

Servlets / JSP

Glaucio Scheibel - GUJava/SC



O que são Servlets?

- Objetos da tecnologia Java que estendem a funcionalidade de um servidor HTTP.
- Comparável aos NSAPI da Netscape, ISAPI da Microsoft, ou Módulos do Apache.
- Independente de plataforma.
- Independente de servidor.



Por que Servlets?

- HTTP é o transporte universal da internet através de *Firewalls* restritivos.
- Navegadores, aplicações web, aplicações Java, e outros programas podem usar HTTP.
- Qualquer tipo de dado pode ser transmitido via HTTP, não só HTML.



Servlets são leves!

- Servlets podem rodar no mesmo processo servidor que o HTTP Server.
- Pode suportar maiores cargas de usuários com menos recursos de máquina.
- Servlets pode ser carregado de qualquer lugar.
 - Sistema local de arquivos.
 - WebSite remoto.



Arquiteturalmente mais rápido que CGI

- Servlets não criam um novo processo a cada requisição.
- Servlets são carregados e ficam prontos para receberem as requisições.
- O mesmo servlet pode atender muitas requisições simultâneas.



Fácil de desenvolver

- É programação Java.
- Sem ponteiros como no código C.
- Não há dialetos diferentes para diferentes bancos de dados.
- Orientado à objetos.
- Servlets adequam-se à práticas de reutilização de código.



Muitas bibliotecas Java disponíveis

- A inacreditável funcionalidade das bibliotecas da plataforma Java faz do Servlet a mais flexível camada intermediária.
- Servlets usam as vantagens de JDBC, EJB, JMS, JavaMail, JavaIDL, RMI, etc.
- E claro, qualquer tecnologia de terceiros feita em Java é facilmente acessível.



Write Once, Run Anywhere!

- Servlets podem rodar em qualquer plataforma onde o runtime Java existe.
- O mesmo servlet pode rodar em qualquer servidor HTTP, desde Apache até Zeus.
- Desenvolva numa pequena máquina e execute no maior dos servidores.



Servidores que suportam Servlets

- Apache Tomcat
- ATG Dynamo Application Server
- Bajie Server
- BEA WebLogic Server
- Borland AppServer
- Brokat Server Technologies
- Caucho Technology Resin
- EasyThings Web Server
- ExOffice Intalio
- Gefion WebServer
- HP Bluestone Total-e-Server
- IBM WebSphere Application Server 4.0
- IONA iPortal Application Server
- iPlanet Web Server
- Lutris Technologies Enhydra
- Macromedia Jrun



Servidores que suportam Servlets

- Jetty
- New Atlanta ServletExec
- Novocode NetForge
- Oracle 9i JServer
- Orion Application Server
- ParaChat Server 3.0
- Pramati Server
- Secant Technologies
ModelMethods Enterprise
Server
- Servotec Internet Server
- Silverstream Application
Server
- Sybase EAServer
- Tagtraum Industries jo!
- Trifork Enterprise
Application Server
- vqSoft vqServer
- W3C Jigsaw
- Zeus Technology Web
Server



JAVA™

Netcraft Survey (abr/02)

Active sites:

- Apache: 64,38%
- Microsoft: 27,15%
- iPlanet: 1,71%
- Zeus: 1,12%

Top developers:

- Apache: 56,38%
- Microsoft: 31,96%
- Zeus: 2,26%
- iPlanet: 2,21%



Um Servlet simples

```
public class HelloServlet extends HttpServlet {  
    public void doGet(HttpServletRequest request,  
                        HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/plain");  
        PrintWriter out = response.getWriter();  
        out.println("Hello World!");  
    }  
}
```



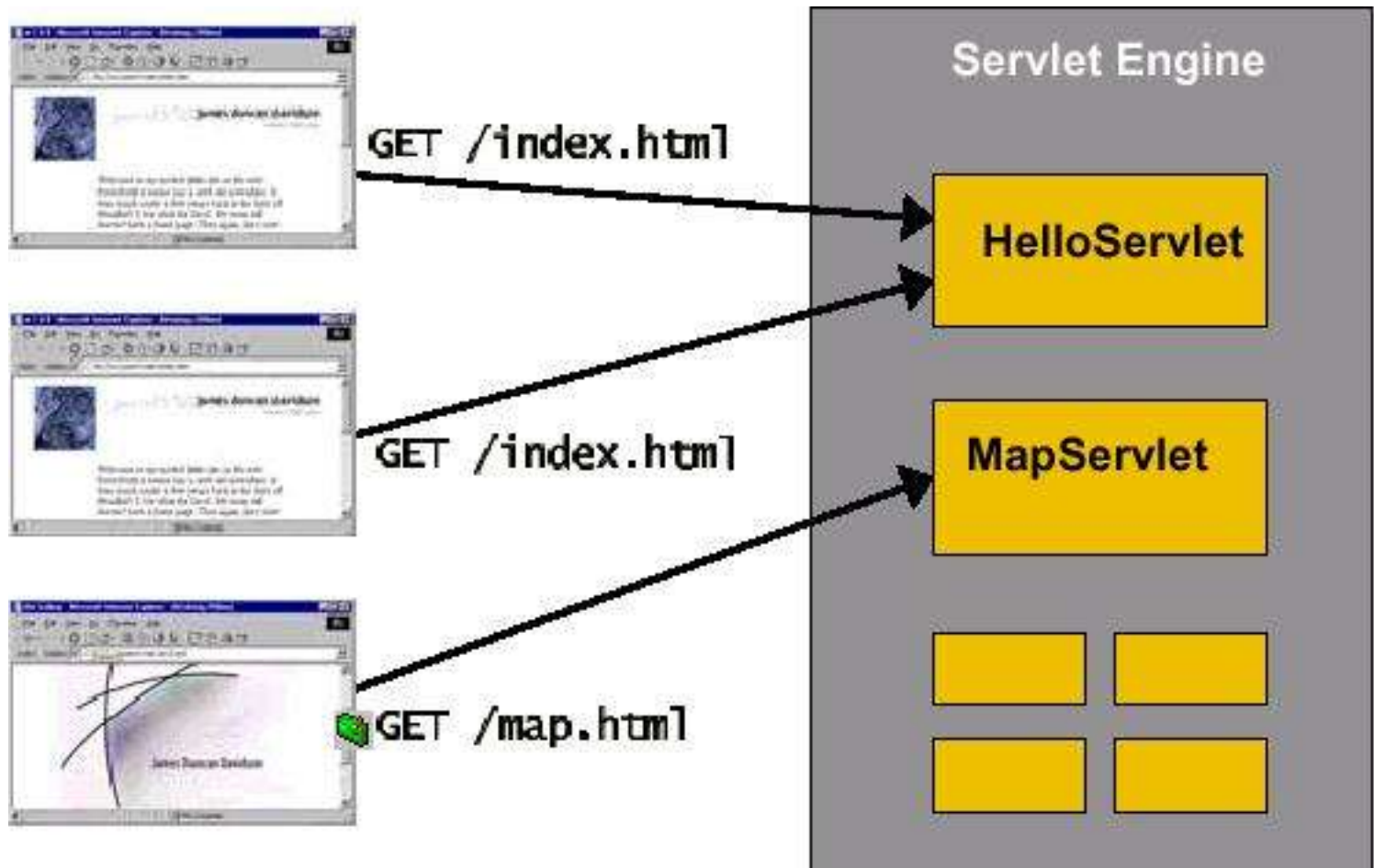
Servlets

- Servlets tem um ciclo de vida bem definido.
- Administrado pelo container.
- Carregado sob demanda.
- Descarregado a qualquer momento.
- Pode ser mapeada para qualquer parte da URL.
- Multi-Thread



JAVA™

Request's em ação



Ciclo de vida do Servlet

- Um servlet é instanciado pelo container.
- É inicializado através do método *init*.
- O método *service* pode ser chamado de 0 a n vezes.
- Pode ser limpo através do método *destroy*, que é invocado quando o container o remove.

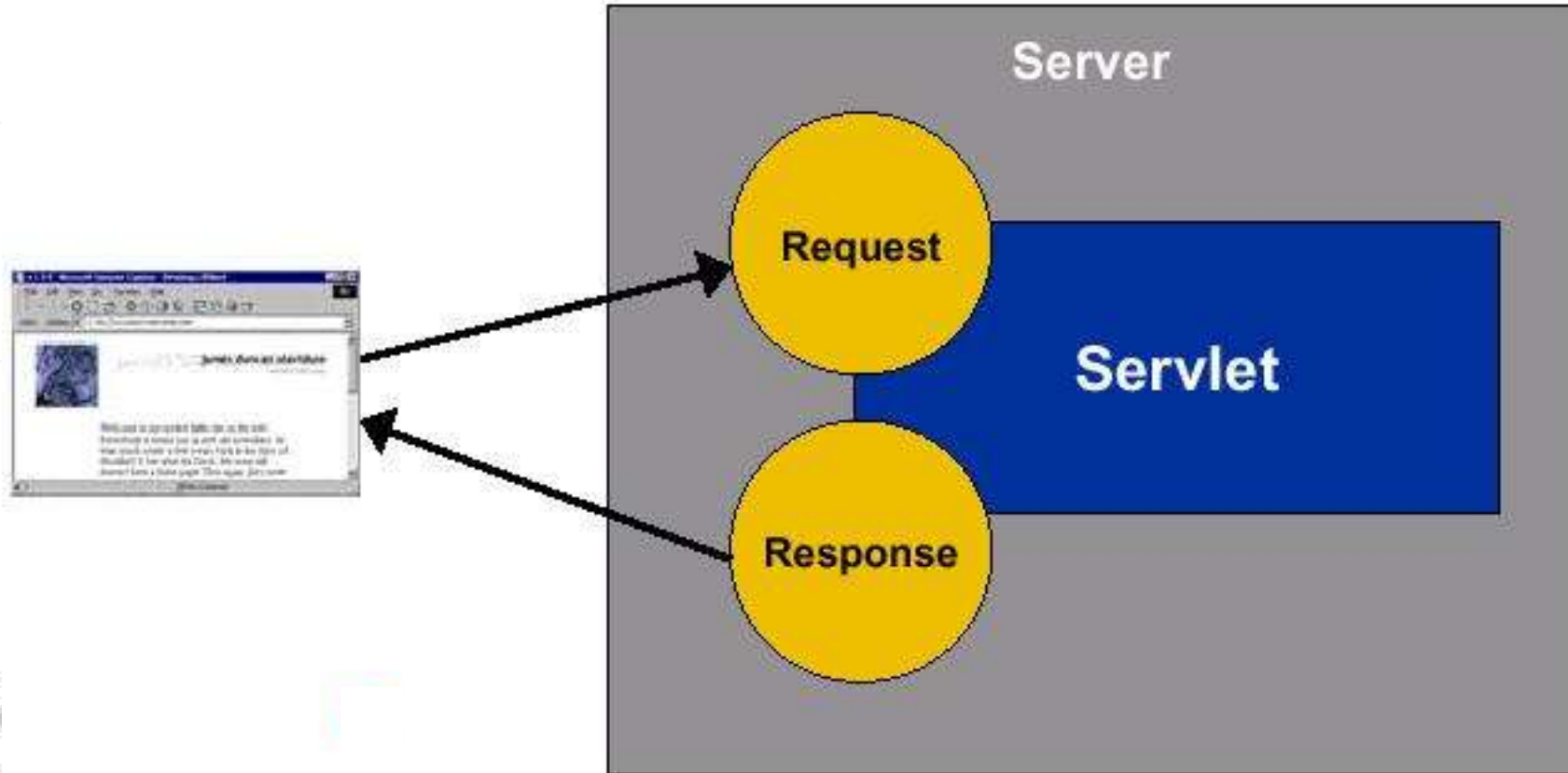


Anatomia de um Request

- Um cliente envia um request a um servidor.
- O request é resolvido para um servlet pelo container.
- O método service do servlet é chamado com um objeto Request e um Response.
- O servlet devolve um response preenchido para o request.



Um Request



O objeto Request

- Encapsula todas as informações do cliente.
- Acesso ao cabeçalho da requisição.
- Acesso a um *InputStream* ou *Reader*.
- Acesso a informações CGI-like.
- Acesso aos dados do formulário.



Métodos frequentes do Request

- Enumeration `getParameterNames();`
- `String getParameter(String paramName);`
- `String getRemoteAddr();`
- `String getRequestURI();`
- Enumeration `getHeaderNames();`
- `String getHeader(String headerName);`
- `HttpSession getSession();`
- `Cookie[] getCookies();`



O objeto Response

- Encapsula toda comunicação para o cliente.
- Acesso aos cabeçalhos da resposta.
- Acesso a um *OutputStream* ou *Writer*.
- Acesso para setar *cookies*.
- Métodos para enviar redirecionamentos, páginas de erro, etc



Métodos frequentes do Response

- `ServletOutputStream getOutputStream();`
- `PrintWriter getWriter();`
- `void setContentType(String type);`
- `void setContentLength(int length);`
- `void addCookie(Cookie cookie);`
- `void setStatus(int statusCode);`
- `void sendError(int statusCode);`
- `Void sendRedirect(String url);`



Rastreamento de sessão

- Sessões são uma série de requisições feitas por um cliente durante um período de tempo.
- Muitas maneiras foram feitas pelos desenvolvedores. Todas complicadas para trabalhar.
- Camada unificada para trabalhar com sessões incluída na API Servlet.



Sessões em ação

```
public void doGet( HttpServletRequest request,
    HttpServletResponse response) throws
    ServletException, IOException {
    HttpSession session = request.getSession();
    Cart cart = (Cart) session.getValue(cart);
    // Lógica do carrinho
    AddItem(inventoryNumber, quantity);
}
```



Não somente HTML

- Muitos programadores utilizam servlet's para HTML, mas...
- Servlets podem gerar imagens usando AWT e Java2D.
- Servlets podem gerar dados customizados.
- Servlets podem enviar objetos serializados.



Geração de imagem

```
public void doGet(HttpServletRequest request,
    HttpServletResponse response) throws
    ServletException, IOException {
    image = this.buildImage(request);
    response.setContentType("image/gif");
    ServletOutputStream out = response.getOutputStream
        ();
    GifEncoder encoder = new GifEncoder(image, out);
    encoder.encode();
}
```



JSP

- Java Server Pages
- Template para um servlet.
- Permite programar através da saída.
- Inclusão de código Java em fontes HTML, XML, etc...
- Ao primeiro acesso, é gerado um servlet que representa cada JSP.



Página JSP

```
<html>
<jsp:useBean id="clock" class="calendar.JspCalendar"/>
<ul>
<li>Day of month: <%=clock.GetDayOfMonth()%>
<li>Year: <%=clock.getYear()%>
</ul>
</html>
```



Elementos JSP

- Texto estático
- Diretivas padrões
- Elementos de script
- Ações padrões
- Tag's customizadas



Diretivas

- `<%@directive attribute="value" ...%>`
- `page`
 - `language`
 - `extends`
 - `imports`
 - `buffer`
 - `errorPage`
 - `isThreadSafe`



Diretivas

- include
- tag library
 - uri: semântica
 - prefix: colisão de nomes
 - Bibliotecas conhecidas
 - Mecanismo portátil do JSP 1.1



Elementos de Script

- Declarações
 - `<%! int i; %>`
- Scriptlets
 - `<% myBean.setSomething(foo); %>`
- Expressões
 - `<%= myBean.getResult() %>`



Objetos implícitos

- Objetos definidos automaticamente:
 - request
 - response
 - session
 - application
 - out (bufferizado)



Tags padrões

- useBean
- setProperty
- getProperty
- include
- forward
- plugin



Tags customizadas

- Encasula funcionalidades
- Facilita:
 - manutenção
 - uso de ferramentas
- Provê
 - semânticas portáteis



Tags customizadas - Exemplo

```
<connection id="con1" ref="foo.xml" userid="<%=request.userid%>" />

<query id="q1" connection="con1">
SELECT account, balance...
</query>

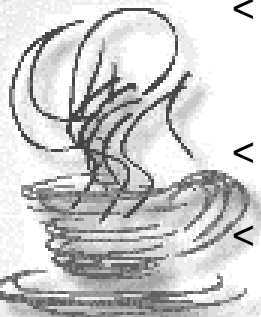
<ul>

<foreach row="row" in="q1">

<li> The balance for <%=row.account%> is <%=row.balance%>

</foreach>

</ul>
```



JAVA[™]

Tags customizadas

- Portáteis
- Tags aninhadas
- Corpo pode ter elementos de script
- Simples de usar
- Simples de definir



Scripts, Beans e Tags

- Complementares.
- Estilos diferentes.
- Scripts: cola.
- Beans: componentes genéricos da plataforma Java.
- Tags: interface adicional para uso nas páginas.



Obrigado!

Perguntas?

