

JavaServer Pages (JSP)



Especificação/IDE/Implementação

- Esse curso foi preparado em 03/2015 usando a seguinte especificação, IDE e implementação
- Especificação
 - JavaServer Pages 2.3 (06/2013, JEE 7)
 - JavaServer Pages 2.2 (12/2009, JEE 6)
 - JavaServer Pages 2.1 (05/2006, JEE 5)
 - JavaServer Pages 2.0 (11/2003, JEE 1.4)
- IDE
 - JDK 8u40
 - NetBeans 8.0.2 na distribuição Java EE
- Implementação
 - GlassFish 4.1 (vem no NetBeans)

Agenda

- O que são JSP?
- Elementos de script
- Variáveis predefinidas
- Cookies
- Inclusão de arquivos
- Encaminhamento de requisições

O que são JSP

- Páginas HTML com códigos adicionais (.jsp)
 - Códigos tratam requisições e geram **conteúdo dinâmico** (ex.: montar listagem de clientes lidos da base de dados)
 - O conteúdo dinâmico é **apresentado** no cliente como uma **página HTML convencional**
 - As páginas JSP são e são **transformadas em Servlets automaticamente**

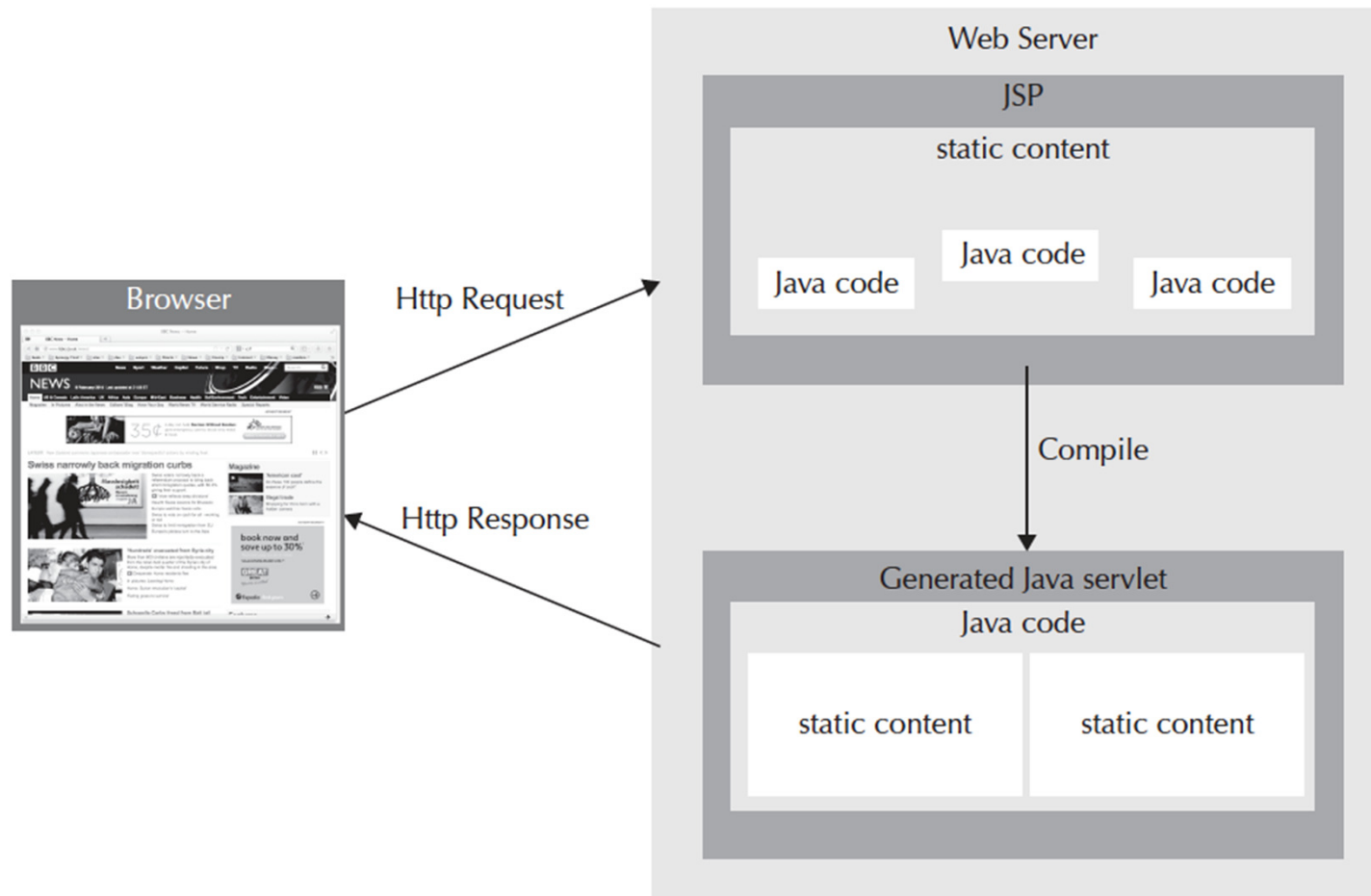
Comparação com Servlets

- Facilita o desenvolvimento
 - Criar páginas JSP é mais fácil do que criar aplicações completas
 - Código Java é escrito no HTML (no Servlet é o oposto)
 - Mais fácil de alterar do que um Servlet (classe) e não precisam ser compiladas pelo desenvolvedor antes de entrarem em produção, como em um Servlet
- Indicado para apresentação, quando se tem **muito HTML e pouco Java**

Não confundir!!!

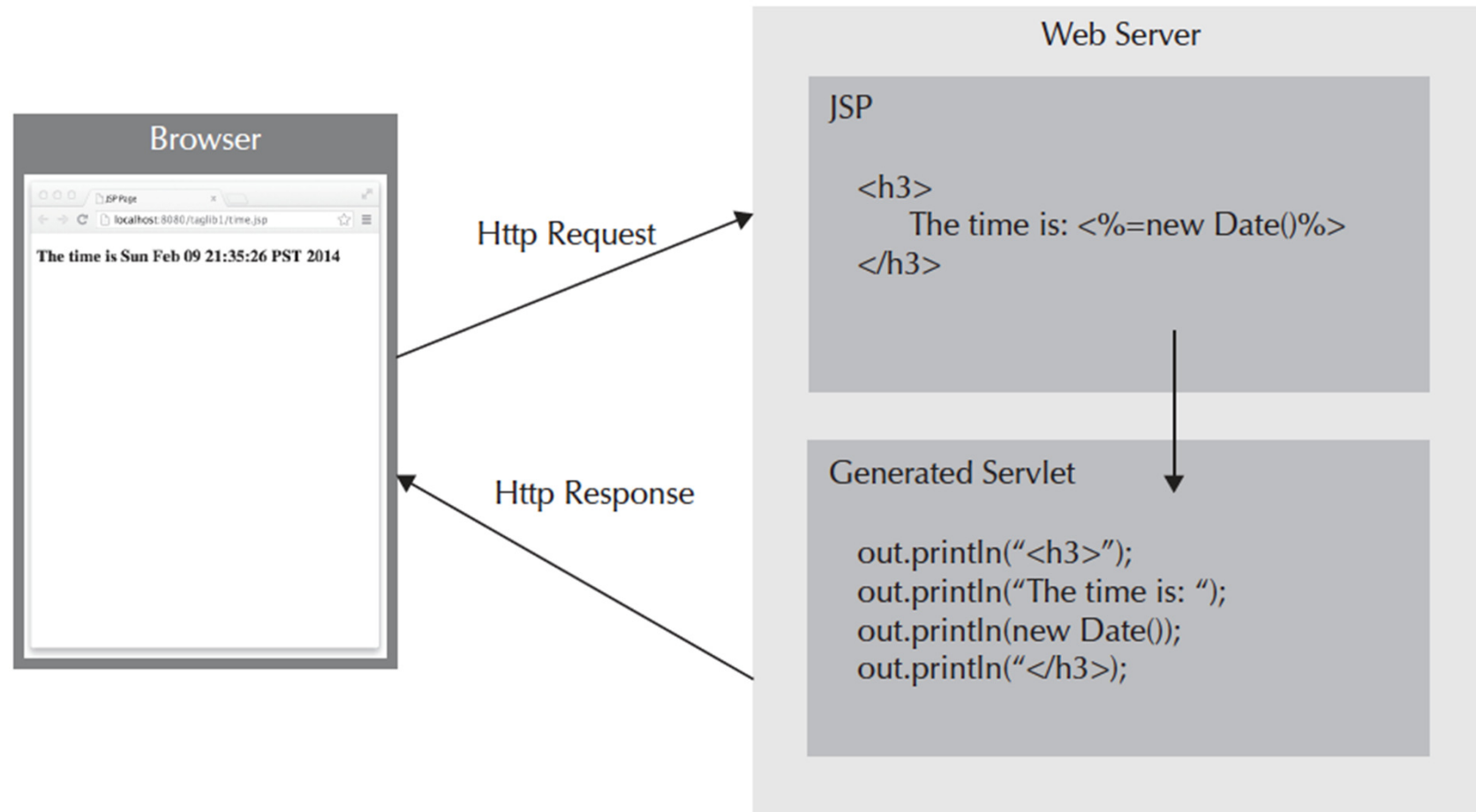
- JSP é uma tecnologia do lado do servidor (*server-side*)
- Tempo de tradução
 - Primeiro acesso
 - Transformação em Servlet
- Tempo de requisição
 - Todo acesso
 - Execução do Servlet associado
- Pode combinar com HTML, Servlets e outras classes Java

Compilação JSP → Servlet



Fonte: livro Java EE 7: The Big Picture

Compilação JSP → Servlet



Fonte: Livro Java EE 7: The Big Picture

Elementos de script

- Comentários
 - HTML: São exibidos na página HTML gerada
`<!-- isto é um comentário -->`
 - JSP: Não são inseridos na página HTML gerada
`<%-- isto é um comentário escondido --%>`
- Declarações
 - Definem variáveis ou métodos para uso subsequente
`<%! int totalVisitas = 0; %>`
`<%! public int getTime() { return time; } %>`

Elementos de script

- Scriptlets

- Código Java a ser executado

```
<% for (int i = 0; i < 10; i++) { %>
```

```
<p>Teste de script</p>
```

```
<% } %>
```

- Expressões

- Valores inseridos na página HTML gerada

O total de visitas é `<%= ++total %>`

A data atual é `<%= new java.util.Date() %>`

Elementos de script

- Diretivas
 - Permite definir propriedades gerais do JSP processadas no momento da tradução para Servlet

```
<%@ ... %>
```
- Include
 - Inclusão de arquivos na página

```
<%@ include file="rodape.htm" %>
```
- Page
 - Importação de pacotes de classes Java utilizadas

```
<%@ page import="java.util.*,java.io.*" %>
```

 - Informações de configuração para geração

```
<%@ page contentType="text/plain" %>
```

```
<%@ page language="java" %>
```

```
<%@ page session="true" %>
```

- Outras opções disponíveis: buffer, autoflush, info, errorPage, etc.

Variáveis implícitas

- É possível usar variáveis internas pré-definidas
- **request** – representa a requisição HTTP
 - Tipo: `HttpServletRequest`
- **response** – representa a resposta HTTP
 - Tipo: `HttpServletResponse`
- **session** – representa a sessão HTTP associada à requisição
 - Tipo: `HttpSession`
- **out** – representa a saída de escrita na página gerada
 - Tipo: `JSPWriter`
- **application** – estrutura de dados compartilhada
 - Tipo: `ServletContext`
- Confira outros objetos internos pré-definidos !
 - `config`, `exception`, `pageContext`, `page`

Objeto interno *request*

- Recepção de dados
 - Permite a recepção de dados provenientes de formulários dispostos em páginas HTML (métodos ***getParameter***, ***getParameterNames***, ***getParameterValues***)
 - Permite a verificação do método de envio (POST/GET) dos dados de formulário (método ***getMethod***)
 - Permite verificar se a conexão entre o cliente e o servidor é segura (método ***isSecure***)

```
nome = request.getParameter ("Nome")  
endereco = request. getParameter ("Endereco")  
bairro = request.getParameter ("Bairro")  
telefone = request.getParameter ("Telefone")
```

Nomes de controles
do formulário que
disparou o script JSP

Objeto interno *response*

- Responsável pela manipulação do cabeçalho HTML
 - **addHeader (name, value), setHeader (name, value) e addCookie(cookie)**, entre outros métodos
- Direcionamento da aplicação para outras páginas
 - **sendRedirect(location)**

Objeto interno *session*

- Gerenciamento da memória de sessão:
 - Permite armazenar (***setAttribute***) e recuperar valores (***getAttribute***) da memória de sessão da aplicação
 - Cada valor é referenciado por seu nome, e a lista de todos os nomes pode ser obtida com ***getAttributeNames***
 - O acesso ao objeto ***session*** deve ser realizado antes de qualquer escrita na página de resposta
 - O método ***setMaxInactiveInterval*** do objeto *session* configura o tempo máximo de atividade da sessão
 - O método ***invalidate*** finaliza a sessão, eliminando todos os valores em memória

Objeto interno *out*

- Permite a impressão de código HTML para a formação de uma página no cliente (método *println*)

```
out.println("<p>Testando 1, 2, 3</p>");
```

- É equivalente a fechar a *tag* de script (%>), escrever HTML e reabrir a *tag* de script (<%)

Exemplo de elementos de script e variáveis predefinidas

```
<%@ page import="java.util.*" %>
...
<%-- Check for AM or PM --%>
<%! int time = Calendar.getInstance().get(Calendar.AM_PM); %>
<%
    String nome = request.getParameter("nome");
    out.println("Olá, " + nome);
    if (time == Calendar.AM) {
        %> Bom dia ! <%
    } else {
        out.println("Boa tarde !");
    }
%>
...
```

Exercício

- Criar JSP Alo mundo (alomundo.jsp)
 - listando números de 0 a 99
- Acessar o JSP em
 - <http://localhost:8080/exercicio/alomundo.jsp>

Inclusão de arquivos

- “<%@ include ... %>”
 - Inclui a página alvo em **tempo de tradução**
 - Precisa traduzir novamente a página se uma página incluída for modificada
 - Pode usar as definições feitas nas páginas incluídas
- “<jsp:include .../>”
 - Inclui a página alvo em **tempo de requisição**
 - Não precisa traduzir a página se uma página incluída for modificada
 - Exemplo
`<jsp:include page="rodape.html" />`

Inclusão de arquivos

- Passagem de parâmetros para “<jsp:include .../>”
 - Quando a página a ser incluída é um JSP, pode ser necessário passar parâmetros para esse página
 - O request original é automaticamente passado para a página incluída
 - É possível passar novos parâmetros com
 - <jsp:param name=... Value=... />
 - Exemplo

```
<jsp:include page="rodape.jsp">  
  <jsp:param name="cor" value="azul" />  
</jsp:include>
```
- As páginas incluídas podem ficar em WEB-INF
 - Não serão acessadas diretamente pelo cliente

Encaminhamento

- “jsp:forward ... />”
 - Redirecionar para uma nova página
`<jsp:forward page="http://www.google.com" />`
 - A página JSP origem não pode ter iniciado a escrita de resposta se for encaminhar a requisição

Raciocínio: Esse tipo de operação, assim como acesso a cookies e sessão, precisa enviar dados pelo cabeçalho. Se o corpo da página já tiver começado a ser respondido ao cliente, não será possível mais alterar o cabeçalho.

Lembre-se: A comunicação entre cliente e servidor (e vice-versa) é feita por demanda.

Exercício

- Criar um JSP para somatório, onde o valor inicial e o valor final são informados
- Informar o número de vezes que
 - O usuário acessou o serviço na mesma sessão
 - O usuário acessou o serviço no mesmo browser
 - Todos os usuários acessaram o serviço desde quando o servidor entrou no ar
- Incluir as páginas padrões cabeçalho.html e rodape.html no JSP de somatório
- Encaminhar para uma página padrão de erro caso algum parâmetro não tenha sido informado

Exercício

- Faça um JSP que registra lembretes para cada um dos usuários, e que permita a listagem de todos os lembretes do usuário
 - Use sessão para guardar os lembretes
 - Crie uma classe Lembretes

JavaServer Pages (JSP)

