

01

## Configurando segundo tomcat

### Transcrição

[00:00] Nós configuramos o balanceador de cargas para que ele enviasse para o nosso usuário, um cookie, que a partir desse momento, todas as outras requisições dos usuários seriam enviadas pelo balanceador para mesma instância, evitando aquele caso que vimos, do usuário coloca um livro no carrinho de compras e o balanceador enviar requisição para o servidor que não tem atribuição da sessão desse nosso usuário.

[00:26] Então com isso, com a sticky session que fizemos na etapa anterior, conseguimos resolver esse problema e estamos sempre mandando usuário para mesma instância enquanto durar esse cookie. Só que não estamos sendo muito justo com esse balanceamento, estamos sempre mandando o usuário para o mesmo servidor, para mesma instância.

[00:46] O ideal é o que? Que o balanceamento continue sendo feito, como estava sendo realizado nas etapas anteriores, logo no início que ele estava dividindo realmente as requisições entre os dois servidores, e encontramos uma forma de resolver esse problema da sessão. Vamos ver como poderíamos resolver isso?

[01:02] Vamos ver aqui o cenário que temos hoje? Então estou aqui novamente com esse meu computador essa gravação, no estúdio, e na Amazon temos esses dois servidores com a aplicação da Casa do Código. Quando colocamos no browser, para acessar a aplicação da Casa do Código, como vimos, ocorre aquele processo de requisição.

[01:22] Quando fazemos essa requisição para o servidor, o servidor que vai responder, vai abrir como se fosse um túnel de comunicação, vai ser a sessão, como vimos. E essa sessão é uma tabela que vai ter as informações de "chave" e "valor", respectivas a esse nosso usuário.

[01:37] Inicialmente, essa configuração, essa sessão, estava sendo armazenada na instância, no servidor, o qual, respondeu a essa requisição do nosso usuário. Então a ideia agora é encontrar uma forma de tirar essa sessão da instância do servidor da aplicação da Casa do Código, e que armazene essa seção em um ambiente externo. Esse ambiente externo que vamos utilizar é um banco de dados não relacional, que chama "redis".

[02:07] Então a ideia é mudar a nossa aplicação da Casa do Código, para que o armazenamento da sessão desse nosso usuário passe a ser feito no "redis". Quando fizermos a requisição para esse servidor, que tem a aplicação da Casa do Código, a nossa aplicação vai encontrar uma forma de enviar isso e armazenar essa sessão no "redis".

[02:30] E com isso, quando o cookie retornar para o nosso usuário no browser, e tiver uma requisição, por exemplo, para esse segundo servidor, esse segundo servidor vai fazer também a consulta da sessão no "redis". Como a sessão do usuário armazenada no "redis", pouco importa qual servidor ele utilizou para acessar a aplicação da Casa do Código.

[02:50] Porque a sessão dele estará armazenada nesse ambiente externo, o qual a nossa aplicação vai fazer essa consulta para verificar se usuário, se a sessão dele está, ou não, ativa, e não devemos ter mais esse problema que havíamos tido na etapa inicial, que tentamos adicionar um livro no carrinho de compras, e vimos que o livro não estava mais presente.

[03:12] Então agora nossa tarefa vai ser trabalhar com o "redis", fazer toda essa configuração da nossa aplicação, para salvar a sessão do usuário no "redis". Para fazer as etapas, vamos fazer toda a configuração aqui na nossa máquina local para ver se funciona em nosso ambiente de desenvolvimento, e depois, fazemos a alteração para levar ao nosso ambiente de produção na Amazon.

[03:36] Para poder fazer essa simulação em nosso ambiente de desenvolvimento, vamos voltar para o Eclipse, e agora, temos o seguinte cenário. Na Amazon temos os dois servidores, cada um com uma aplicação da Casa do Código. Para poder recriar esse cenário para ver se conseguimos armazenar a sessão do usuário no "redis", e se temos duas instâncias que conseguem fazer essa verificação, o que vou fazer? Vou subir no Eclipse mais um Tomcat, para simular os dois servidores que temos na Amazon.

[04:07] Clico com o botão direito do mouse e vou adicionar um servidor, e colocamos que vamos trabalhar com o Tomcat mesmo, vou deixar o número "dois", para saber que esse é o segundo Tomcat que estamos trabalhando, vou colocar "next", e vamos acionar a aplicação da Casa do Código nesse segundo Tomcat que vamos configurar.

[04:26] Como estou colocando esses dois Tomcat na minha máquina local, por padrão eles vão trabalhar nas mesmas portas. Então para não ter nenhum conflito com o primeiro Tomcat que tínhamos, com esse segundo que acabamos de configurar, vou mudar as portas desse segundo Tomcat para não ter nenhum tipo de conflito com primeiro.

[04:45] Vou clicar duas vezes nesse segundo Tomcat, vamos esperar ele abrir para fazer as configurações. Nas portas desse segundo Tomcat, vou alterar todas elas para um número acima do original. Então aqui é "8005", eu vou colocar "8006", aqui vou colocar "8081", e vou colocar aqui "8010". Assim evitando de ter algum conflito com as portas desse primeiro Tomcat. Então eu só fechar e confirmar que eu quero salvar essas alterações.

[05:18] Com isso, temos aqui a nossa da Casa do Código nesses dois Tomcat, simulando o que temos em nosso ambiente de produção da Amazon. Agora o que acontece? Deveríamos inicializar o Tomcat para poder fazer esses ajustes. Mas se lembremos bem, o que tínhamos feito nas etapas anteriores, vamos só no nosso pacote "configuration", e vamos na nossa classe "JPA configuration".

[05:48] Tínhamos feito a configuração dessa nossa aplicação, para se comunicar com banco de dados que está na Amazon. Mas fizemos a mudança, e esse banco agora não tem mais essa permissão de acesso público, e queremos fazer esse teste localmente no nosso computador da gravação.

[06:05] Eu teria que ficar apagando, trocando esses usuários e senhas, colocando aqui de novo que eu quero trabalhar com o localhost, muito trabalhoso ficar fazendo essa mudança, para depois ter que voltar para esse cenário, para levar na Amazon.

[06:21] Teríamos que trocar para fazer as alterações localmente, depois para levar para produção, teríamos que mudar toda essa informação do banco para levar para a Amazon novamente. Então para evitar isso vamos trabalhar com o recurso de "perfis" no "spring", para que consiga configurar as informações do banco que nós vamos utilizar no ambiente de desenvolvimento e no ambiente de produção.

[06:39] Para podermos trabalhar com esse recurso de perfis do Spring, vamos na classe "ServletSpringMCV", e quando o servidor que for inicializado, vamos pedir para trabalhar com esse perfil de desenvolvimento. Então para isso vamos aqui na classe "servlet SpringMCV", e vamos sobreescrivar um método que é o "onStartup", e fazemos a sobreescrita desse nosso método. Vamos chamar o "servletContext", e vamos setar esse parâmetro, dizendo que queremos trabalhar com o ambiente de desenvolvimento.

[07:17] Chamamos o método para colocar o parâmetro "setInitParameter", e temos que colocar a "chave", que é do "Spring" para ele poder trabalhar com os "perfis". Colocamos "spring.profiles.active" de perfis ativos, e configuramos que queremos trabalhar com o nosso ambiente de "dev", de desenvolvimento, por que vamos fazer esse teste na nossa máquina local.

[07:49] Feito isso colocamos Ctrl+S, e temos o nosso ambiente de desenvolvimento configurado. Vou voltar para a classe "JPA configuration", vou copiar esse método que nós tínhamos, que está no nosso ambiente de produção, e vou colocar esse como sendo o nosso "dataSourceDev" e esse aqui como nosso "dataSourceProd". E no "dataSourceDev", no

ambiente de desenvolvimento, eu vou colocar que nós temos o nosso usuário "Root" senha vazia, e vou colocar localhost.

[08:19] Agora temos já os nossos dois bancos configurados para o ambiente de desenvolvimento e para o ambiente de produção. Só que para falar que esse é desenvolvimento, e esse é do ambiente de produção, temos que colocar anotação do "spring" que é o "profile". Então esse daqui é o perfil de dev, e esse aqui vamos chamar ele como sendo o profile de produção, vou colocar só "prod".

[08:45] Então com isso, não temos que ficar mais alterando toda hora o username, password e a URL do banco. Basta vir aqui falar se queremos trabalhar com o ambiente de desenvolvimento, ou com o ambiente de produção, e já temos um menor retrabalho na hora de levar essa aplicação novamente para o ambiente de produção.

[09:08] Agora que fizemos essa alteração, vamos configurar o "redis" para que a nossa aplicação passe a armazenar a sessão do nosso usuário no "redis". Vamos fazer isso na próxima etapa.