

Tutorial: Creating Applications Using MySQL, JDBC, and JSP

By: Eiman Elnahrawy

September 16th 2003
Department of Computer Science
Rutgers University

© Eiman Elnahrawy 2003

Outline

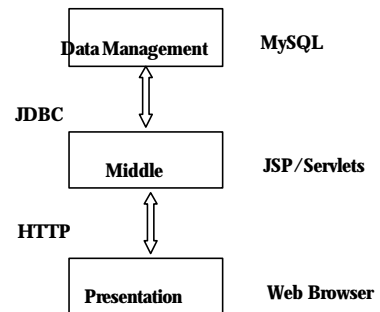
- Introduction
- Needed Tools and Installation
 - JRE, IDE
 - MySQL
 - J2EE
 - JDBC
- How to use and Examples
 - MySQL
 - MySQL + JDBC
 - MySQL + JDBC + JSP

© Eiman Elnahrawy 2003

Three-Tier Application

- Presentation Tier: user interface to make requests, provide input and see results
- Middle Tier: application logic
- Data Management Tier: database management

© Eiman Elnahrawy 2003



© Eiman Elnahrawy 2003

- MySQL
 - DBMS
 - Standard SQL language
- JSP vs. Servlets
 - Servlets - HTML embedded into Java code
 - JSP - java code embedded into HTML
 - JSP better for building interfaces with simple logic, Servlets better for complex application logic
 - We will use JSP in this project

© Eiman Elnahrawy 2003

- What is a Java Server page (JSP)?
 - Combination of JSP scriptlets/tags and HTML/XML tags
 - Has the extension .jsp or .jspx
 - Extension means JSP engine at the web server will process elements on the page
 - JSP engine interprets tags, generates required content, then sends results back as an HTML/XML page to the browser
 - Typically compiled into Java servlet classes the first time called
 - Classes remain in server memory, so subsequent calls have very fast response times
- From <http://java.sun.com/products/jsp/faq.html>

© Eiman Elnahrawy 2003

- **JDBC**
 - Accessing DB from Java, executing SQL queries
 - Results retrieved as a set object called “result set” (Cursor)
 - We can move forward, in reverse, jump to specific part, etc., in the result set

© Eiman Elnahrawy 2003

Outline

- Introduction
- **Needed Tools and Installation**
 - **JRE, IDE**
 - **MySQL**
 - **J2EE**
 - **JDBC**
- How to use and Examples
 - MySQL
 - MySQL + JDBC
 - MySQL + JDBC + JSP

© Eiman Elnahrawy 2003

Needed Tools

- JRE, Java Runtime Environment: compiling, running
- MySQL: creating and storing data
- JDBC Driver: accessing the DB
- J2EE, Java 2 SDK Enterprise Edition: running JSP
- Integrated development environment (IDE), optional: developing, debugging

© Eiman Elnahrawy 2003

JRE, IDE

- I assume you have them
- If not
 - go to <http://java.sun.com/j2se/downloads.html> and install J2SE (Standard Edition)
 - Any IDE, e.g., Forte for Java
 - Download Forte for Java (Currently Sun ONE Studio) from <http://java.sun.com>

© Eiman Elnahrawy 2003

MySQL

- Download from <http://www.mysql.com/downloads/mysql-3.23.html>
- Windows: Source 12 MB, installed 30MB
- Follow installation instructions

© Eiman Elnahrawy 2003

J2EE

- Download from <http://java.sun.com/products/servlet/download.html>
- Install J2EE SDK (1.3.1 FCS Release)
- Follow installation instructions at http://java.sun.com/j2ee/sdk_1.3/install.html regarding setting the environment variables
- Windows: Source 17 MB, installed 27 MB

© Eiman Elnahrawy 2003

JDBC

- Download from <http://www.mysql.com/downloads/api-jdbc-stable.html>
- < 1MB
- File name: Mysql-connector-java-3.0.8-stable
- Un-archive (unzip) the file

© Eiman Elnahrawy 2003

Configure JDBC for Java programs

- One of two approaches
- Approach I
 - Copy mysql-connector-java-3.0.7-stable-bin.jar, your "com" and "org" subdirectories and all their contents to anywhere you like
 - Put the directory holding them in your CLASSPATH environment variable
- Approach II
 - Put "com", "org" directory in JAVA_HOME/jre/lib/ext, I believe you should put every un-archived file

© Eiman Elnahrawy 2003

Configure JDBC for JSP (J2EE)

- Copy the JDBC driver (JAR file + com, org, everything else) to the J2EE_HOME\lib\system directory
- Include the classpath to the JAR file, e.g., C:\j2sdee1.3.1\lib\system in a J2EE_CLASSPATH environment variable

© Eiman Elnahrawy 2003

- Set J2EE_CLASSPATH in the user configuration script as follows
 - Open the corresponding file
 - On UNIX, the configuration script is in bin/userconfig.sh
 - On Windows, it is in bin/userconfig.bat
 - Set the J2EE_CLASSPATH to the path where the jar file and other files exist
 - Set the JAVA_HOME variable to your JRE as well
- Note: after configuring the JDBC driver restart the J2EE server for the new configuration to take effect

© Eiman Elnahrawy 2003

Outline

- Introduction
- Needed Tools and Installation
 - JRE, IDE
 - MySQL
 - J2EE
 - JDBC
- How to use and Examples
 - MySQL
 - MySQL + JDBC
 - MySQL + JDBC + JSP

© Eiman Elnahrawy 2003

Java and HTML

- We assume you have a background about Java, HTML
- Don't know Java
 - A minimal knowledge is needed, nothing fancy
 - Can use CGI but Support is not guaranteed!
- Don't know HTML
 - Refer to text book: chapter 7.6.1, 7.6.2, 7.6.3
 - Cheat!
 - Open similar web pages or design using MS FrontPage
 - View the source code
 - Match source code with the layout

© Eiman Elnahrawy 2003

Example 1: MySQL

- Resources
 - Tutorial "Beginning MySQL" by W.J. Gilmore, include concise information about creating, manipulating tables
http://www.devshed.com/Server_Side/MySQL/Intro
 - Introduction to MySQL and JDBC
<http://www.ils.unc.edu/~lindgren/190/mysql-jdbc/>
 - Detailed documentation "MySQL reference manual"
 - <http://mysql.dyntex.com/>
 - Not necessary
 - Contains section about installation in case you have problems

© Eiman Elnahrawy 2003

Example 1: MySQL

- Run your MySQL server if not already running (MySQL_HOME\bin\winmysqladmin.exe) or from program menu
- Open command window
MySQL_HOME\bin\mysql.exe

© Eiman Elnahrawy 2003

- Create a new database, type at command line
 - create database jsp;
- Select this database, type
 - use jsp;
- Create a new table, type
 - create table test(id varchar(5), name varchar(20));
- Check
 - show tables; //see all tables in the DB
 - desc test; //describe the selected table "test"

© Eiman Elnahrawy 2003

- Populate the table
 - insert into test values("1", "name a");
 - insert into test values("2", "name b");
 - insert into test values("3", "name c");
 - select * from test;
- Delete rows
 - delete from test where id =1;
- Drop table
 - Drop table test;

© Eiman Elnahrawy 2003

Example 2: Java + JDBC+ MySQL

```
import java.sql.*;
import java.util.*;
import java.io.*;
```

```
public class test {
    private static String driverName = "com.mysql.jdbc.Driver"; // or
    "org.gjt.mm.mysql.Driver";
    private static String protocol = "jdbc:mysql";
    private static String host = "localhost:3306";
    private static String db = "jsp";
    private static String user = "root";
    private static String pswd = "";

    public test() {
    }
}
```

© Eiman Elnahrawy 2003

```
public static void main(String[] args) throws Exception{
```

```
    Connection con;
    Statement stmt;
    ResultSet rs;
```

```
String createString = "create table MOVIES (id INTEGER NOT NULL,
movie_name VARCHAR(30),primary key(id))";
String insertString1 = "insert into MOVIES values(1, \"My Fair Lady\")";
String insertString2 = "insert into MOVIES values(2, \"Finding Nemo\")";
String insertString3 = "insert into MOVIES values(3, \"Sound of Music\")";
String selectString = "select * from MOVIES";
```

```
try { Class.forName(driverName).newInstance(); //load jdbc driver
    } catch (java.lang.ClassNotFoundException e) {
        System.err.println(" ClassNotFoundException: ");
        System.err.println(e.getMessage());
    }
try { (String url = protocol + "://" + host + "/" + db;
con = DriverManager.getConnection(url, user, pswd); //connect to the DB
stmt = con.createStatement(); //creating SQL Statement object
//creates a "movies" table in MySQL
stmt.executeUpdate(createString);
    System.out.println("Movies table created");
```

© Eiman Elnahrawy 2003

```

//populate the created table with values
stmt.executeUpdate(insertString1);
stmt.executeUpdate(insertString2);
stmt.executeUpdate(insertString3);
System.out.println("Movies table populated");
//display all tuples
int resLength = 0;
rs = stmt.executeQuery(selectString); //creating a ResultSet
while(rs.next() ) {
    System.out.println("row 1: id = " + rs.getInt("id") + ", movie
name = " + rs.getString("movie_name"));
    resLength++;
}
System.out.println("Select statement executed, " + resLength + " rows
retrieved");
//close everything
stmt.close();
con.close();
} catch(SQLException ex1) {
    System.err.println("SQLException: " + ex1.getMessage());
}
} //main } //test

```

© Eiman Elnahrawy 2003

How to Run a Java program?

- Use your IDE
- Or from command line
 - Compile it using javac test.java
 - Run it using java test

© Eiman Elnahrawy 2003

Examples 3, 4: JSP, JDBC, and MySQL

- Already created the MOVIES table
- Now, access data in this table using a JSP
 - Example 3: Display all available movies
 - Example 4: Interactive, user enters movie id and movie name is then displayed

© Eiman Elnahrawy 2003

- JSP
 - Scriptlet or JSP Tags
 - Scriptlet are easier to understand
- Tutorials
 - <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/>
 - http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/JSPIntro.html
 - <http://java.sun.com/j2ee/tutorial>
 - http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/GettingStarted.html
 - Marty Hall, "Core Servlets and Java Server Pages" <http://pdf.coreservlets.com/>
 - FAQ: <http://java.sun.com/products/jsp/faq.html>

© Eiman Elnahrawy 2003

Example 3

```

<%@page contentType="text/html"%>
<html>
  <head> <title> Example 3</title> </head>
  <body>
    <%@ page import="java.sql.*" %>

    <h1>This is a JSP that queries MySQL using
    JDBC</h1>
    <hr>

```

© Eiman Elnahrawy 2003

```

<%
try {
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    java.sql.Connection conn;
    conn =
    DriverManager.getConnection("jdbc:mysql://localhost:33
    06/jsp", "root", "r");
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery("select * from
    MOVIES");
%>
    <table border=2 cellpadding=5 cellspacing=5>
    <tr>
    <th>Movie Id</th>
    <th>Name</th>
    </tr>

```

© Eiman Elnahrawy 2003

```

<%
while (rs.next() {
    int mid = rs.getInt("id");
    String name = rs.getString("movie_name");%>
    <tr>
        <td><%= mid%></td>
        <td><%= name%></td>
    </tr>
}
rs.close();
stmt.close();
conn.close();
%>
</table>
<% catch (Exception e) {
    %><begin Exception Dump<P><%=
    e.getMessage();
    e.printStackTrace();
    %><%= e %><P><End Exception<%=
}
%>
<hr>
</body>
</html>

```

© Eiman Elnahrawy 2003

Example 4: search page

```

<%@page contentType="text/html"%>
<html>
<head><title>Example 4: Interactive JSP</title></head>
<body>
<h1> Interactive Database Access Test using JSP/MySQL </h1>
<hr>
<p>
<form action="finder.jsp" method=get>
This is a simple example to find information about a specific movie given its ID
number
<hr>
<br> <br> Please enter movie id <input type=int name=id>
<br> <br> <input type=submit value="Search database">
</form>
<p>
</body>

```

© Eiman Elnahrawy 2003

Example 4: finder page

```

<%@page contentType="text/html"%>
<html>
<head>
<%@ page
import = "java.io.*"
import = "java.lang.*"
import = "java.sql.*"
%>
<title>Result of Query</title>
</head>
<body>
<h1>Retrieving the corresponding Movie</h1>
<hr>

```

© Eiman Elnahrawy 2003

```

<%
try {
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    java.sql.Connection conn;
    conn =
    DriverManager.getConnection("jdbc:mysql://localhost:3306/jsp","root","");
    Statement stmt = conn.createStatement();
    String id = request.getParameter("id");
    ResultSet rs = stmt.executeQuery("select movie_name from
MOVIES where id=" + id);
    if (rs.next()){
        String name = rs.getString("movie_name");
        out.println("<table border=2>");
        out.println("<tr><td>" + name);
        out.println("</table>");
        else out.println("Sorry, no matches for " + id);
    } catch (Exception e) {
        out.println("Error!!<br>");
    }
}
%>
</body>
</html>

```

© Eiman Elnahrawy 2003

How to Run a JSP

- Place files under J2EE_HOME/public_html
- Make sure J2EE and MySQL are running
 - To start the J2EE Server, open a terminal window and type `j2ee -verbose`
 - Verbose option is useful for debugging
 - To stop the server, type `j2ee -stop` (use another window)
 - To start the MySQL server run `MySQL_HOME\bin\winmysqladmin.exe`

© Eiman Elnahrawy 2003

How to Run a JSP

- Access <http://localhost:8000/filename.jsp> from your Internet explorer
- For the provided examples use
 - <http://localhost:8000/example3.jsp>
 - <http://localhost:8000/example4.jsp>

© Eiman Elnahrawy 2003

FAQ: How do I debug my J2EE applications running in the J2EE SDK?

- Use Forte for Java
- Put the following as one line in `setenv.sh/setenv.bat` in `J2EE_HOME/bin`
 - `JAVACMD="SJAVA_HOME/bin/java -Xmx128m -Xdebug -Xrunjdwp:transport=dt_socket,server=y,address=9876 SSSL_OPTIONS SJAAS_OPTIONS "`

© Eiman Elnahrawy 2003

- In Forte go to the Debug menu
- Select "Attach to Vm"
- Give the hostname, port to connect to the vm
- Make sure Debugger Type is selected as "Default Debugger (JPDA)"
- Now ready to debug
- These instructions are copied from http://java.sun.com/j2ee/sdk_1.3/faq.html#debug

© Eiman Elnahrawy 2003

That's All!

This tutorial + examples are available from the
"Teaching" link on my webpage
http://paul.rutgers.edu/~eiman/cs541_fall03.html

© Eiman Elnahrawy 2003