JSP & EL & JSTL

1. jsp

> Java Server Page

- 什么是jsp

> 从用户角度看待，就是是一个网页， 从程序员角度看待， 其实是一个java类， 它继承了servlet，所以可以直接说jsp 就是一个Servlet.

- 为什么会有jsp?

> html 多数情况下用来显示静态内容， 一成不变的。 但是有时候我们需要在网页上显示一些动态数据， 比如： 查询所有的学生信息， 根据姓名去查询具体某个学生。 这些动作都需要去查询数据库，然后在网页上显示。 html是不支持写java代码 ， jsp里面可以写java代码。

1.1 怎么用JSP

1.1.1 指令写法

<%@ 指令名字 %>

1.1.2 page指令

- language

> 表明jsp页面中可以写java代码

- contentType

> 其实即使说这个文件是什么类型，告诉浏览器我是什么内容类型，以及使用什么编码

```
contentType="text/html; charset=UTF-8"

text/html  MIMEType 这是一个文本，html网页
```

- pageEncoding jsp内容编码
- extends 用于指定jsp翻译成java文件后，继承的父类是谁，一般不用改。
- import 导包使用的，一般不用手写。
- session

> 值可选的有true or false .

> 用于控制在这个jsp页面里面，能够直接使用session对象。

> 具体的区别是，如果该值是true , 那么在代码里面会有getSession（）的调用，如果是false : 那么就不会有该方法调用，也就是没有session对象了。在页面上自然也就不能使用session了。

- errorPage

> 指的是错误的页面， 值需要给错误的页面路径

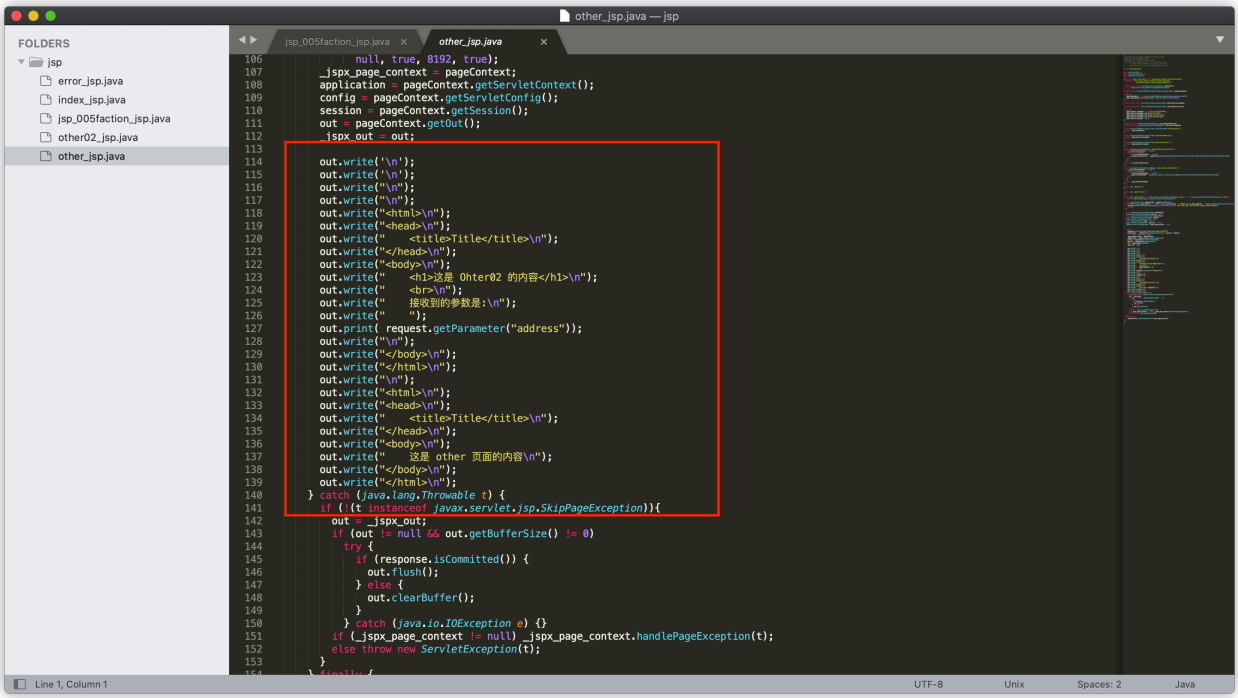
- isErrorPage

> 上面的errorPage 用于指定错误的时候跑到哪一个页面去。 那么这个isErroPage , 就是声明某一个页面到底是不是错误的页面。

1.1.3 include

> 包含另外一个jsp的内容进来。

```
<%@ include file="other02.jsp"%>
```



- 背后细节:

> 把另外一个页面的所有内容拿过来一起输出。 所有的标签元素都包含进来。

1.1.4 taglib

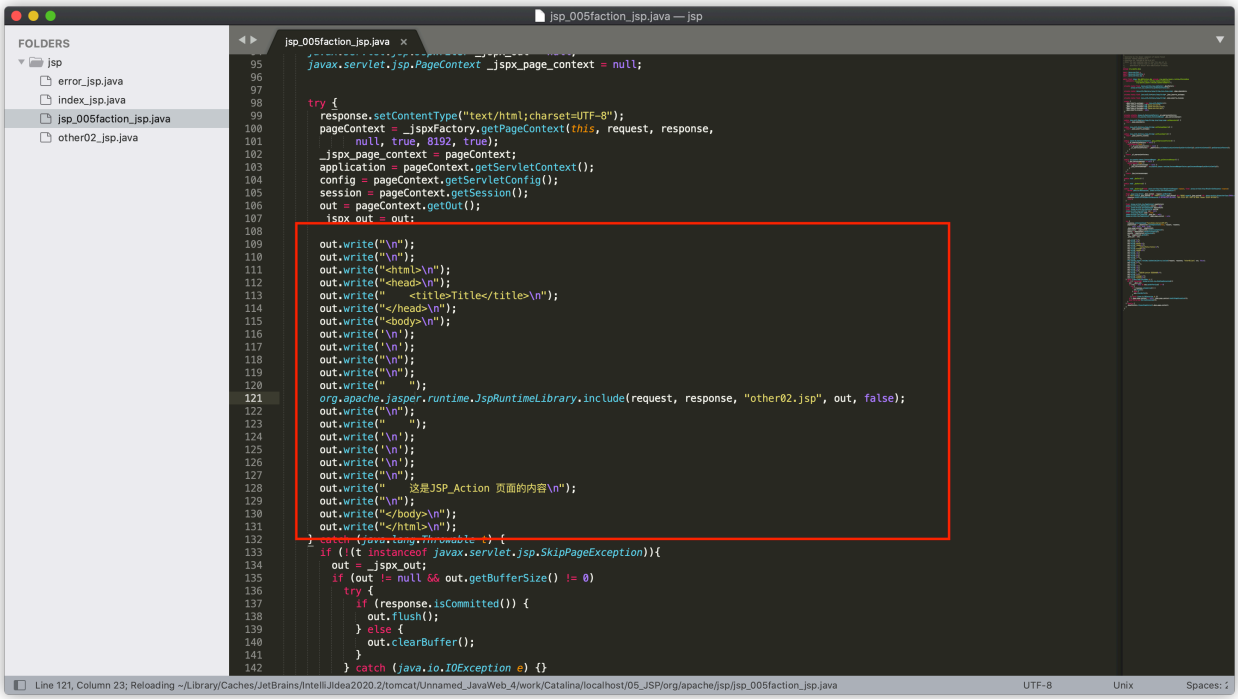
```
<%@ taglib prefix=""  uri=""%>
uri: 标签库路径
prefix : 标签库的别名
```

1.2 JSP 动作标签

```
<jsp:include page=""></jsp:include>
<jsp:param value="" name=""/>
<jsp:forward page=""></jsp:forward>
```

- jsp:include

```
<jsp:include page="other02.jsp"></jsp:include>
```



> 包含指定的页面， 这里是动态包含。 也就是不把包含的页面所有元素标签全部拿过来输出，而是把它的运行结果拿过来。

- jsp:forward

  <jsp:forward page="">/jsp:forward

> 前往哪一个页面。

```
<%
  //请求转发
  request.getRequestDispatcher("other02.jsp").forward(request, response);
%>
```

- jsp:param

> 意思是： 在包含某个页面的时候，或者在跳转某个页面的时候，加入这个参数。

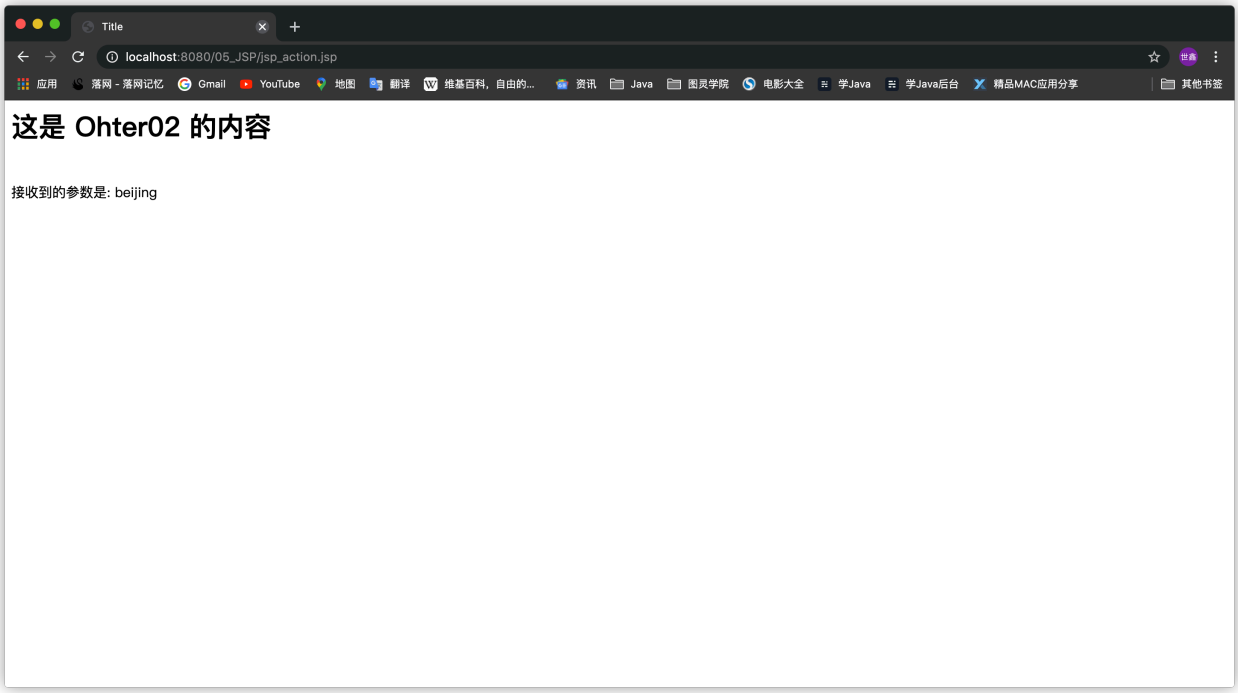
Jsp_action.jsp

```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 2:25 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
```

```
    <title>Title</title>
</head>
<body>
    <%--    <jsp:include page=""></jsp:include>--%>
    <%--    <jsp:forward page=""></jsp:forward>--%>
    <%--    <jsp:param name="" value=""/>--%>

    <%--    <jsp:include page="other02.jsp"></jsp:include>--%>
    <%--  跳转到other02页面 带着参数  --%>
    <jsp:forward page="other02.jsp">
        <jsp:param name="address" value="beijing"/>
    </jsp:forward>
    这是JSP_Action 页面的内容

</body>
</html>
```

other02.jsp

```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 2:27 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <h1>这是 Ohter02 的内容</h1>
    <br>
    接收到的参数是:
    <%= request.getParameter("address")%>
</body>
</html>
```



1.3 JSP内置对象

> 所谓内置对象，就是我们可以直接在jsp页面中使用这些对象。 不用创建。

- pageContext
- request
- session
- application

以上4个是作用域对象，

- 作用域

> 表示这些对象可以存值，他们的取值范围有限定。 setAttribute 和 getAttribute

```
使用作用域来存储数据<br>

<%
  pageContext.setAttribute("name", "page");
  request.setAttribute("name", "request");
  session.setAttribute("name", "session");
  application.setAttribute("name", "application");
%>

取出四个作用域中的值<br>

<%=pageContext.getAttribute("name")%>
<%=request.getAttribute("name")%>
<%=session.getAttribute("name")%>
<%=application.getAttribute("name")%>
```

作用域范围大小：

```
pageContext -- request --- session -- application
```

1.4 四个作用域的区别

- pageContext 【PageContext】

> 作用域仅限于当前的页面。

> 还可以获取到其他八个内置对象。

- request 【HttpServletRequest】

> 作用域仅限于一次请求， 只要服务器对该请求做出了响应。 这个域中存的值就没有了。

- session 【HttpSession】

> 作用域限于一次会话（多次请求与响应） 当中。

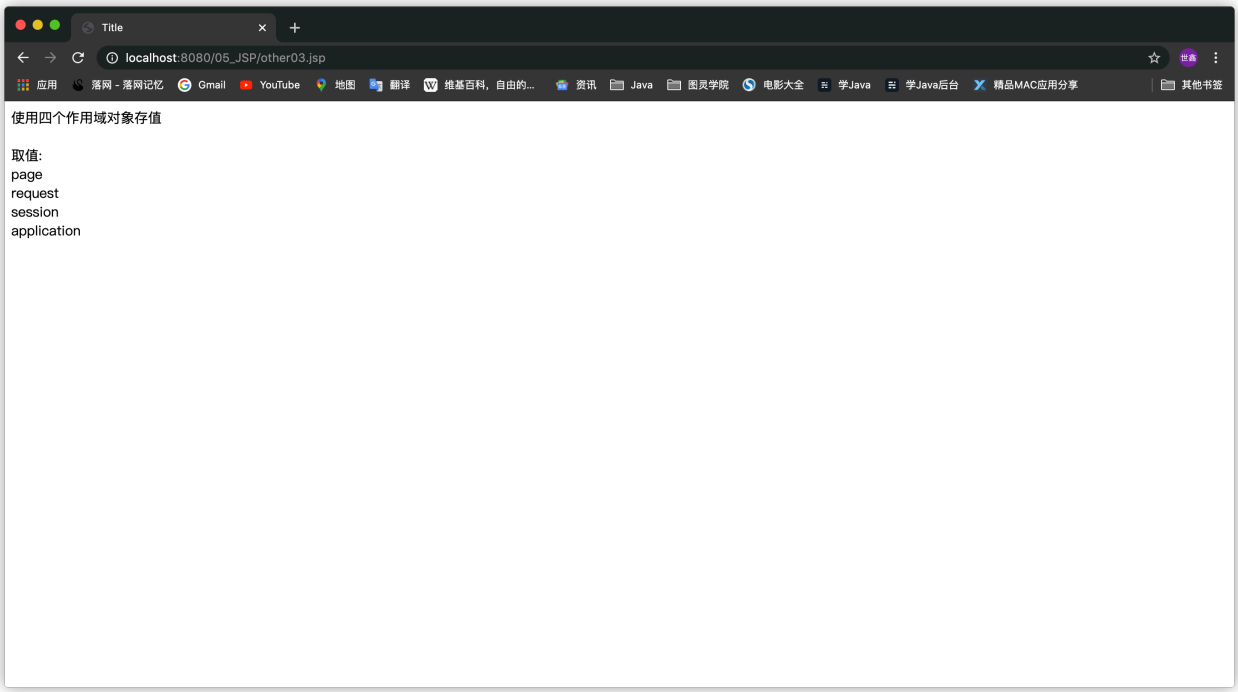
- application 【ServletContext】

> 整个工程都可以访问， 服务器关闭后就不能访问了。

other03.jsp

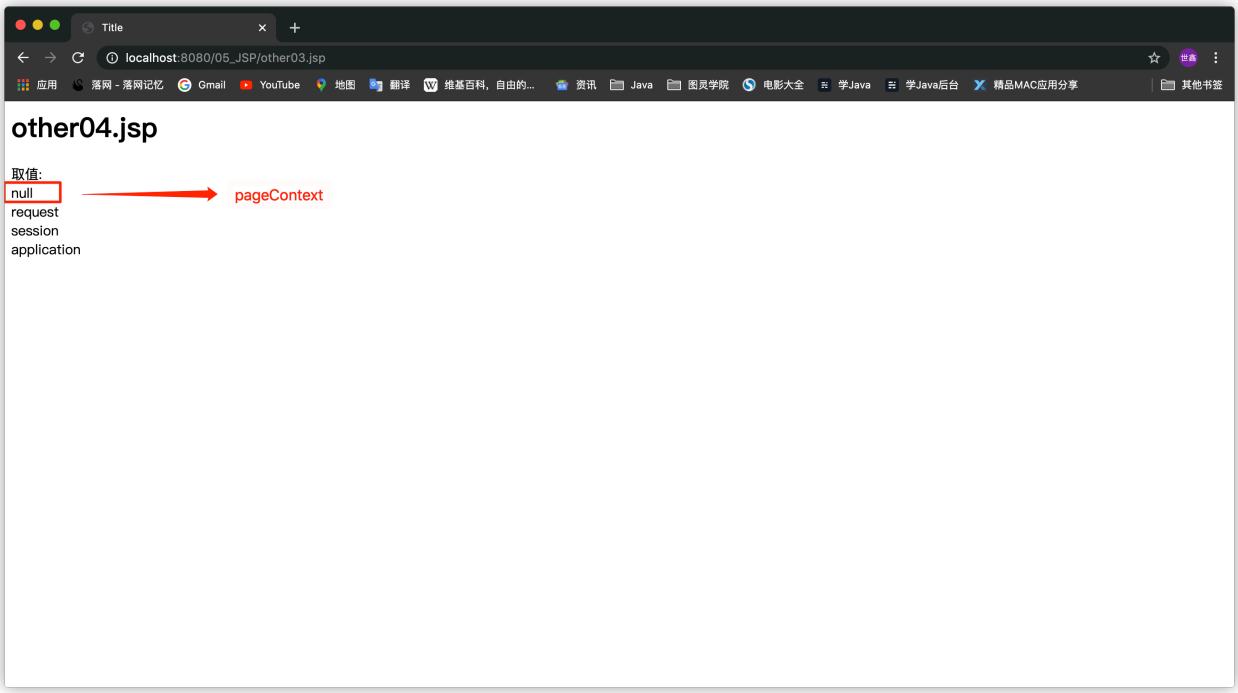
```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 3:00 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    使用四个作用域对象存值
    <% pageContext.setAttribute("name","page");%>
    <% request.setAttribute("name","request");%>
    <% session.setAttribute("name","session");%>
    <% application.setAttribute("name","application");%>

    <br>
    <br>
    取值：
    <br>
    <%= pageContext.getAttribute("name")%><br>
```

```
    <%= request.getAttribute("name")%><br>
    <%= session.getAttribute("name")%><br>
    <%= application.getAttribute("name")%><br>
</body>
</html>
```



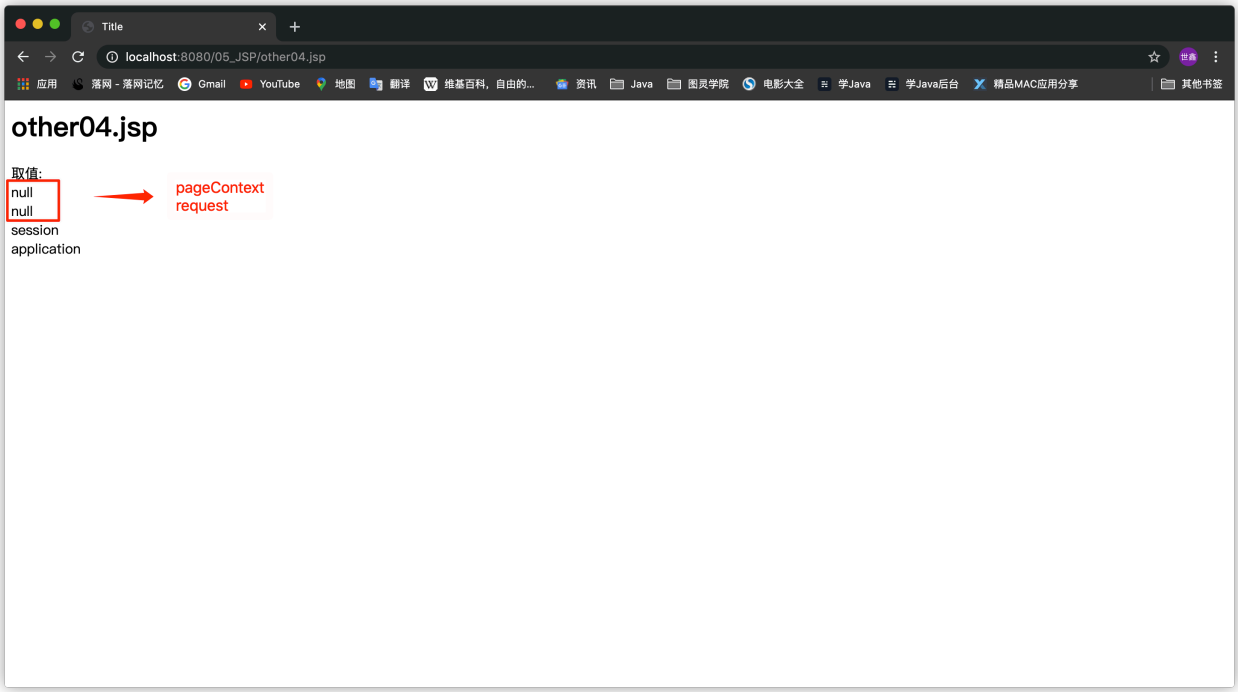
在other03.jsp中添加代码跳转到other04.jsp

```
<%
        request.getRequestDispatcher("other04.jsp").forward(request,response);
%>
```

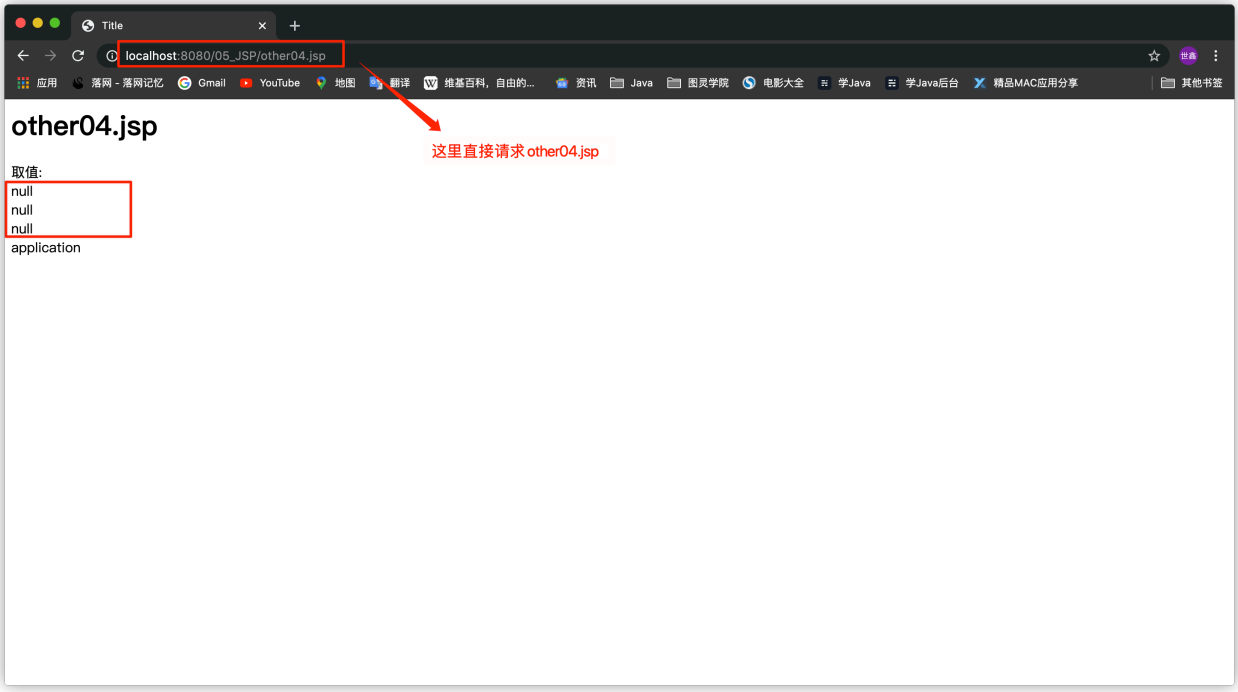


继续修改other03.jsp

```
    <%
//
request.getRequestDispatcher("other04.jsp").forward(request,response);
        response.sendRedirect("other04.jsp");
    %>
```

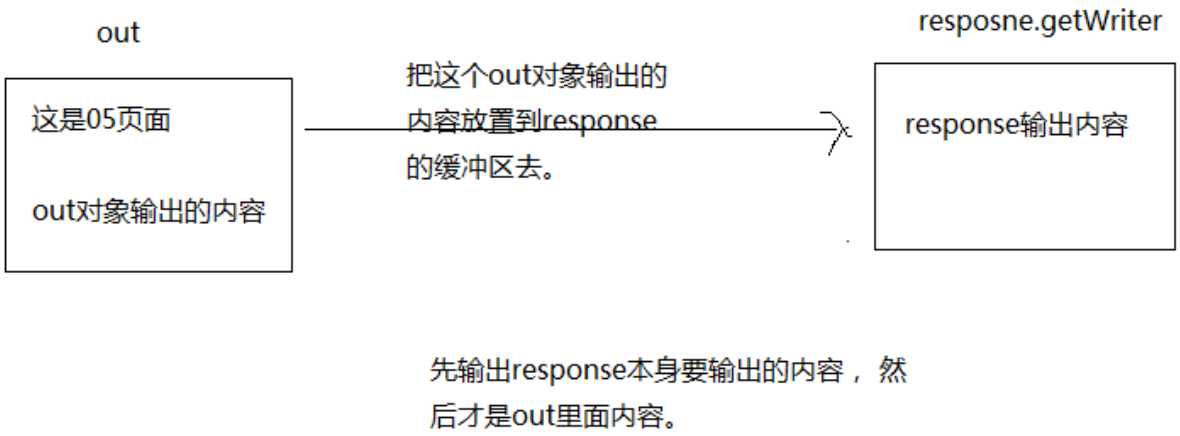


关闭浏览器,重新打开.继续测试 session



- out 【JspWriter】

- response 【HttpServletResponse】



- exception 【Throwable】
  - page 【Object】 ---就是这个jsp翻译成的java类的实例对象
    - config 【ServletConfig】

2. EL表达式

> 是为了简化咱们的jsp代码，具体一点就是为了简化在jsp里面写的那些java代码。

- 写法格式

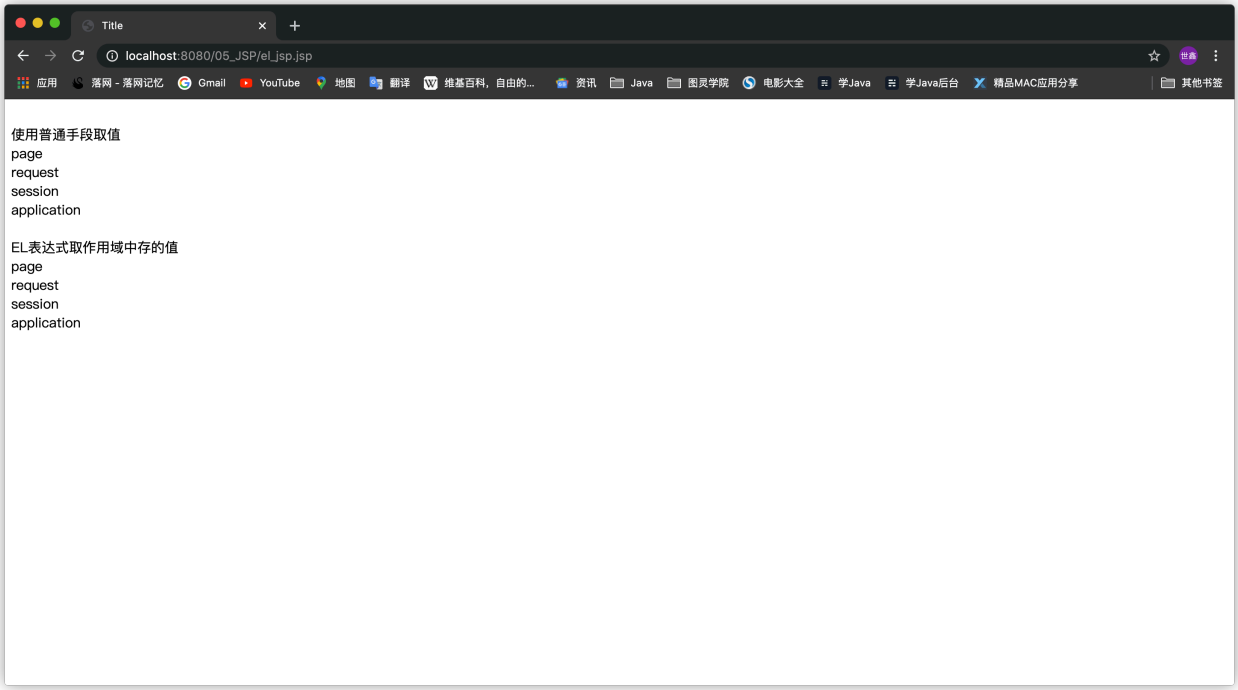
${表达式 }

> 如果从作用域中取值，会先从小的作用域开始取，如果没有，就往下一个作用域取。 一直把四个作用域取完都没有， 就没有显示。

2.1 取出4个作用域中存放的值。

```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 3:58 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
```

```
    <%
        pageContext.setAttribute("name","page");
        request.setAttribute("name","request");
        session.setAttribute("name","session");
        application.setAttribute("name","application");
    %>
    <br>
    使用普通手段取值<br>
    <%= pageContext.getAttribute("name")%><br>
    <%= request.getAttribute("name")%><br>
    <%= session.getAttribute("name")%><br>
    <%= application.getAttribute("name")%><br>
    <br>
    EL表达式取作用域中存的值<br>
    ${pageScope.name}<br>
    ${requestScope.name}<br>
    ${sessionScope.name}<br>
    ${applicationScope.name}<br>

</body>
</html>
```



2.2 如果域中所存的是数组

```
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.Map" %>
<%@ page import="java.util.HashMap" %><%--
```

```
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 3:58 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>

    <%
        pageContext.setAttribute("name","page");
        request.setAttribute("name","request");
        session.setAttribute("name","session");
        application.setAttribute("name","application");
    %>
    <br>
    使用普通手段取值<br>
    <%= pageContext.getAttribute("name")%><br>
    <%= request.getAttribute("name")%><br>
    <%= session.getAttribute("name")%><br>
    <%= application.getAttribute("name")%><br>
    <br>
    EL表达式取作用域中存的值<br>
    ${pageScope.name}<br>
    ${requestScope.name}<br>
    ${sessionScope.name}<br>
    ${applicationScope.name}<br>

    <br>------------取数组中的值-------------<br>
    <%
        String[] args = {"aa","bb","cc","dd"};
        pageContext.setAttribute("array",args);
    %>
    ${array[0]},${array[1]},${array[2]},${array[3]}
</body>
</html>
```

2.3 如果域中锁存的是集合

```
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.Map" %>
```

```jsp
<%@ page import="java.util.HashMap" %><%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 3:58 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>

    <%
        pageContext.setAttribute("name","page");
        request.setAttribute("name","request");
        session.setAttribute("name","session");
        application.setAttribute("name","application");
    %>
    <br>
    使用普通手段取值<br>
    <%= pageContext.getAttribute("name")%><br>
    <%= request.getAttribute("name")%><br>
    <%= session.getAttribute("name")%><br>
    <%= application.getAttribute("name")%><br>
    <br>
    EL表达式取作用域中存的值<br>
    ${pageScope.name}<br>
    ${requestScope.name}<br>
    ${sessionScope.name}<br>
    ${applicationScope.name}<br>

    <br>------------取数组中的值-------------<br>
    <%
        String[] args = {"aa","bb","cc","dd"};
        pageContext.setAttribute("array",args);
    %>
    ${array[0]},${array[1]},${array[2]},${array[3]}


    <br>------------取集合中的值-------------<br>
    <%
        List<String> list = new ArrayList<>();
        list.add("11");
        list.add("22");
        list.add("33");
        list.add("44");
        pageContext.setAttribute("list",list);
```

```
    %>
    ${list.get(0)},${list.get(1)},${list.get(2)},${list.get(3)}

</body>
</html>
```

2.4 取出Map集合的值

```
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.Map" %>
<%@ page import="java.util.HashMap" %><%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 3:58 上午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>

    <%
        pageContext.setAttribute("name","page");
        request.setAttribute("name","request");
        session.setAttribute("name","session");
        application.setAttribute("name","application");
    %>
    <br>
    使用普通手段取值<br>
    <%= pageContext.getAttribute("name")%><br>
    <%= request.getAttribute("name")%><br>
    <%= session.getAttribute("name")%><br>
    <%= application.getAttribute("name")%><br>
    <br>
    EL表达式取作用域中存的值<br>
    ${pageScope.name}<br>
    ${requestScope.name}<br>
    ${sessionScope.name}<br>
    ${applicationScope.name}<br>
```

```
    <br>------------取数组中的值-------------<br>
    <%
        String[] args = {"aa","bb","cc","dd"};
        pageContext.setAttribute("array",args);
    %>
    ${array[0]},${array[1]},${array[2]},${array[3]}


    <br>------------取集合中的值-------------<br>
    <%
        List<String> list = new ArrayList<>();
        list.add("11");
        list.add("22");
        list.add("33");
        list.add("44");
        pageContext.setAttribute("list",list);
    %>
    ${list.get(0)},${list.get(1)},${list.get(2)},${list.get(3)}


    <br>------------取Map中的值-------------<br>
    <%
        Map map = new HashMap();
        map.put("name","admiral");
        map.put("age",18);
        map.put("address","beijing");
        pageContext.setAttribute("map",map);
    %>
    ${map.name},${map.age},${map.address}

</body>
</html>
```

2.5 取值细节：

1. 从域中取值。 得先存值。
   <%

   //pageContext.setAttribute("name", "zhangsan");
   session.setAttribute("name", "lisi...");
   %>

直接指定说了，到这个作用域里面去找这个name

${ pageScope.name }

//先从page里面找，没有去request找，去session，去application

${ name }

指定从session中取值

${ sessionScope.name }

2. 取值方式

如果这份值是有下标的，那么直接使用[]

```
<%
  String [] array = {"aa","bb","cc"}
  session.setAttribute("array",array);
%>
```

```
${ array[1] } --> 这里array说的是attribute的name
```

如果没有下标， 直接使用 .的方式去取

```
<%
  User user = new User("zhangsan",18);

  session.setAttribute("u", user);
%>

${ u.name }  , ${ u.age }
```

> 一般使用EL表达式，用的比较多的，都是从一个对象中取出它的属性值，比如取出某一个学生的姓名。

2.6 EL表达式 的11个内置对象。

${ 对象名.成员 }

- pageContext

作用域相关对象

- pageScope
- requestScope
- sessionScope
- applicationScope

头信息相关对象

- header

- headerValues

参数信息相关对象

- param
- paramValues

- cookie
  全局初始化参数
- initParam

3. JSTL

> 全称： JSP Standard Tag Library jsp标准标签库

> 简化jsp的代码编写。 替换 <%%> 写法。 一般与EL表达式配合

3.1 怎么使用

1. 导入jar文件到工程的WebContent/Web-Inf/lib jstl.jar standard.jar
2. 在jsp页面上，使用taglib 指令，来引入标签库
3. 注意： 如果想支持 EL表达式，那么引入的标签库必须选择1.1的版本，1.0的版本不支持EL表达式。

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

常用标签

c:set/c:set
<c:if test="">/c:if
c:forEach/c:forEach

- c:set

```
  <!-- 声明一个对象name，  对象的值 zhangsan , 存储到了page（默认） ，  指定是session -->
  <c:set var="name" value="zhangsan" scope="session"></c:set>

  ${sessionScope.name }
```

- c:if

> 判断test里面的表达式是否满足，如果满足，就执行c:if标签中的输出 ， c:if 是没有else的。

```
<c:set var="age" value="18" ></c:set>
<c:if test="${ age > 26 }">
  年龄大于了26岁...
</c:if>

<c:if test="${ age <= 26 }">
  年龄小于了26岁...
</c:if>
```

```
------------------------------

定义一个变量名 flag  去接收前面表达式的值，然后存在session域中

<c:if test="${ age > 26 }" var="flag" scope="session">
  年龄大于了26岁...
</c:if>
```

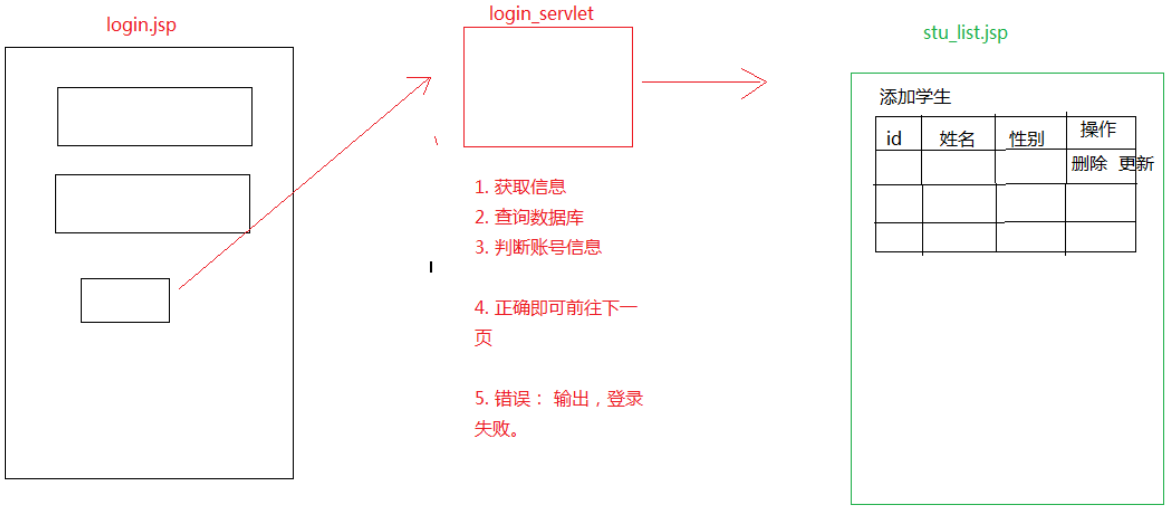
- c:forEach

```
从1 开始遍历到10 ，得到的结果 ，赋值给 i ,并且会存储到page域中， step , 增幅为2,
<c:forEach begin="1" end="10" var="i" step="2">
  ${i }
</c:forEach>
```

```
---------------------------------------------

<!-- items : 表示遍历哪一个对象，注意，这里必须写EL表达式。
var: 遍历出来的每一个元素用user 去接收。 -->
<c:forEach var="user" items="${list }">
  ${user.name } ----${user.age }
</c:forEach>
```

学生信息管理系统

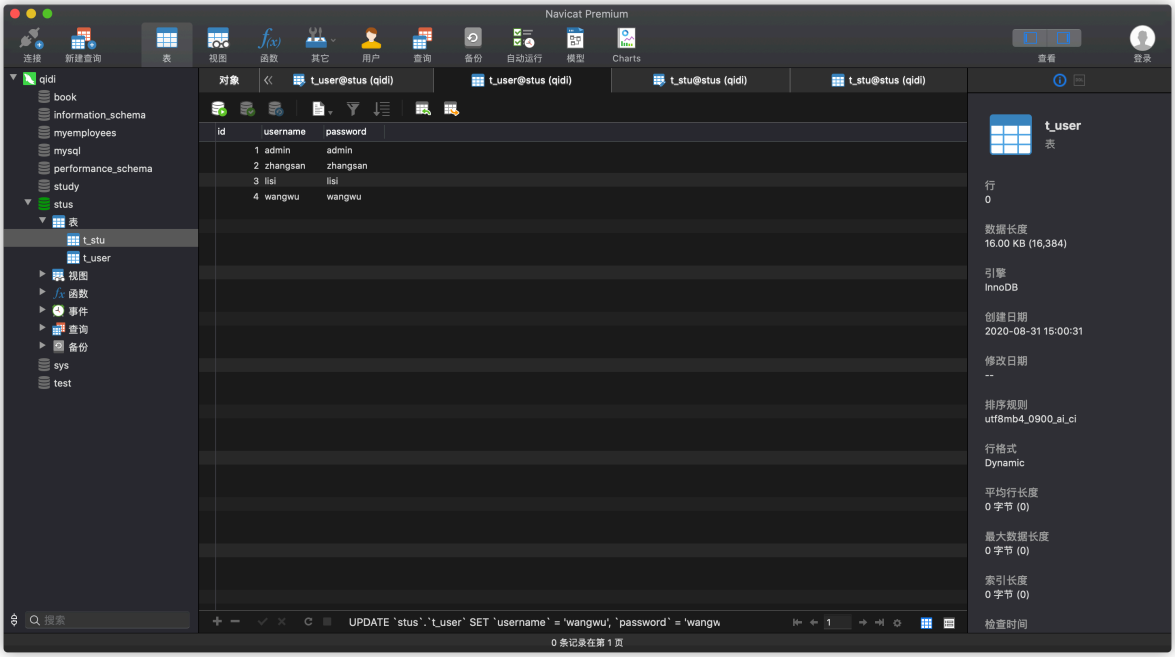
- 需求分析



1. 先写 login.jsp , 并且搭配一个LoginServlet 去获取登录信息。

```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 12:24 下午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <h2>欢迎使用学生管理系统</h2>
    <form action="http://localhost:8080/05_StudentManager/login"
method="post">
        账号:<input type="text" name="username"><br>
        密码:<input type="password" name="password">
        <input type="submit" value="登陆">
    </form>
</body>
</html>
```

2. 创建用户表， 里面只要有id , username 和 password



3. 创建UserDao, 定义登录的方法

```
package com.admiral.dao;

import com.admiral.pojo.User;

/**
 * @author 白世鑫
 * @description
 * @date 2020/8/31
 */
public interface UserDao {

    public User login(String username, String password);
}
```

4. 创建UserDaoImpl , 实现刚才定义的登录方法。

```
package com.admiral.dao.impl;

import com.admiral.dao.BaseDao;
import com.admiral.dao.UserDao;
import com.admiral.pojo.User;

/**
 * @author 白世鑫
 * @title: UserDaoImpl
 * @projectName JavaWeb
 * @description:
 * @date 2020/8/31  12:37 下午
```

```
 */
public class UserDaoImpl extends BaseDao implements UserDao {
    @Override
    public User login(String username, String password) {
        String sql = "select id,username,password from t_user where username=?
and password=?";
        return queryForOne(User.class, sql, username, password);
    }
}
```

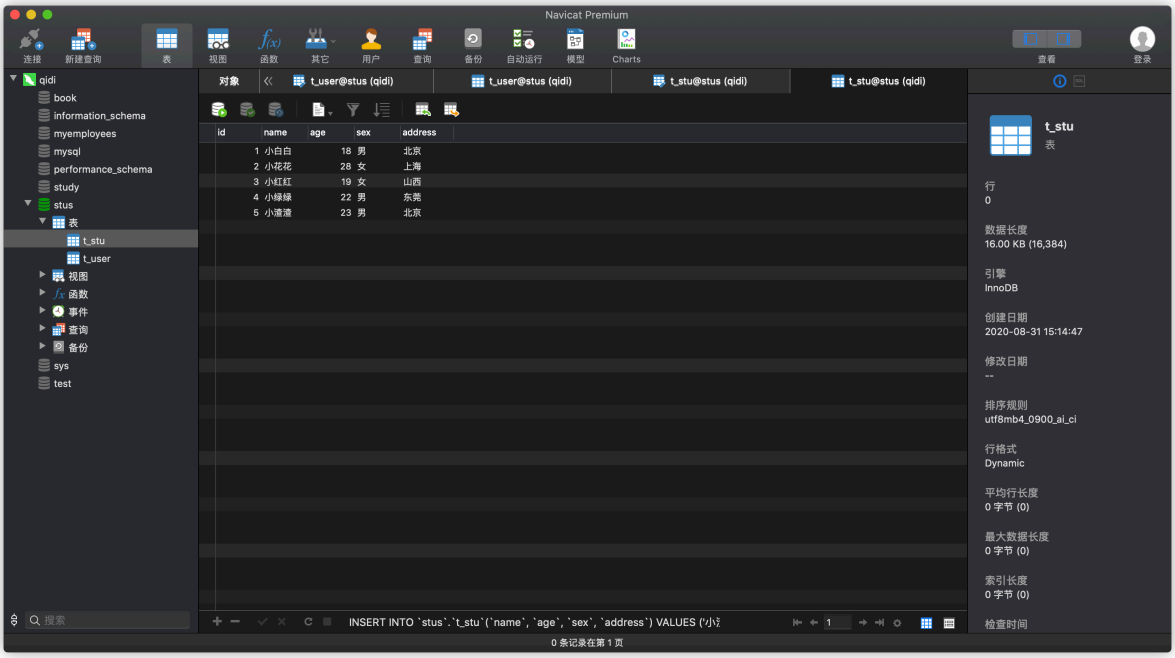
5. 在LoginServlet里面访问UserDao， 判断登录结果。 以区分对待

```
package com.admiral.web;

import com.admiral.dao.StudentDao;
import com.admiral.dao.UserDao;
import com.admiral.dao.impl.StudentDaoImpl;
import com.admiral.dao.impl.UserDaoImpl;
import com.admiral.pojo.Student;
import com.admiral.pojo.User;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.List;

public class LoginServlet extends HttpServlet {
    UserDao userDao = new UserDaoImpl();
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        User login = userDao.login(username, password);
        if (login != null) {
            //登录成功
            System.out.println("登录成功");
            response.getWriter().write("登录成功");

        }else {
            //登录失败
            System.out.println("登录失败");
            response.getWriter().write("登录失败");
        }
    }

}
```

6. 创建stu_list.jsp , 让登录成功的时候跳转过去。

```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
  Time: 1:07 下午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <h1>学生信息列表</h1>

</body>
</html>
```

7. 创建学生表， 里面字段随意。



8. 定义学生的Dao . StuDao

```
package com.admiral.dao;

import com.admiral.pojo.Student;
```

```
import java.util.List;

/**
 * @author 白世鑫
 * @description
 * @date 2020/8/31
 */
public interface StudentDao {

    public List<Student> findAll();
}
```

9. 对上面定义的StuDao 做出实现 StuDaoImpl

```
package com.admiral.dao.impl;

import com.admiral.dao.BaseDao;
import com.admiral.dao.StudentDao;
import com.admiral.pojo.Student;

import java.util.List;

/**
 * @author 白世鑫
 * @title: StudentDaoImpl
 * @projectName JavaWeb
 * @description:
 * @date 2020/8/31  1:00 下午
 */
public class StudentDaoImpl extends BaseDao implements StudentDao {
    @Override
    public List<Student> findAll() {
        String sql = "select id,name,age,gander,address from t_stu";
        return queryForList(sql, Student.class);
    }
}
```

10. 在登录成功的时候，完成三件事情。

- 查询所有的学生
- 把这个所有的学生集合存储到作用域中。
- 跳转到stu_list.jsp

```
package com.admiral.web;

import com.admiral.dao.StudentDao;
import com.admiral.dao.UserDao;
```

```
import com.admiral.dao.impl.StudentDaoImpl;
import com.admiral.dao.impl.UserDaoImpl;
import com.admiral.pojo.Student;
import com.admiral.pojo.User;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.List;

public class LoginServlet extends HttpServlet {
    UserDao userDao = new UserDaoImpl();
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        User login = userDao.login(username, password);
        if (login != null) {
            //登录成功
            System.out.println("登录成功");
            response.getWriter().write("登录成功");

            //1.查询所有学生信息
            StudentDao studentDao = new StudentDaoImpl();
            List<Student> all = studentDao.findAll();
            //2.将查询到的结果存储到域对象中
            request.getSession().setAttribute("list",all);
            //2.重定向
            if (all != null) {
                response.sendRedirect("stu_list.jsp");
            }
        }else {
            //登录失败
            System.out.println("登录失败");
            response.getWriter().write("登录失败");
        }
    }

}
```

11. 在stu_list.jsp中，取出域中的集合，然后使用c标签 去遍历集合。

```
<%--
  Created by IntelliJ IDEA.
  User: baishixin
  Date: 2020/8/31
```

```
  Time: 1:07 下午
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <h1>学生信息列表</h1>
    <table border="1" width="700" >
        <tr align="center">
            <th>编号</th>
            <th>姓名</th>
            <th>年龄</th>
            <th>性别</th>
            <th>地址</th>
            <th>操作</th>
        </tr>
        <c:forEach var="stu" items="${list}">
            <tr align="center">
                <td>${stu.id}</td>
                <td>${stu.name}</td>
                <td>${stu.age}</td>
                <td>${stu.gander}</td>
                <td>${stu.address}</td>
                <td><a href="#">更新</a><a href="#">删除</a> </td>
            </tr>
        </c:forEach>

    </table>
</body>
</html>
```

总结：

- JSP

  三大指令

  page
  include
  taglib

  三个动作标签

  jsp:include
  jsp:forward

jsp:param

九个内置对象

四个作用域
pageContext
request
session
application

out
exception
response
page
config

- EL

  ${ 表达式 }

  取4个作用域中的值

  ```
  ${ name }
  ```

  有11个内置对象。

  ```
  pageContext

  pageScope
  requestScope
  sessionScope
  applicationScope

  header
  headerValues

  param
  paramValues

  cookie
  initParam
  ```

- JSTL

> 使用1.1的版本， 支持EL表达式， 1.0不支持EL表达式

> 拷贝jar包， 通过taglib 去引入标签库

```
<c:set>
<c:if>
<c:forEach>
```