05

Mãos na massa: Implementando o carrinho

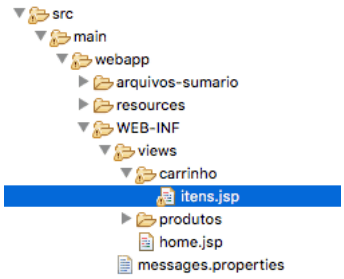
Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Se você ainda não baixou, faça o download do código fonte da página JSP para renderizar os itens do carrinho aqui (https://s3.amazonaws.com/caelum-online-public/spring-mvc-1-criando-aplicacoes-web/springmvc-arquivos-extras-aula13.zip).

Após baixar, extraia o ZIP. Você encontrará dois arquivos JSP e uma imagem:

- O arquivo JSP limpo **mas sem *expression languages*** (**itens-limpo.jsp**).
- O arquivo JSP finalizado com **todas as alterações já aplicadas** (**itens-pronto.jsp**).
- Uma imagem **excluir.png** para adicionar no projeto.

2) Escolha qual JSP usar, renomeie-a para **itens.jsp** e copie-a para a pasta **src/main/webapp/WEB-INF/views/carrinho** (que deve ser criada) do seu projeto:

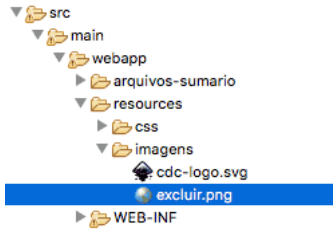


3) Se você usou o arquivo **itens-limpo.jsp**, ainda é preciso incluir as *expression languages* nos lugares indicados. Lembre-se de alterar os locais onde deseja que as informações de URL, *título*, *preço*, *total*, *quantidade*, além do `<c:forEach />` para listar os itens do carrinho:

```
<tbody>
    <c:forEach items="${carrinhoCompras.itens}" var="item">
        <tr>
            <td class="cart-img-col">
                <img src="http://cdn.shopify.com/s/files/1/0155/7645/products/css-eficiente-feat
                    width="71px" height="100px" />
            </td>
            <td class="item-title">${item.produto.titulo}</td>
            <td class="numeric-cell">${item.preco}</td>
            <td class="quantity-input-cell">
                <input type="number" min="0" id="quantidade" name="quantidade"
                    value="${carrinhoCompras.getQuantidade(item)}" />
            </td>
            <td class="numeric-cell">${carrinhoCompras.getTotal(item)}</td>
            <td class="remove-item">
                <form action="" method="POST">
                    <input type="image" src="${contextPath}/resources/imagens/excluir.png"
                        alt="Excluir" title="Excluir" />
                </form>
            </td>
        </tr>
```

```
    </c:forEach>
</tbody>
```

4) Copie a imagem **excluir.png** para a pasta **src/main/webapp/resources/imagens** da aplicação.



5) Crie o método `precoPara` na classe `Produto` para descobrir o preço de um produto de acordo com o tipo:

```
public BigDecimal precoPara(TipoPreco tipoPreco) {
    return precos.stream()
        .filter(preco -> preco.getTipo().equals(tipoPreco))
            .findFirst().get().getValor();
}
```

6) Na classe **`CarrinhoItem`**, crie os métodos `getTotal`, apenas realizando uma multiplicação, e **`getPreco`**, que chama o método **`precoPara`** da classe `Produto`:

```
public class CarrinhoItem {

    public BigDecimal getPreco() {
        return produto.precoPara(tipoPreco);
    }

    public BigDecimal getTotal(int quantidade) {
        return this.getPreco().multiply(new BigDecimal(quantidade));
    }

    // restante do código omitido
}
```

7) Na classe **`CarrinhoCompras`**, crie o método **`getItens`**, que irá retornar uma coleção de itens. Na mesma classe, crie também dois métodos **`getTotal`**, um que apenas repassa a chamada para o método de mesmo nome da classe **`CarrinhoItem`** e outro para saber o total que o usuário está pagando:

```
@Component
@Scope(value=WebApplicationContext.SCOPE_SESSION)
public class CarrinhoCompras {

    public Collection<CarrinhoItem> getItens() {
        return itens.keySet();
    }

    public BigDecimal getTotal(CarrinhoItem item) {
        return item.getTotal(getQuantidade(item));
    }
```

```
    public BigDecimal getTotal() {
        BigDecimal total = BigDecimal.ZERO;

        for (CarrinhoItem item : itens.keySet()) {
            total = total.add(getTotal(item));
        }

        return total ;
    }

    // restante do código omitido
}
```

8) No `CarrinhoComprasController`, crie o método `itens`:

```
@RequestMapping(method=RequestMethod.GET)
public ModelAndView itens(){
    return new ModelAndView("carrinho/itens");
}
```

9) Ainda em `CarrinhoComprasController`, altere o método `add` para que, em vez do usuário ser redirecionado para a página de produtos, ele seja redirecionado para o carrinho:

```
@RequestMapping("/add")
public ModelAndView add(Integer produtoId, TipoPreco tipo){

    ModelAndView modelAndView = new ModelAndView("redirect:/carrinho");
    CarrinhoItem carrinhoItem = criaItem(produtoId, tipo);

    carrinho.add(carrinhoItem);

    return modelAndView;
}
```

10) Por fim, todo componente do Spring que possua escopo de sessão precisa implementar a interface `Serializable`, então faça isso com a classe `CarrinhoCompras`:

```
@Component
@Scope(value=WebApplicationContext.SCOPE_SESSION)
public class CarrinhoCompras implements Serializable {

    private static final long serialVersionUID = 1L;

    // restante do código omitido
}
```