SÃOJUDAS UNIVERSIDADE

FTCE

Faculdade de Tecnologia e Ciências Exatas

USJT - 2017 - Práticas de Programação

Professores	Teoria: Bonato, Hamilton Laboratório: Calvetti, Hamilton, Keity e Rodrigo

Aula: 7
Assunto: Expression Languages - JSTL

Conceitos básicos

Como vimos nas últimas aulas, as páginas JSP que fizemos usando código Java em scriptlets <% %> e expressões <%= %> tornam-se complexas e difíceis de manter a medida que a quantidade de código HTML, CSS e Java crescem na página. Por isso foram criadas pela Sun outras formas mais fáceis de se escrever código nas JSP: a Expression Language (EL) e as Tag Libraries (JSTL, ou JSP Tag Library). Se puder opte sempre pelo uso de EL e JSTL. Porém é importante dominar os dois conceitos, pois em sua vida profissional de programador você irá encontrar muito código escrito em scriptlets para dar manutenção.

Expression Language

Uma expressão em JSP EL tem o seguinte formato:

```
${expr}
```

onde `expr` é a expressão em si.

Você pode, por exemplo, pegar um parâmetro nome na request usando a expressão `${param.nome}`. Ou pode pegar um atributo de um *bean* (que é um Transfer Object - TO) colocado pelo Servlet na request usando `${cliente.fone}`.

Em uma expressão você pode usar os operadores lógicos, relacionais e aritméticos do Java, com a mesma sintaxe ou com uma sintaxe diferenciada. Pode também chamar funções definidas em tag libraries (na próxima seção) e ter acesso a objetos implícitos. Veja nas tabelas abaixo:

Operador	Descrição
`.`	acessa uma propriedade de um bean ou uma entrada em um Map
`[ ]`	acessa um elemento em array ou uma lista
`( )`	agrupa expressões e altera a precedência
`+`	adição
`-`	subtração
`*`	multiplicação
`/ ou div`	divisão
`% ou mod`	módulo (resto da divisão)
`== ou eq`	igual
`!= ou ne`	diferente

`< ou lt`	menor
`> ou gt`	maior
`<= ou le`	menor ou igual
`>= ou ge`	maior ou igual
`&& ou and`	e
`|| ou or`	ou
`! ou not`	não
`empty`	testa se a variável é vazia

Objeto Implícito	Descrição
pageScope	variáveis no escopo da página
requestScope	variáveis no escopo da request
sessionScope	variáveis no escopo da session
applicationScopo	variáveis no escopo da aplicação
param	parâmetros da request como String
paramValues	parâmetros da request como coleções de String
header	cabeçalho da requisição HTTP como String
headerValues	cabeçalho da requisição HTTP como coleção de String
initParam	parâmetro de inicialização do contexto
cookie	valor de cookies
pageContext	o contexto da página do objeto JSP da página atual

Funções

Você pode chamar uma função desde que ela tenha sido pré definida em uma biblioteca de tags. Por exemplo, `${ns:soma(1,2)}` retorna o resultado da função soma definida em uma taglib customizada chamada ns. Já a tag `${fn:length("Olá!!")}` retorna 5, que é o comprimento da string passada como parêmatro para função length definida na taglib de funções padrão fn, que já vem com a JSTL.

JSTL

As tag libraries, ou **taglibs**, são bibliotecas de tags, que podem ser padrão ou standard, cujo objetivo é o de permitir a escrita de código Java por meio do uso de tags, tornando assim o código mais semelhante ao HTML .

Usando taglibs você pode instanciar objetos, codificar loops e desvios condicionais e formatar datas e valores, por exemplo. Combinando as taglibs com as expressões da EL, é possível evitar tranquilamente o uso dos scriptlets na criação das JSP.

Instanciando Beans

Um Javabean é um objeto Java que contém um construtor vazio, atributos e métodos gets e sets, como vimos usando ao longo do curso. Outros autores ainda se referem a eles como POJOs (Plain Old Java Objects). Não é necessário instanciar um Cliente, mas basta se referir a ele, na EL, pelo nome usado quando ele foi passado para a JSP via request ou sessions pelo servlet.

Por exemplo, para imprimir os dados do cliente, que foram colocados na request via `request.setAttribute("cliente", cliente)`, use a Expression Language:

```
${cliente.id}
```

```
${cliente.nome}
${cliente.fone}
${cliente.email}
```

Veja que é desnecessário usar o método get. Se você criou os gets e os sets usando direitinho a convenção de nomenclatura dos métodos, basta escrever o nome dos atributos, que nos beans são chamados de propriedades. Veja também que o nome do objeto é o valor que você colocou em id.

Instalação

Os JARs da JSTL não vem com o Tomcat. Para instalá-los baixe os arquivos javax.servlet.jsp.jstl-1.2.1.jar e javax.servlet.jsp.jstl-api-1.2.1.jar de http://jstl.java.net/ e copiá-los para o diretório WEB-INF/libs do WebContents do seu projeto Eclipse. A mesma pasta onde está gravado o jar do MySQL.

Importando para a página

Para usar taglips na sua JSP você precisa colocar o import no topo da página. Para usar as tags core, coloque

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

Para usar formatação, inclua também:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
```

Você já deve ter notado que o prefix serve para escolher a biblioteca correta a ser carregada.

For Each

Vamos fazer um loop que percorra um ArrayList<ClienteTO> e imprima dos dados de cada cliente. Para isso, basta se referenciar ao ArrayList usando o mesmo nome que ele foi colocado na request. O código fica assim:

```
<c:forEach var="cliente" items="${lista}">
    ${cliente.id}
    ${cliente.nome}
    ${cliente.fone}
    ${cliente.email}
</c:forEach>
```

Se precisar de um contador, acrescente a propriedade varStatus="contador" na tag forEach. A variável contador será incrementada a cada volta do loop.

If

Para fazer um desvio condicional if use a tag abaixo:

```
<c:if test="${not empty cliente.fone}">
     <strong>Fone:</strong>${cliente.fone}
</c:if>
```

Note que não existe um else. Para fazermos as vezes do else devemos escrever outro if.

```
<c:if test="${empty cliente.fone}">
      <strong>Telefone não informado.</strong>
</c:if>
```

Switch-case

O comando, na verdade, é choose:

```
<c:choose>
    <c:when test="${not empty cliente.fone}">
        <strong>Fone:</strong>${cliente.fone}
    </c:when>
    <c:otherwise>
      <strong>Telefone não informado.</strong>
    </c:otherwise>
</c:choose>
```

Importando páginas

No código que estamos fazendo teremos que repetir o navbar do Bootstrap em cada uma das páginas HTML e nas JSP copiando e colando o código de modo a termos um menu. Para modularizar melhor, podemos criar um JSP que contenha somente as tags do navbar (sem as demais tags da estrutura do html) e importar esta JSP nas outras JSP, colocando a tag abaixo no lugar onde se quer que aquele código apareça:

```
<c:import url="menu.jsp"/>
```

Formatação de datas

Para formatar a saída de uma objeto do tipo java.uti.Date, use a tag abaixo:

```
<fmt:formatDate value="${cliente.dataNascimento.time}"
    pattern="dd/MM/yyyy" />
```

Para saber mais sobre tags e como criar taglibs customizadas, veja o capítulo referente a este assunto no tutorial de JEE (está na bibliografia). Há também um bom guia de referência rápida na Universidade de Iowa (também está na bibliografia).

Exemplo: Clientes.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Visualizar Cliente</title>

    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
</head>
```

```
<body>
        <!-- Barra superior com os menus de navegação -->
             <c:import url="Menu.jsp"/>
        <!-- Container Principal -->
        <div id="main" class="container">
            <h3 class="page-header">Visualizar Cliente #${cliente.id}</h3>
            <div class="row">
                <div class="col-md-12">
                    <p><strong>Nome</strong>
                    </p>
                    <p>
                        ${cliente.nome}
                    </p>
                </div>
            </div>
            <div class="row">
                <div class="col-md-6">
                    <p><strong>Celular</strong>
                    </p>
                    <p>
                        ${cliente.fone}
                    </p>
                </div>
                <div class="col-md-6">
                    <p><strong>E-Mail</strong>
                    </p>
                    <p>
                        ${cliente.email}
                    </p>
                </div>
            </div>
            <hr />
            <div id="actions" class="row">
                <div class="col-md-12">
                    <a href="index.jsp" class="btn btn-default">Voltar</a>
                </div>
            </div>
        </div>
        <script src="js/jquery.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
</body>

</html>
```

Exemplo: index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>cerveja.biz - Criar Cliente</title>
```

```
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
</head>

<body>
    <!-- Barra superior com os menus de navegacao -->
       <c:import url="Menu.jsp"/>
    <!-- Container Principal -->
    <div id="main" class="container">
        <h3 class="page-header">Incluir Cliente</h3>
        <!-- Formulario para inclusao de clientes -->
        <form action="ManterCliente.do" method="post">
            <!-- area de campos do form -->
            <div class="row">
                <div class="form-group col-md-12">
                    <label for="nome">Nome</label>
                    <input type="text" class="form-control" name="nome" id="nome"
required maxlength="100" placeholder="nome completo">
                </div>
            </div>
            <div class="row">
                <div class="form-group col-md-6">
                    <label for="fone">Celular</label>
                    <input type="tel" class="form-control" name="fone" id="fone"
maxlength="15" pattern="(?:\(\d{2}\)|\d{2})[- ]?\d{5}[- ]?\d{4}" placeholder="opcional;
celular com ddd no formato (99) 99999-9999">
                </div>

                <div class="form-group col-md-6">
                    <label for="email">E-Mail</label>
                    <input type="email" class="form-control" name="email" id="email"
required maxlength="60" placeholder="email obrigatório">
                </div>
            </div>
            <hr />
            <div id="actions" class="row">
                <div class="col-md-12">
                    <button type="submit" class="btn btn-primary" name="acao"
value="Criar">Salvar</button>
                    <a href="index.html" class="btn btn-default">Cancelar</a>
                </div>
            </div>
        </form>
    </div>
    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
</body>

</html>
```

Exemplo: Menu.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
    <nav class="navbar navbar-inverse navbar-fixed-top">
        <div class="container-fluid">
```

```
            <div class="navbar-header">
                <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="index.jsp">Cadastro</a>
            </div>
            <div id="navbar" class="navbar-collapse collapse">
                <ul class="nav navbar-nav navbar-right">
                    <li><a href="index.jsp">Clientes</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
```

Exercícios

1. Revisão da Teoria

a. Copie os arquivos jar da pasta WebContent >WEB-INF>lib do exemplo para a pasta WebContent>WEB-INF>lib do seu projeto de cadastro de clientes para habilitar o uso de JSTL.

b. Copie o Cliente.jsp, Menu.jsp e index.jsp para a pasta WebContent. Apague o index.html desta mesma pasta. É necessário que seja JSP para importar o menu.

3. Faça deployment e rode o projeto.

2. Exercício para Nota (correção em aula)

a. Copie os arquivos jar da pasta WebContent >WEB-INF>lib do exemplo para a pasta WebContent>WEB-INF>lib do seu projeto de cadastro de países para habilitar o uso de JSTL.

2. Refatore todos seu JSP usando Expression Language e JSTL e removendo todos os scriptlets <%%> e as expressões <%=%>. Use também os imports para não precisar mais copiar e colar o menu. Crie um Menu.jsp para importar. Veja que terá que transformar todas as suas páginas em JSP, mesmo as estáticas.

Bibliografia:

SIGGELKOW, Bill. **JSTL Quick Reference**. Disponível online em <http://homepage.cs.uiowa.edu/~slonnegr/wpj/jqr.pdf>. Acessado em 10/04/16.

Oracle. **The Java EE 5 Tutorial: using JSTL**. Disponível online em <http://docs.oracle.com/javaee/5/tutorial/doc/bnake.html>. Acessado em 10/04/16.