05

Utilizando Cache na Página

Transcrição

Vamos continuar com o cache do Hibernate, chamado de **Cache de segundo nível**, no qual fazemos com que o JPA e o Hibernate passem funções para o infinispan. Mas um outro cache que ajuda bastante é o da página em si, a qual é renderizada pelo JSF. Cada livro da home está dentro de um loop renderizando não só os dados como as imagens e o código HTML gerado do resultado da busca no banco de dados. Vejamos como podemos melhor isso.

Existe um projeto chamado OmniFaces (http://showcase.ominifaces.org) o qual possui diversos componentes e possibilidade para ajudar o JSF a ser um framework mais completo. E um dos componentes é justamente o cache (http://showcase.ominifaces.org/components/cache), que pode ser usado de uma forma bem simplificada.

Primeiramente abrimos o `index.xhtml` . O Omnifaces vem atravez do namespace

```
xmlns:o="http://omnifaces.org/ui"
```

Vamos colocá-lo em nossa aplicação pelo `pom.xml` :

```
<dependency>
    <groupId>javax.faces</groupId>
    <artifactId>javax.faces-api</artifactId>
    <version>2.2</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>org.omnifaces</groupId>
    <artifactId>omnifaces</artifactId>
    <version>2.1</version>
</dependency>
```

Iremos abrir e fechar com `<o:cache> as duas seções de livros da nossa página, no` index.xhtml`:

```
<o:cache>
    //Últimos Lançamentos

    //Todos os nossos livros
</o:cache>
```

Se recarregarmos a aplicação não veremos nenhuma diferença em relação a antes pois já havíamos implementado o cache e a query do Hibernate não aparece mais. Porém façamos um teste.

Entramos no `HomeBean.java` e façamos com que cada um dos métodos imprima no log o seguinte:

```
public List<Livro> ultimosLancamentos() {
    System.out.println("Entrando nos ultimos lancamentos")
    return dao.ultimosLancamentos();
}
```

Se estamos fazendo o cache da tela, não é para entrar nos métodos, ou seja, se estiver tudo certo não veremos as mensagens no console. E de fato está tudo ok!

Não deixe de pesquisar os inúmeros recursos do Omnifaces, os quais deixarão sua aplicação mais perfomática.

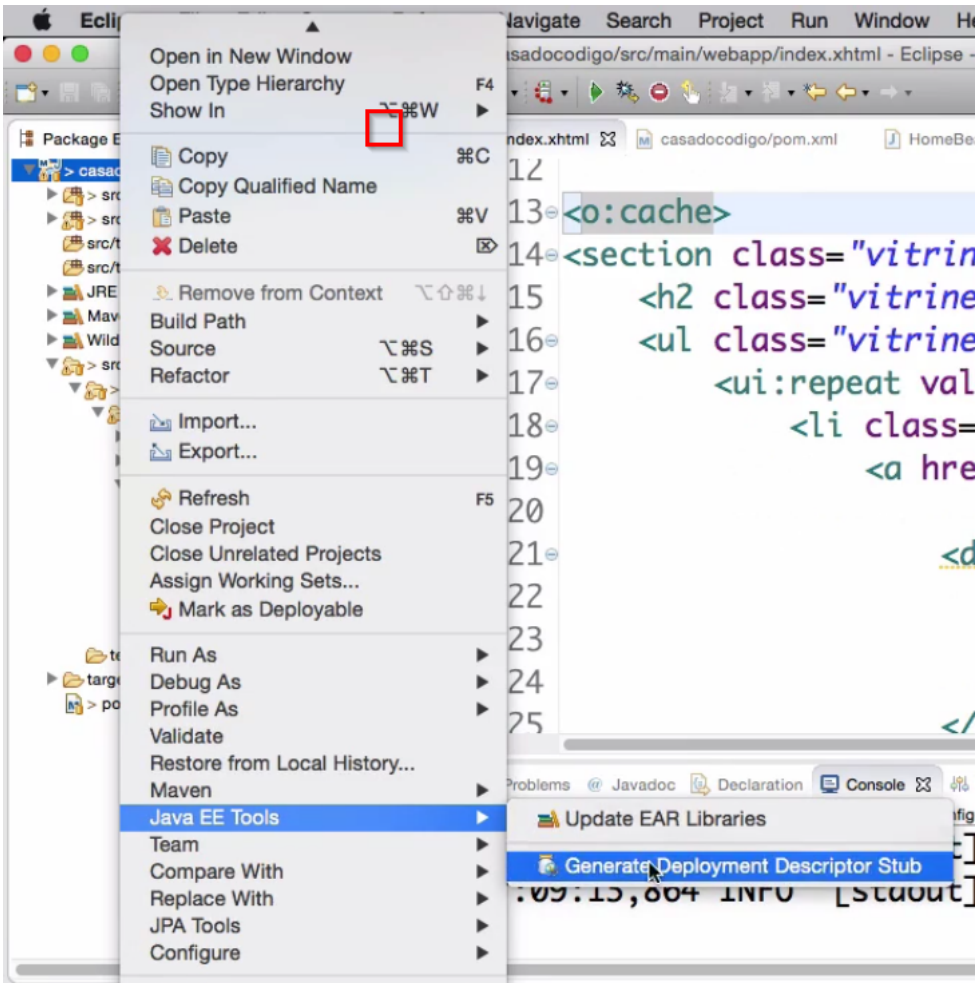
Continuando, trabalharemos em cima de um escopo com o cache do Omnifaces. Quando não o definimos, ele é um escopo de sessão, ou seja, cada usuário terá o valor cacheado apenas para ele. Para nossa aplicação não faz muito sentido pois se por um lado um usuário talvez nem entre de novo na home, outro com certeza entrará.

Quando fazemos:

```
<o:cache scope="application">
```

A nossa home ficará cacheada para todos os usuários. Podemos testar isso carregando a home no navegador normal e no navegador em modo anônimo. Sem o código acima, o cache acontecerá apenas para a segunda vez que o primeiro usuário entrar. Para os outros usuários, na primeira vez que entrarem o html será renderizado novamente.

Vamos agora definir o tempo de vida do cache do Omnifaces. Para isso devemos criar o `web.xml` . Fazemos isso clicando com o botão direito em cima do projeto vamos em "Java EE Tools > Generate Deployment Descriptor Stub":



O xml será criado dentro da pasta "src/main/webapp/WEB-INF". Dentro dele iremos definir um `context-param` que servirá para criarmos um parâmetro de inicialização indicando para o Omnifaces o tempo de cache:

```
<context-param>
    <param-name>org.omnifaces.CACHE_SETTINGS_APPLICATION_TTL</param-name>
    <param-value>900</param-value>
</context-param>
```

Aqui estamos indicando 900 segundos de vida útil de cache, ou seja, 15 minutos.