02 Escopo sessão

Nas aulas anteriores, criamos o carrinho de compras da nossa aplicação e fizemos aparecer na página de detalhes o número de produtos que já adicionamos no carrinho. Faça alguns testes, adicione produtos para se certificar que tudo está funcionando perfeitamente.

Nossos testes até agora, não consideraram a situação na qual múltiplos usuários acessam a loja. Nós testamos considerando apenas um usuário. Faça o seguinte teste: Abra a aplicação em um navegador e adicione alguns itens no carrinho de compras. Após isso, abra um outro navegador e acesse a página de detalhes de um produto. O segundo navegador apresenta o carrinho com o mesmo número de itens do primeiro navegador. Isso não deveria acontecer.

Neste caso, dois navegadores diferentes deveriam representar dois usuários diferentes. Aparentemente, a aplicação está compartilhando o carrinho com todos os usuários que a acessam. Seria a mesma situação de estar em uma loja física e outra pessoa colocasse produtos em nossa sacola de compras. Isso não parece certo.

Note que quando um usuário escolhe um livro e depois, uma segunda pessoa escolhe outro item, os dois clientes estão adicionando produtos no mesmo carrinho.

Isso acontece porque ao marcarmos a classe `CarrinhoCompras` com a anotação `@Component`, transformando nossa classe em um ***Bean*** do **Spring** estamos também configurando que este objeto será **Singleton**. Por padrão, o **Spring** tem esta configuração para os seus componentes.

Para corrigir o problema, devemos especificar o escopo do componente através da anotação `@Scope`. Ela recebe um parâmetro chamado `value` que pode receber valores das constantes da interface `WebApplicationContext`.

Quando se faz necessário que um recurso seja individual, ou seja, único para cada usuário, definimos os recursos com o escopo de sessão. Este é criado a partir do momento em que o usuário entra em uma determinada aplicação e segue até o encerramento da mesma - ou ao fechar do navegador em alguns casos.

Então, configure a classe `CarrinhoCompras` para que o componente seja criado no escopo de sessão e assim seja único para cada usuário ao entrar na aplicação. A constante da interface `WebApplicationContext` neste caso será a `SCOPE_SESSION`.

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

Mas fazer somente isso no **Spring** não será o suficiente. Se iniciarmos o servidor agora teremos um erro porque o escopo de sessão não é permitido dentro do `CarrinhoComprasController` que está dentro do escopo de aplicação. Resolveremos este problema com a mesma solução, mas com valores diferentes. Em seguida, entenderemos o motivo.

Controller no geral tem um papel bem definido, ele simplesmente trata a requisição. Ele recebe os dados, repassa para um outro objeto e devolve uma resposta. Pensando assim, podemos concluir que após a requisição ser atendida, não faz sentido que o ***controller*** ainda esteja "vivo". Geralmente, o escopo dos ***controllers*** é o de ***request***. Isto significa que quando há uma requisição, o ***controller*** é criado e depois, ela deixa de existir. Podemos configurar isso através da anotação `@Scope` com o valor da constante `SCOPE_REQUEST` da interface `WebApplicationContext`.

```
@Controller
@RequestMapping("/carrinho")
@Scope(value=WebApplicationContext.SCOPE_REQUEST)
public class CarrinhoComprasController {
    [...]
}
```

Inicie o servidor, faça os testes novamente. Faça verificação tanto com um usuário quanto com vários. O carrinho de compras não deve mais estar compartilhado e cada usuário deve ter o seu próprio com os produtos selecionados.