



UNIVERSIDADE
ESTADUAL DE LONDRINA

Introdução a Servlets

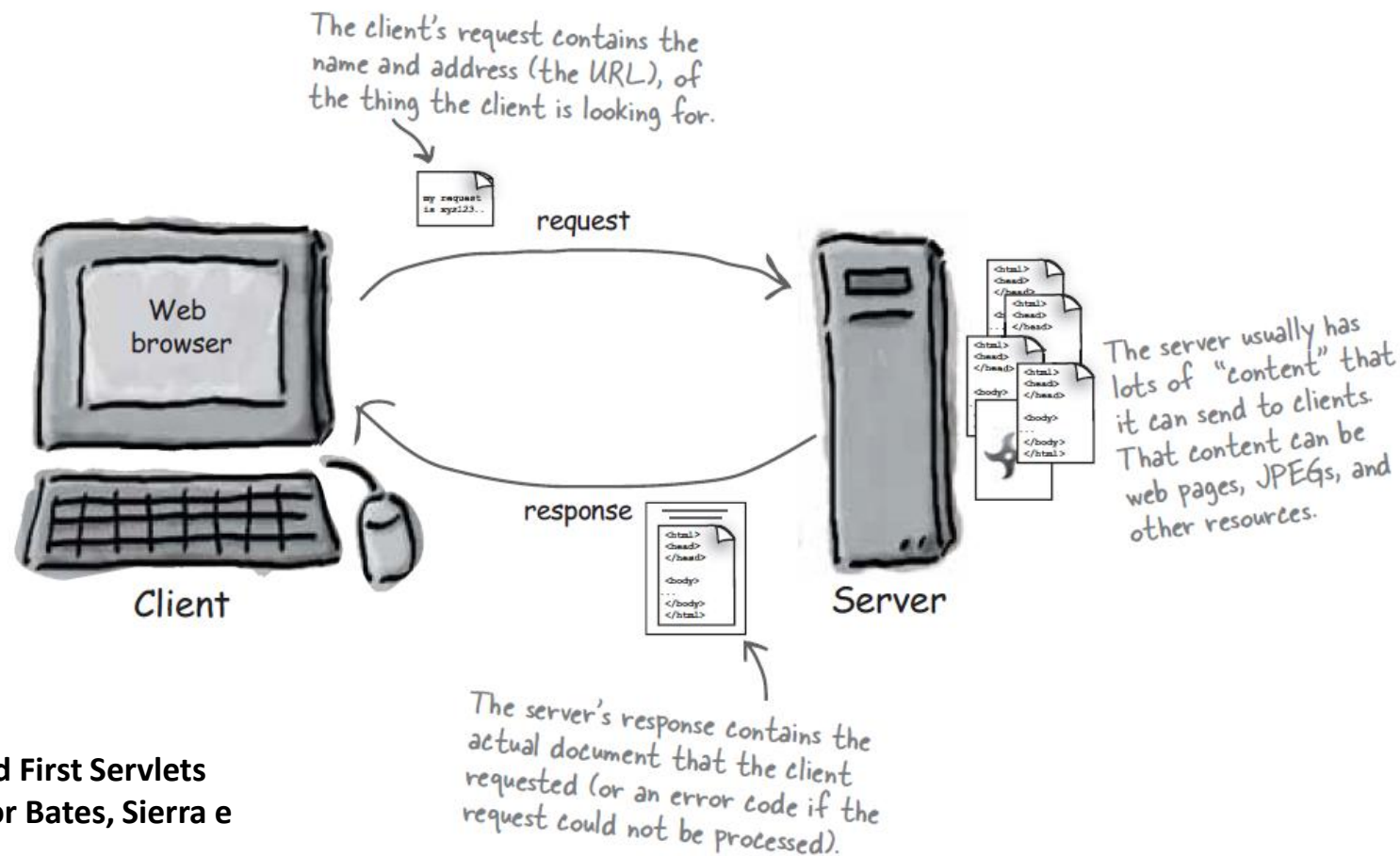
Laboratório de Programação (5COP011)

Prof. Bruno Bogaz Zarpelão

Objetivos

- Conceitos básicos:
 - interação cliente-servidor;
 - HTTP;
 - páginas estáticas x páginas dinâmicas;
 - Servlets
- Primeiro servlet.

Interação cliente servidor



Fonte: Head First Servlets
and JSPs por Bates, Sierra e
Basham

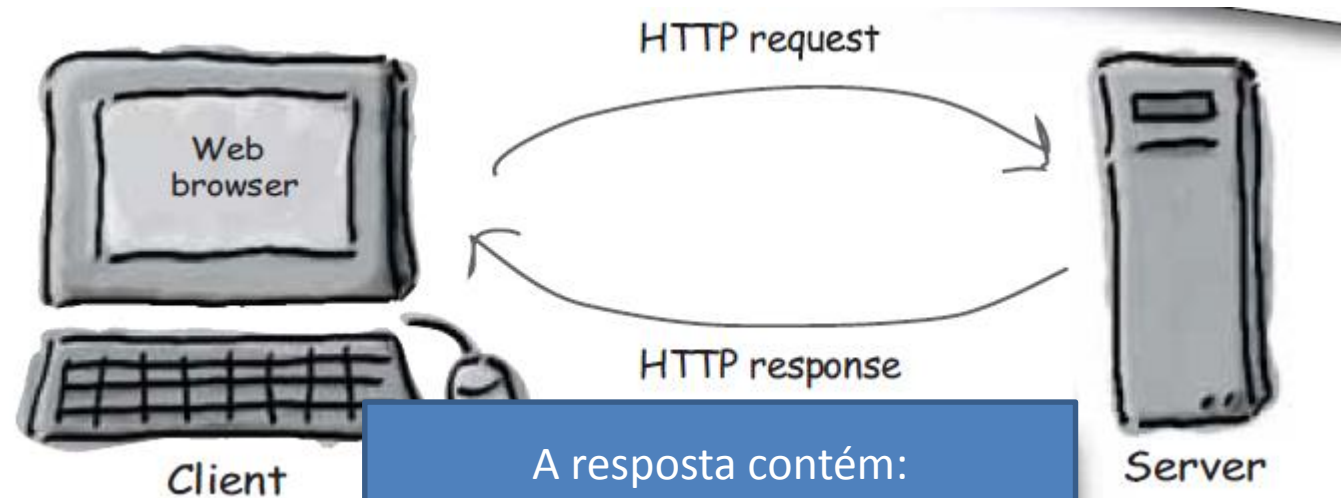
Interação cliente servidor

- O cliente e o servidor falam dois “idiomas”:
 - HTML: o conteúdo retornado pelo servidor é composto por um conjunto de instruções em HTML;
 - HTTP: tanto as requisições quanto as respostas são sempre escritas no formato do HTTP;

HTTP

A requisição contém:

- método HTTP (get, post, etc)
- Objeto a ser acessado (URL)
- parâmetros

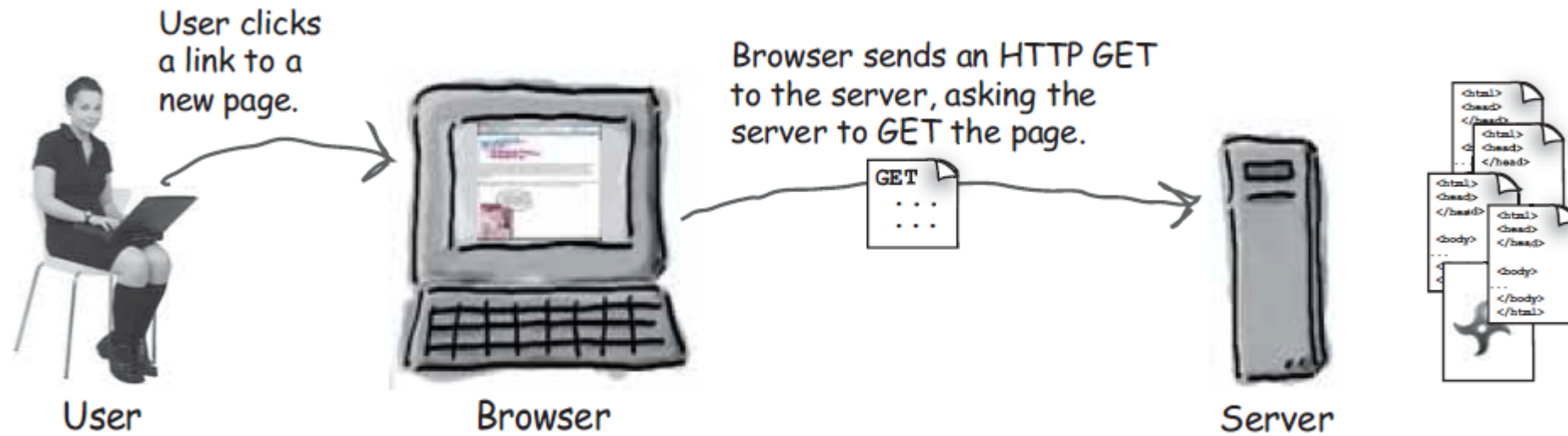


A resposta contém:

- Código de resposta (ex.: 404)
- Tipo de conteúdo (texto, fotos, HTML)
- O conteúdo

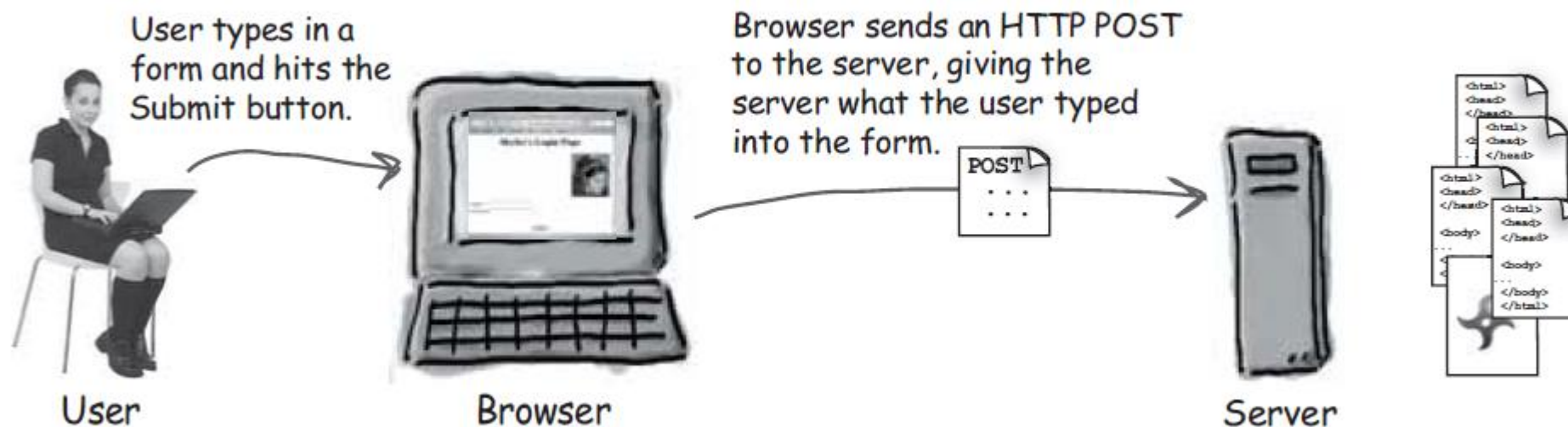
HTTP - Método *get*

GET

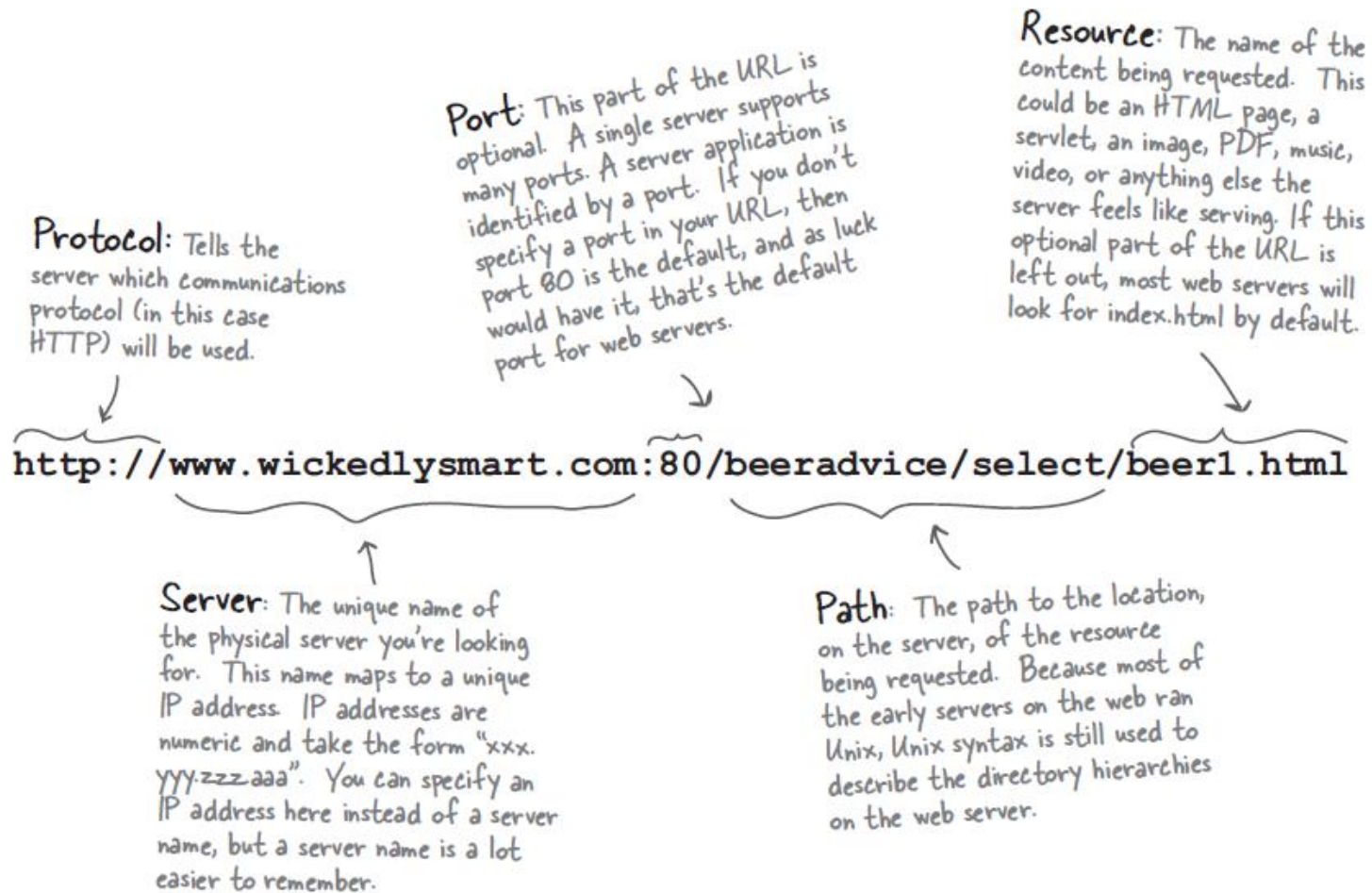


HTTP – Método *post*

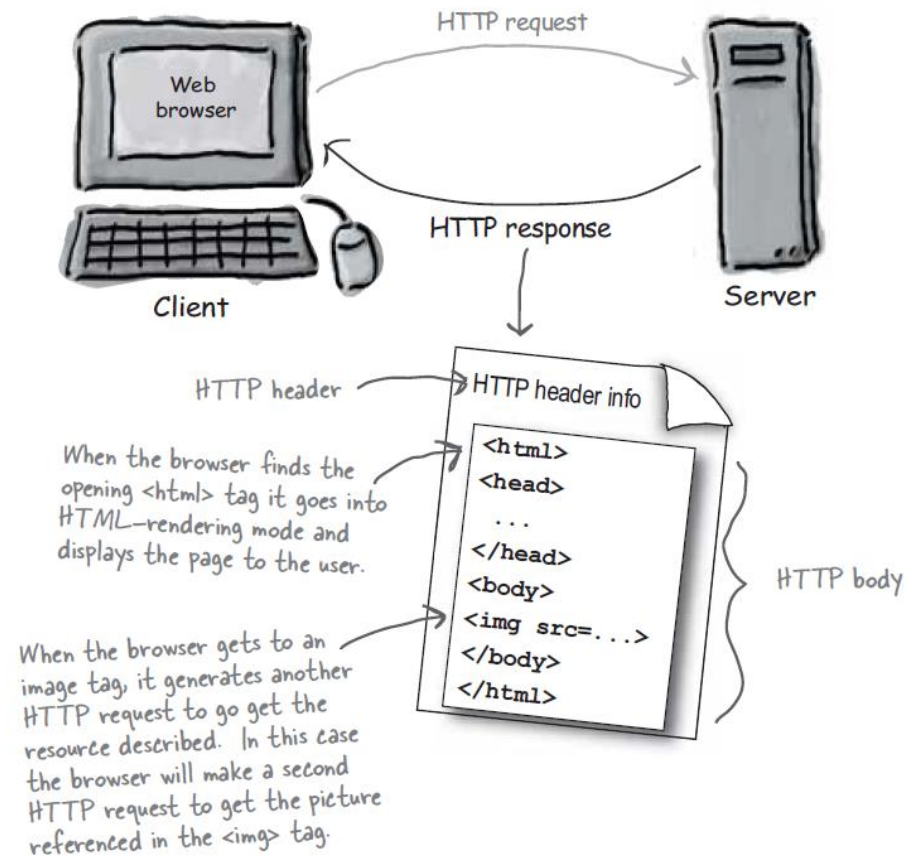
POST



URL (Uniform Resource Locator)



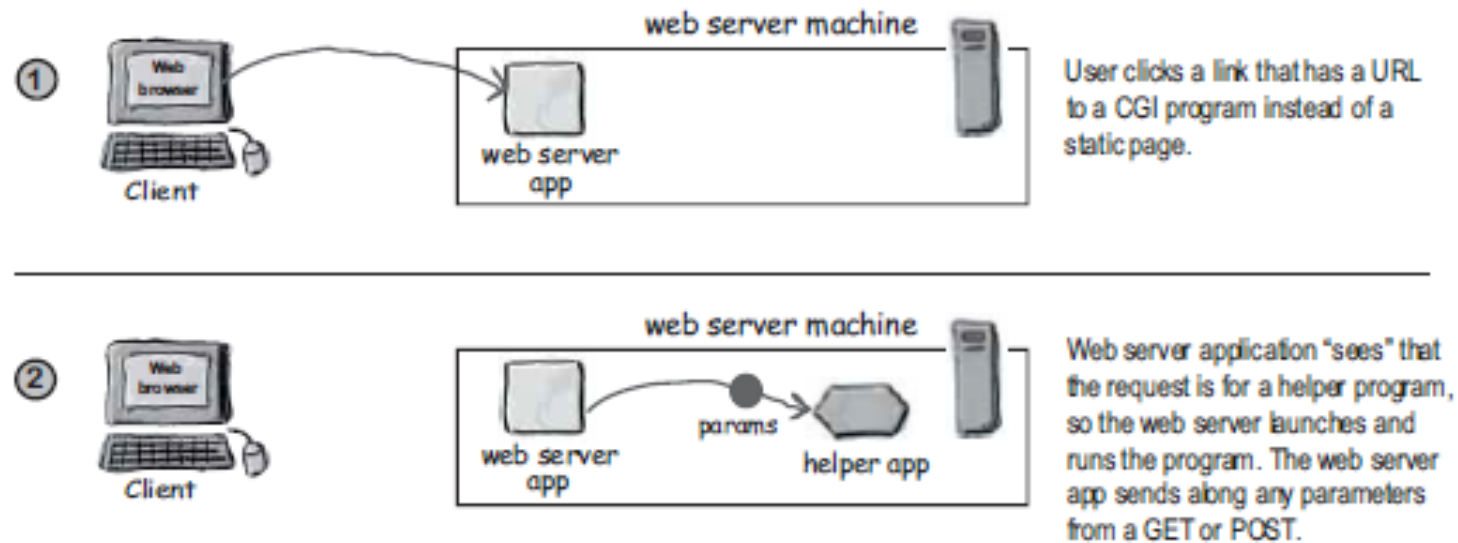
HTML



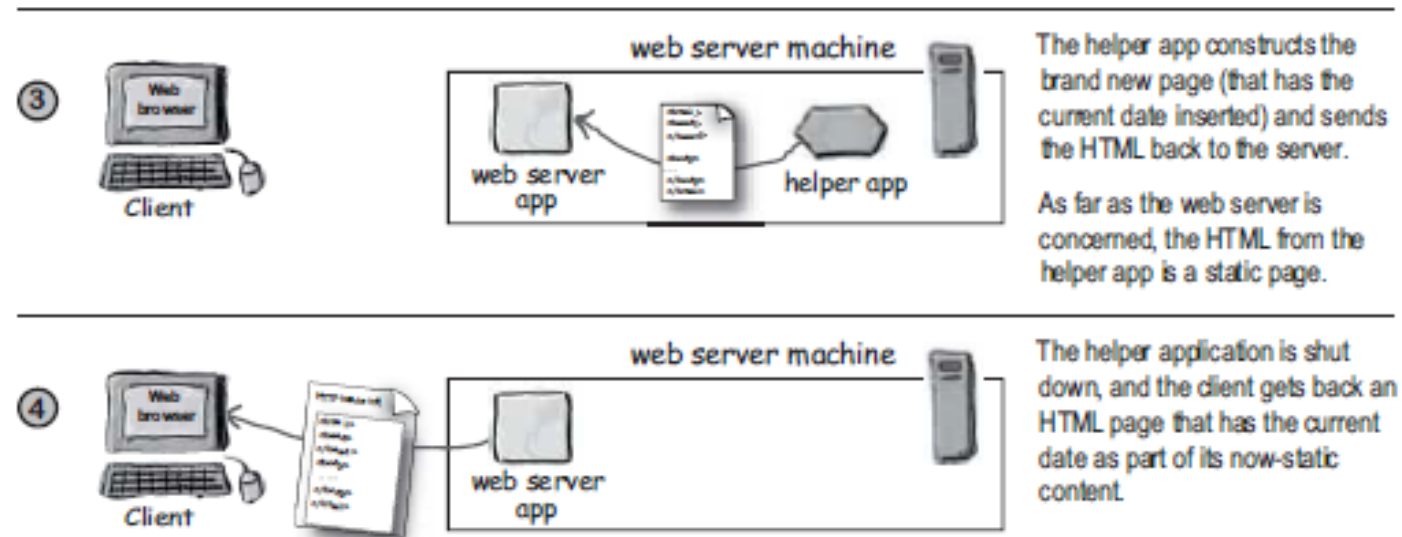
Páginas estáticas, páginas dinâmicas e CGI

- O HTML, o HTTP e os servidores Web “puros” (ex.: Apache HTTP Server) só conseguem trabalhar com páginas estáticas.
- Para ter conteúdo dinâmico, precisamos de software extra (CGI – Common Gateway Interface):
 - PHP, Pearl, C, etc.

CGI



CGI



Java e aplicações Web

- Também podemos construir páginas dinâmicas (aplicações Web) utilizando Java.
- Para executar uma aplicação Web Java, precisamos de um container Web.
- O container Web é uma aplicação executada em conjunto com o servidor HTTP para trabalhar com conteúdos dinâmicos.

Java e aplicações Web

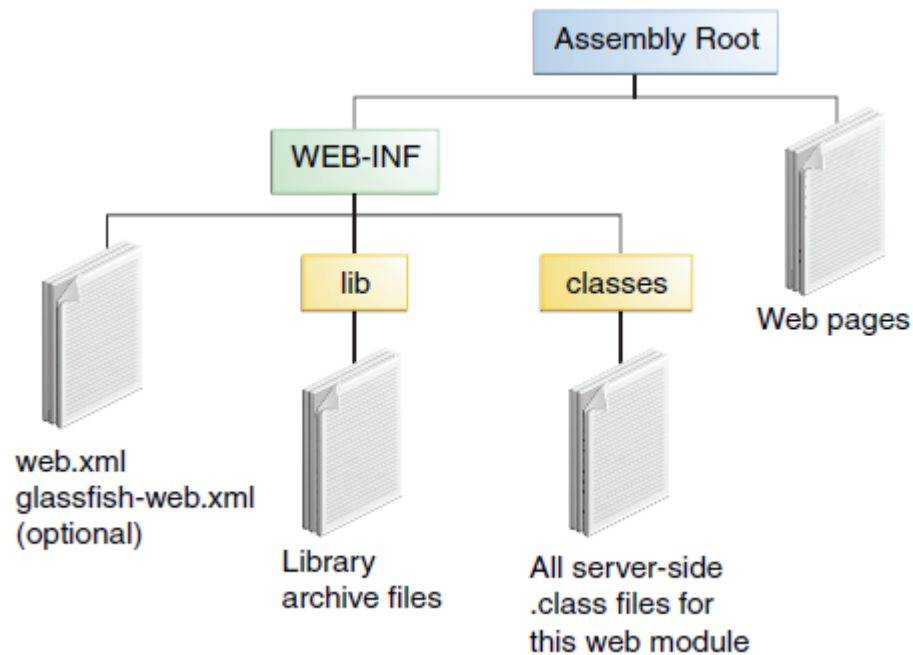
- A aplicação Web que desenvolvemos na linguagem Java é implantada em um container Web.
- O container Web executa essa aplicação Java para tratar requisições HTTP e respondê-las.
- O Apache Tomcat é um exemplo de container Web bem tradicional.

Estrutura de uma aplicação Web Java

- Para implantar uma aplicação Web Java em um container Web, algumas regras devem ser seguidas.
- Os códigos fonte e demais arquivos referentes à aplicação Web devem ser colocados em um arquivo no formato .war (Web Archive).

Estrutura de uma aplicação Web Java

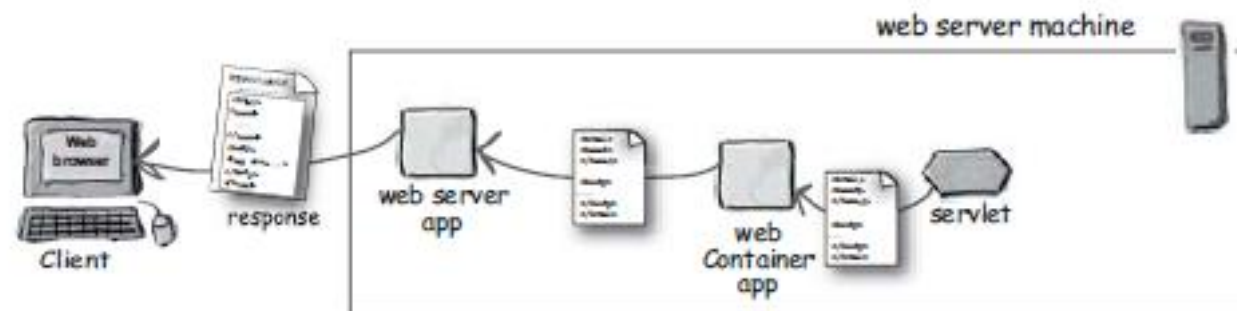
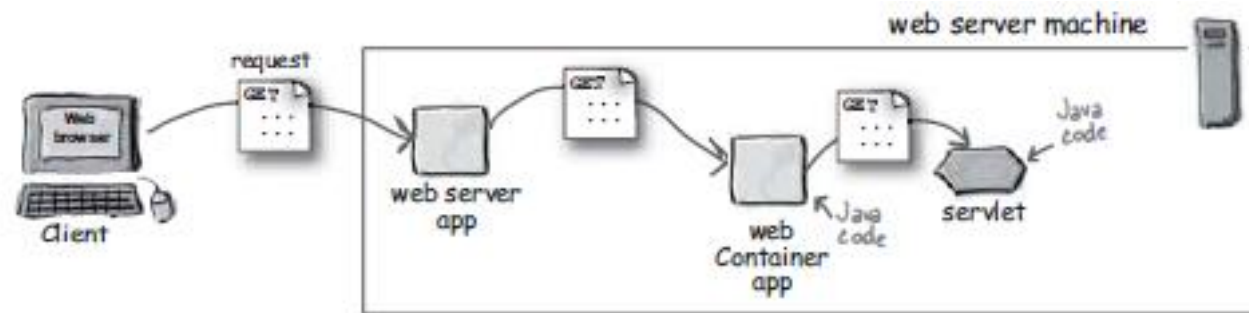
- O conteúdo do Web Archive deve seguir essa estrutura:



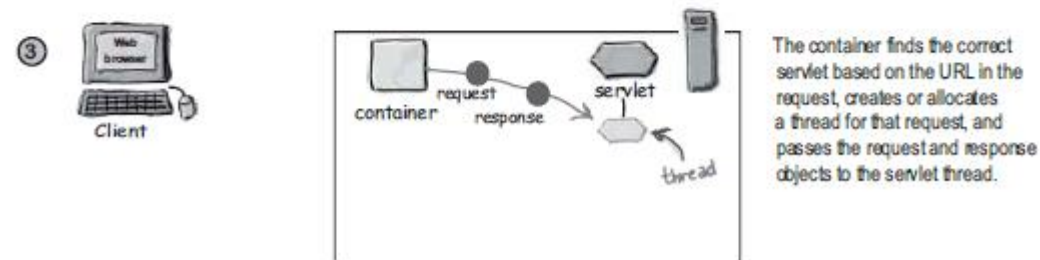
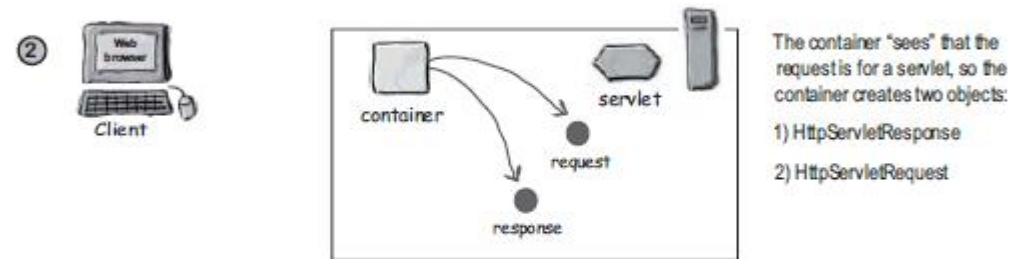
Servlets

- Os Servlets são uma solução presente no J2EE que permitem a construção de aplicações Web com Java.

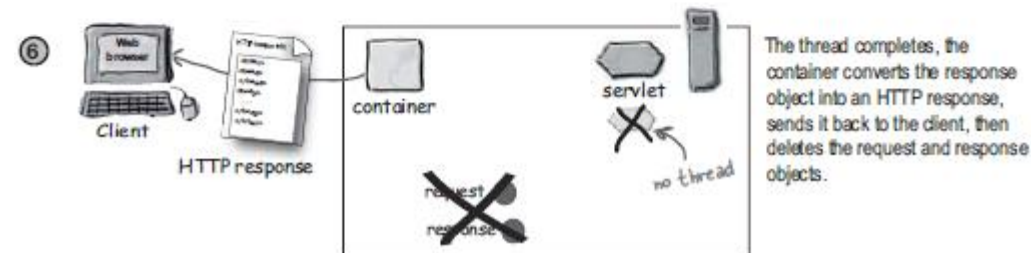
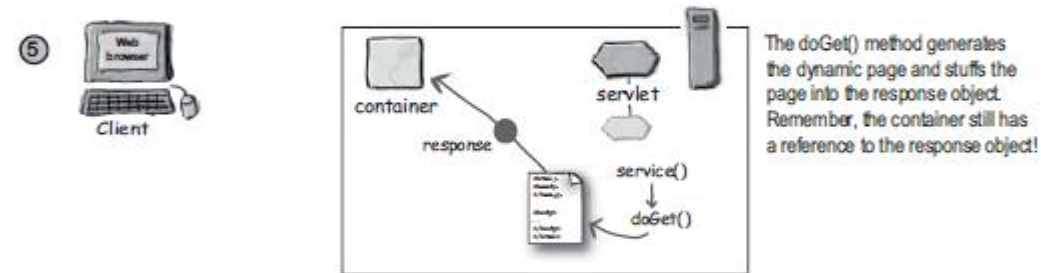
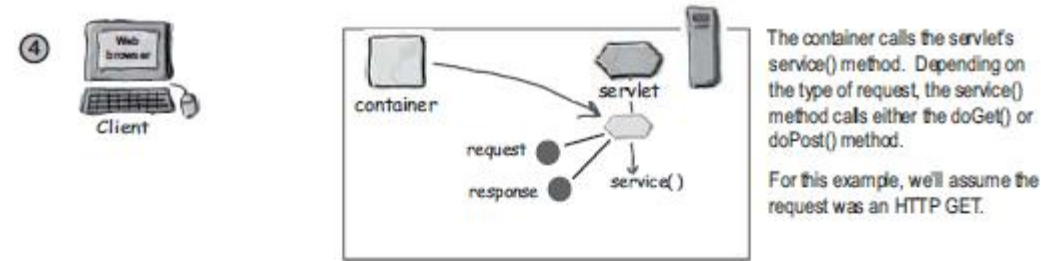
Servlets



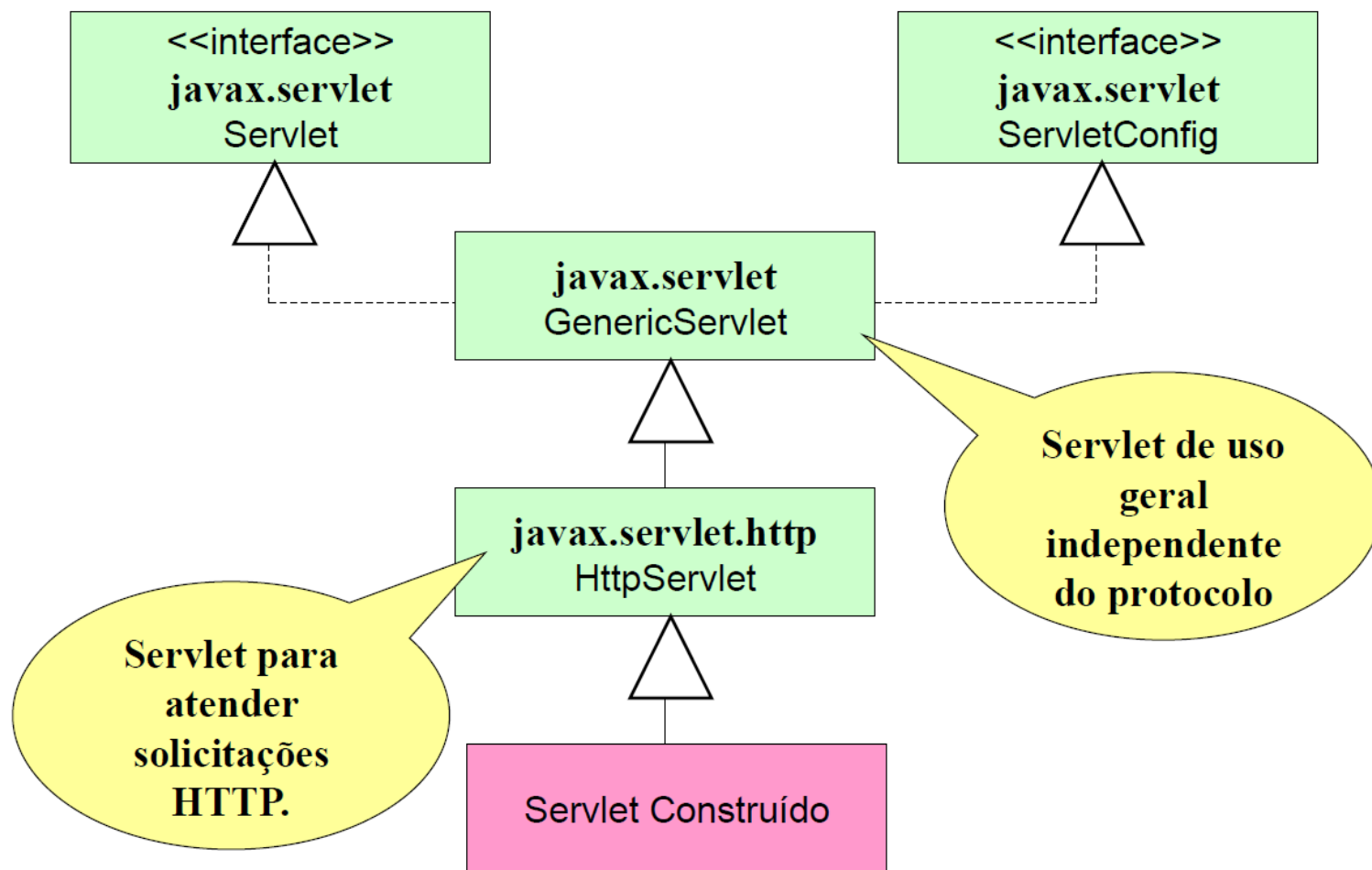
Manipulação de um *request*



Manipulação de um *request*



Servlets



Exemplo de código de Servlet

SERVLET EXAMPLE

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

```
public class ServWelcome extends HttpServlet
```

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

```
{
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
```

```
    out.println("<HTML>");
    out.println("<HEAD><TITLE>First Servlet Program</TITLE></HEAD>");
    out.println("<BODY>");
    out.println("<H1>Welcome to Servlets</H1>");
    out.println("</BODY>");
    out.println("</HTML>");
    out.close();
```

```
}
```

Servlets are not part of the standard SDK, they are part of the J2EE

Servlets normally extend HttpServlet

The response to be sent to the client

Details of the HTTP request from the client

Set the response type to text/html (this is normal)

This HTML text is sent to the client

Don't forget to close the connection with the client

Servlets

- Os métodos *doGet* e *doPost* são usados para processar requisições HTTP do tipo get e post, respectivamente;
- Para fazer o mapeamento entre a URL e o Servlet que deve ser executado, nós utilizamos um arquivo conhecido como deployment-descriptor. O nome deste arquivo é “web.xml”;

Mapeamento de uma URL para o Servlet

Primeiro, declara-se o servlet, indicando a classe e dando um nome:

```
<servlet>
  <servlet-name> primeiraServlet </servlet-name>
  <servlet-class> br.uel.HelloWorldServlet </servlet-class>
</servlet>
```

Depois, define-se o padrão de URL que deverá ser tratado por esse servlet:

```
<servlet-mapping>
  <servlet-name> primeiraServlet </servlet-name>
  <url-pattern> /oi </url-pattern>
</servlet-mapping>
```


Mapeamento de uma URL para o Servlet

- Também podemos fazer isso com as annotations do Java (à partir do J2EE 6/Servlets 3):

```
@WebServlet(name = "primeiraServlet", urlPatterns = {"/oi"})  
public class HelloWorldServlet extends HttpServlet {  
    ...  
}
```