

Lição 6



Páginas JSP Avançadas

Objetivos

Ao final desta lição, o estudante será capaz de:

- Compreender a linguagem de expressão das páginas JSP
- Utilizar a biblioteca JSTL



Linguagem de Expressão JSP

- Foi introduzida com a especificação JSP 2.0
- Fornece uma sintaxe simples e clara para escrever expressões
- Executam lógica simples ou acesso a variáveis de escopo
- Para acessar à propriedade de um *JavaBean*:

```
<% String name = user.getLastName(); %>
```

ou

```
<jsp:getProperty name="user"  
  property="lastname"/>
```



Linguagem de Expressão JSP

- Usando o primeiro modo:
 - Expostos a programação Java
 - Usar os métodos `get()` dos beans
 - O desenvolvedor precisa estar consciente do tipo da propriedade Java
- Usando o segundo modo:
 - Este método é mais programação – neutro
 - A propriedade de um *JavaBean* é longa e pesada
 - O método apresenta a sua saída diretamente ao navegador cliente
 - Manipulações não podem ser feitas através do valor recuperado



Linguagem de Expressão JSP

- Usando o modelo EL, pode ser acessada como:

`$\{user.lastName\}$`

- Programação neutra
- Curta e ao ponto
- Pode ser usada para exibir a saída diretamente ao usuário
- Pode ser usada para apresentar o valor de *tags* personalizadas para serem processadas

Syntaxe EL

- Construtor EL pode ser um literal ou uma expressão fechada entre em \$ (and)
- Literais
 - Literais suportados e tratadas como em Java: Boolean, Long, Float, String e null
- Define operadores padrão que podem ser aplicados em literais
 - Operadores aritméticos básicos (+, -, *, /, %)
 - Operadores lógicos (& &, | |, !)
 - Operadores de comparação(==, !=, <, <=, >, > =)



Syntaxe EL

- As palavras reservadas de EL são:
 - and (&&), eq (==), gt (>), ge (>=), or (||), ne (!=), le (<=), lt (<), div (/), mod (%)
 - true e false
 - instanceof
 - null
 - Empty
- **Exemplos de expressões EL**
 - `${'JEDI'}` – formato com o valor da String "JEDI"
 - `${5 + 37}` - formato com o valor 42
 - `$ ((10% 5) == 2)` - formato com o valor true
 - `${empty ''}` - formato com o valor true



Acessando Variáveis de Escopo e Propriedades

- Embora seja capaz de formatar expressões simples de literais
- EL mostra a sua utilidade quando cessa e processa variáveis em diferentes escopos
- Acessar variáveis é simples com EL:
 - As variáveis são simplesmente referenciadas pelo nome
 - Propriedades de um *JavaBean*, métodos, e coleções podem ser acessados a partir de um nome de variável usando a notação "."
 - Métodos e propriedades são acessadas da mesma forma



Acessando Variáveis de Escopo e Propriedades

- Exemplo:

`${user.lastName}`

- Acessa um *JavaBean* que pode ser referenciado pelo nome *user* e recupera o valor de sua propriedade *lastName*
- O escopo do *JavaBean* não importa
 - EL executa busca de escopo
 - Se foi especificado com o nome de um *JavaBean* que não existe em nenhum escopo, null será retornado



Objetos Implícitos EL

- Embora o âmbito automático procure tornar mais fácil para os desenvolvedores a escrita de código, especificando uma variável do escopo
- Explicitamente facilita a manutenção do nosso construtor por futuros desenvolvedores
- EL nos fornece vários objetos implícitos que representam um Map de objetos dentro dos diferentes escopos:
 - pageScope
 - requestScope
 - sessionScope
 - applicationScope



Objetos Implícitos EL

- Além disso, EL também define as seguintes objetos implícitos:
 - **param**: um Map que representa uma requisição a parâmetros de nomes e seus valores
 - **paramValues**: um Map que representa o nomes dos parâmetros para arrays de String e seus valores
 - **header**: um Map que representa os cabeçalhos disponíveis em uma dada requisição
 - **headerValues**: um Map que representa os cabeçalhos disponíveis em uma dada requisição
 - **Cookies**: Mapeia os cookies disponíveis em um pedido



Notação []

- Além da Notação ".", EL prevê também o "[]"
- Notação de acesso a variáveis, métodos e arrays
- Em muitos aspectos, as duas notações são semelhantes:

`${user.lastName}`

`${user[lastName]}`

JSTL

- Afasta do uso de *scriptlets* dentro das páginas JSP
- Tags personalizadas:
 - Permite graus de personalização e extensibilidade através do uso de *tags* personalizadas
 - Prevêem um caminho de apresentar funcionalidades reusáveis dentro da página JSP
 - Série de bibliotecas disponíveis
- É fornecido um padrão de biblioteca de tags, chamado Java Standard Tag Library ou JSTL

Incluindo JSTL na nossa Aplicação

- Passaremos agora para o NetBeans



Biblioteca Core

- Fornece funcionalidade básica para o projeto WEB
- Pode ser dividida em subcategorias:
 - Tags de uso geral
 - Repetição
 - Condicionais
 - Manipulação de URL
- Em cada página JSP que fizer uso da funcionalidade da biblioteca Core
- A seguinte diretiva deve ser acrescentada:

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

 - Usa as tags da biblioteca Core através do prefixo c



Tags de Uso Geral

- Executa tarefas simples e comuns, embora um deles tenha se tornado irrelevante com a liberação da especificação JSP 2.0
- As *tags* incluídas no conjunto de uso geral são:
 - out
 - set
 - remove
 - catch

<c:out>

- Avalia o resultado de uma expressão
- Envia o resultado diretamente a saída da página
- Funciona da mesma maneira da ação padrão <jsp:getProperty>, não sendo necessário um *JavaBean* para acessar à funcionalidade
- Obsoleta com a liberação da especificação JSP 2.0:
 - expressões EL podem ser avaliadas no stream de saída em qualquer parte da página JSP sem utilizar qualquer tags JSP 2.0

<c:set>

- Executa quase a mesma funcionalidade da ação <jsp:setProperty>
- Capaz de definir valores diretamente em um *JavaBean*
- Capaz de definir uma variável em um determinado escopo
- Pode ser usada posteriormente pelo JSP ou em outra parte da aplicação

<c:remove>

- Fornece uma maneira de remover variáveis de um escopo definido
 - Tem dois atributos:
 - scope – O escopo da variável a ser removida.
 - var – O nome da variável a ser removida do escopo definido
- `<c:remove var="myString" scope="session"/>`
- Assim como para a tag <c:set>, o uso dessa *tag* é limitado



<c:catch>

- Fornece capacidade de manipulação de erros em áreas específicas de uma JSP
- É simples de usar:
 - Coloque o conteúdo JSP que poderia potencialmente gerar erros no corpo da tag <c:catch>
- Tem apenas um atributo:
 - **var**: Define o nome que será utilizado para expor a exceção gerada
 - A variável assim criada terá escopo *page*
 - Pode ser acessada mesmo depois que o bloco de captura encerrou



Tags de Uso Geral

- Passaremos agora para o NetBeans



Tags de Repetição

- Proporciona alternativas para embutir laços de repetição em uma página JSP como *scriptlets*
- Seguintes *tags*:
 - forEach
 - forToken



<c:forEach>

- *tag* de repetição mais comumente usada
- Permite iteração em:
 - arrays primitivos
 - instancias de:
 - java.util.Collection
 - java.util.Iterator
 - java.util.Map
 - java.util.Enumeration
- Executa a repetição através de cada elemento do alvo
- Expõe o valor atual da repetição para o código JSP
- Contido no corpo da *tag*

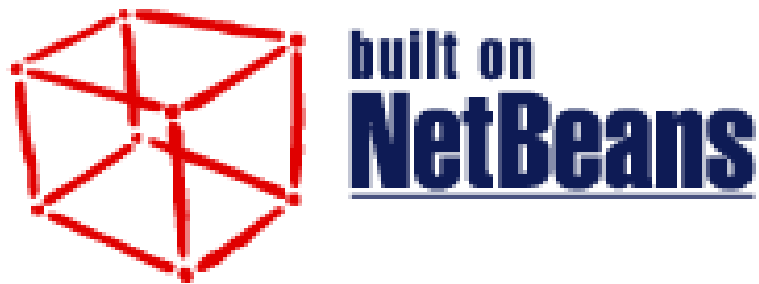


<c:forTokens>

- As outras interações tag são fornecidas por JSTL
- Tag capta uma String e analisa em tokens com base em um delimitador
- Todos os atributos da tag forEach são compartilhados por esta tag, com um acréscimo:
 - *delims*: Define o delimitador a ser usado quando analisar o alvo na String em *tokens*
- Os itens dos atributo tem agora um novo objetivo para esta tag
- String que define o atributo separador do laço

Tags de Repetição

- Passaremos agora para o NetBeans



Condicionais

- Realizam a função fornecida pela regra if, else-if, ...
- Afirmações que podem ser encontradas no padrão *Java*
- O uso destas *tags* no padrão de declaração Java conduzem a limpeza do código
- Dois conjuntos de condicionais principais:
 - `<c:if>`
 - `<c:choose>`, `<c:when>` e `<c:otherwise>`



<c:if>

- Permite a execução do conteúdo de sua descrição com base na resposta do valor
- Se a expressão avaliada for verdadeiro o código interno será executado
- Se a expressão avaliada for falso, o código interno não será executado

<c:choose>, <c:when>, <c:otherwise>

- As *tags* trabalham em conjunto para fornecer funcionalidades de alterar o fluxo de código dentro da aplicação
- Uma ou mais *tags* são colocadas para a escolha de uma delas
 - A escolha da *tag* para avaliar o teste dos atributos acontece quando a *tag* executa o corpo dela para verificar se é verdadeiro
 - Se não houver sucesso na *tag* correspondente, outra *tag* é escolhida, se esta estiver incluída

Condicionais

- Passaremos agora para o NetBeans



Manipulando URL

- JSTL fornece algumas tags personalizadas que possuem capacidades para manipulação de URL
- Estas *tags* constroem e melhoram ações que estão disponíveis nas páginas *JSPs*:
 - import
 - param
 - url

<c:import>

- Funciona da mesma maneira que o include JSP
- A ação no JSP inclui apenas as páginas no interior da aplicação WEB incluído no conteúdo atual da página
- A *tag* <c:import> permite a especificação de URLs absolutas apontar para recursos externos

<c:import>

- O atributo que usa esta *tag* é:
 - url: O valor da URL de importação, pode ser:
 - **Relativa**, aponta para um recurso da aplicação WEB
 - **Absoluta**, capaz de apontar para recursos externos
- Contém uma série de outros atributos
- São utilizados para modificações em modo de programação do conteúdo do recurso

<c:param>

- Utilizada em conjunto com uma série de outras *tags* dentro da URL para manipulação de todo grupo
- Não pode ser utilizada sozinha, apenas como uma *tag* de outras *tags* disponíveis em JSTL
- Usada para fornecer a forma de incluir parâmetros e valores dentro de uma URL

<c:url>

- Usado para fornecer de forma automática uma URL com as informações necessárias para gerenciar as sessões
- Atributos relevantes:
 - value: O URL para ser processada
- Esta *tag* NÃO é recomendada a URLs
- Se faz necessária quando se detecta que o cliente navegador não suporta *cookies*
- Esta tag pode ter zero a muitas *tags* <c:param>, para inclusão parâmetro



Manipulando URL

- Passaremos agora para o NetBeans



Sumário

- Linguagem de Expressão JSP
- Sintaxe EL
- Acessando Variáveis de Escopo e Propriedades
- Objetos Implícitos EL
- Biblioteca Core da JSTL

Parceiros

- Os seguintes parceiros tornaram JEDITM possível em Língua Portuguesa:



University of the Philippines
Java
Research and
Development
Center

