06

A primeira conexão

Transcrição

Você pode baixar o driver do MySQL aqui (https://caelum-online-public.s3.amazonaws.com/1451-jdbc/01/mysql-connector-java-8.0.17.jar).

[00:00] Bom, aluno, tudo bom? Agora que nós temos um entendimento de como a nossa aplicação Java comunica com o banco de dados, é hora de botarmos a mão na massa. Eu criei no Eclipse um projeto chamado "loja-virtual-repository" e nele nós vamos clicar com o botão direito do mouse, selecionar "New > Project" e criar uma classe chamada "TestaConexão" no campo "Name:" da janela "New Java Class". Nessa classe, eu vou querer que tenha o método `main`.

[00:30] Então uma vez que eu mande criar, nós temos uma estrutura de classe bem simples, nada que já não conhecemos. A primeira coisa que eu quero aqui é recuperar a conexão. Então eu pego a `Connection connection =`, eu pego a interface *connection*, que está dentro de java.sql. Vou pegar o `Connection connection = DriverManager`, o nosso gerenciador de drivers.

[01:00] Dentro do gerenciador eu pego o método `Connection connection = DriverManager.getConnection(url, user, password);`, que possui a string URL, a string user e a string password. Aqui, nesse arquivo texto, eu já separei a nossa string de conexão com o banco de dados que nós configuramos nas aulas anteriores. Então a nossa string, aqui entre parênteses, ela vai ser bem simples.

[01:25] E aqui, a string é bem ilegível mesmo, nós temos `jdbc:`, da especificação JDBC, o banco de dados, qual é o banco de dados, então no nosso

caso nós configuramos o MySQL. Vai estar na nossa máquina mesmo, o `localhost`, a `loja_virtual` foi a Database que nós criamos e tenho que fazer essas configurações de `Timezone` e `serverTimezone`. E usuário e senha, `"root", "root"`, que já tínhamos configurado anteriormente.

[01:56] Aqui, esse alerta que o Eclipse está dando, é porque quando você vai usar a aplicação Java para se comunicar com o banco de dados, nós temos N maneiras de ter uma exceção. Se o banco de dados, que eu quero me conectar, estiver fora, eu vou ter uma exceção.

[02:17] Se essa minha string de conexão estiver errada, eu vou ter uma exceção. Então o Eclipse só pede para adicionarmos um `throws SQLException`. O que eu quero fazer é só recuperar a conexão e depois fechar essa conexão, porque nós vamos ver, ao longo do curso, que tudo que nós abrimos, nós temos que fechar, então com a conexão não é diferente.

[02:45] Então a conexão que eu recuperei, se estiver tudo certo, se eu não tiver nenhuma exceção na execução, eu fecho essa conexão com `connection.close();`. O meu código, ele fica bem sucinto aqui, bem simples. E na hora de testar, é para ele só terminar o processo: ele vai executar, fecha a conexão e termina o processo do programa que estamos executando.

[03:11] Tivemos um erro aqui. Olha só, o que ele está falando, que não foi encontrado o driver. E é de fato que nós falamos anteriormente, para o Java se comunicar com o MySQL, com o SQL Server, ele precisa de um driver que conheça tudo o que tem no SQL, tudo o que tem no banco de dados. A nossa aplicação, ela não vai conseguir se comunicar de forma nativa, então precisamos desse driver.

[03:45] E, como foi falado, ele é uma lib, então ele vai ficar dentro do Build Path, eu tenho um "Build Path > Add External Archives...". Então, quando eu for selecionar, e aqui eu peço para vocês usarem a versão do arquivo que está no repositório da Alura, que é o "mysql-connector-java-8.0.17".

[04:08] Vamos trabalhar sempre na mesma versão para evitar qualquer tipo de incompatibilidade de versões mais novas, versões mais antigas. Pode ser que lá na frente nós não consigamos executar os mesmos comandos por conta de uma versão, então, para não ter nenhum tipo de problema, use esta versão que está no repositório.

[04:29] Agora vou mandar executar de novo o código. Uma vez que eu mando, beleza, não tivemos nenhuma exceção, o processo aqui, o nosso programa rodou com sucesso. Para ficar melhor de visualizar essa execução, vou botar aqui `System.out.println("Fechando conexão! !");` depois que ele recuperou a conexão.

[04:52] Então se eu vir em "Run as > Java Application" e mandar executar de novo, vocês viram, ele demorou um pouco aqui, recuperou a conexão, fechou a conexão e depois deu um *close*. Por que sabemos que deu tudo certo? Porque nós não tivemos nenhuma exceção, nenhum SQL *exception*.

[05:12] Então é isso, aluno. Agora nós estamos caminhando já para partes mais avançadas dessa comunicação da nossa aplicação com o banco de dados. Agora com essa conexão em mãos, nós vamos conseguir executar aqueles comandos SQL, como Insert, como Update, como Select, tudo isso que nós estamos ansiosos já para ver. Então espero que tenham gostado e até a próxima aula.