

JSP Fragment

Transcrição

Apesar de não termos comentado antes, a página de itens do carrinho, também tem o mesmo problema e ainda está com o código antigo. Faremos com que ela também use o `pageTemplate` que acabamos de criar. Vejamos como fica a página de itens do carrinho após uso do `pageTemplate`:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://www.springframework.org/tags" prefix="s" %>
<%@ taglib tagdir="/WEB-INF/tags" prefix="tags" %>

<tags:pageTemplate titulo="Livros de Java, Android, IOs, Mobile e muito mais...">

<section class="container middle">
    <h2 id="cart-title">Seu carrinho de compras</h2>
    <table id="cart-table">
        <colgroup>
            <col class="item-col" />
            <col class="item-price-col" />
            <col class="item-quantity-col" />
            <col class="line-price-col" />
            <col class="delete-col" />
        </colgroup>
        <thead>
            <tr>
                <th class="cart-img-col"></th>
                <th width="65%">Item</th>
                <th width="10%">Preço</th>
                <th width="10%">Quantidade</th>
                <th width="10%">Total</th>
                <th width="5%"></th>
            </tr>
        </thead>
        <tbody>
            <c:forEach items="${carrinhoCompras.itens }" var="item">
                <tr>
                    <td class="cart-img-col">
                        
                    </td>
                    <td class="item-title">${item.produto.titulo}</td>
                    <td class="numeric-cell">${item.preco}</td>
                    <td class="quantity-input-cell">
                        <input type="number" min="0" readonly="readonly" id="quantidade" name="quantidade" value="1" />
                    </td>
                    <td class="numeric-cell">${carrinhoCompras.getTotal(item)}</td>
                    <td class="remove-item">
                        <form action="${s:mvcUrl('CCC#remover').arg(0, item.produto.id).arg(1, item.id)}>
                            <input type="image" src="/excluir.png" alt="Excluir" title="Excluir" />
                        </form>
                    </td>
                </tr>
            </c:forEach>
        </tbody>
    </table>
</section>
```

```

        </c:forEach>
    </tbody>
    <tfoot>
        <tr>
            <td colspan="4">
                <form action="${s:mvcUrl('PC#finalizar').build()}" method="post">
                    <input type="submit" class="checkout" name="checkout" value="Finalizar" />
                </form>
            </td>
            <td class="numeric-cell">${carrinhoCompras.total}</td>
            <td></td>
        </tr>
    </tfoot>
</table>

<h2>Você já conhece os outros livros da Casa do Código?</h2>
<ul id="collection" class="related-books">
    <li class="col-left">
        <a href="/products/livro-plsql" class="block clearfix book-suggest" data-book="livro-plsql">
            
        </a>
    </li>
</ul>
<h2>
    <a href="http://www.casadocodigo.com.br">Veja todos os livros que publicamos!</a>
</h2>
</section>

</tags:pageTemplate>

```

E tudo funciona perfeitamente, podemos adicionar produtos ao carrinho, ver a página de detalhes de cada produto, mas ao tentar finalizar uma compra, teremos um erro:

```

HTTP Status 403 - Invalid CSRF Token 'null' was found on the request parameter '_csrf' or header 'X-CSRF-TOKEN'.


---


type Status report
message Invalid CSRF Token 'null' was found on the request parameter '_csrf' or header 'X-CSRF-TOKEN'.
description Access to the specified resource has been forbidden.


---


Apache Tomcat/7.0.70

```

Isso acontece porque não atualizamos os formulários da página do carrinho depois de termos adicionado o **Spring Security** em nosso projeto. Para resolver o problema, basta fazer o import da `taglib` de formulário do *Spring* e substituir a tag `form` do HTML pela tag `form:form` do *Spring*.

fazendo o import da `taglib` na página `itens.jsp`

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
```

Alterando as tags de formulários da página `itens.jsp`. No formulário de remover produtos do carrinho.

```

<form:form action="${s:mvcUrl('CCC#remover').arg(0, item.produto.id).arg(1, item.tipoPreco).build()}" method="post">
    <input type="image" src="/excluir.png" alt="Excluir" title="Excluir" />
</form:form>

```

E também no de finalização da compra

```
<form:form action="${s:mvcUrl('PC#finalizar').build()}" method="post">
    <input type="submit" class="checkout" name="checkout" value="Finalizar compra" />
</form:form>
```

Trabalhando com Fragments JSP

Uma necessidade comum em alguns projetos web é que determinados scripts sejam executados em apenas algumas páginas. Imaginemos que ao entrar na página de finalização da compra de um produto, queremos que um determinado script seja executado. Como fariamos?

A especificação *JSP* determina para estes casos, um recurso chamado de **fragment**, onde dentro de uma *tag* podemos incluir um outro trecho de código que é fornecido pela própria página.

Para efeito de exemplo, imprimiremos no console do navegador a seguinte mensagem: Finalização de compra de X itens . Onde o X será o total de itens no carrinho.

O uso de *fragments* no nosso *pageTemplate* será feito através da tag *jsp:invoke* onde por meio do atributo *fragment* definimos o nome do mesmo. Além disso precisamos definir um atributo em nosso template de mesmo nome do *fragment* e configurar o *fragment* deste como *true* .

```
<%@ tag language="java" pageEncoding="ISO-8859-1"%>
<%@ attribute name="titulo" required="true" %>
<%@ attribute name="bodyClass" required="false" %>

<%@ attribute name="extraScripts" fragment="true" %>

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"/>

    <title>${titulo} - Casa do Código</title>

    <!-- imports de css aqui -->
</head>
<body class="${bodyClass}">

<%@ include file="/WEB-INF/views/cabecalho.jsp" %>

<jsp:doBody />

<jsp:invoke fragment="extraScripts"></jsp:invoke>

<%@ include file="/WEB-INF/views/rodape.jsp" %>
</body>
</html>
```

A partir do momento em que fizemos isso, todas as páginas da nossa aplicação podem ter fragmentos de código específicos. Mas para utilizar este recurso, precisa-se seguir algumas regras.

1 - As páginas que usarão *fragments* precisarão envolver o conteúdo das mesmas com a tag `jsp:body`. 2 - O fragmento de código a ser exibido/executado na página deverá estar envolvido pela tag `jsp:attribute` indicando o nome do *fragment* e deve ainda ser declarado antes da tag `jsp:body`.

Seguindo estas regras na página de itens do carrinho e adicionando o código referente a mensagem que queremos exibir, teremos:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://www.springframework.org/tags" prefix="s" %>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<%@ taglib tagdir="/WEB-INF/tags" prefix="tags" %>

<tags:pageTemplate titulo="Seu carrinho de compras">

    <jsp:attribute name="extraScripts">
        <script> console.log("Finalização de compra de ${carrinhoCompras.quantidade} itens"); <
    </jsp:attribute>

    <jsp:body>
        <!-- todo o conteúdo da página aqui -->
    </jsp:body>
</tags:pageTemplate>
```

E agora, apenas na página de carrinho de compras, o script é executado. Caso verifiquemos no console do navegador, veremos algo como:



Finalização de compra de 1 itens

carrinho:153:2

Considerações Finais

Não vimos nada de específico do *Spring* nesta aula. Porém aprendemos bastante sobre alguns problemas comuns que aparecem no dia a dia do desenvolver uma aplicação e mais, vimos recursos e detalhes importantíssimos da especificação *JSP*. Agora chegou a hora de fazer alguns exercícios!