**物联网中间件**

**实验手册**

目录

[实验一 用户登陆 2](#_Toc8399372)

[实验目的： 2](#_Toc8399373)

[实验环境： 2](#_Toc8399374)

[实验内容： 2](#_Toc8399375)

[实验二 用户注册 2](#_Toc8399376)

[实验目的： 2](#_Toc8399377)

[实验环境： 2](#_Toc8399378)

[实验内容： 2](#_Toc8399379)

[实验三 Apache MQ的使用 3](#_Toc8399380)

[实验目的： 3](#_Toc8399381)

[实验环境： 3](#_Toc8399382)

[实验内容： 3](#_Toc8399383)

# 实验一 用户登陆

用户登陆 4月25日

## 实验目的：

1、熟悉Eclipse的使用；

2、熟悉简单动态Jsp页面的搭建；

3、熟悉MySQL的基本操作；

## 实验环境：

1、配置有JAVA EE开发环境的计算机；

## 实验内容：

1、配置Eclipse，配置Tomcat，建立JAVA WEB项目；

1）修改Eclipse的默认字符编码，Jsp模板；

2）进入官网下载Tomcat 8并解压到任意目录；

3）配置JAVA 服务运行环境；

a.导入Tomcat jar包；

b.导入MySQL connector包；

c.配置JAVA 服务运行环境；

4）创建一个JAVA WEB项目；

2、完成用户登陆的代码实现；

doLogin.jsp主要逻辑代码：

<%

request.setCharacterEncoding("UTF-8");

response.setCharacterEncoding("UTF-8");

String username = request.getParameter("name");

String userpwd = request.getParameter("pwd");

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

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/iot", "root", "root");

String sql = "select \* from user where name=? and pwd=?";

PreparedStatement pstmt = conn.prepareStatement(sql);

pstmt.setString(1, username);

pstmt.setString(2, userpwd);

ResultSet rs = pstmt.executeQuery();

UserBean bean = null;

if (rs.next()) {

bean = new UserBean();

bean.setId(rs.getInt("id"));

bean.setName(rs.getString("name"));

bean.setPwd(rs.getString("pwd"));

bean.setType(rs.getString("type"));

bean.setImage\_path(rs.getString("image\_path"));

session.setAttribute("user", bean);

response.sendRedirect("index.jsp");

} else {

response.sendRedirect("login.jsp");

}

%>

# 实验二 用户注册

用户注册 4月30日

## 实验目的：

1、熟悉Eclipse的使用；

2、熟悉简单动态Jsp页面的搭建；

3、熟悉MySQL的基本操作；

## 实验环境：

1、配置有JAVA EE开发环境的计算机；

## 实验内容：

1、完成用户注册的代码实现；

doRegister.jsp主要逻辑代码：

<%

request.setCharacterEncoding("UTF-8");

response.setCharacterEncoding("UTF-8");

String username=request.getParameter("name");

String userpwd=request.getParameter("pwd");

String useremail=request.getParameter("email");

String usertype=request.getParameter("type");

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/iot","root","root");

String sql="insert into user (name,pwd,email,type) values(?,?,?,?)";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1, username);

pstmt.setString(2,userpwd);

pstmt.setString(3,useremail);

pstmt.setString(4,usertype);

int i=pstmt.executeUpdate();

if(i>0){

response.sendRedirect("login.jsp");

}else{

response.sendRedirect("register.jsp");

}

%>

2、完成用户查询显示的代码实现；

doFind.jsp主要逻辑代码：

<%

request.setCharacterEncoding("UTF-8");

response.setCharacterEncoding("UTF-8");

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/iot","root","root");

String sql="select \* from user";

PreparedStatement pstmt=conn.prepareStatement(sql);

ResultSet rs=pstmt.executeQuery();

List<UserBean>list=new ArrayList<UserBean>();

UserBean bean=null;

while(rs.next()){

bean=new UserBean();

bean.setId(rs.getInt("id"));

bean.setName(rs.getString("name"));

bean.setPwd(rs.getString("pwd"));

bean.setEmail(rs.getString("email"));

bean.setType(rs.getString("type"));

bean.setImage\_path(rs.getString("image\_path"));

list.add(bean);

}

request.setAttribute("users", list);

request.getRequestDispatcher("showUser.jsp").forward(request, response);

%>

# 实验三 Apache MQ的使用

Apache MQ的使用 5月7日

## 实验目的：

1、熟悉ActiveMQ的使用；

2、掌握使用JAVA操作ActiveMQ；

## 实验环境：

1、配置有JAVA开发环境的计算机；

2、ActiveMQ 5 的压缩包；

## 实验内容：

1、运行ActiveMQ；

2、通过浏览器创建ActiveMQ消息队列；

3、编写AppProducer代码，向ActiveMQ中添加消息；

public class AppProducer{

private static final String URL="tcp://127.0.0.1:61616";

private static final String QUEUENAME="queue-test";

public static void run() throws JMSException{

ConnectionFactory connectionFactory=new ActiveMQConnectionFactory(URL);

Connection connection = connectionFactory.createConnection();

connection.start();

Session session = connection.createSession(false,Session.AUTO\_ACKNOWLEDGE);

Destination destination=session.createQueue(QUEUENAME);

MessageProducer producer =session.createProducer(destination);

for(int i=0;i<100;i++) {

TextMessage textMessage=session.createTextMessage("test "+i);

try {

producer.send(textMessage);

System.out.println("发送消息："+textMessage.getText());

}catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

}

}

}

}

4、编写AppConsumer代码，消费ActiveMQ中的消息；

public class AppConsumer{

private static final String URL="tcp://127.0.0.1:61616";

private static final String QUEUENAME="queue-test";

public static void run() throws JMSException{

ConnectionFactory connectionFactory=new ActiveMQConnectionFactory(URL);

Connection connection = connectionFactory.createConnection();

connection.start();

Session session = connection.createSession(false,Session.AUTO\_ACKNOWLEDGE);

Destination destination=session.createQueue(QUEUENAME);

MessageConsumer consumer=session.createConsumer(destination);

consumer.setMessageListener(new MessageListener() {

@Override

public void onMessage(Message arg0) {

// TODO Auto-generated method stub

TextMessage textMessage=(TextMessage) arg0;

try {

System.out.println("接收消息："+textMessage.getText());

}catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

}

}

});

}

}