



Conhecendo Expression Language

Transcrição

Neste curso, como já sabemos, estamos aprendendo sobre o uso de Servlets ao passo que construímos uma aplicação web. Neste ponto, já temos o Servlet com uma responsabilidade e o JSP com outra, e também já criamos o modelo de cadastro de empresas. O Servlet se encarrega do protocolo HTTP, e então chamamos o modelo que controlará fluxo da requisição. Finalmente, usamos o *Dispatcher* para evocar o JSP, isto é, a visualização do conteúdo na página.

Nas últimas aulas, refatoramos `ListaEmpresasServlet.java` e utilizamos o `RequestDispatcher` para chamar o arquivo JSP e realizar o laço para listar as empresas utilizando scriptlets. Nossa próxima etapa é simplificar esse código, pois não é uma boa prática utilizar código Java por meio de scriptlets dentro do HTML. Devemos sempre pensar na construção de aplicações maiores e refletir sobre sua manutenção.

Vamos inserir o código Java no HTML para realizarmos ações dinâmicas no momento de listar as empresas. Começaremos a refatorar `novaEmpresaCriada.jsp` :

```
<%  
    //scriptlet  
    String nomeEmpresa = (String)request.getAttribute("empresa");  
    System.out.println(nomeEmpresa);  
%>
```

```
<html>
```

```
<body>
    Empresa <%= nomeEmpresa %> cadastrada com sucesso!
</body>
</html>
```

[COPIAR CÓDIGO](#)

Buscaremos uma forma de imprimir `nomeEmpresa` utilizando menos scriptlets. Para isso, usaremos uma outra linguagem bastante simples - mais precisamente de uma sintaxe, marcada por `${ }`, e utilizada para definir expressões. Uma expressão escrita entre as chaves `{ }` é interpretada pelo navegador e impressa na tela. Faremos um pequeno teste incluindo a expressão `${ 3 + 3 }` em nosso código:

```
<%
//scriptlet
String nomeEmpresa = (String)request.getAttribute("empresa");
System.out.println(nomeEmpresa);
%>

<html>
    <body>
        Empresa ${ 3 + 3 } cadastrada com sucesso!
    </body>
</html>
```

[COPIAR CÓDIGO](#)

No navegador, usaremos a URL

<http://localhost:8080/gerenciador/novaEmpresaCriada.jsp>

(<http://localhost:8080/gerenciador/novaEmpresaCriada.jsp>). Perceba que a mensagem exibida na tela será `Empresa 6 cadastrada com sucesso!`. O conteúdo expresso foi interpretado por esta pequena linguagem, e, por fim, impresso na tela.

O que queremos é exibir o conteúdo do atributo `empresa`, que faz parte da requisição. Substituiremos `${ 3 + 3 }` por `${ empresa }`. Trata-se de uma linguagem simples, porém poderosa: ela analisa a requisição e busca o atributo `empresa`, isto é, trata-se da mesma ação que estamos realizando na parte superior do código, portanto iremos apagá-la.

Dessa forma teremos apenas HTML em nosso arquivo, o que o torna mais elegante.

```
<html>
  <body>
    Empresa ${ empresa } cadastrada com sucesso!
  </body>
</html>
```

[COPIAR CÓDIGO](#)

Ao acessarmos <http://localhost:8080/gerenciador/novaEmpresaCriada.jsp> (<http://localhost:8080/gerenciador/novaEmpresaCriada.jsp>), teremos a mensagem:

Empresa cadastrada com sucesso!

Ou seja, nenhum nome de empresa. Isso ocorreu porque não definimos o valor da variável. Acessaremos o formulário por meio da URL <http://localhost:8080/gerenciador/formNovaEmpresa.html> (<http://localhost:8080/gerenciador/formNovaEmpresa.html>), e partir desse formulário enviaremos uma requisição para definir no nome da empresa a ser cadastrada, no caso, a empresa **Google**.

Lembrando que o Servlet lerá o parâmetro e inserirá o atributo na requisição. Quando a linguagem de expressão tentar ler o atributo, ele já estará lá.

Ao acessarmos novamente

<http://localhost:8080/gerenciador/novaEmpresaCriada.jsp>
(<http://localhost:8080/gerenciador/novaEmpresaCriada.jsp>), teremos a mensagem:

Empresa Google cadastrada com sucesso!

Isso significa Servlet guardou no nome da empresa dentro da requisição e a linguagem de expressão conseguiu acessar o valor.

Desse modo, simplificamos nosso código utilizando essa linguagem. Contudo, a *expression language* não realiza laços, portanto precisamos aprender outras ferramentas para substituir de fato todo o scriptlet do código.