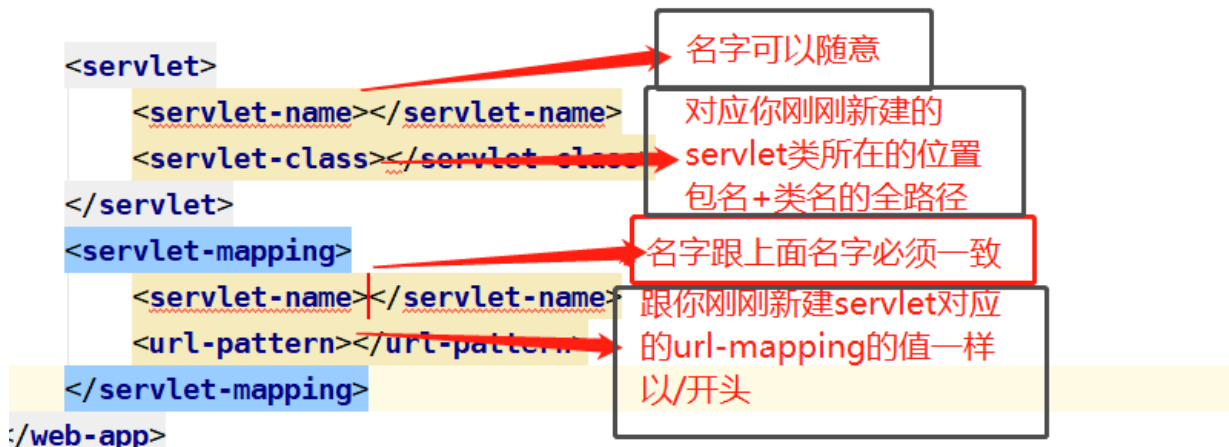


2020/2/14 20:16:20

1.从index.jsp页面跳转到admin/admin.jsp页面，通过servlet的形式跳转

1.1 servlet的配置,新建一个类，搜索servlet(然后进行配置,里面的url-mapping对应的是index.jsp form表单对应的地址，比如"/login" 对应的action应该为"login")

1.2 web.xml配置 servlet



2.通过index.jsp输入的用户名和密码经过你刚刚新建的servlet，在里面调用连接数据库的类的方法，然后按结果进行跳转

2.1.连接数据库的工具类

2.2 连接数据库的接口和实现类

2.3 连接数据库的实体类

3.取得用户名密码用`request.getParameter("页面的name属性的值");`

4.跳转使用`request.getRequestDispatcher().forward(request,response)`(重定向使用`response.sendRedirect()`)

2020/2/14 17:38:00

主要代码

servlet内写的东西

```
BizDao bizDao=new BizDaoImple();
String username=req.getParameter("username");
String password=req.getParameter("password");
boolean b = bizDao.validateUser(username, password);
if (b){
    req.getRequestDispatcher("admin/admin.jsp").forward(req,resp);
}else {
    resp.sendRedirect("index.jsp");
}
```

数据库工具类

```
/**
 * 连接和关闭数据库
 */
public class BaseDao {
//数据库连接通道 connection
protected Connection con=null;
//圣旨
protected PreparedStatement ps=null;
//结果集
protected ResultSet rs=null;

/**
 * 得到连接
@return
 */
protected Connection getCon(){
//1导jar包
try {
    Class.forName("com.mysql.jdbc.Driver");
    con= DriverManager.getConnection("jdbc:mysql://localhost:3306/news","root","996649855");
} catch (Exception e) {
    e.printStackTrace();
}
return con;
}

protected void closeAll(Connection con,PreparedStatement ps,ResultSet rs){
try {
    if (rs!=null){
        rs.close();
    }
    if (ps!=null){
        ps.close();
    }
    if (con!=null){
        con.close();
    }
} catch (Exception e) {
    e.printStackTrace();
}
}
}
```

连接数据库的接口和实现类

```
/**
 * 版权所有(C)，2020，所有权利保留。
 * 项目名: News
 * 文件名: BizDao.java
 * 模块说明:
 * 修改历史:
 * 2020-2-12 15:52:21 - WeiBin - 创建。
 */
```

```
package wb.dao;

public interface BizDao {

    boolean validateUser(String username, String password);
}
```

```
package wb.dao.impl;
```

```
import wb.dao.BaseDao;
import wb.dao.BizDao;
import wb.pojo.UserInfo;

import java.util.ArrayList;
import java.util.List;

public class BizDaoImpl extends BaseDao implements BizDao {
    @Override
    public boolean validateUser(String username, String password) {
        boolean flag;
        List<UserInfo> list=new ArrayList<>();

        try {
            ps = getCon().prepareStatement("select * from user_info where username=? and password=?"
            ps.setString(1,username);
            ps.setString(2,password);
            rs = ps.executeQuery();
            while (rs.next()){
                UserInfo u=new UserInfo();
                u.setUser_id(rs.getInt(1));
                u.setUsername(rs.getString(2));
                u.setPassword(rs.getString(3));
                list.add(u);
            }
            if (list.size()==1){
                flag=true;
            }else {
                flag=false;
            }
        } catch (Exception e) {
            throw new RuntimeException();
        }
        return flag;
    }
}
```

用户名和密码的实体类 对应数据库

```
package wb.pojo;

public class UserInfo {
    private int user_id;
    private String username;
```

```
private String password;

public int getUser_id() {
    return user_id;
}

public void setUser_id(int user_id) {
    this.user_id = user_id;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}
}
```