

## Lab 07.1 - JSTL

**JSTL** é um conjunto padrão de **bibliotecas de tags** que estão disponíveis a partir da **versão 1.4 do Java**. O núcleo **JSTL** inclui **tags de internacionalização, formatação, XML e outras**. Neste laboratório, você vai aprender a usar estas **tags** em suas aplicações web, é necessário fazer o download das bibliotecas **jstl.jar**, **standart.jar**, após o download adicione-as no diretório **lib** em **WebContent**.

### Exercícios

**Exercício 1:** Usando algumas **tags jstl**.

**Exercício 2:** Alterando a aplicação Bancaria para utilizar tags **jstl** e **Internacionalização** .

### Exercício 1 Tags jstl

#### 1.1. Utilizando a tag <c:if>

1. Crie um novo **Projeto Dynamic Web** chamado **JSTL** e adicione a biblioteca.
2. Para o uso das tags **jstl** deve ser adicionado no inicio das paginas jsp a linha abaixo.

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

3. Crie um arquivo **verifica.jsp** e adicione o código abaixo. Analise o código e veja como esta sendo recuperado os dados.

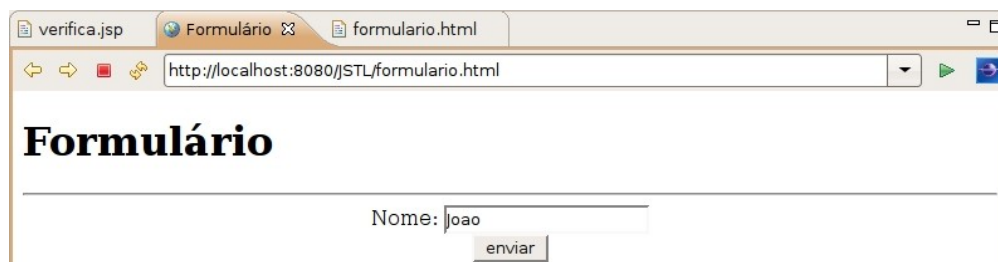
```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Verifica</title>
</head>
<body>
    <h1>Verifica se o campo nome foi preenchido</h1>
    <hr>
    <c:if test="${empty param.nome}">
        Você não preencheu seu nome!
    </c:if>
    <c:if test="${not empty param.nome}">
        O seu nome é ${param.nome}
    </c:if>
</body>
</html>
```

3. Crie outro arquivo chamado **formulario.html** conforme abaixo. Veja que o **action** do formulário está referenciando a **JSP**, ou seja, quando for feito **submit** no formulário, o resultado será o retorno da página **verifica.jsp**.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Formulário</title>
</head>
<body>
    <h1>Formulário</h1><hr>
    <form action="verifica.jsp">
```

```
<center>Nome: <input type="text" name="nome" size="20"><br>
<input type="submit" value="enviar"></center>
</form>
</body>
</html>
```

4. Execute o arquivo **formulario.html** e teste a aplicação.



## 1.2. Usando a Tag <c:import>

1. Crie um arquivo chamado **import.jsp** conforme o código abaixo.

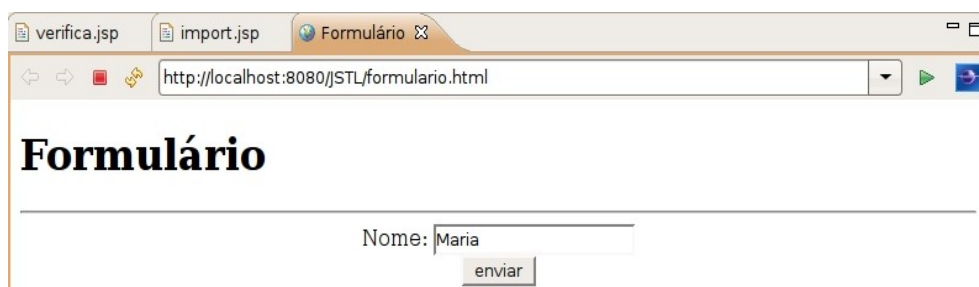
```
<%@ taglib uri="http://java.sun.com/jstl/core" prefix="c"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Exemplo de Imports</title>
</head>
<body>
<h1>Exemplo de Imports</h1>
```

```
<hr>
<c:import url="verifica.jsp" />
</body>
</html>
```

2. Modifique o arquivo **formulario.html** para que o valor do atributo **action** da **tag form** seja **import.jsp** conforme abaixo.

```
<form action="import.jsp">
  <center>Nome: <input type="text" name="nome" size="20"><br>
  <input type="submit" value="enviar"></center>
</form>
```

3. Execute o arquivo **formulario.html** novamente e veja o resultado.



4. Modifique o arquivo **import.jsp** para que o valor do atributo **url** seja **http://www.google.com.br** conforme abaixo.

```
<c:import url="http://www.google.com.br" />
```

5. Execute o arquivo **import.jsp** e veja o resultado.



### 1.3. Trabalhando com <c:forEach> e <c:forTokens>

1. Crie o arquivo **foreach.jsp** e adicione o código abaixo. Analise e tente entender o código.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Exemplo - <c:forEach></title>
</head>
<body>
<h1>Exemplo - <c:forEach></h1>
<hr>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page import="java.util.Vector" %>

<h3>Interagindo sobre um faixa de numeros</h3>
<c:forEach var="item" begin="1" end="10">
    ${item}
</c:forEach>

<% Vector<String> v = new Vector<String>();
    v.add("um"); v.add("dois"); v.add("tres");
    v.add("quatro"); v.add("cinco");
    pageContext.setAttribute("vector", v);
%>

<h3>Interagindo sobre um Vetor</h3>

<c:forEach items="${vector}" var="item" >
    ${item}
</c:forEach>
</body>
</html>
```

## Exercício 2 *jstl* e internacionalização

### 1.1. Aplicando tags *jstl* a aplicação Banco-WebView

1. Agora vamos aplicar o uso de **jstl** a nossa aplicação. É necessário adicionar as bibliotecas **jstl.jar** e **standard.jar** a nosso projeto dentro do diretório **WebContent > lib** como foi feito no exercício 1.
2. Vamos ajustar as nossas paginas **jsp** adicionando as seguintes linhas abaixo para utilizarmos as tags **jstl**. Note que usaremos a biblioteca **fmt**, esta biblioteca contem as propriedades necessárias para trabalharmos com internacionalização.

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

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

3. Após adicionar o código acima nos arquivos **jsp**, localize e remova os trecho de código abaixo no arquivo **cabecalho.jsp**.

```
<%@page import="java.util.ArrayList"%>  
<%@page import="threeway.projeto.modelo.Log"%>  
<%@page import="java.util.List"%>
```

4. Agora não utilizaremos mais código **JAVA** dentro do **jsp**, vamos alterar a listagem de logs dentro do arquivo **cabecalho.jsp**, localize e remova o código abaixo.

```
<tbody>  
<%  
List<Log> dados = (ArrayList<Log>) request.getSession().getAttribute("dadosLog");  
  
for (Log log : dados){  
  
%>  
    <tr>  
        <td><% out.print(log.getDescricao()); %></td>  
        <td><% out.print(log.getDataFormatada()); %></td>  
    </tr>  
    <%}%>  
</tbody>
```

5. No lugar do código removido iremos adicionar o código.

```
<tbody>  
<c:forEach items="${dadosLog}" var="log">  
    <tr>  
        <td>  
            ${log.descricao}  
        </td>  
        <td>  
            ${log.dataFormatada}  
        </td>  
    </tr>  
</c:forEach>  
</tbody>
```

Não utilizamos código java na tela jsp, apenas a tag `<c:forEach>` da biblioteca jstl. Foi recuperado da sessão o item **dadosLog**, note que utilizamos linguagem de expressão para mapear a propriedade **items** do `<c:forEach>` e para imprimir os valores da lista dentro de cada `<td>`.

## 1.2. Internacionalização.

1. Vamos criar um Servlet com o nome **MudarIdioma** responsável por alterar o idioma de nossa aplicação.

```
@WebServlet("/MudarIdioma")
public class MudarIdioma extends AppController{

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        //Recupera a linguagem selecionada na tela
        final String locale = request.getParameter("linguagem");

        //Definindo o locale
        Config.set(request.getSession(), Config.FMT_LOCALE, locale);

        response.sendRedirect("index.jsp");

    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        doGet(request, response);

    }

}
```

2. Crie um pacote dentro **src** com nome **threeway.projeto.messages** e adicione dois arquivos **.properties** **message\_en\_US.properties**, **message\_pt\_BR.properties** dentro dos arquivos **adicione o código conforme abaixo**.

1. **message\_pt\_BR.properties**

```
titulo.cabecalho = Projeto Banco 3Way NetWorks
titulo.aba.cliente = Manter Cliente
titulo.aba.operacoesBancarias = Operações Bancarias

saudacao = Bem vindo, Sr.(a)

botao.brasil = Brasil
botao.eua = Estados Unidos
botao.sair = Sair
```

2. **message\_en\_US.properties**

```
titulo.cabecalho = Project Bank 3Way NetWorks
titulo.aba.cliente = Register Client
titulo.aba.operacoesBancarias = banking

saudacao = Welcome, Mr.(a)

botao.brasil = Brazil
botao.eua = United States
botao.sair = Exit
```

Todos os textos e labels da aplicação devem ser mantidos dentro dos arquivos properties, assim a aplicação conseguirá traduzir de acordo com a localização, o detalhe maior é que as Keys devem ser iguais em todos os arquivos properties.

3. Agora vamos altera os arquivos **jsp** para que todos os campos de texto sejam os textos dos arquivos properties, altere o arquivo **cabecalho.jsp** para que fique conforme abaixo.

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav navbar-right">
    <li><a>
      <form action="Logout" id="formIcons">
        <span>
          <fmt:message key="saudacao" />
        </span>
        <b>
          <script type="text/javascript">
            document.write(getCookie());
          </script>
        </b>
      <input type="submit" name="action" class="btn btn-xs btn-danger" value="<fmt:message key="botao.sair" />" />
      <button type="submit" name="action" class="btn btn-xs btn-default" value="info" data-toggle="modal"
        data-target="#modalInfo">
        
      </button>
    </form>

    <!-- Adição do formulario de alteração de idioma -->
    <form method="post" action="MudarIdioma">

      <button type="submit" title="<fmt:message key="botao.brasil"/>" name="linguagem" class="btn btn-xs
        btn-default" value="pt_BR">
        
      </button>
      <button type="submit" name="linguagem" title="<fmt:message key="botao.eua"/>" class="btn btn-xs btn-
        default" value="en_US">
        
      </button>

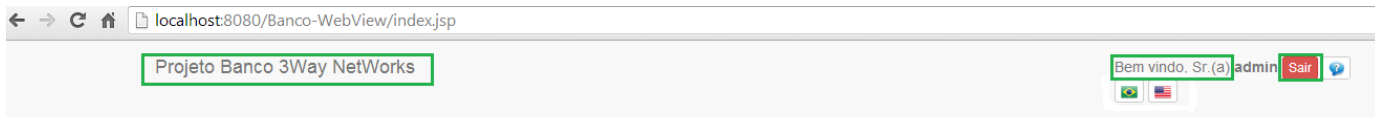
    </form>
  </a></li>
</ul>
</div>
```

Agora os textos serão referenciados pelo componente **<fmt:message key="" />**, onde deve ser preenchida a propriedade key.

4. Altere o arquivo **web.xml** adicionando o código abaixo para fazer o mapeamento do arquivo properties.

```
<context-param>
  <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
  <param-value>threeway.projeto.messages.message</param-value>
</context-param>
```

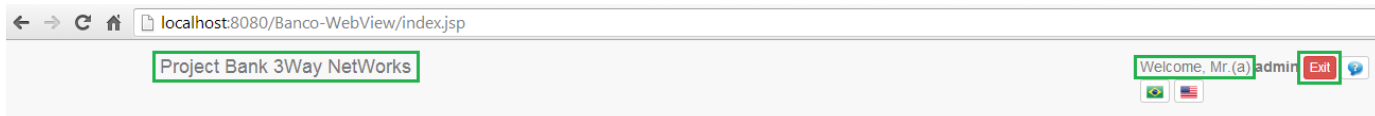
5. Pronto, agora reinicie o servidor, acesse a aplicação e realize os testes de mudança de idioma.



Banco:  
Agência:

Manter Clientes

Operações Bancárias



Banco:  
Agência:

Manter Clientes

Operações Bancárias

6. O sistema já está convertendo os textos do cabeçalho para o inglês agora vamos alterar os outros texto das paginas usando os arquivos *properties* e a tag **jstl** `<fmt:message key=""/>`, lembre-se que todo texto adicionado em um *propertie* português deve ser adicionado no *propertie* inglês com a mesma **key**=**value**.