**Prepare MySQL Connection for non-web Applications**

**Part 1:** Download Connector

1. Download the java connector for MySQL from the following site:

<https://dev.mysql.com/downloads/connector/j/>



1. Select the “Platform Independent” version
2. Unzip to retrieve the JAR file



**Part 2:** Add Dependency to Project

1. Right click project Dependencies



1. Select “Add Dependency…”
2. Fill up the following info:



Select Add.

1. Right click to newly created Dependency, and select “Manually install artifacts”:



1. Select “Browse” to the extracted JAR file:



1. Install Locally:



**Part 3:** Prepare Database Connection class

package db;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

  public static Connection initializeDatabase()

          throws SQLException, ClassNotFoundException {

    // Initialize all the information regarding Database Connection

    String dbDriver = "com.mysql.jdbc.Driver";

    String dbURL = "jdbc:mysql://localhost:3306/";

    // Database name to access

    String dbName = "MyDB";

    String dbUsername = "root";

    String dbPassword = "P@$$w0rd";

    Class.forName(dbDriver);

    return DriverManager.getConnection(dbURL + dbName,

            dbUsername,

            dbPassword);

  }

}

**Part 4:** Test

package com.mycompany.db;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import db.DatabaseConnection;

public class DB {

  static void ReadData() {

    try (Connection con = DatabaseConnection.initializeDatabase()){

      PreparedStatement st = con.prepareStatement("SELECT \* FROM State");

      ResultSet rs = st.executeQuery();

      while(rs.next()){

            //Display values

        System.out.printf(

      "ID:%d Country:%s Name:%-20s Local Name:%-20s%n",

                rs.getInt("ID"),

                rs.getString("Country"),

                rs.getString("Name"),

                rs.getString("LocalName"));

      }

    }

    catch(Exception e){

      System.out.println("ERROR:"+e.getMessage());

    }

  }

  public static void main(String[] args) {

    ReadData();

  }

}

**Prepare MySQL Connection for Web Applications**

**Part 1:** Setup Glassfish

1. Copy Connector to Glassfish folder



1. Restart the Server



1. Goto Admin Console (http://localhost:4848/)



1. Go to JDBC Connection Pools



1. Create New



1. Prepare these info, then Next



1. Type “com.mysql.cj.jdbc.MysqlDataSource” as class name. Don’t use the default. Our new connection Factory changed to this class



1. Scroll down to the bottom of page, fill in these



1. Click “Add Property” to add 3 more new properties



1. Scroll back up, Click Finish



1. New Pool added. Now click on it



1. Click Ping



1. Succeeded. Now Click Additional Properties



1. After review the properties settings, press Cancel to end the test



1. Select JDBC Resources



1. Select New



1. Fill in these then press OK



1. Done



**Part 2:** Connecting MySQL from Java Web Application

1. Create DB class in the web project

package db;

import java.sql.Connection;

import java.sql.SQLException;

import javax.naming.NamingException;

public class DB {

  public static Connection getConnection()

          throws SQLException, ClassNotFoundException, NamingException {

    javax.naming.InitialContext ctx = new javax.naming.InitialContext();

    var ds = (javax.sql.DataSource) ctx.lookup("jdbc/MySQL");

    return ds.getConnection();

  }

}

1. Write a test Servlet

@WebServlet(name = "ReadData", urlPatterns = {"/ReadData"})

public class ReadtData extends HttpServlet {

  protected void processRequest(HttpServletRequest request, HttpServletResponse response)

          throws ServletException, IOException {

    response.setContentType("text/html;charset=UTF-8");

    try (PrintWriter out = response.getWriter()) {

      try (Connection con = DB.getConnection()) {

        PreparedStatement st = con.prepareStatement("SELECT \* FROM State");

        ResultSet rs = st.executeQuery();

        StringBuilder sb = new StringBuilder("<table border='1'>");

        sb.append("<tr><th colspan='4'>State Information</th></tr>");

        sb.append("<tr><th>ID</th><th>Country</th><th>Name</th><th>LocalName</th></tr>");

        while (rs.next()) {

          sb.append("<tr><td>");

          sb.append(rs.getInt("ID"));

          sb.append("</td><td>");

          sb.append(rs.getString("Country"));

          sb.append("</td><td>");

          sb.append(rs.getString("Name"));

          sb.append("</td><td>");

          sb.append(rs.getString("LocalName"));

          sb.append("</td></tr>");

        }

        sb.append("</table>");

        out.print(sb.toString());

      } catch (Exception e) {

        out.println("ERROR:" + e.getMessage());

      }

    }

  }

1. Compile and Run

