08

JSP para listar empresas

Transcrição

Iremos extrair o HTML de `ListaEmpresaSevlet.java`, assim como fizemos em `NovaEmpresaServlet.java`, com o auxílio do arquivo JSP, que contém um scriptlet que renderiza o código HTML dinamicamente.

Não teremos novidades, faremos o `requestDespacher()` e as requisições para incluir valores e assim por diante. Você pode, inclusive, interromper a aula e tentar realizar a criação do JSP sozinho e depois comparar os resultados. Não explicaremos nenhum conceito novo neste capítulo, mas poderemos praticar e sedimentar os conhecimentos que já adquirimos.

Faremos a refatoração do código de `ListaEmpresasServlet.java`, mas antes faremos uma pequena recapitulação: a requisição do navegador chega até o Servlet, onde ocorre o processamento com o banco de dados e em seguida usamos um método despachador `requestDispatcher()` para chamar o arquivo JSP.

Lembrando que é possível, no Servlet, acoplar o valor da requisição e no JSP podemos acessar esse valor. Dessa forma teremos a divisão das responsabilidades entre os arquivos, uma vez que o Servlet armazenará o código Java e o JSP o código de visualização.

Na pasta `WebContent` criaremos um novo arquivo "JSP File", que chamaremos de `listaEmpresas.jsp`.

```
<%@ page language="java" contentType="text/html; charset=ISO-8
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

</body>
</html>
```

COPIAR CÓDIGO

Teremos acima do código um pequeno scriptlet que define o `charset`. Relembrando o Java IO, estamos nos referindo ao *input* e ao *output*, portanto aqui também é importante definirmos o *encoding *de como interpretar os caracteres. No mundo web isso pode ser delicado, pois os caracteres podem surgir de maneira errada caso o `charset` não esteja definido corretamente. Como estamos no sistema operacional Windows, foi inserido o `ISO-8859-1`.

Em `ListaEmpresasServlet` coletaremos o trecho de código a partir da linha `PrintWriter out = reponse.getWriter();` e o inseriremos dentro de `listaEmpresas.jsp`.

```
@WebServlet("/listaEmpresas")
public class ListaEmpresasServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServle
        throws ServletException, IOException {

    Banco banco = new Banco();
    List<Empresa> lista = banco.getEmpresas();
```

```
        PrintWriter out = reponse.getWriter();
        out.println("<html><body>");
        out.println("<ul>");
        for (Empresa empresa : lista) {
            out.println("<li>" + empresa.getNome() + "<li>");
        }
        out.println("</ul>");
        out.println("</body></html>");
        }
    }
```

COPIAR CÓDIGO

Retiraremos a linha do `PrintWriter()`, afinal ela não é necessária nesse contexto. Em seguida, retiraremos `out.println("<html><body>")`, e retiararemos `<ul>` do método em que estava inserido. Faremos algumas formatações até o que código fique com este aspecto:

```
<%@ page language="java" contentType="text/html; charset=ISO-8
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <ul>
            for (Empresa empresa : lista) {
            out.println("<li>" + empresa.getNome() + "</li>");
        }
    </ul>

</body>
</html>
```

COPIAR CÓDIGO

Precisamos fazer o laço e imprimir o nome da empresa. Para o laço, usaremos um pequeno scriptlet. A ideia é que para cada empresa é preciso renderizar um `<li>`. Usaremos o método `getNome()` para realizar a impressão sem a necessidade do uso do `out`.

```
<%@ page language="java" contentType="text/html; charset=ISO-8
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <ul>
    <%
        for (Empresa empresa : lista) {
    %>
        <li><%= empresa.getNome() %></li>
    <%
        }
    %>
    </ul>

</body>
</html>
```

COPIAR CÓDIGO

Ainda precisamos resolver a `Empresa` e `lista`, que apresentam erro segundo o compilador. Ao acessarmos `ListaEmpresasServlet`, notaremos a existência da variável `lista`, que contem o nome das empresas. Precisamos inseri-la na requisição, e faremos isso por meio de `request.setAttribute()`, e o atributo

será o apelido `empresas` e objeto `lista`. Estamos inserindo a lista inteira de empresas cadastradas em nosso "banco de dados".

Em seguida, usaremos o `RequestDispacher() rd`, proveniente da requisição, portanto usaremos `getRequestDispacher()` e passaremos o local de destino, no caso, o arquivo JSP `listaEmpresas.jsp`. Para finalizar, acionaremos `rd.forward(request,response)` para que a requisição seja de fato enviada.

```
@WebServlet("/listaEmpresas")
public class ListaEmpresasServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServle
        throws ServletException, IOException {

    Banco banco = new Banco();
    List<Empresa> lista = banco.getEmpresas();

    request.setAttribute("empresas", lista);

    RequestDispatcher rd = request.getResquestDispatcher("/list
    rd.forward(request,response);
```

COPIAR CÓDIGO

Em teoria, nosso Servlet está pronto. Voltaremos até `listaEmpresas.jsp` e continuaremos trabalhando com `lista` e `Empresa`. No arquivo `novaEmpresaCriada.jsp`, havíamos utilizado o `request.getAttribute("empresa")` para solucionar a questão, e no caso, tomaremos a mesma medida.

Porém, dessa vez utilizaremos `List` com o atributo `Empresa`, sendo a variável `lista` proveniente de `request.getAttribute()`, que receberá o atributo `empresas`.

Por fim, faremos a importação dos elementos por peio de um scriptlet seguido de `@`, que significa que trata-se de uma declaração da página. Escreveremos, na parte superior do código `page import="java.util.List, br.com.alura.gerenciador.servlet.Empresa"` .

```
<%@ page language="java" contentType="text/html; charset=ISO-8
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.util.List, br.com.alura.gerenciador.servl
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <ul>
    <%
        List<Empresa> lista = (List<Empresa>)request.getAttribu
        for (Empresa empresa : lista) {
    %>
        <li><%= empresa.getNome() %></li>
    <%
        }
    %>
    </ul>

</body>
</html>
```

COPIAR CÓDIGO

Podemos fazer o nosso código se estruturar de uma maneira mais elegante e que facilite sua manutenção, mas por ora conseguimos separar as responsabilidades e não há mais código HTML no Servlet, nossa preocupação principal. Iniciaremos o Tomcat e testaremos as modificações no navegador.

Acessaremos a URL http://localhost:8080/gerenciador/listaEmpresas (http://localhost:8080/gerenciador/listaEmpresas), e veremos as duas empresas cadastradas em nosso banco: **Alura** e **Caelum**, e esta lista é baseada no Servlet.

Para provar esse estado, iremos inserir uma mensagem `Lista de empresas` na tela via arquivo JSP:

```
<body>
    Lista de empresas: <br />
    <ul>
    <%
        List<Empresa> lista = (List<Empresa>)request.getAttribu
        for (Empresa empresa : lista) {
    %>
        <li><%= empresa.getNome() %></li>
    <%
        }
    %>
    </ul>

</body>
</html>
```

COPIAR CÓDIGO

Ao recarregarmos a página no navegador, teremos a seguinte mensagem exibida:

> Lista de empresas:
>
> Alura
>
> Caelum

O conteúdo do arquivo JSP está sendo exibido. Podemos ver essa ligação também acessando o código fonte da página, com o conteúdo apresentado de maneira formatada:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    Lista de empresas: <br />
    <ul>

        <li>Alura</li>

        <li>Caelum</li>
    </ul>

</body>
</html>
```

COPIAR CÓDIGO

Nossa próxima tarefa durante as aulas será melhorar o arquivo JSP, de forma que não seja mais necessário escrever scriptlets a todo momento, o que não parece ser uma boa prática - afinal, ao escrevermos códigos mais complexos, isso pode se tornar um pesadelo de manutenção.