Java Web com Servlets e JSPs

Java Server Pages

Softblue
cursos online

1

Tópicos Abordados

- O que é
- Elementos de um JSP
  - Scriplets
  - Expressions
  - Declarations
- Diretivas do JSP
  - Include
  - Taglib
  - Page
- JSP Actions
- Objetos implícitos
- Páginas de erro
- Inicialização e destruição de JSPs
- Passando parâmetros para JSPs

2

O Que É

- Um documento HTML é estático
- Para criar respostas dinâmicas em Java, existem os servlets
  - Dependendo da resposta, é bastante difícil codificá-la usando servlets
- Surgiram os JSPs
  - **J**ava **S**erver **P**ages
  - HTML + Código Java

3

Exemplo de JSP

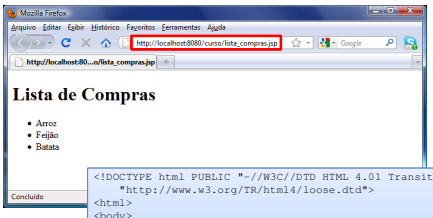
Softblue

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<%@page import="java.util.List"%>
<%@page import="java.util.ArrayList"%>
<%
  List<String> l = new ArrayList<String>();
  l.add("Arroz");
  l.add("Feijão");
  l.add("Batata");
%>
<html>
<body>
<H1>Lista de Compras</H1>

<ul>
<% for (String item : l) { %>
<li><%= item %></li>
<% } %>
</ul>
</body>
</html>
```

4

Exemplo de JSP

Softblue




```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<body>
<H1>Lista de Compras</H1>
<ul>
    <li>Arroz</li>
    <li>Feijão</li>
    <li>Batata</li>
</ul>
</body>
</html>
```

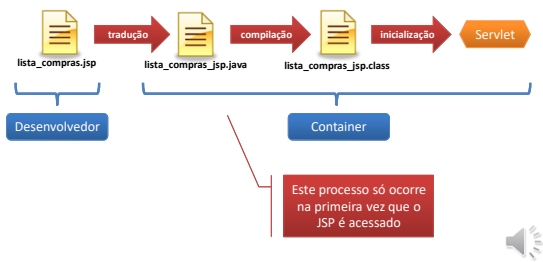
Na resposta que chega ao cliente, o documento é 100% HTML

5

O Que é Realmente um JSP?

Softblue

- Um JSP é na verdade um servlet!



6

Scriptlets

- São códigos Java inseridos no JSP
- Não estão presentes no HTML resultante
- Um scriptlet deve começar com **<%** e terminar com **%>**

```
<%
  List<String> l = new ArrayList<String>();
  l.add("Arroz");
  l.add("Feijão");
  l.add("Batata");
%>
```

O ponto-e-vírgula é necessário aqui

7

Expressions

- Também são códigos Java inseridos no JSP
- São convertidas em texto no HTML resultante
- Uma expression deve começar com **<%=** e terminar com **%>**

```
<ul>
<% for (String item : l) { %>
<li><%= item %></li>
<% } %>
</ul>
```

Uma expression deve sempre resultar numa string

Expressions não recebem ponto-e-vírgula

8

Declarations

- Também são códigos Java inseridos no JSP
- Usadas para declarar atributos e métodos de instância
- Uma declaration deve começar com **<%!** e terminar com **%>**

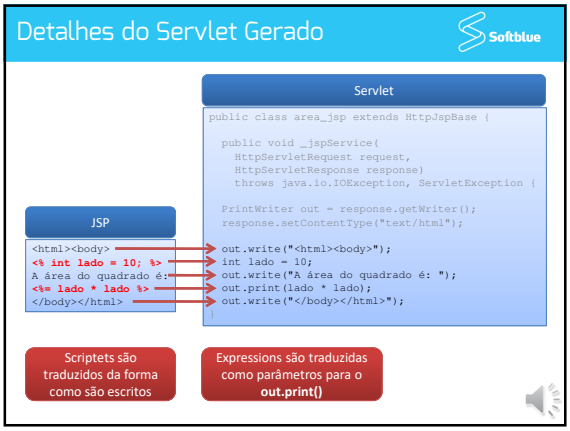
```
<%! int id = 0; %>
```

```
<%!
  int getId() {
    return id;
  }
%>
```

Não esqueça de colocar ponto-e-vírgula

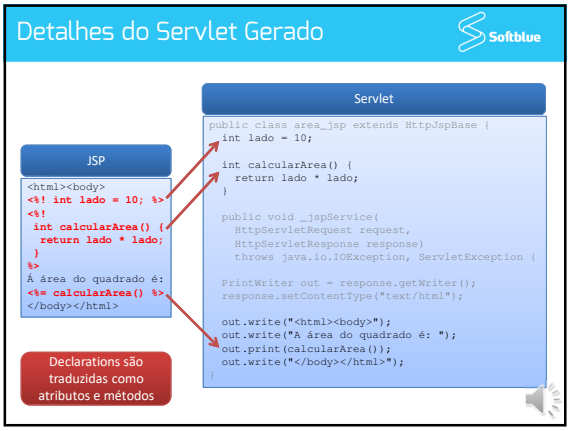
9

Detalhes do Servlet Gerado



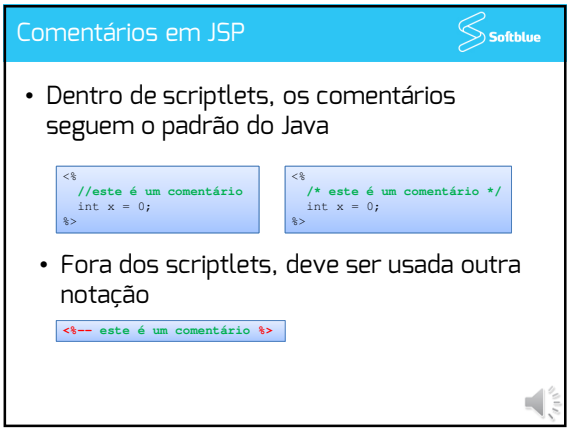
10

Detalhes do Servlet Gerado



11

Comentários em JSP



12

Diretivas do JSP

- JSP possui 3 tipos de diretivas (directives)
- Elas são identificadas por começarem por **<%@** e terminarem por **%>**

Diretiva	Descrição
include	Inclui código de um arquivo externo no JSP
taglib	Define uma tag library
page	Define propriedades da página

13

A Diretiva Include

- Permite incluir um arquivo externo na criação do JSP
- A inclusão é feita durante a fase de tradução
  - O servlet gerado já contém o conteúdo incluído

```
<%@ include file="inc/header.jsp" %>
```

14

A Diretiva Taglib

- Permite referenciar tag libraries na página
  - Tag libraries são bibliotecas de tags
  - Utilizadas para simplificar algumas tarefas e esconder o código Java

```
<%@ taglib uri="http://www.softblue.com.br/tags" prefix="sb" %>

<sb:loop id="item" value="lista">
  ...
</sb:loop>
```

15

A Diretiva Page

- Define propriedades específicas da página JSP
- É composta por diversos atributos
  - language
  - contentType
  - pageEncoding
  - import
  - isErrorPage
  - errorPage

16

A Diretiva Page

- Os atributos **language**, **contentType** e **pageEncoding** definem a linguagem e codificação

```
<%@ page language="java"
         contentType="text/html; charset=ISO-8859-1"
         pageEncoding="ISO-8859-1" %>
```

17

A Diretiva Page

- O atributo **import** é utilizado para importar classes e/ou pacotes que serão usados no JSP
- Funciona de forma bastante semelhante ao *import* do Java

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

18

A Diretiva Page

- Por padrão, alguns imports já são realizados
  - java.lang.*
  - javax.servlet.*
  - javax.servlet.jsp.*
  - javax.servlet.http.*

19

JSP Actions

- Funcionalidades para melhorar a produtividade no desenvolvimento
- São definidas pelas tags no formato **<jsp:*action*>**

Action	Descrição
<jsp:include>	Inclui outro JSP para renderização
<jsp:forward>	Redireciona a requisição para outro local
<jsp:param>	Cria parâmetros no JSP
<jsp:getProperty>	Recupera a propriedade de um Java Bean
<jsp:setProperty>	Atribui um valor a uma propriedade de um Java Bean
<jsp:useBean>	Referencia um Java Bean no JSP

20

A action <jsp:include>

- Inclui o conteúdo de outro arquivo (HTML, JSP, servlet, etc.)
- A inclusão é feita durante a renderização
  - A diretiva *<%@ include %>* faz a inclusão na fase de tradução

```
<html>
<body>
<jsp:include page="header.jsp" />
...
</body>
</html>
```

Inclui o arquivo *header.jsp* na geração do HTML de retorno

21

A action <jsp:forward>

- Permite redirecionar a requisição para outro local
  - HTML, JSP, servlet, etc.

```
<jsp:forward page="result.jsp" />
```

Redireciona para o arquivo *result.jsp*

22

Objetos Implícitos

- Como um JSP é um servlet, ele possui acesso à objetos que um servlet acessaria
- Estes objetos existem de forma implícita no JSP

23

Objetos Implícitos

Objeto Implícito	Classe do Objeto	
out	JspWriter	Dados na saída
application	ServletContext	Configuração
config	ServletConfig	
exception	JspException	Apenas para páginas de erro
request	HttpServletRequest	Escopo de dados
response	HttpServletResponse	
session	HttpSession	
pageContext	PageContext	
page	Object	

24

Páginas de Erro

- Os atributos **isErrorPage** e **errorPage** possibilitam o direcionamento para uma página de erro caso alguma exceção inesperada ocorra
- **isErrorPage** deve ser usado pela página que representa a página de erro
- **errorPage** deve indicar uma página de erro para que haja o redirecionamento no caso de erro

25

Páginas de Erro

lista_compras.jsp

```
<%@ page errorPage="error.jsp" %>

<html>
<body>
<%
  Object o = null;
  o.toString();
%>
</body>
</html>
```

Este código vai gerar uma *NullPointerException*

ibuto **errorPage** define JSP chamar em caso de exceção

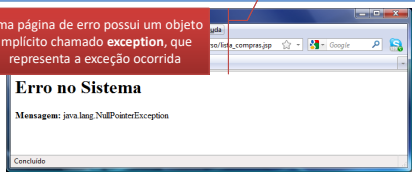
26

Páginas de Erro

error.jsp

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

<html>
<body>
<H1>Erro no Sistema</H1>
<STRONG>Mensagem: </STRONG><%= exception.toString() %>
</body>
</html>
```

Uma página de erro possui um objeto implícito chamado **exception**, que representa a exceção ocorrida



27

Inicialização e Destruição de JSPs

- Em servlets, o container chama os métodos
  - **init()**: ao inicializar o servlet
  - **destroy()**: ao destruir o servlet
  - **service()**: ao atender uma requisição
- Como um JSP é um servlet, o container também chama métodos semelhantes
  - **jspInit()**: ao inicializar o JSP
  - **jspDestroy()**: ao destruir o JSP
  - **_jspService()**: ao atender uma requisição

28

Inicialização e Destruição de JSPs

- É possível sobrescrever os métodos **jspInit()** e **jspDestroy()**
- O método **_jspService()** não deve ser sobrescrito

Usar declaration

```
<%!
  public void jspInit() {
    //inicializar o que for necessário
  }

  public void jspDestroy() {
    //destruir o que for necessário
  }
%>
```

29

Passando Parâmetros para JSPs

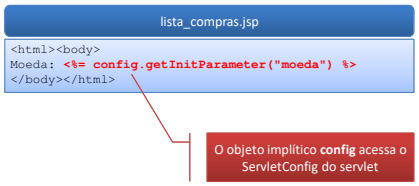
- Assim como servlets, JSPs também podem receber parâmetros de inicialização

web.xml

```
<servlet>
    <servlet-name>ListaCompras</servlet-name>
    <jsp-file>/lista_compras.jsp</jsp-file>
    <init-param>
        <param-name>moeda</param-name>
        <param-value>R$</param-value>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>ListaCompras</servlet-name>
    <url-pattern>/lista_compras.jsp</url-pattern>
</servlet-mapping>
```

30

Passando Parâmetros para JSPs



31

Não use código Java em JSPs

- Apesar de JSPs terem sido criados para possibilitar a mistura de HTML e código Java, escrever código Java em JSPs não é uma boa prática
  - Dificulta o trabalho de web designers, que não são programadores
  - Para páginas complexas, o código fica confuso
  - Dificuldade de manutenção

32

Não use código Java em JSPs

- Qual a alternativa?
  - EL (Expression Language)
  - JSTL (Java Server Pages Standard Tag Library)
  - Tag libraries customizadas

33