Introdução a Servlets

Laboratório de Programação (5COP011)

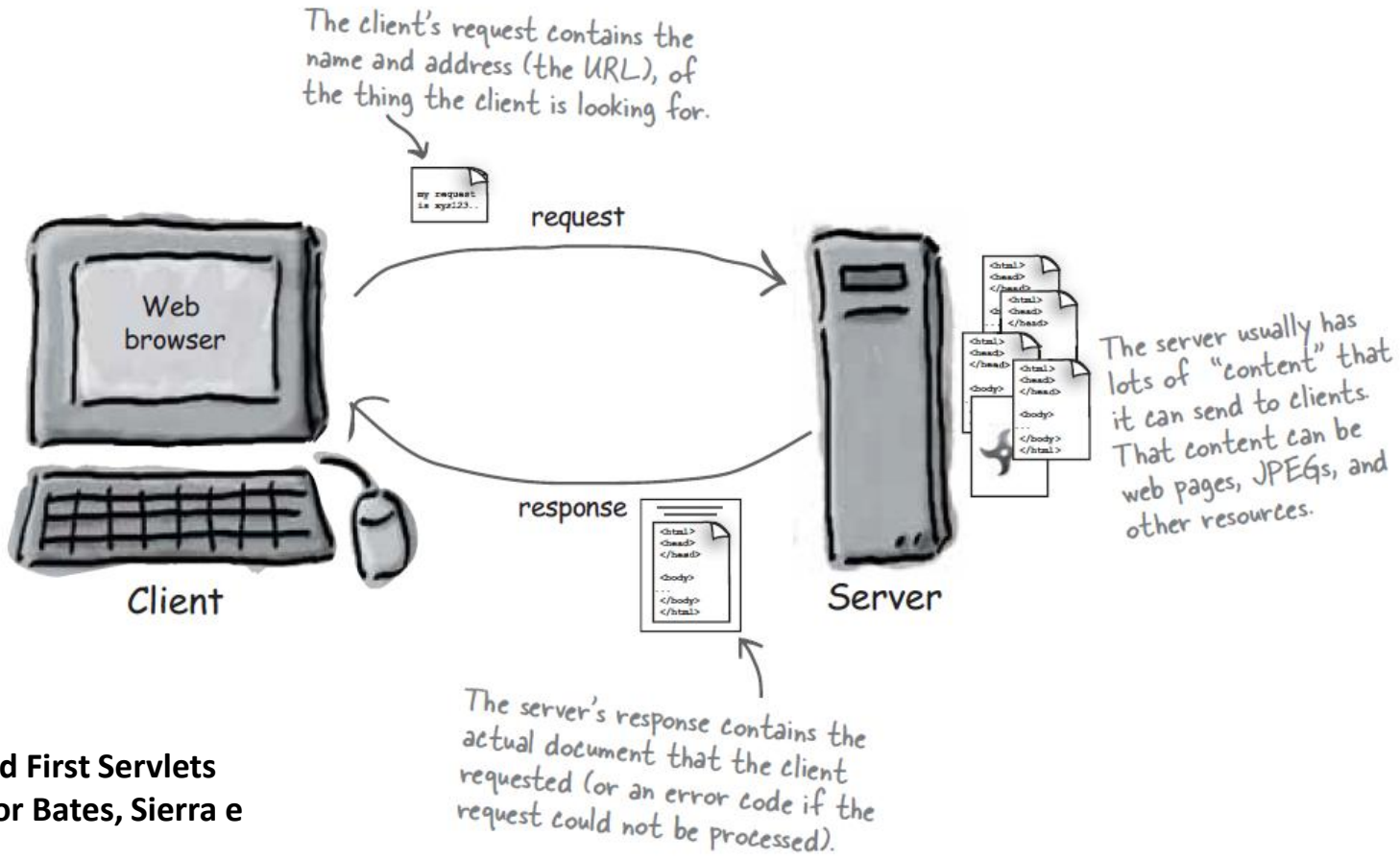
Prof. Bruno Bogaz Zarpelão

Departamento de Computação - 2016

Objetivos

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

Interação cliente servidor



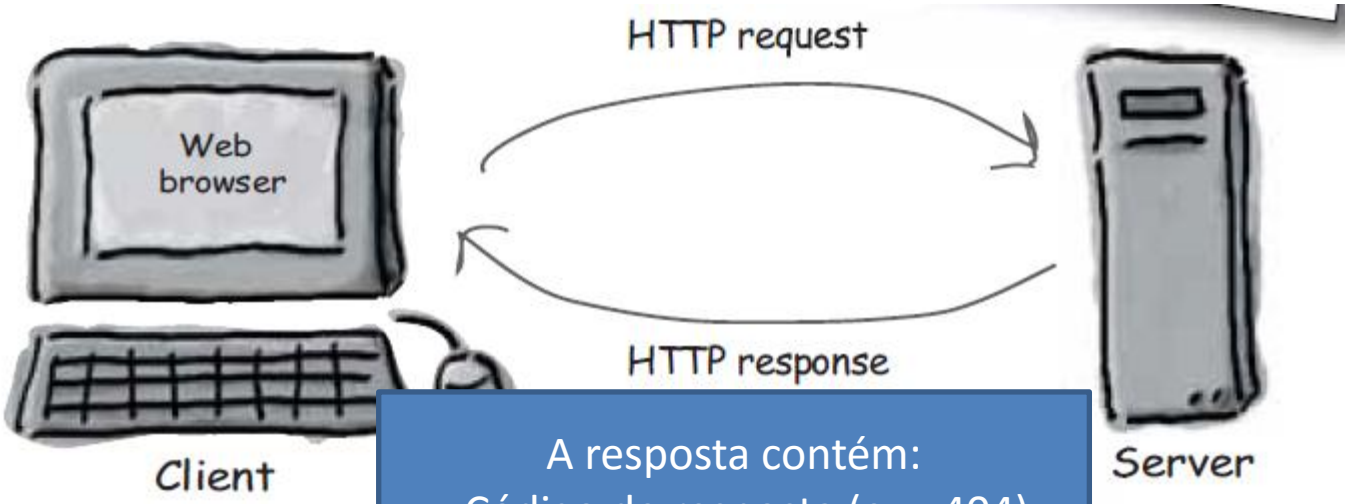
Fonte: Head First Servlets and JSPs por Bates, Sierra e Basham

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;

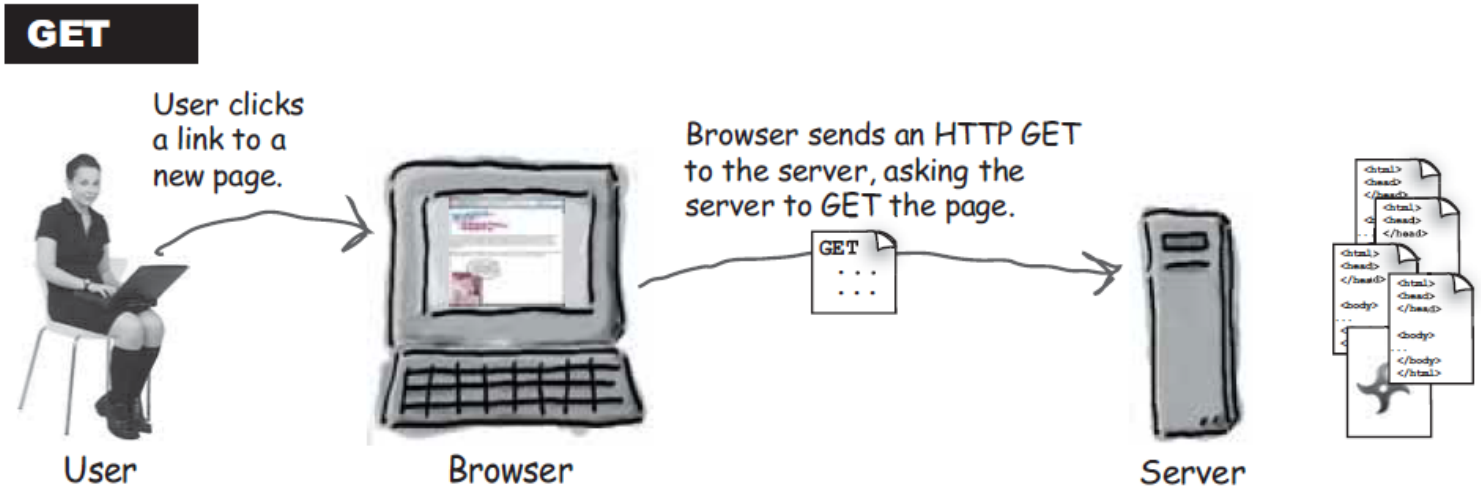
HTTP

A requisição contém:
- método HTTP (get, post, etc)
-Objeto a ser acessado (URL)
- parâmetros

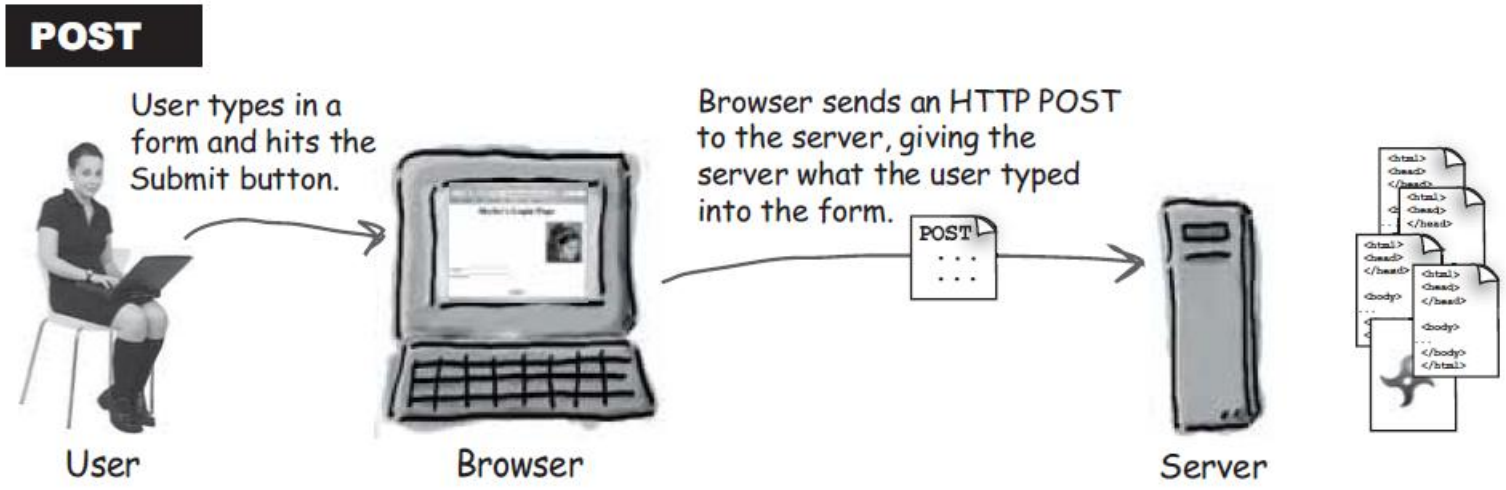


A resposta contém:
- Código de resposta (ex.: 404)
- Tipo de conteúdo (texto, fotos, HTML)
- O conteúdo

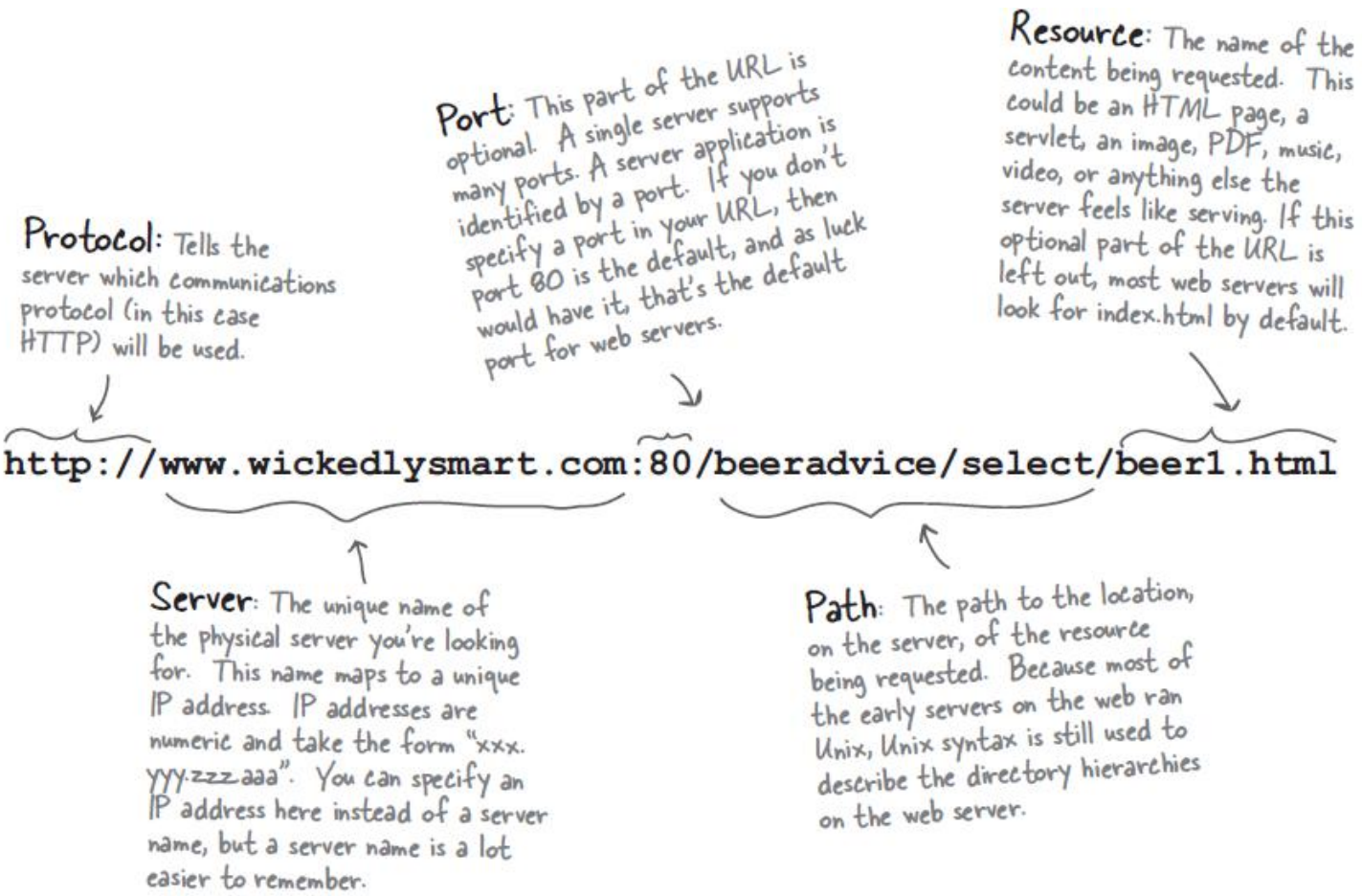
HTTP - Método *get*



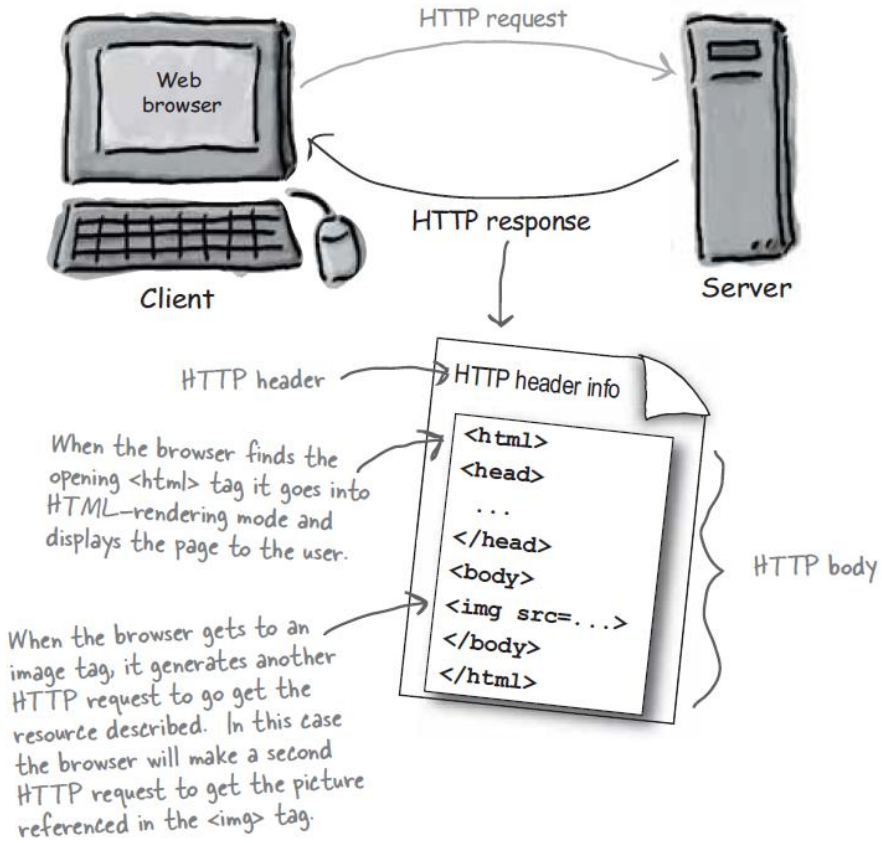
HTTP – Método *post*



URL (Uniform Resource Locator)



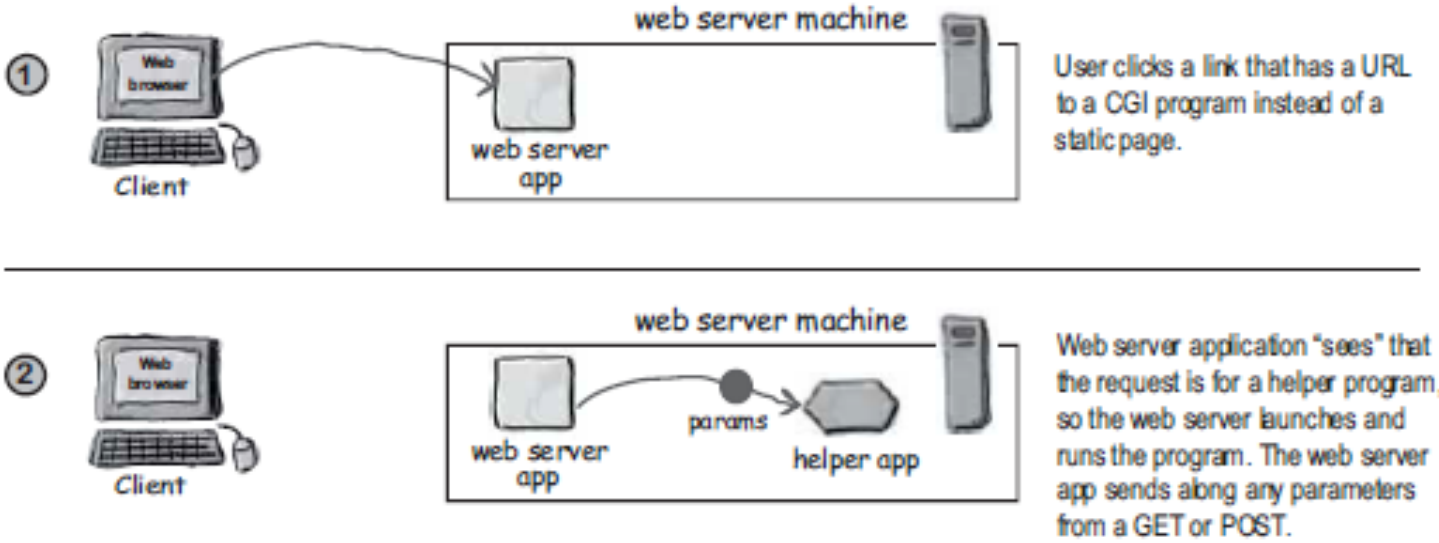
HTML


HTTP request
Web browser
HTTP response
Client
Server
HTTP header
HTTP header info
When the browser finds the opening <html> tag it goes into HTML-rendering mode and displays the page to the user.
<html>
<head>
...
</head>
<body>
<img src=...>
</body>
</html>
HTTP body
When the browser gets to an image tag, it generates another HTTP request to go get the resource described. In this case the browser will make a second HTTP request to get the picture referenced in the <img> tag.

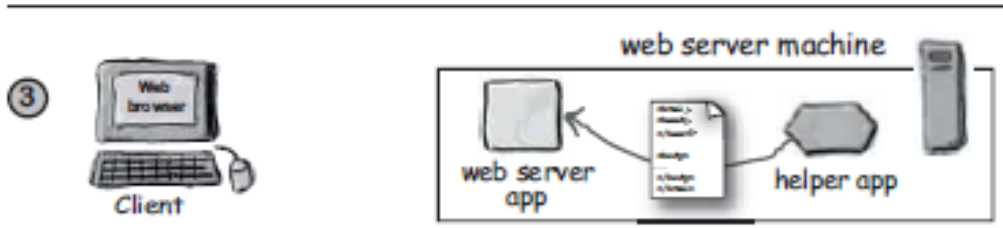
Páginas estáticas, páginas dinâmicas e CGI

- 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

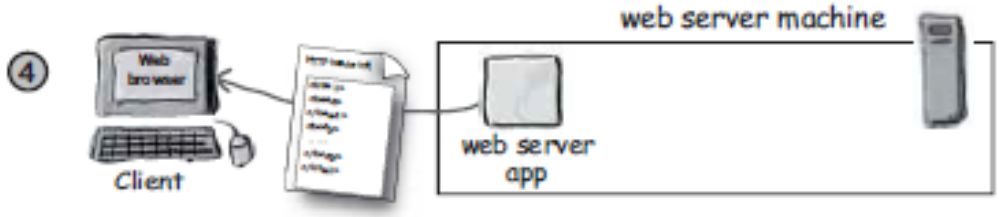

1
Web browser
Client
web server machine
web server app
User clicks a link that has a URL to a CGI program instead of a static page.
2
Web browser
Client
web server machine
web server app
params
helper app
Web server application "sees" that the request is for a helper program, so the web server launches and runs the program. The web server app sends along any parameters from a GET or POST.

CGI



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.

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

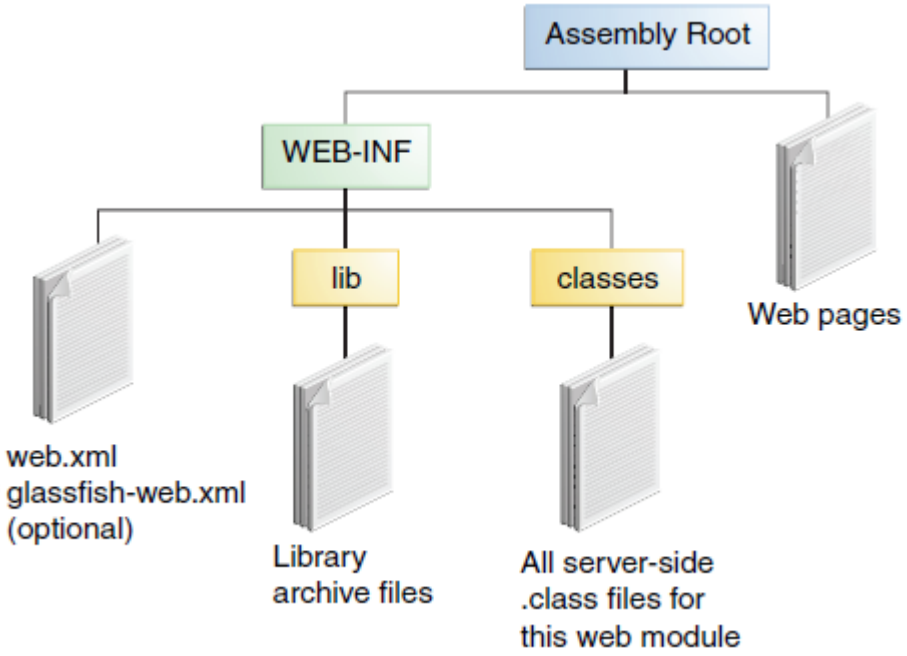
- A aplicação Web que desenvolvemos na linguagem Java é implantada em um container Web.
- O container Web executa essa aplicação Java para tratar requisições HTTP e respondê-las.
- O Apache Tomcat é um exemplo de container Web bem tradicional.

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

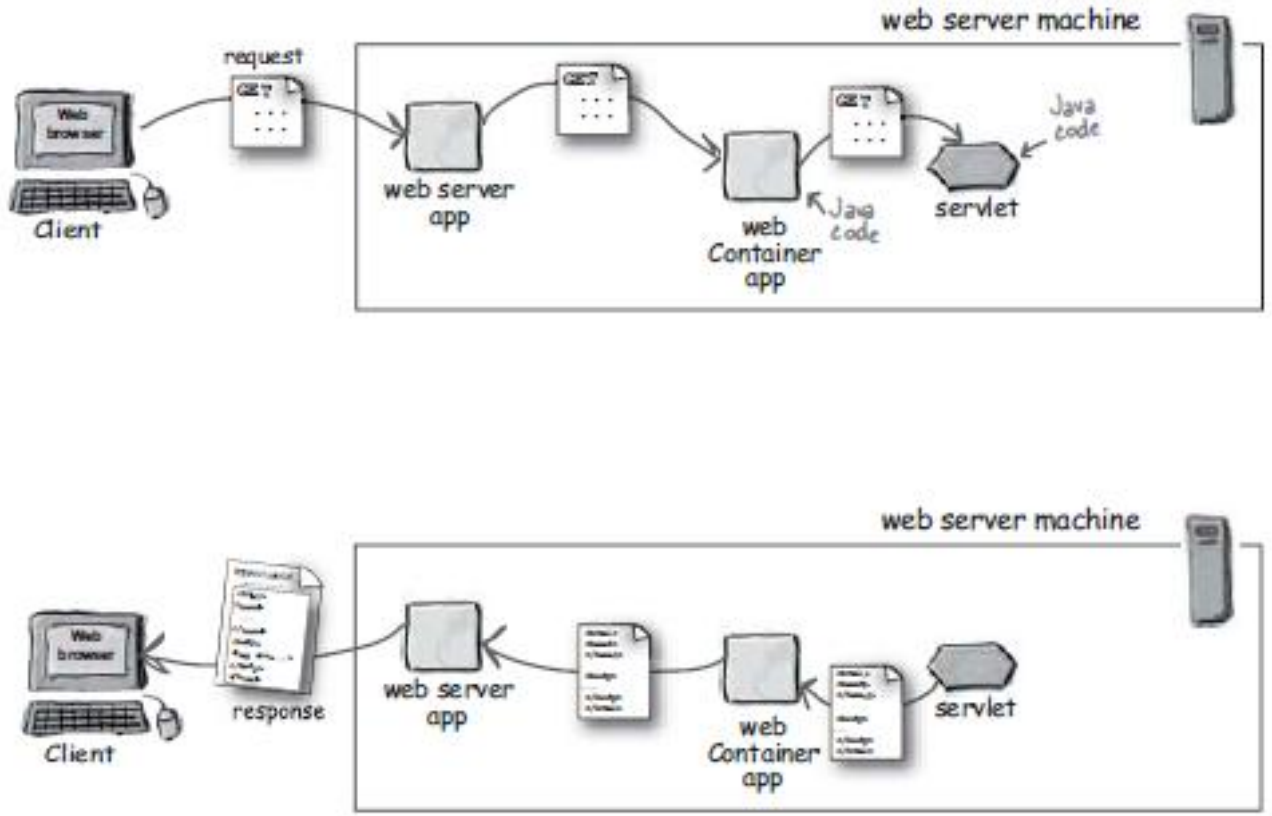
- O conteúdo do Web Archive deve seguir essa estrutura:



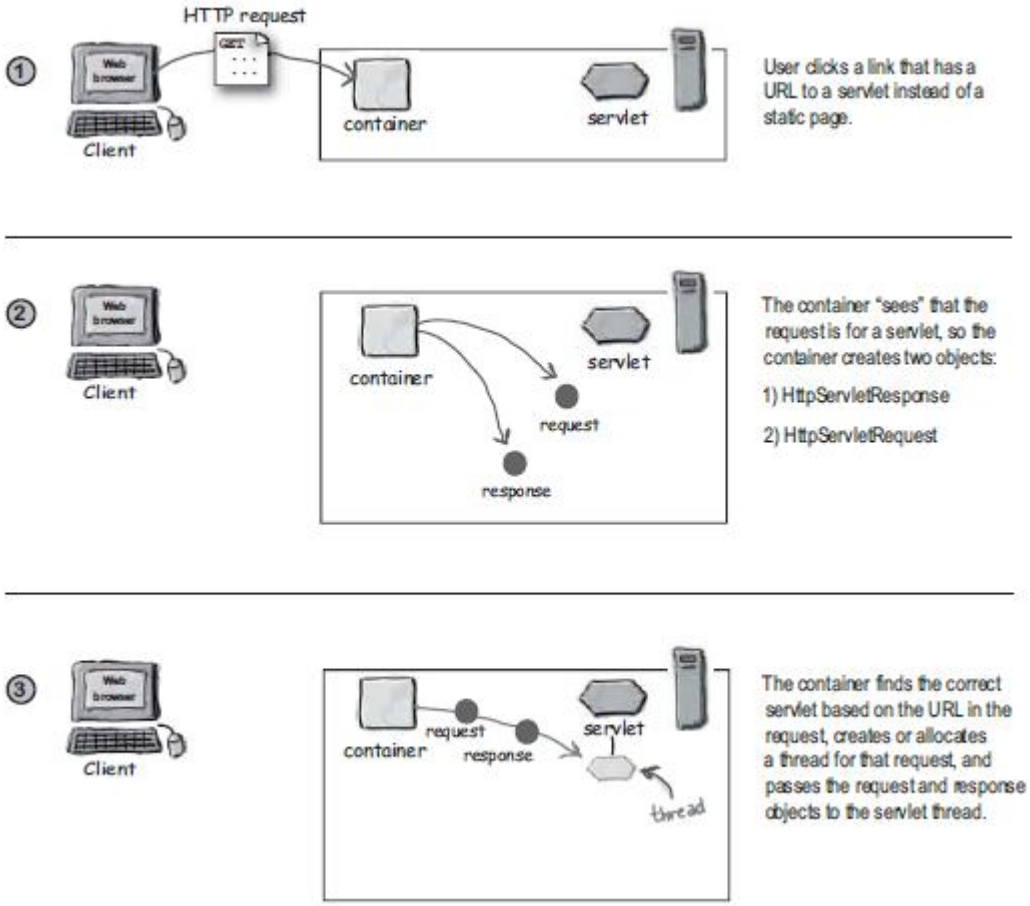
Servlets

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

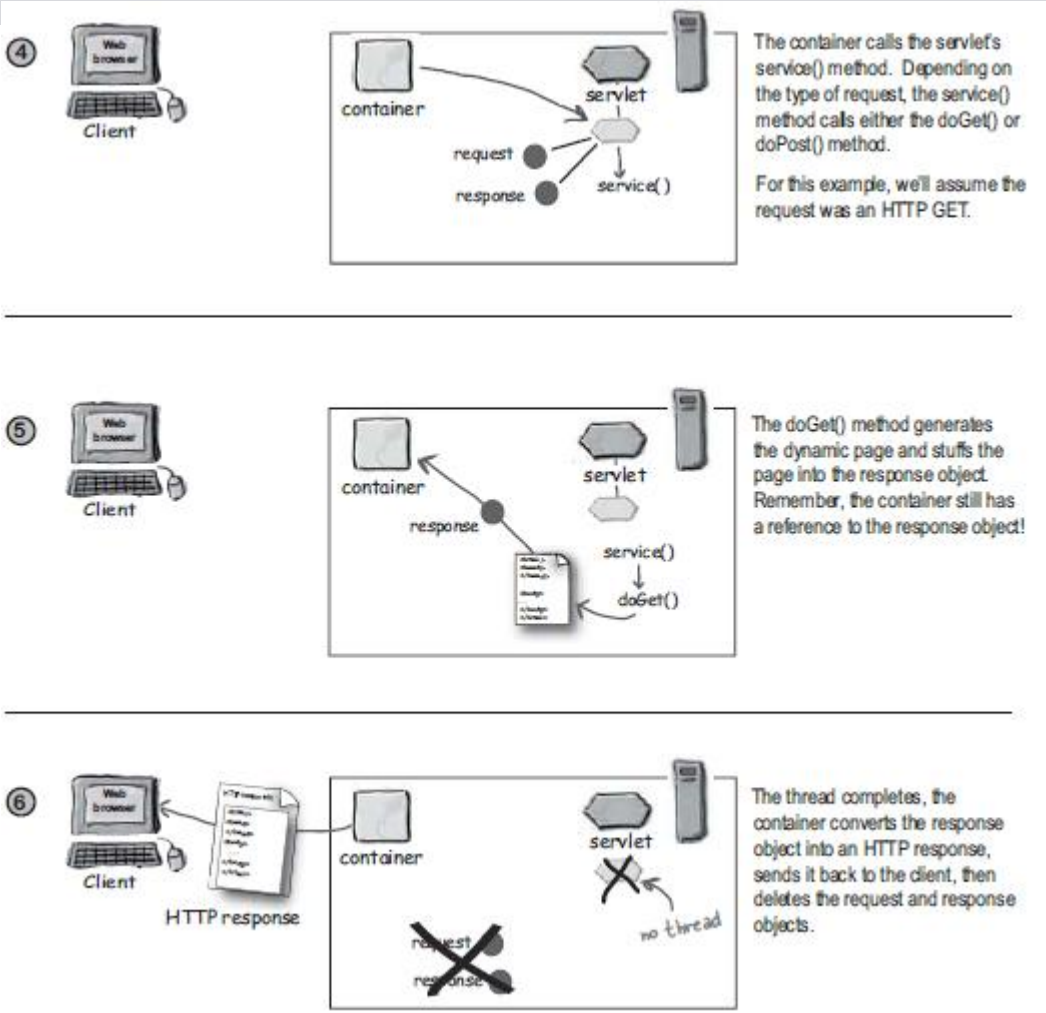
Servlets


web server machine
request
Web browser
Client
web server app
web Container app
Java code
servlet
Java code
web server machine
Web browser
Client
response
web server app
web Container app
servlet

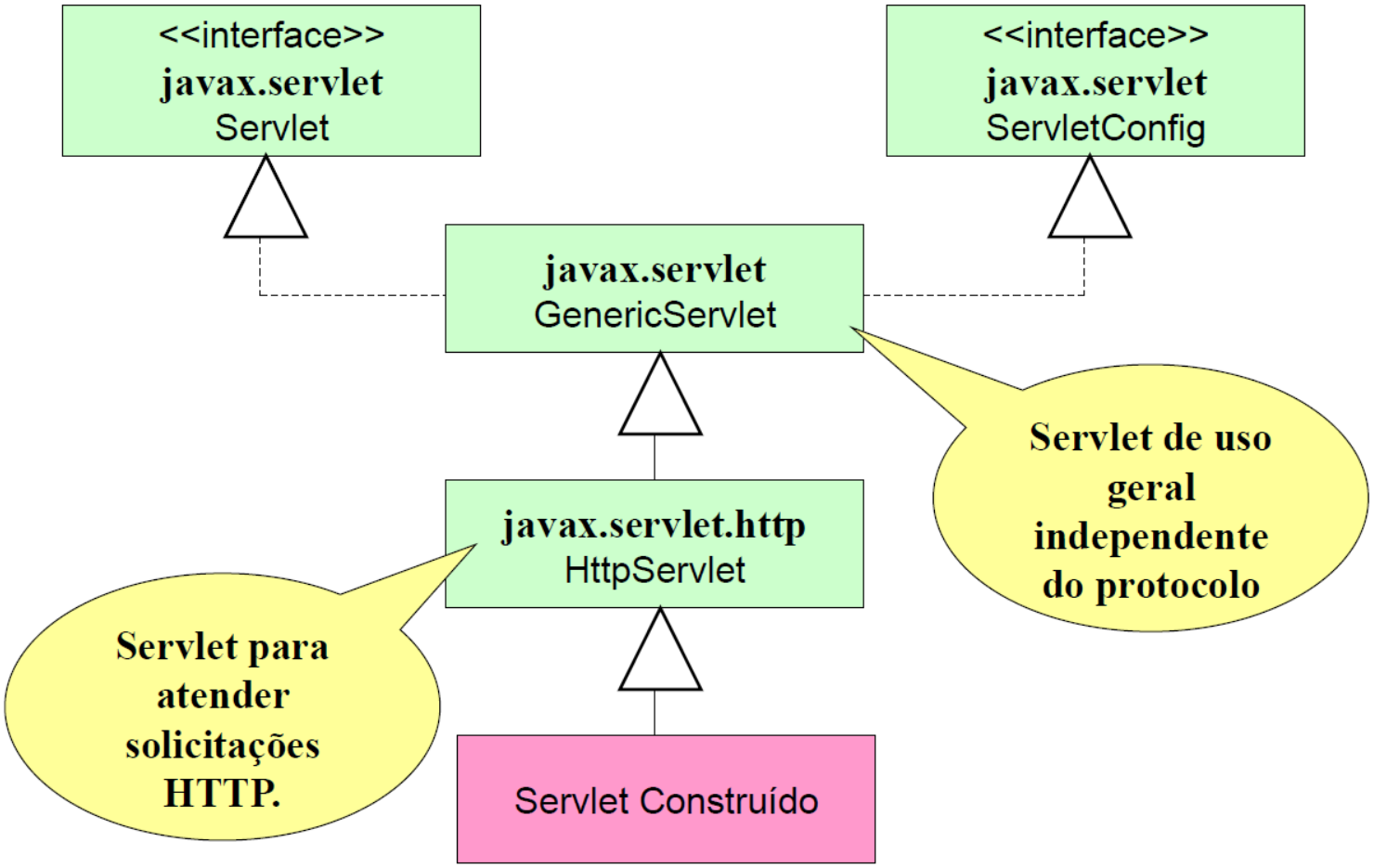
Manipulação de um *request*



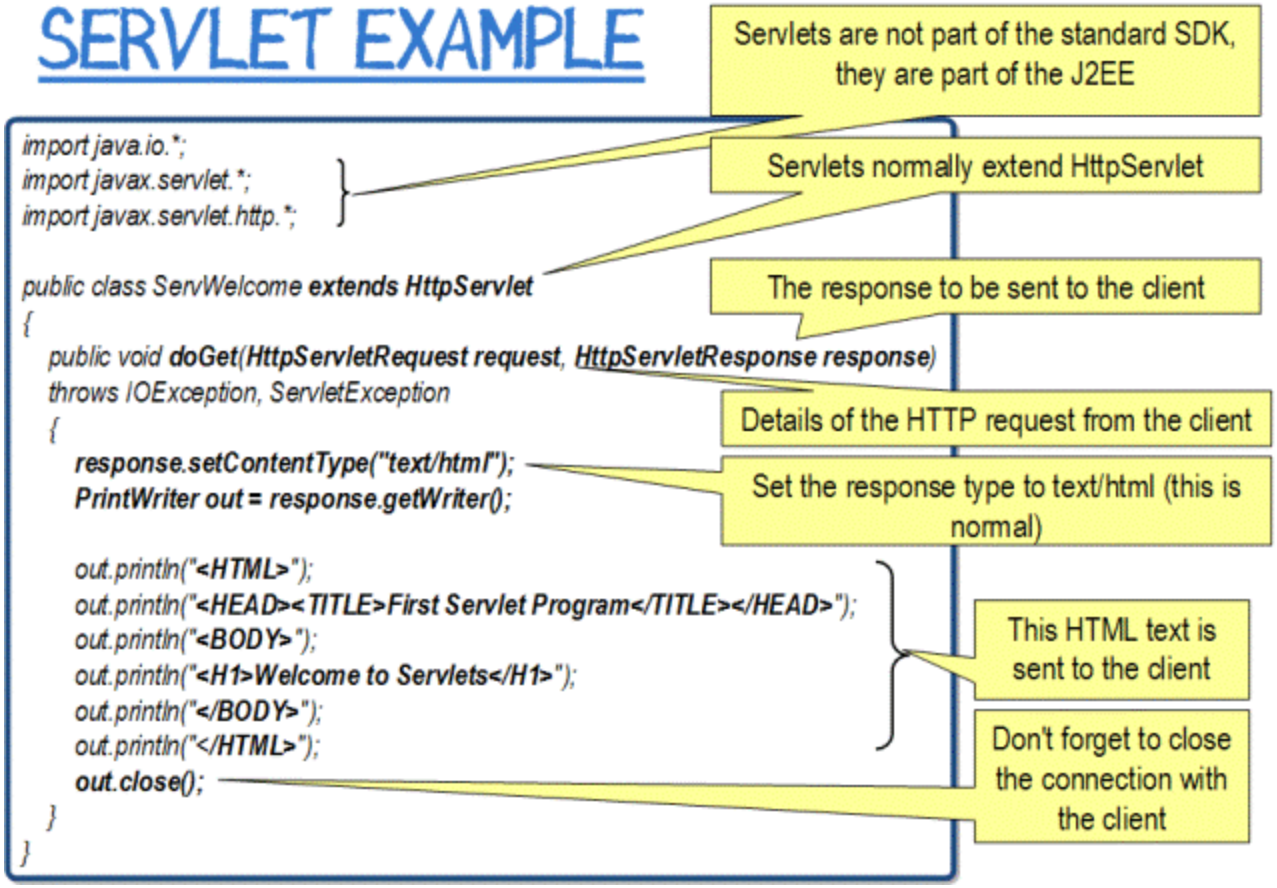
Manipulação de um *request*



Servlets


<<interface>>
javax.servlet
Servlet
<<interface>>
javax.servlet
ServletConfig
javax.servlet
GenericServlet
Servlet de uso geral independente do protocolo
javax.servlet.http
HttpServlet
Servlet para atender solicitações HTTP.
Servlet Construído

Exemplo de código de Servlet



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 {
…
}
```