JDBC

Java Database Connectivity

Visual of what JDBC does for us

Visual of what we will be doing?

Ok

No

JDBC Overview

What is JDBC?

Features

Database Support

Architecture

Development Process

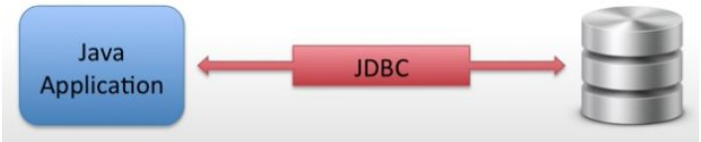
What is JDBC?

Allows Java applications to connect to a relational database

Queries, commands, result sets

Release with JDK 1.1 back in 1997

Supported by major databases; Oracle, IBM DB2, MS SQL Server, MySQL, PostgreSQL, etc.



Features

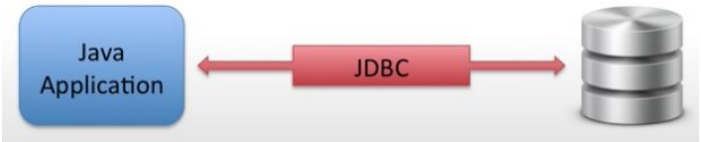
Standard API

Provides portable access to various databases

JDBC Architecture

JDBC Driver

- Provides connection to a database
- Converts JDBC calls to for specific databases
- Provided by database vendor



JDBC Driver Manager

DriverManager helps to connect to an application based on the database connection string

JDBC API

JDBC API is defined in the following packages

java.sql and javax.sql

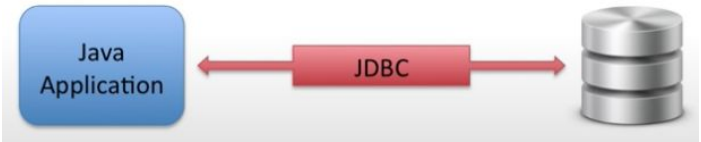
Key classes

DriverManager - define connection to database

Connection - connect to database

Statement - execute sql statement

ResultSet - get results from executing sql statement



Development Process

Step 1: Get a connection to a database

Step 2: Create a statement object

Step 3: Execute SQL query

Step 4: Process Result Set

Step 1: Get a connection to a database

Need connection string in form of JDBC URL

jdbc:<driver protocol>:<driver connection details>

jdbc:mysql://localhost:3306/movie

jdbc:postgresql://localhost:5432/movie

Step 1: Get a connection to a database

```
import java.sql.*;

public static final String URL = "jdbc:postgresql://localhost:5432/movie";
public static final String USER = "postgres";
public static final String PASS = "bigredarmy";

Connection connection = DriverManager.getConnection(URL, USER, PASS);
```

Step 2: Create a Statement object

```
import java.sql.*;

public static final String URL = "jdbc:postgresql://localhost:5432/movie";
public static final String USER = "postgres";
public static final String PASS = "bigredarmy";

Connection connection = DriverManager.getConnection(URL, USER, PASS);

Statement statement = connection.createStatement();
```

Step 3: Execute SQL Query

```
import java.sql.*;

public static final String URL = "jdbc:postgresql://localhost:5432/movie";
public static final String USER = "postgres";
public static final String PASS = "bigredarmy";

Connection connection = DriverManager.getConnection(URL, USER, PASS);

Statement statement = connection.createStatement();
ResultSet resultSet = statement.executeQuery("SELECT * FROM movies;");
```

Step 4: Process the Result Set

```
import java.sql.*;

public static final String URL = "jdbc:postgresql://localhost:5432/movie";
public static final String USER = "postgres";
public static final String PASS = "bigredarmy";

Connection connection = DriverManager.getConnection(URL, USER, PASS);

Statement statement = connection.createStatement();
ResultSet resultSet = statement.executeQuery("SELECT * FROM movies;");

while (resultSet.next()) {
   // read data from each row
   String movieName = resultSet.getString("movie_name");
}
```

Step 4: Process the Result Set

Methods for reading data

- getString(columnName)
- getString(columnIndex)
- get...

```
while (resultSet.next()) {
   // read data from each row
   String movieName = resultSet.getString("movie_name");
   String movieLength = resultSet.getString(2);
}
```

Recap: Development Process

1. Get a connection to database
2. Create a Statement object
3. Execute SQL query
4. Process Result Set

JDBC Summary

What is JDBC?

Features

Database Support

Architecture

Development Process

Lets go code

Using JDBC, connect to your movie database and print out all movie names