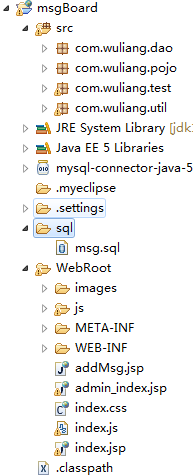
# Java 留言板

思路：需求



1. 模板—index.jsp，index.css，phtot
2. 根据index.html中的留言板中需要写入数据，新建Mysql数据库，新建表t\_msg，

建表如下：

create table t\_msg(

id int auto\_increment,

usetname varchar(12),

theme varchar(30),

emali varchar(18),

address varchar(5),

photo int ,

iqcq varchar(20),

hpage varchar(20),

message text,

times date,

primary key(id)

);

1. 写入UDButil类，数据库工具类，

package com.wuliang.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class UDButil {

public UDButil(){}

static {

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static Connection getConn(){

Connection conn = null;

try {

conn = DriverManager.getConnection("jdbc:mysql:///test", "test", "test");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return conn;

}

public static PreparedStatement getPstmt(Connection conn,String sql){

PreparedStatement pstmt = null;

try {

pstmt = conn.prepareStatement(sql);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return pstmt;

}

public static ResultSet getRs(PreparedStatement pstmt){

ResultSet rs = null;

try {

rs = pstmt.executeQuery();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return rs;

}

public static void getClose(Connection conn,PreparedStatement pstmt,ResultSet rs){

if(rs==null){

try {

rs.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

rs = null;

}

if(pstmt==null){

try {

pstmt.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

pstmt = null;

}if(conn==null){

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

conn = null;

}

}

}

1. 写入实体类，现在进入了 javaBean 类型
2. 用Dao类就是用来实现数据库的更新和查询

public class Dao {

public static void getUpdate(Msg msg) throws SQLException{

Connection conn = UDButil.getConn();

PreparedStatement pstmt = UDButil.getPstmt(conn, "insert into t\_msg values(null,?,?,?,?,?,?,?,?,now())");

pstmt.setString(1,msg.getUsetname());

pstmt.setString(2,msg.getTheme());

pstmt.setString(3,msg.getEmali());

pstmt.setString(4,msg.getAddress());

pstmt.setInt(5,msg.getPhoto());

pstmt.setString(6,msg.getIqcq());

pstmt.setString(7,msg.getHpage());

pstmt.setString(8,msg.getMessage());

pstmt.executeUpdate();

}

用insert into t\_msg values(null,?,?,?,?,?,?,?,?,now()); 用？来代表其中的一个元素，当然pstmt.setUsername(8,msg.getUsername());给插入数据，用pstmt.executeUdate();来表示是否成功插入，返回为0时则失败

public static List<Msg> getMsg() throws Exception{

List<Msg> msgs = new ArrayList<Msg>();

Connection conn = UDButil.getConn();

PreparedStatement pstmt = UDButil.getPstmt(conn,"select \* from t\_msg");

ResultSet rs = UDButil.getRs(pstmt);

while(rs.next()){

Msg msg = new Msg();

msg.setId(rs.getInt(1));

msg.setUsetname(rs.getString(2));

msg.setTheme(rs.getString(3));

msg.setEmali(rs.getString(4));//

msg.setAddress(rs.getString(5));

msg.setPhoto(rs.getInt(6));

msg.setIqcq(rs.getString(7));

msg.setHpage(rs.getString(8));

msg.setMessage(rs.getString(9));

msg.setTimes(rs.getTimestamp(10));

msgs.add(msg);

}

rs.getString()可以找到查询到的列数的数据，成为msg.set(参数);

return msgs;

}

1. AddMsg.jsp 的数据

try{

Msg msg = new Msg();

msg.setUsername(request.getParameter("username"));

msg.setTheme(request.getParameter("theme"));

msg.setTitle(request.getParameter("title"));

msg.setEmail(request.getParameter("email"));

msg.setAddres(request.getParameter("addres"));

msg.setId(Integer.parseInt(request.getParameter("photo")));

msg.setIqcq(request.getParameter("iqcq"));

msg.setHpage(request.getParameter("hpage"));

msg.setMessage(request.getParameter("message"));

Dao.msgDao(msg);

request.setAttribute("msg","留言表发表成功！！！1");

request.getRequestDispatcher("./admin\_index.jsp").forward(request,response);

}catch (Exception e){

e.printStackTrace();

request.setAttribute("msg","留言表发表失败！！！");

request.getRequestDispatcher("./admin\_index.jsp").forward(request,response);

return;

}

