



Despachando a requisição

Transcrição

Seguiremos aprendendo mais sobre JSP, isto é, *Java Server Pages*. A motivação de usarmos esse tipo de formato, é de não escrevermos códigos HTML dentro do código Java, como era o caso de `NovaEmpresaServlet.java`. Escrevemos o código HTML em uma página específica: `novaEmpresaCriada.jsp`, em que podemos gerar uma interação dinâmica.

```
<%  
    //scriptlet  
    String nomeEmpresa = "Alura";  
    System.out.println(nomeEmpresa);  
%>  
  
<html><body>  
Empresa <%= nomeEmpresa %> cadastrada com sucesso!  
</body></html>
```

[COPIAR CÓDIGO](#)

Já que temos a mensagem HTML separada, podemos excluir àquela que estava em `NovaEmpresaServlet.java`.

```
@WebServlet("/novaEmpresa")  
public class NovaEmpresaServlet extends HttpServlet {  
  
    private static final long serialVersionUID = 1L;  
  
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        // ...  
    }  
}
```

```
        throws ServletException, IOException {  
            System.out.println("Cadastrando nova empresa");  
  
            String nomeEmpresa = request.getParameter("nome");  
            Empresa empresa = new Empresa();  
            empresa.setNome(nomeEmpresa);  
  
            Banco banco = new Banco();  
            banco.adiciona(empresa);  
  
        }  
  
    }
```

[COPIAR CÓDIGO](#)

Testaremos no navegador e verificaremos se o cadastro de empresas continua operando normalmente após retirarmos esse trecho de HTML.

Acessando <http://localhost:8080/gerenciador/formNovaEmpresa.html> (<http://localhost:8080/gerenciador/formNovaEmpresa.html>), cadastraremos a empresa Google no formulário. Lembrando que será chamado NovaEmpresa() neste processo, como podemos verificar no código fonte:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="ISO-8859-1">  
<title>Insert title here</title>  
</head>  
<body>  
  
    <form action="/gerenciador/novaEmpresa" method="post">  
  
        Nome: <input type="text" name="nome" />  
  
        <input type="submit" />
```

```
</form>

</body>
</html>
```

[COPIAR CÓDIGO](#)

Ao pressionarmos "Enviar", seremos direcionados para uma página em branco, não há qualquer mensagem de confirmação de cadastro ou de erro. Ao analisarmos o código fonte, veremos que ele também está vazio.

Na console teremos a mensagem `Cadastrando nova empresa`, isto é, o Servlet foi chamado, mas não o arquivo `novaEmpresCriada.jsp`, que contém a informação a ser impressa na página. Vamos entender com detalhes o aconteceu.

Uma requisição é enviada pelo servidor para o Servlet, então é lido o parâmetro e feito o processamento no banco de dados. Contudo, todo conteúdo de interface do HTML está armazenado no arquivo JSP, portanto de nada adiantará devolvermos uma resposta direta por meio do Servlet.

O que faremos é criar uma conexão entre o Servlet e o JSP, para que este último envie a resposta ao navegador. Isto é, primeiro a requisição chega até o Servlet e nele é executado todo o processamento, mas o trabalho de resposta é feito pelo arquivo JSP. Com isso, realizamos uma separação de responsabilidades.

Para executarmos esse processo usaremos a `request`, que possui um objeto específico para despachar a requisição para o JSP, trata-se do `getRequestDispatcher()`. Precisamos indicar ao despachador o destino da requisição, isto é, o arquivo `/novaEmpresaCriada.jsp`. Nos será devolvido um `RequestDispatcher`, portanto pegaremos a referência `rd` e realizaremos a importação.

Resta adicionarmos um item que ativa o percurso dessa requisição, o método `forward()` que receberá os parâmetros de `request` e `response`.

```
@WebServlet("/novaEmpresa")
public class NovaEmpresaServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        System.out.println("Cadastrando nova empresa");

        String nomeEmpresa = request.getParameter("nome");
        Empresa empresa = new Empresa();
        empresa.setNome(nomeEmpresa);

        Banco banco = new Banco();
        banco.adiciona(empresa);

        //chamar o JSP
        RequestDispatcher rd = request.getRequestDispatcher("/index.jsp");
        rd.forward(request, response);
    }
}
```

[COPIAR CÓDIGO](#)

Neste ponto não importa mais se o código HTML terá uma linha ou mil, já que estamos direcionando a requisição para o arquivo JSP. Agora, iremos testar a aplicação.

De volta ao formulário no navegador, tentaremos cadastrar a empresa "Facebook". Ao pressionarmos o botão "Enviar", teremos a seguinte mensagem exibida na tela:

Empresa Alura cadastrada com sucesso!

Isto é, o JSP foi chamado, mas ainda estramos nos referindo à empresa **Alura** - ou seja, temos um sucesso parcial, porque a requisição chegou de fato ao despachador. Ao analisarmos o código de `novaEmpresaCriada.jsp`, perceberemos que o nome `Alura` está fixo. O nome da empresa, de forma dinâmica, é acessado via Servlet por meio de `nomeEmpresa`.

Em `NovaEmpresaServlet` estamos acionando o método `forward()` que leva `request` e `response` como parâmetros. Dentro do Servlet temos o nome da empresa que devemos passar para o JSP, e precisamos encontrar um elemento em comum entre esses dois arquivos, que no caso é requisição.

No Servlet, agruparemos o nome da empresa em `request`. Para isso utilizaremos `setAttribute("empresa", empresa.getNome())`, contudo perceba que assinatura do método `setAttribute()` recebe um string `name` e um `Object`. A string `name` funciona como apelido, que será `empresa` e em seguida o objeto `empresa`.

```
@WebServlet("/novaEmpresa")
public class NovaEmpresaServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        System.out.println("Cadastrando nova empresa");

        String nomeEmpresa = request.getParameter("nome");
        Empresa empresa = new Empresa();
        empresa.setNome(nomeEmpresa);

        Banco banco = new Banco();
        banco.adiciona(empresa);

        //chamar o JSP
        RequestDispatcher rd = request.getRequestDispatcher("novaEmpresaCriada.jsp");
        rd.forward(request, response);
    }
}
```

```
request.setAttribute("empresa", empresa.getNome());  
rd.forward(request, response);  
  
}  
  
}
```

[COPIAR CÓDIGO](#)

O apelido `empresa` será usado no arquivo JSP. Usaremos a `request` e o método `getAttribute()` que receberá `empresa`. O método devolverá um objeto, e sabemos que se trata de uma string, indicando uma referência mais específica.

```
<%  
//scriptlet  
String nomeEmpresa = (String)request.getAttribute("empresa");  
System.out.println(nomeEmpresa);  
%>  
  
<html><body>  
Empresa <%= nomeEmpresa %> cadastrada com sucesso!  
</body></html>
```

[COPIAR CÓDIGO](#)

Salvaremos as modificações. No Servlet, estamos inserindo informações e no arquivo JSP estamos recolhendo os valores, dessa forma os dois funcionam em conjunto e poderemos definir uma variável que o JSP sozinho não reconheceria.

Testaremos a aplicação no navegador: no formulário, cadastraremos a empresa "Facebook" e teremos a mensagem:

Empresa Facebook cadastrada com sucesso!

Isto é, conseguimos passar uma informação do Servlet para o JSP.

Aproveitamos o melhor do Servlet e do JSP, o que precisamos trabalhar ainda é o no uso do scriptlet, pois, da forma como está, teremos muita dificuldade com a manutenção do sistema. Outro ponto é que `ListaEmpresaServlet` ainda está utilizando HTML. Nas próximas aulas nos debruçaremos mais sobre este problema e refatoraremos o Servlet.