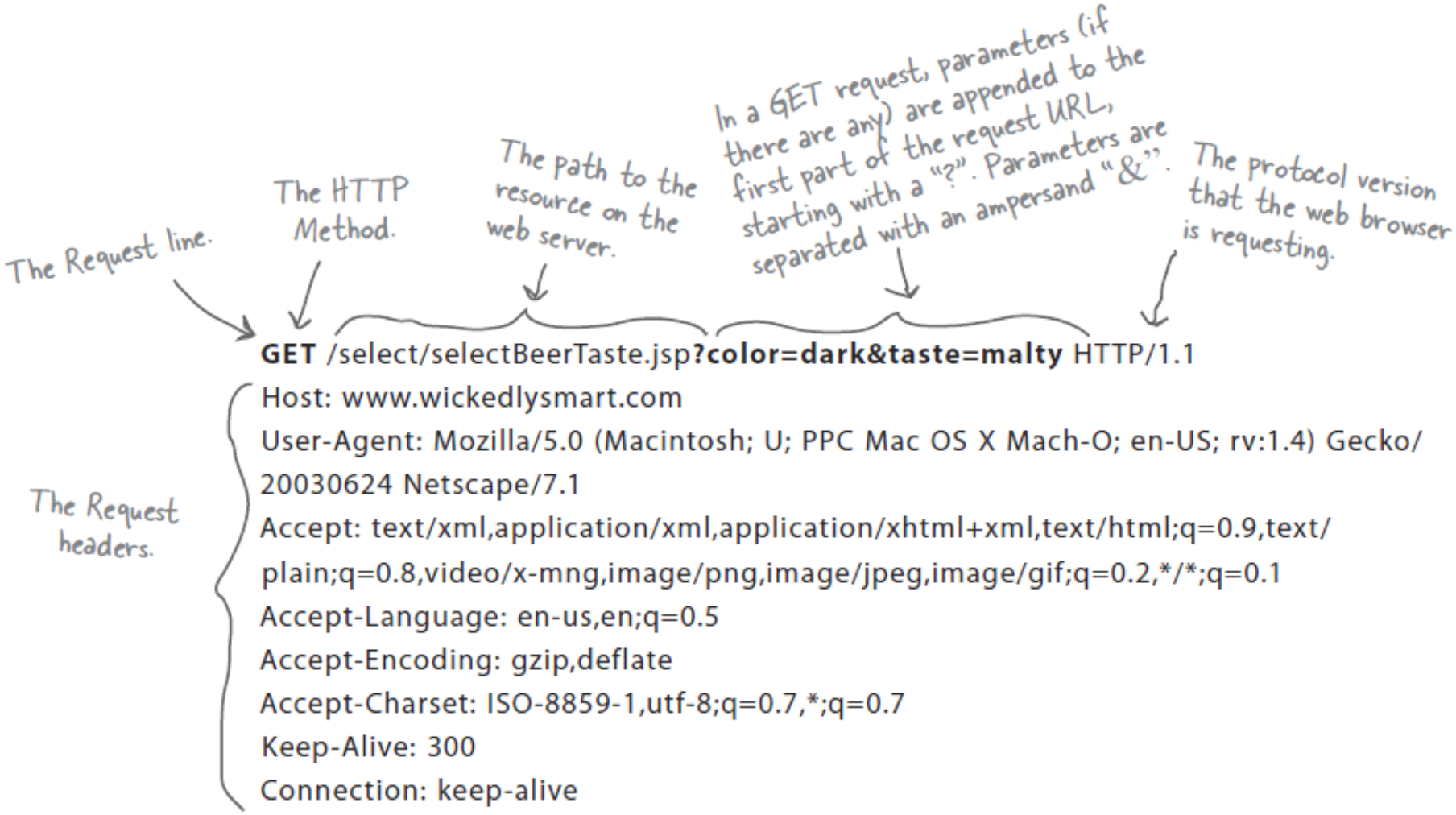
HTTP *method*



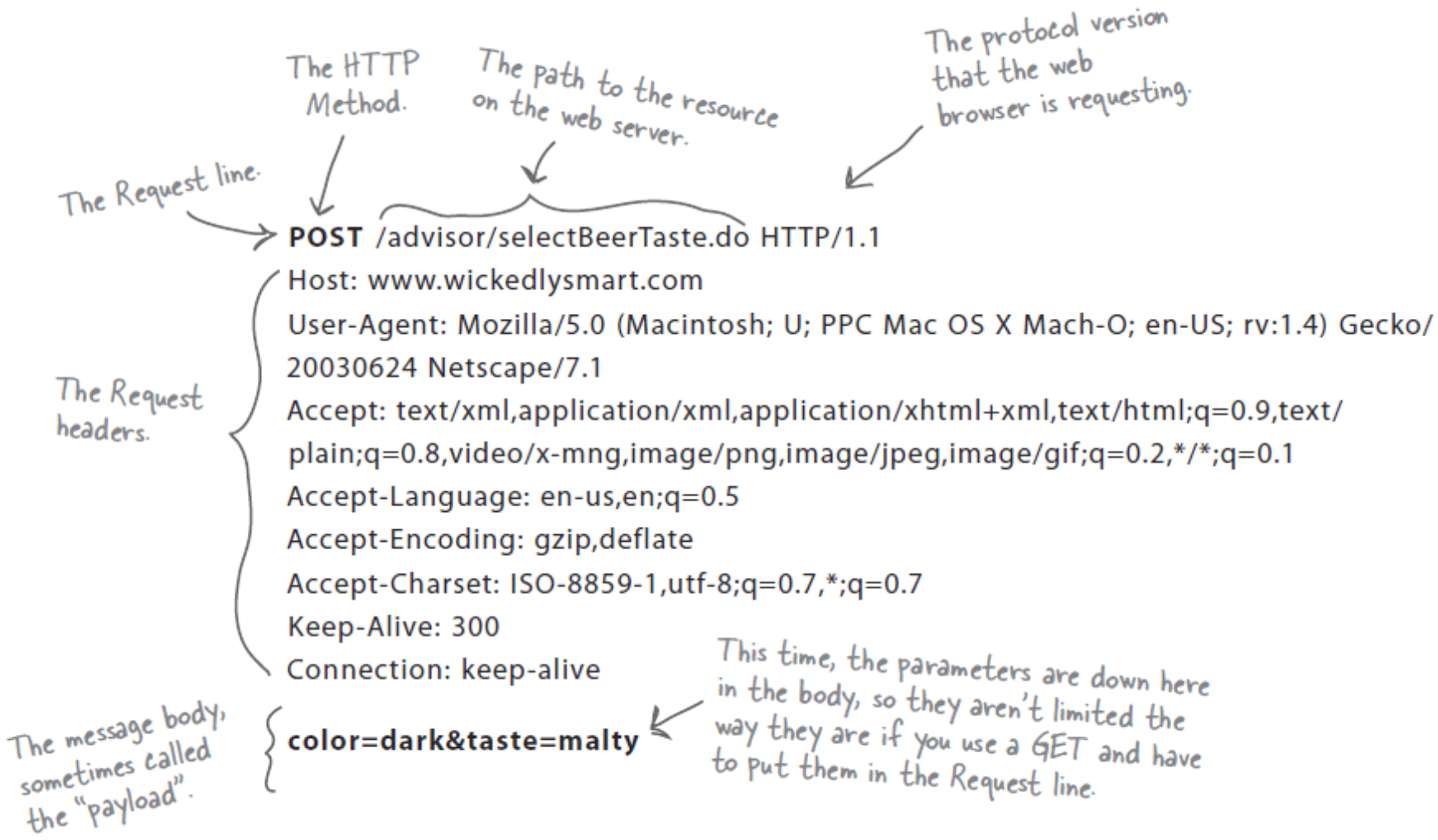
you can send a little data with HTTP GET

- The total amount of characters in a GET is really limited (depending on the server).
- The data you send with the GET is appended to the URL up in the browser bar, so whatever you send is exposed. Better not put a password or some other sensitive data as part of a GET!

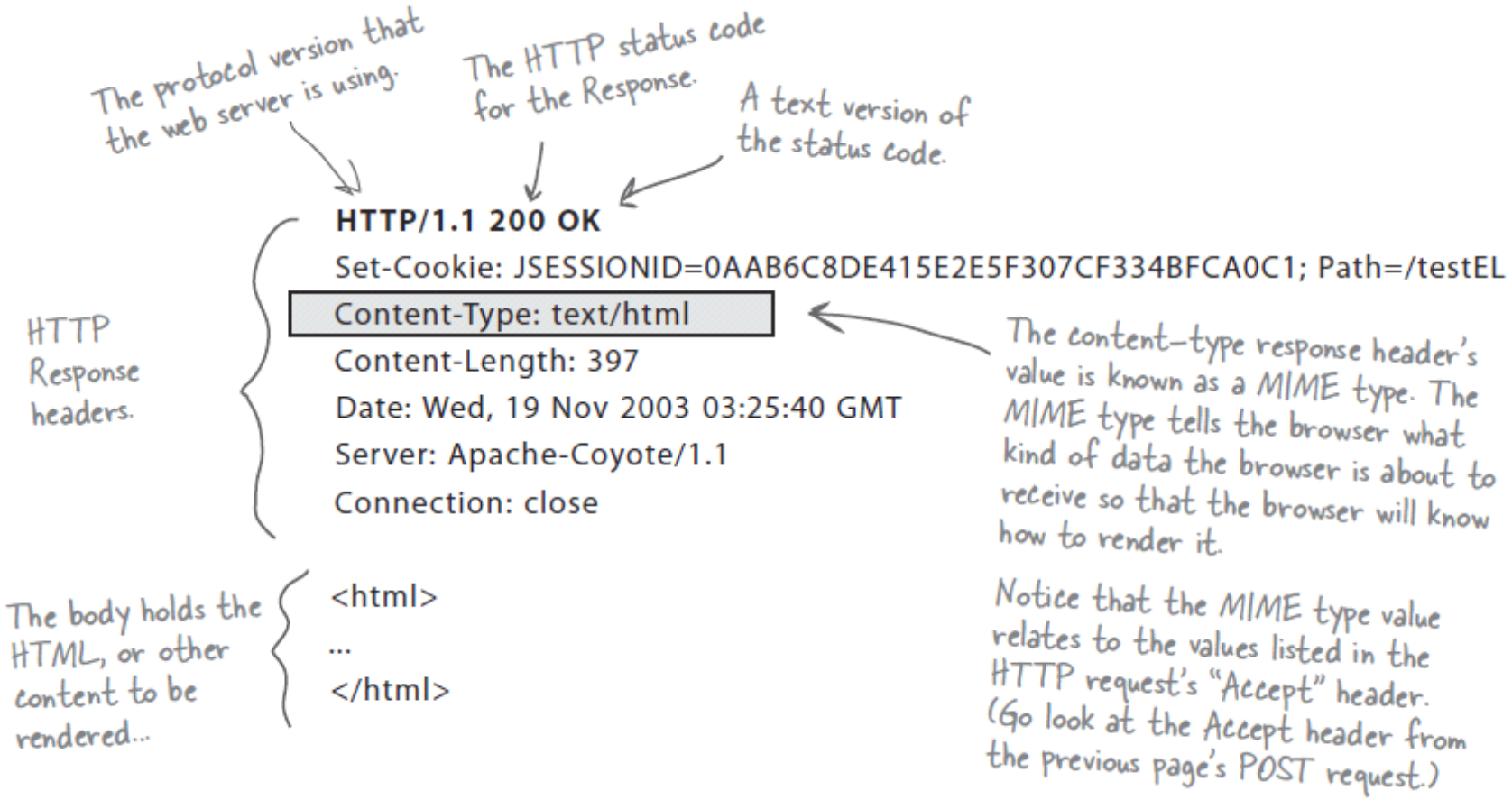
Anatomy of an HTTP GET request

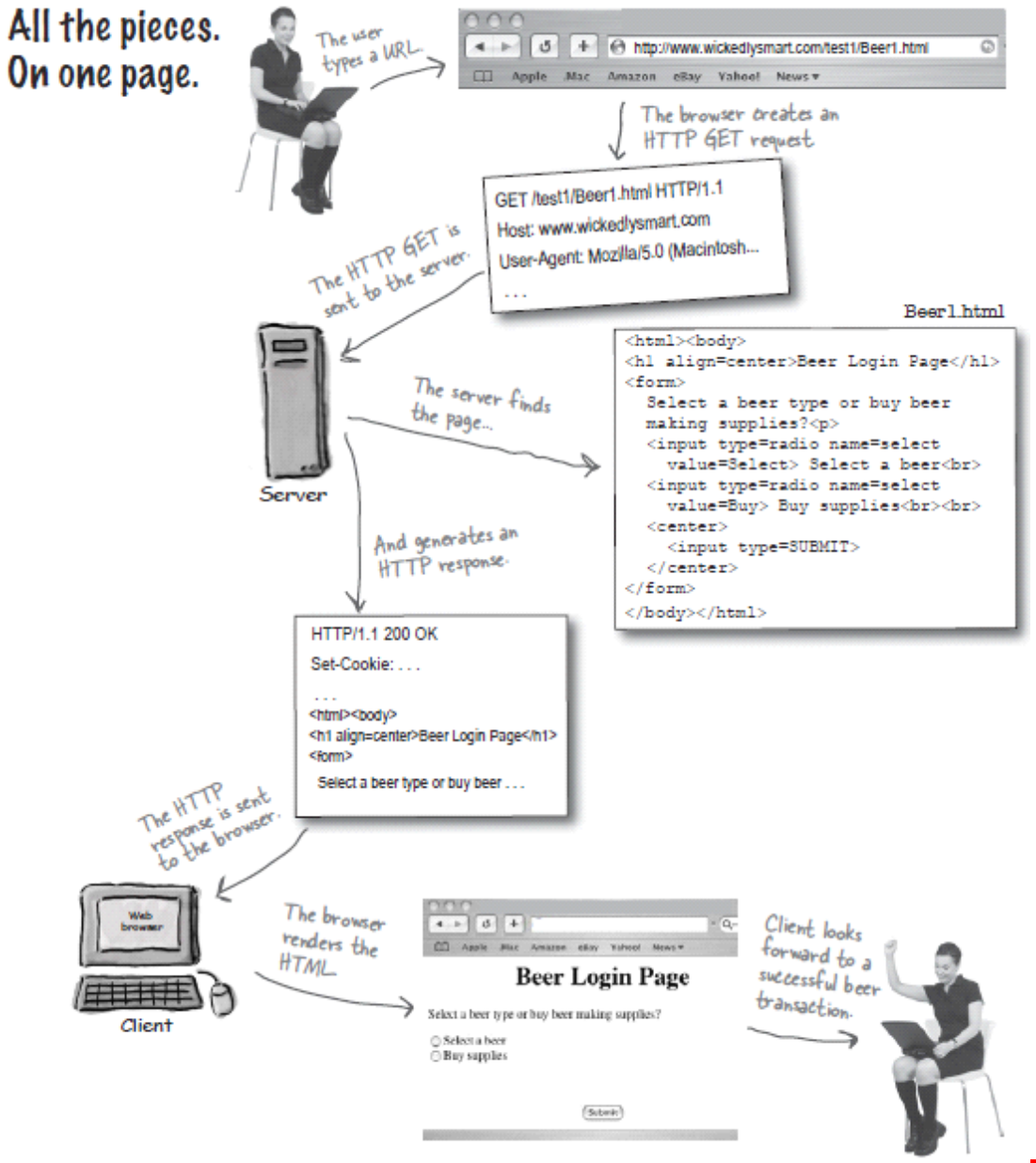


Anatomy of an HTTP POST request



Anatomy of an HTTP response

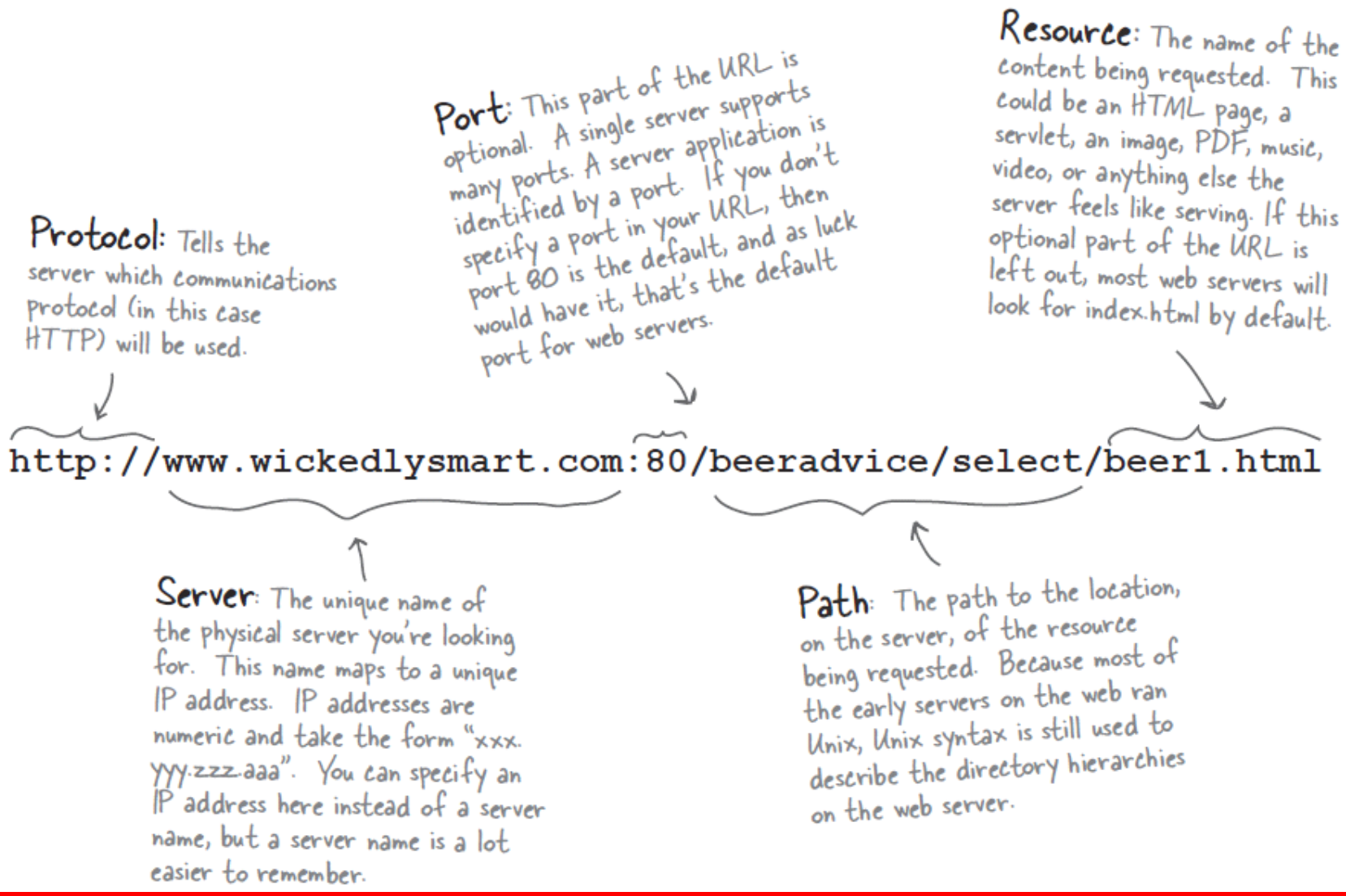




Why use Servlets & JSPs: *an introduction*

- What web servers and clients do, and how they talk?
- What is the HTTP protocol?
- Anatomy of HTTP GET and POST requests and HTTP responses
- Locating web pages using URLs
- Web servers, static web pages, and CGI
- Servlets Demystified: write, deploy, and run a servlet
- JSP is what happened when somebody introduced Java to HTML

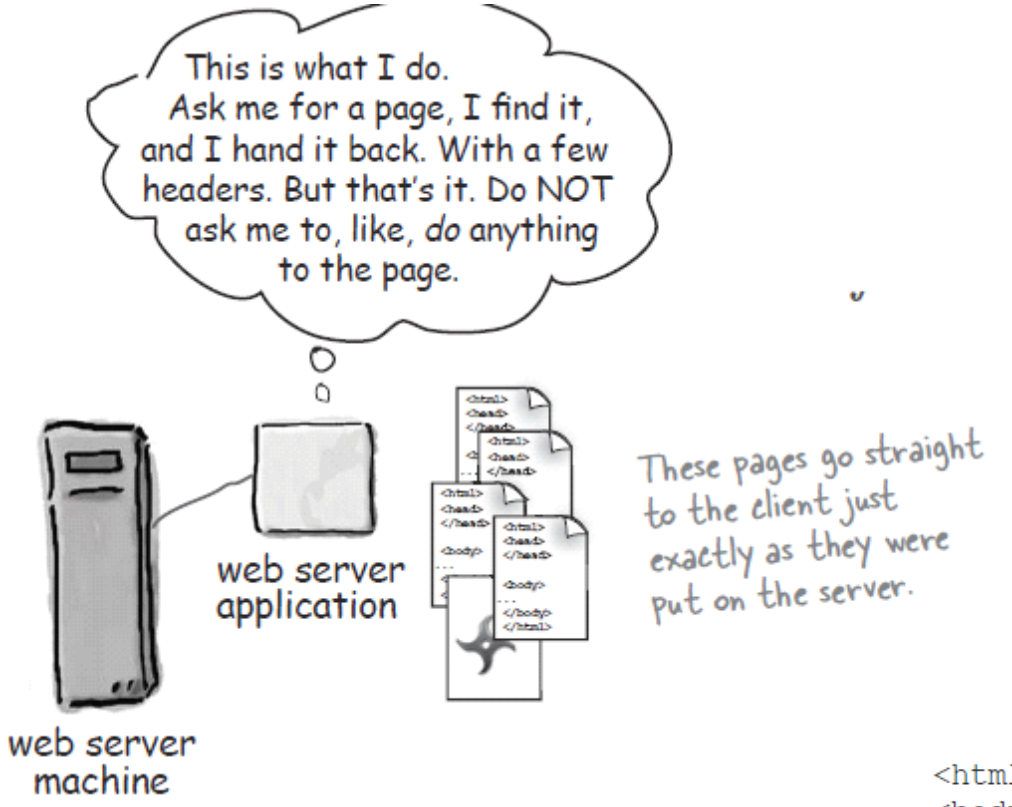
Uniform Resource Locators



Why use Servlets & JSPs: *an introduction*

- What web servers and clients do, and how they talk?
- What is the HTTP protocol?
- Anatomy of HTTP GET and POST requests and HTTP responses
- Locating web pages using URLs
- Web servers, static web pages, and CGI
- Servlets Demystified: write, deploy, and run a servlet
- JSP is what happened when somebody introduced Java to HTML

Web servers love serving static web pages



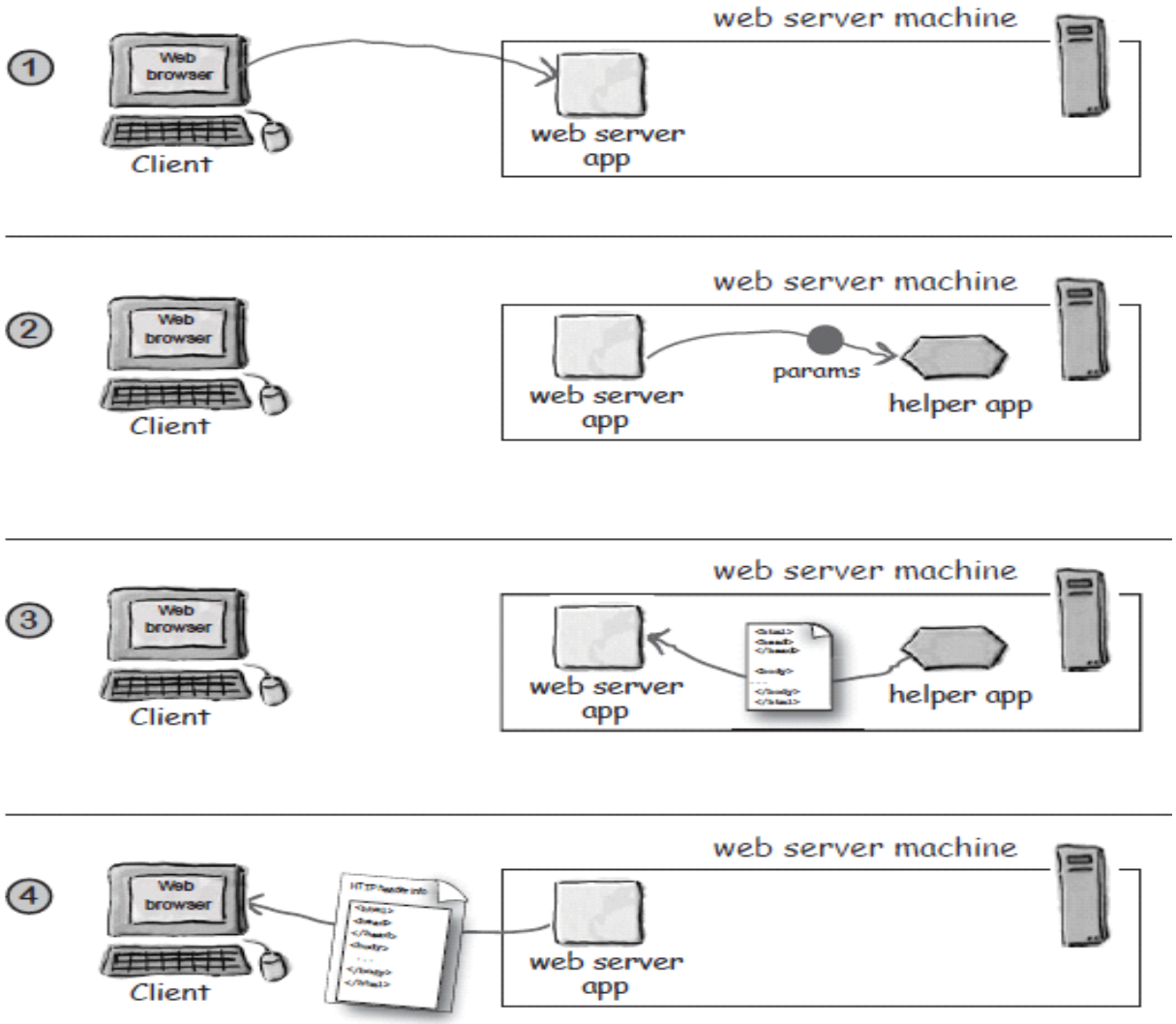
What if we want to stick something variable inside the HTML page?

```
<html>
<body>
The current time is [insertTimeOnServer].
</body>
</html>
```

Two things the web ser ver alone won't do

- **Dynamic content**
- **Saving data on the server**

CGI



Servlets

- Servlets and CGI both play the role of a helper app in the web server
- A Servlet can generate HTML **dynamically**

Why use Servlets & JSPs: *an introduction*

- What web servers and clients do, and how they talk?
- What is the HTTP protocol?
- Anatomy of HTTP GET and POST requests and HTTP responses
- Locating web pages using URLs
- Web servers, static web pages, and CGI
- Servlets Demystified: write, deploy, and run a servlet
- JSP is what happened when somebody introduced Java to HTML

A Servlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class Ch1Servlet extends HttpServlet {

  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
                    throws IOException {

    PrintWriter out = response.getWriter();
    java.util.Date today = new java.util.Date();
    out.println("<html> " +
                "<body>" +
                "<h1 align=center>HF\'s Chapter1 Servlet</h1>"
                + "<br>" + today +  "</body>" + "</html>");
  }
}
```

Standard servlet declarations (there will be about 400 pages describing this stuff).

HTML embedded in a Java program. Looks lovely, doesn't it?

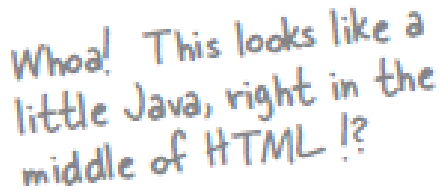
Why use Servlets & JSPs: *an introduction*

- What web servers and clients do, and how they talk?
- What is the HTTP protocol?
- Anatomy of HTTP GET and POST requests and HTTP responses
- Locating web pages using URLs
- Web servers, static web pages, and CGI
- Servlets Demystified: write, deploy, and run a servlet
- JSP is what happened when somebody introduced Java to HTML

Introduced Java to HTML

```
<html>
<body>
<h1>Skyler's Login Page</h1>
<br>
<%= new java.util.Date() %>
</body>
</html>
```

skylerlogin.jsp



BULLET POINTS

- HTTP stands for Hypertext Transfer Protocol, and is the network protocol used on the Web. It runs on top of TCP/IP.
- HTTP uses a request/response model—the client makes an HTTP request, and the web server gives back an HTTP response that the browser then figures out how to handle (depending on the content type of the response)
- If the response from the server is an HTML page, the HTML is added to the HTTP response.

BULLET POINTS

- An HTTP request includes the request URL (the resource the client is trying to access), the HTTP method (GET, POST, etc.), and (optionally) form parameter data (also called the “query string”).
- An HTTP response includes a status code, the content-type (also known as MIME type), and the actual content of the response (HTML, image, etc.)
- A GET request appends form data to the end of the URL.
- A POST request includes form data in the body of the request.

BULLET POINTS

- A MIME type tells the browser what kind of data the browser is about to receive so that the browser will know what to do with it (render the HTML, display the graphic, play the music, etc.)
- URL stands for Uniform Resource Locator. Every resource on the web has its own unique address in this format. It starts with a protocol, followed by the server name, an optional port number, and usually a specific path and resource name. It can also include an optional query string, if the URL is for a GET request.

BULLET POINTS

- Web servers are good at serving static HTML pages, but if you need dynamically-generated data in the page (the current time, for example), you need some kind of helper app that can work with the server. The non-Java term for these helper apps (most often written in Perl) is CGI (which stands for Common Gateway Interface).
- Putting HTML inside a println() statement is ugly and error-prone, but JSPs solve that problem by letting you put Java into an HTML page rather than putting HTML into Java code.