Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Curso: Bacharelado em Ciência da Computação
Disciplina: Tópicos de Java para a web
Prof. Corneli Gomes Furtado Júnior
Aula realizada no dia 18/03/2020

JAVA SERVER PAGES

JSP é o acrônimo para Java Server Pages, uma linguagem de script com especificação aberta que tem como objetivo primário a geração de conteúdo dinâmico para páginas da Internet. Podemos ao invés de utilizar HTML para desenvolver páginas Web estáticas e sem funcionalidade, utilizar o JSP para criar dinamismo. É possível escrever HTML com códigos JSP embutidos. Como o HTML é uma linguagem estática, o JSP será o responsável por criar dinamismo. Por ser gratuita e possuir especificação aberta possui diversos servidores que suportam a linguagem. Em nossas aulas faremos uso do Tomcat. O JSP necessita de servidor para funcionar por ser uma linguagem Server-side script, o usuário não consegue ver a codificação JSP, pois esta é convertida diretamente pelo servidor, sendo apresentado ao usuário apenas codificação HTML.

Uma pagina JSP possui extensão *.jsp* e consiste em uma página com codificação HTML e com codificação Java, inserida entre as *tag´s* <% e %>, denominada *scriptlets* e funcionando da seguinte forma: o servidor recebe uma requisição para uma página JSP, interpreta esta página gerando a codificação HTML e retorna ao cliente o resultado de sua solicitação.

BENEFÍCIOS DO JSP

Páginas JSP são traduzidas para servlets. Portanto, quaisquer tarefas que as páginas JSP pudessem executar poderiam também ser executadas por servlets. Todavia, essa equivalência não significa que servlets e páginas jsp sejam igualmente apropriadas para todos os cenários. A questão não é o poder da tecnologia, e a conveniência, a produtividade e o poder de sustentação de um e de outro.

O JSP fornece as seguintes vantagens sobre os servlets puros:

- **É mais fácil escrever e manter o HTML**: o seu código estático é HTML comum; sem barras invertidas ou aspas duplas adicionais.
- **Você pode usar ferramentas padrão de desenvolvimento de sites**: Por exemplo, Macromedia Dreamweaver e MS-Frontpage.
- **Você pode dividir sua equipe de desenvolvimento:** Os programadores Java podem trabalhar sobre o código dinâmico. Os desenvolvedores web podem se concentrar na camada de apresentação.

SCRIPTLETS

São instruções que são executadas cada vez que uma página é requisitada. Para tanto, basta escrevê-lo entre as tags <% e %>.

Um exemplo de scriptlet pode ser observado no quadro abaixo. Sempre que a página primeirapagina.jsp for requisitada, será exibida a data e hora da requisição. Observe que foi utilizado o objeto **out,** que é implicito (`PrintWriter out = response.getWriter();`), para escrever um conteúdo na tela.

primeirapagina.jsp

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Minha primeira página JSP</title>
</head>
<body>
<%
java.util.Date data = new java.util.Date();
out.println(data);
%>
</body>
</html>
```

COMO USAR EXPRESSÕES JSP

Uma expressão JSP é usada para inserir valores diretamente na saída. Ela possui o seguinte formato:

<%= Expressão Java %>

A expressão é avaliada, convertida para uma string, e inserida na página. Por exemplo, a página segundapagina.jsp, apresentada no Quandro 1 apresenta o mesmo resultado da primeirapagina.jsp.

segundapagina.jsp

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Minha primeira página JSP</title>
</head>
<body>
```

```
<%= new java.util.Date() %>

</body>
</html>
```

JSP ACTION

JSP actions são utilizados para controlar o comportamento do servlet engine. Você pode dinamicamente inserir um arquivo, reusar um componente JavaBeans, realizar o encaminhamento para outra página ou gerar uma página HTML.A sintaxe dos JSP Actions é apresentada abaixo:

```
<jsp:action_name attribute="value" />
```

A tabela abaixo apresenta uma lista de ações pré-definidas e suas respectivas funcionalidades.

Action	Propósito
jps:include	Inclui um arquivo no tempo em que a página é requisitada
jsp:useBean	Instancia um JavaBean
jsp:setProperty	Atribui um valor a uma propriedade do JavaBean
jsp:getProperty	Recupera o valor de uma propriedade do JavaBean
jsp:forward	Encaminha o requisitante para uma nova página

Existem dois elementos que são comuns a todas as JSP Actions: os atributos **id** e **scope.**

- id: Identifica de forma única o elemento, e permite que a ação seja referenciada dentro da página JSP. Se a ação cria a instância de um objeto, o id pode ser utilizado como referência a este objeto.

- Scope**:** este atributo define o ciclo de vida de um elemento de ação. Este parâmetro pode assumir quatro valores: (a) page, (b)request, (c)session, and (d) application.

PRATICANDO 01:

- Crie duas páginas JSP denominadas (a)data.jsp and (b) main.jsp. As páginas devem conter o seguinte conteúdo:

data.jsp

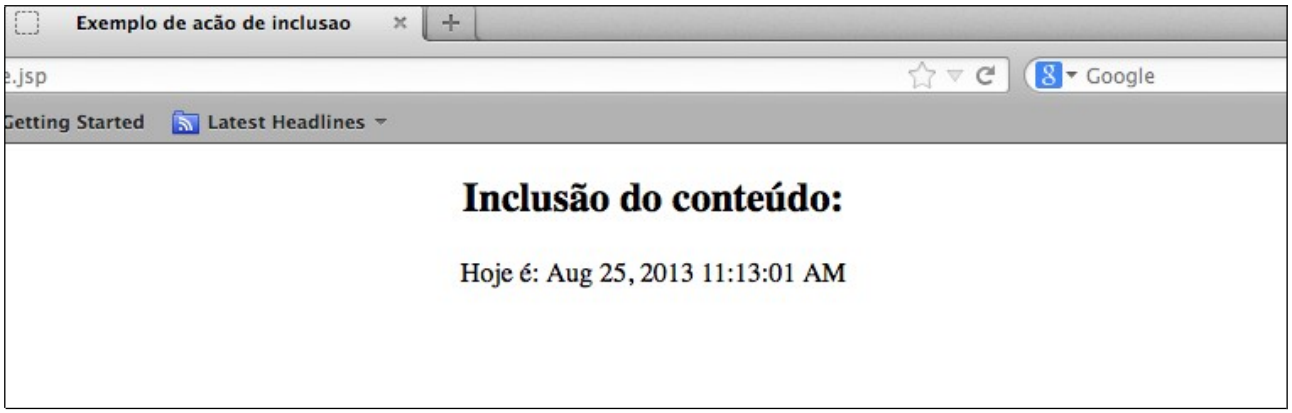
```
<%@page import="java.util.Date" %>
<p>
Hoje é: <%= (new java.util.Date()).toLocaleString()%>
</p>
```

main.jsp

```
<head>
<title>Exemplo de acão de inclusao</title>
</head>
<body>
```

```
<center>
<h2>Inclusão do conteúdo: </h2>
<jsp:include page="data.jsp" />
</center>
</body>
</html>
```

- Agora, invoque a página main.jsp e observe o resultado.



Nessa pequena prática, observamos o uso de duas actions, a *jsp:include* (responsável por fazer a inclusão de um conteúdo de uma pagina em outra) e a *jsp:import* (responsável por importar Classes que serão utilizadas no projeto).

MANIPULANDO BEANS

A ação **useBean** é muito versátil. Esta inicialmente busca por um objeto utilizando o seu id e variáveis de escopo. Se o objeto não existe, então tentará criar o objeto. A sintaxe é exibida a seguir:

```
<jsp:useBean id="name" class="package.class" />
```

A propriedade class especifica o caminho completo para o Bean.

A ação **setProperty** atribui valores às propriedades de um Bean. O Bean, obviamente, deve ter sido definido anteriormente usando a ação **useBean.**

```
<jsp:useBean id="myName" ... >
...
   <jsp:setProperty name="myName" property="someProperty" .../>
</jsp:useBean>
```

Abaixo, segue uma descrição de cada parâmetro utilizado na ação **setProperty.**

Attribute	Description
name	Designates the bean whose property will be set. The Bean must have been previously defined.
property	Indicates the property you want to set. A value of "*" means that all request parameters whose names match bean property names will be passed to the appropriate setter methods.
value	The value that is to be assigned to the given property. The the parameter's value is null, or the parameter does not exist, the setProperty action is ignored.

A ação **getProperty** é utilizada para recuperar o valor de uma dada propriedade e converte-la em string e, por fim, inseri-la em uma saída. A seguir os atributos desta ação.

Attribute	Description
name	The name of the Bean that has a property to be retrieved. The Bean must have been previously defined.
property	The property attribute is the name of the Bean property to be retrieved.

PRATICANDO 02

- Crie uma classe denominada Mensagem.java com o seguinte conteúdo:

```
package br.edu.ifce;

public class Mensagem {
	String texto;

	public String getTexto() {
		return texto;
	}

	public void setTexto(String texto) {
		this.texto = texto;
	}
}
```

- Crie um arquivo .jsp com o seguinte conteúdo:

```
<body>
	<jsp:useBean id="msg" class="br.edu.ifce .Mensagem" />
	<jsp:setProperty property="texto" name="msg" value="Mensagem exibida ao usuário"/>
	<jsp:getProperty property="texto" name="msg"/>
</body>
```

- Execute o arquivo .jsp. Você acabou de instanciar um bean, definir e recuperar informações sobre o atributo texto.

EXERCÍCIO PROPOSTO

Usando a ação <jsp:usebean> e demais ações necessárias, crie uma formulário de envio de e-mails que receba os parâmetros destinatário, assunto e corpo da mensagem e realize corretamente o envio de e-mails. Para o envio de e-mails é possível usar a API Commons Email. Faça uma pesquisa sobre essa API para integrá-la ao seu projeto.

Para retirar eventuais dúvidas, segue um exemplo de projeto semelhante ao proposto. O login é admin e senha admin.