

04

Limpando o Cache

Transcrição

O que faremos agora é pegar as entidades que estão dentro do cache e fazer uma determinada atualização delas. Uma das possibilidades é acessar o próprio cache.

Na lista de livros podemos pedir para ser feita uma consulta antes de fazer o cache:

```
public List<Livro> ultimosLancamentos() {  
    Cache cache = manager.getEntityManagerFactory().getCache();  
    cache.evict(Livro.class);  
    //...  
}
```

Com o `getCache()` conseguimos obter um objeto do próprio cache, e o método `evict()` passando o `Livro` para limpar seu cache. Vamos criar esse método que limpa o cache separadamente:

```
public void limpaCache() {  
    Cache cache = manager.getEntityManagerFactory().getCache();  
    cache.evict(Livro.class);  
}
```

Poderíamos usar outros `evict()`, como o `evict(Livro.class, 11)` que pegaria apenas o primeiro livro da lista, por exemplo. Ou `evictAll()` o que limpa o cache de todos, seja livro, autor, etc. Estas são as possibilidades programáticas por meio do JPA. Como o hibernate é muito mais abrangente, temos outras possibilidades mais interessantes, como:

- `.unwrap(SessionFactory.class) : dá a possibilidade de evitar partes do cache por regiões, fazendo:`
- `factory.getCache().evictAllRegions() , para todas as regiões`
- `factory.getCache().evictQueryRegion("home") , para uma região específica`

Mas como indicar essas regiões? Dentro das listas fazemos:

```
.setHint(QueryHints.HINT_CACHE_REGION, "home")
```

Isso o Hibernate faz, porém o JPA ainda não.

Existem ainda outros momentos os quais queremos que o cache seja evitado. Por exemplo quando adicionamos ou excluímos um livro das listas. Adicionaremos um novo livro, então, pelo
`localhost:8080/casadocodigo/livros/form.xhtml .`

Poderemos perceber que ao atualizar a página, novamente é feito um `SELECT`. Ou seja, o cache foi limpo para que seja feita a atualização! É algo automático feito pelo próprio servidor de cache ao perceber uma mudança das entidades (`UPDATE` ou `INSERT`).

Agora se, por exemplo, deletarmos um livro diretamente do banco de dados, não tem como a aplicação saber que houve uma mudança, este é outro cuidado que se deve ter com o cache. Se seu banco de dados ou outro serviço externo passa

por atualizações constantes talvez seja o caso, dependendo do seu negócio, de diminuir o tempo de cache ou implementar o código para que a aplicação chame os serviços necessários.