package com.nttdata.petstore.dbfw;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public interface ParamMapper {

void mapParams(PreparedStatement preStmt) throws DBFWException,

SQLException;

}

package com.nttdata.petstore.dbfw;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import org.apache.log4j.Logger;

import com.nttdata.petstore.dao.ProductDAO;

public class DBHelper {

public static final Logger LOG = Logger.getLogger(ProductDAO.class);

private DBHelper() {

}

public static int executeUpdate(Connection con, final String sqlStmt,

ParamMapper inMap) throws DBFWException {

PreparedStatement preStmt;

int result = 0;

try {

preStmt = con.prepareStatement(sqlStmt);

inMap.mapParams(preStmt);

result = preStmt.executeUpdate();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return result;

}

public static List executeSelect(Connection con, final String sqlStmt,

ResultMapper outMap) throws DBFWException {

PreparedStatement preStmt = null;

ResultSet resSet = null;

List categoryList = new ArrayList();

try {

preStmt = con.prepareStatement(sqlStmt);

resSet = preStmt.executeQuery();

while (resSet.next()) {

final Object obj = outMap.mapRow(resSet);

categoryList.add(obj);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return categoryList;

}

public static List executeSelect(Connection con, final String sqlStmt,

ResultMapper outMap, ParamMapper inMap) throws DBFWException {

PreparedStatement preStmt = null;

ResultSet resSet = null;

List categoryList = new ArrayList();

try {

preStmt = con.prepareStatement(sqlStmt);

inMap.mapParams(preStmt);

resSet = preStmt.executeQuery();

while (resSet.next()) {

final Object obj = outMap.mapRow(resSet);

categoryList.add(obj);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return categoryList;

}

public static boolean validateUser(Connection con, final String sqlStmt,

final ResultMapper outMap, String custId, String password) {

PreparedStatement prestmt = null;

ResultSet resSet = null;

boolean isValidated = false;

try {

prestmt = con.prepareStatement(sqlStmt);

prestmt.setString(1, custId);

prestmt.setString(2, password);

resSet = prestmt.executeQuery();

while (resSet.next()) {

String user = resSet.getString("CustID");

String pass = resSet.getString("Password");

if (custId.equals(user) && (password.equals(pass))) {

LOG.debug("Validating the user");

isValidated = true;

System.out.println("Valid User");

} else {

isValidated = false;

System.out.println("Not a valid user");

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

LOG.error("Throws a DBFW exception");

}

return isValidated;

}

}

package com.nttdata.petstore.dbfw;

import java.sql.ResultSet;

import java.sql.SQLException;

public interface ResultMapper {

Object mapRow(ResultSet resSet) throws DBFWException, SQLException;

}

package com.nttdata.petstore.dbfw;

public class DBFWException extends Exception {

public DBFWException() {

// TODO Auto-generated constructor stub

}

public DBFWException(String message) {

super(message);

// TODO Auto-generated constructor stub

}

public DBFWException(Throwable cause) {

super(cause);

// TODO Auto-generated constructor stub

}

public DBFWException(String message, Throwable cause) {

super(message, cause);

// TODO Auto-generated constructor stub

}

}