JDBC

Comunicação com banco de dados utilizando JDBC

Prof. Benefrancis do Nascimento

TÓPICOS

- Definição
- Propósito da aula
- Vamos codificar
- Atividades Complementares
- Conclusão

Definição

JDBC significa Java Database Connectivity. No desenvolvimento Java, esta é uma tecnologia bem conhecida e comumente utilizada para a implementação da interação do banco de dados.

A JDBC é uma API do Java que possibilita que uma aplicação construída na linguagem consiga acessar um banco de dados configurado em sua máquina local ou remotamente. A API JDBC é composta pelos pacotes java.sql e javax.sql, incluídos no Java SE. Por meio das classes e interfaces fornecidas por esses dois pacotes, as pessoas podem desenvolver softwares que acessem as mais diversas fontes de dados.

Propósito da aula

Nós vamos implementar a conexão com banco de dados Oracle utilizando a API JDBC.

Para se conectar a um banco de dados utilizando JDBC dependemos de:

a) Driver de Conexão JDBC;
b) A URL de conexão com o banco de dados
c) Saber o usuário e a senha do banco de dados ao qual se deseja conectar.

Vamos codificar!

Criaremos agora nosso projeto Java e a classe para conexão com o banco de dados Oracle

Atividades Complementares

1. Agora que aprendeu como se realiza a conexão com o banco de dados Oracle. Indico que instale outro banco de dados relacional, por exemplo o Maria DB em seu computador pessoal e altere o programa desta aula para que funcione com o novo banco de dados. Anote o que foi necessário modificar para fazer o programa funcionar.

2. Possivelmente o método “findByNome” contém vulnerabilidade a **SQL Injection**. Leia sobre isso no site da OWASP [https://owasp.org/www-community/attacks/SQL_Injection].

Para verificar a vulnerabilidade, execute o método como mostrado abaixo:

```
String nome = "'bla bla bla' OR 'a'='a'";
findByNome(conn, nome);
```

Obs: Você perceberá que o programa está vulnerável se outros nomes forem imprimidos no console. Faça a devida correção conforme orientação do próprio site.

Atividades Complementares

3. Qual a diferença entre o **Statement** e o **PreparedStatement** e quando devo usar um ou outro?

4. Leia o seguinte artigo disponível nestes dois links abaixo:

https://refactoring.guru/pt-br/design-patterns/singleton

https://refactoring.guru/pt-br/design-patterns/singleton/java/example

Proponha mudança na maneira como se obtém a conexão com o banco de dados para evitar que cada usuário do sistema tenha uma nova conexão com o banco de dados somente para ele.

Motivação: Não podemos esquecer que os recursos computacionais são limitados e que você, por meio dos conhecimentos que irá adquirir, será um profissional de prestígio e bem remunerado por possuí-los.

Conclusão

Nesta aula nós conhecemos o JDBC, criamos código para acesso ao banco de dados ORACLE e implementamos os principais métodos de manipulação de dados.

Ao realizar os exercícios propostos, o aluno será capaz de codificar com mais segurança e qualidade aplicando os padrões de projeto e também desenvolvendo o pensamento critico/analítico fundamentais para o profissional de tecnologia da informação.

Agradeço!

benefrancis@gmail.com

Baixe os códigos desta aula em:

https://github.com/Benefrancis/aula-jdbc

Veja o acesso e manipulação de dados respeitando os princípios SOLID:

https://github.com/Benefrancis/aluno