### JAVA第二阶段—DAY18-JAVA案例

1. Cookie快速入门，演示Cookie的创建发送流程

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

创建项目day03\_demo

步骤二：创建Servlet

在src目录下创建包com.bailiban.cookie,分别创建Cookie1Servlet和Cookie2Servlet

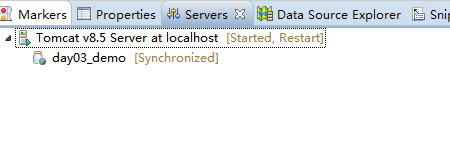
Cookie1Servlet 用于创建Cookie

|  |
| --- |
| **package** com.bailiban.cookie;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.Cookie;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  /\*\*  \* Servlet implementation class Cookie1Servlet  \*/  @WebServlet("/cookie1")  **public** **class** Cookie1Servlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //创建Cookie  Cookie c1 = **new** Cookie("name", "wangcai");    //发送Cookie 到客户端  response.addCookie(c1);    }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

Cookie2Servlet 用于获取Cookie

|  |
| --- |
| **package** com.bailiban.cookie;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.Cookie;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  /\*\*  \* Servlet implementation class Cookie1Servlet  \*/  @WebServlet("/cookie2")  **public** **class** Cookie2Servlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //从请求中获取cookie  Cookie[] cookies = request.getCookies();  //获取数据遍历cookie  **if**(cookies!=**null**) {  **for** (Cookie c : cookies) {  System.***out***.println(c.getName()+"--"+c.getValue());  }  }    }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

步骤三：将项目部署到Tomcat,启动服务器

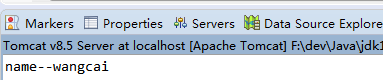


步骤四：浏览器访问

在浏览器地址输入：http://localhost:8080/day03\_demo/cookie1

再输入地址：http://localhost:8080/day03\_demo/cookie2

控制台打印：



1. 基于案例1演示 Cookie细节。

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

步骤二：在Cookie1Servlet中在添加代码

分别测试Cookie的保存时间，保存中文，一次发送多个Cookie

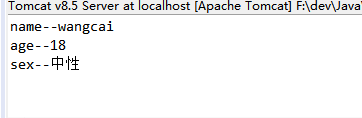
|  |
| --- |
| **package** com.bailiban.cookie;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.Cookie;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  /\*\*  \* Servlet implementation class Cookie1Servlet  \*/  @WebServlet("/cookie1")  **public** **class** Cookie1Servlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //创建Cookie  Cookie c1 = **new** Cookie("name", "wangcai");  //设置Cookie的生命周期 30秒  c1.setMaxAge(30);  //再创建Cookie  Cookie c2 = **new** Cookie("age", "18");  //再创建Cookie  Cookie c3 = **new** Cookie("sex", "中性");  //发送Cookie 到客户端  response.addCookie(c1);  response.addCookie(c2);  response.addCookie(c3);  }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

步骤四：浏览器访问

在浏览器地址输入：http://localhost:8080/day03\_demo/cookie1

再输入地址：http://localhost:8080/day03\_demo/cookie2

控制台打印



1. HttpSession快速入门，熟悉session中数据的存和取

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

步骤二：创建Servlet

在src目录下创建包com.bailiban.session,分别创建Session1Servlet和Session2Servlet

在Session1Servlet向session中存储数据

|  |
| --- |
| **package** com.bailiban.session;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  @WebServlet("/session1")  **public** **class** Session1Servlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  //使用session共享数据  //获取session  HttpSession session = request.getSession();  //System.out.println(session);  //存储数据  session.setAttribute("msg", "session数据");  }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

在Session2Servlet从session取数据

|  |
| --- |
| **package** com.bailiban.session;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  @WebServlet("/session2")  **public** **class** Session2Servlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //使用session获取数据  //获取session  HttpSession session = request.getSession();  //获取数据  Object msg = session.getAttribute("msg");  System.***out***.println(msg);    }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

步骤三：创建session.jsp页面

在该页面取出session中保存的数据

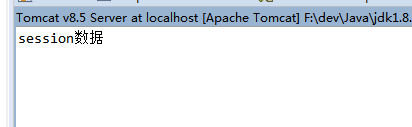
|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>获取session数据</title>  </head>  <body>  <%=session.getAttribute("msg") %>  </body>  </html> |

步骤五：浏览器访问 查看效果

在浏览器地址输入：http://localhost:8080/day03\_demo/session1

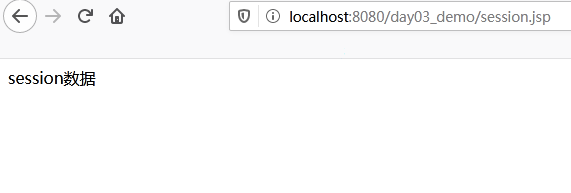
输入：http://localhost:8080/day03\_demo/session2

查看效果



输入：http://localhost:8080/day03\_demo/session.jsp

查看效果



1. 演示4大域对象的生命周期特点

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

步骤二：定义JSP页面

在web根目录WebContent 中创建目录include,在该目录中分别创建三个JSP文件

test1.jsp、test2.jsp、test3.jsp

test1.jsp中加入转发动作 请求test1.jsp 转发到test2.jsp

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>域对象生命周期1</title>  </head>  <body>  <%  //分别向四大域对象中放数据  pageContext.setAttribute("page", "pageContext数据");  request.setAttribute("rq", "request数据");  session.setAttribute("se", "session数据");  application.setAttribute("app", "application");  %>  <!-- 从四大域对象中取数据 -->  <%=pageContext.getAttribute("page") %><br/>  <%=request.getAttribute("rq") %><br/>  <%=session.getAttribute("se") %><br/>  <%=application.getAttribute("app") %>  <jsp:forward page=*"test2.jsp"*></jsp:forward>  </body>  </html> |

test2.jsp

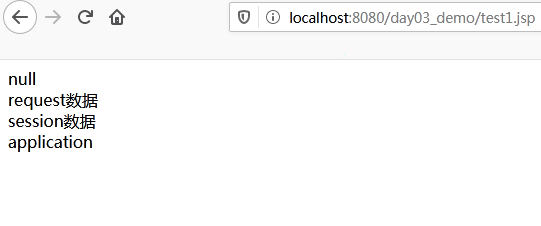
|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>域对象生命周期2</title>  </head>  <body>  <!-- 从四大域对象中取数据 -->  <%=pageContext.getAttribute("page") %><br/>  <%=request.getAttribute("rq") %><br/>  <%=session.getAttribute("se") %><br/>  <%=application.getAttribute("app") %>  </body>  </html> |

test3.jsp

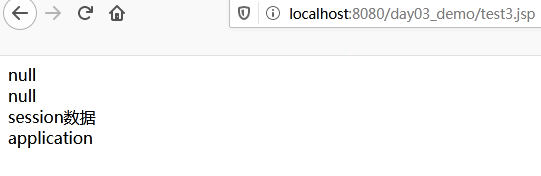
|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>域对象生命周期3</title>  </head>  <body>  <!-- 从四大域对象中取数据 -->  <%=pageContext.getAttribute("page") %><br/>  <%=request.getAttribute("rq") %><br/>  <%=session.getAttribute("se") %><br/>  <%=application.getAttribute("app") %>  </body>  </html> |
|  |

步骤三：浏览器访问查看效果

在浏览器地址输入：http://localhost:8080/day03\_demo/test1.jsp



在浏览器地址输入：http://localhost:8080/day03\_demo/test3.jsp



1. 演示EL表达式运算符的使用

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

步骤二：定义JSP页面

在web根目录WebContent 中创建目录el,在该目录中创建el1.jsp 页面

使用EL运算符

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>运算符</title>  </head>  <body>  <h1>算术运算符</h1>  ${8+3 }<br/>  ${8-3 }<br/>  ${8/3 }<br/>  ${8\*3 }<br/>  ${8%3 }  <h1>比较运算符</h1>  ${3>8}  <h1>逻辑运算符</h1>  ${3>1&&3>8}  <h1>三元运算符</h1>  ${3<8?**true**:**false**}  </body>  </html> |

步骤三：浏览器访问 查看效果

在浏览器地址输入：http://localhost:8080/day03\_demo/el/el1.jsp



1. 演示EL取值

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

步骤二：定义JSP页面

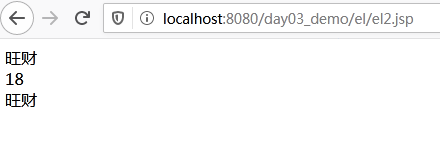
在web根目录WebContent 中创建目录el,在该目录中创建el2.jsp 页面

使用EL取值

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>获取域中数据</title>  </head>  <body>  <%  //向域中放数据  session.setAttribute("name", "如花");  request.setAttribute("name", "旺财");  session.setAttribute("age", 18);  %>  <!-- 使用EL 从指定域获取数据 -->  ${requestScope.name} <br/>  ${sessionScope.age} <br/>  <!-- 自动通过键名取值 优先取小范围 -->  ${name}    </body>  </html> |

步骤三：浏览器访问 查看效果

在浏览器地址输入：http://localhost:8080/day03\_demo/el/el2.jsp



1. 演示EL获取对象、List集合、Map集合的值

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

步骤二：定义实体类User

在src下定义包com.bailiban.domain，在该包下创建实体类User

|  |
| --- |
| **package** com.bailiban.domain;  **import** java.util.Date;  **public** **class** User {    **private** String name;  **private** **int** age;  **private** Date birthday;      **public** User() {  **super**();  // **TODO** Auto-generated constructor stub  }    **public** User(String name, **int** age, Date birthday) {  **super**();  **this**.name = name;  **this**.age = age;  **this**.birthday = birthday;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** **int** getAge() {  **return** age;  }  **public** **void** setAge(**int** age) {  **this**.age = age;  }  **public** Date getBirthday() {  **return** birthday;  }  **public** **void** setBirthday(Date birthday) {  **this**.birthday = birthday;  }      } |

步骤三：定义JSP页面

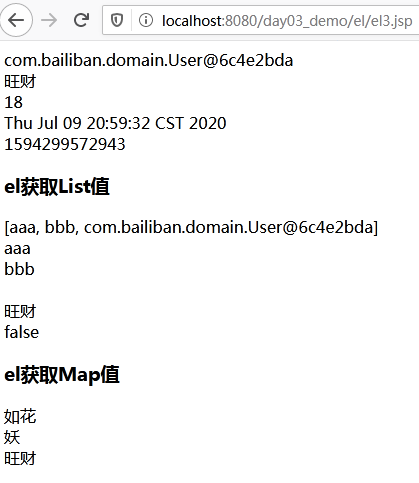
在web根目录WebContent 中创建目录el,在该目录中创建el3.jsp 页面

使用EL取对象、List集合、Map集合的值

|  |
| --- |
| <%@page import=*"java.util.Map"*%>  <%@page import=*"java.util.HashMap"*%>  <%@page import=*"java.util.ArrayList"*%>  <%@page import=*"java.util.List"*%>  <%@page import=*"java.util.Date"*%>  <%@page import=*"com.bailiban.domain.User"*%>  <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>获取对象属性值</title>  </head>  <body>  <%  //创建用户对象  User user = **new** User();  user.setName("旺财");  user.setAge(18);  user.setBirthday(**new** Date());  //将对象放入 request域对象  request.setAttribute("u", user);    List<Object> list = **new** ArrayList<>();  list.add("aaa");  list.add("bbb");  list.add(user);  request.setAttribute("list",list);    Map map = **new** HashMap();  map.put("sname","如花");  map.put("sex","妖");  map.put("user",user);  request.setAttribute("map",map);  %>  <%--  使用el 表达式 获取 域中对象  --%>  ${u} <br/>  <%--获取对象属性值 --%>  ${u.name}<br/>  ${u.age}<br/>  ${u.birthday}<br/>  ${u.birthday.time}<br/>    <h3>el获取List值</h3>  ${list}<br>  ${list[0]}<br>  ${list[1]}<br>  ${list[10]}<br>  ${list[2].name}  ${empty list}  <h3>el获取Map值</h3>  ${map.sname}<br>  ${map["sex"]}<br>  ${map.user.name}  </body>  </html> |

步骤五：浏览器访问 查看效果

在浏览器地址输入：<http://localhost:8080/day03_demo/el/el3.jsp>



1. 使用el内置对象pageContext动态获取虚拟目录

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

步骤二：定义JSP页面

在web根目录WebContent 中创建目录el,在该目录中创建el4.jsp 页面

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>el隐式对象</title>  </head>  <body>  <h3>在jsp页面动态获取虚拟目录</h3>  ${pageContext.request.contextPath}  </body>  </html> |

步骤三：浏览器访问

在浏览器地址输入：[http://localhost:8080/day03\_demo/el/el4.jsp](http://localhost:8080/day03_demo/el/el3.jsp)



1. 使用if标签判定集合是否为空，和数字的奇偶

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

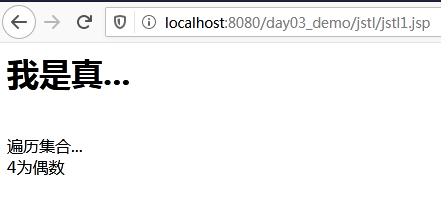
步骤二：定义JSP页面

在web根目录WebContent 中创建目录jstl,在该目录中创建jstl1.jsp 页面

|  |
| --- |
| <%@ page import=*"java.util.List"* %>  <%@ page import=*"java.util.ArrayList"* %>  <%@ page contentType=*"text/html;charset=UTF-8"* language=*"java"* %>  <%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <html>  <head>  <title>if标签</title>  </head>  <body>  <%--  c:if标签  1. 属性：  \* test 必须属性，接受boolean表达式  \* 如果表达式为true，则显示if标签体内容，如果为false，则不显示标签体内容  \* 一般情况下，test属性值会结合el表达式一起使用  2. 注意：c:if标签没有else情况，想要else情况，则可以在定义一个c:if标签  --%>  <c:if test=*"true"*>  <h1>我是真...</h1>  </c:if>  <br>  <%  //判断request域中的一个list集合是否为空，如果不为null则显示遍历集合  List list = **new** ArrayList();  list.add("aaa");  request.setAttribute("list",list);  request.setAttribute("num",4);  %>  <c:if test=*"*${**not empty** list}*"*>  遍历集合...  </c:if>  <br>  <c:if test=*"*${num% 2 != 0}*"*>  ${num}为奇数  </c:if>  <c:if test=*"*${num% 2 == 0}*"*>  ${num}为偶数  </c:if>  </body>  </html> |

步骤三：浏览器访问 查看效果

在浏览器地址输入：http://localhost:8080/day03\_demo/jstl/jstl1.jsp



1. 使用choose标签 完成数字编号对应星期

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

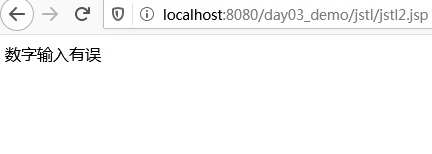
步骤二：定义JSP页面

在web根目录WebContent 中创建目录jstl,在该目录中创建jstl2.jsp 页面

|  |
| --- |
| <%@ page contentType=*"text/html;charset=UTF-8"* language=*"java"* %>  <%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <html>  <head>  <title>choose标签</title>  </head>  <body>  <%--  完成数字编号对应星期几案例  1.域中存储一数字  2.使用choose标签取出数字 相当于switch声明  3.使用when标签做数字判断 相当于case  4.otherwise标签做其他情况的声明 相当于default  --%>  <%  request.setAttribute("num",38);  %>  <c:choose>  <c:when test=*"*${num == 1}*"*>星期一</c:when>  <c:when test=*"*${num == 2}*"*>星期二</c:when>  <c:when test=*"*${num == 3}*"*>星期三</c:when>  <c:when test=*"*${num == 4}*"*>星期四</c:when>  <c:when test=*"*${num == 5}*"*>星期五</c:when>  <c:when test=*"*${num == 6}*"*>星期六</c:when>  <c:when test=*"*${num == 7}*"*>星期天</c:when>  <c:otherwise>数字输入有误</c:otherwise>  </c:choose>  </body>  </html> |

步骤三：浏览器访问 查看效果

在浏览器地址输入：http://localhost:8080/day03\_demo/jstl/jstl2.jsp



1. 演示foreach标签的使用

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

直接使用案例1项目

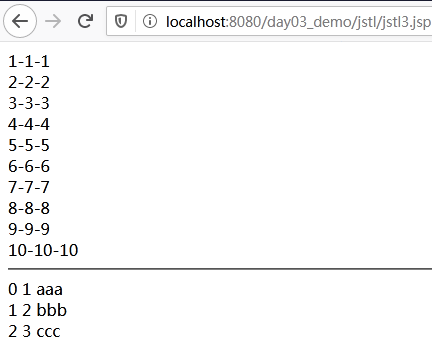
步骤二：定义JSP页面

在web根目录WebContent 中创建目录jstl,在该目录中创建jstl3.jsp 页面

|  |
| --- |
| <%@ page import=*"java.util.ArrayList"* %>  <%@ page import=*"java.util.List"* %>  <%@ page contentType=*"text/html;charset=UTF-8"* language=*"java"* %>  <%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <html>  <head>  <title>foreach标签</title>  </head>  <body>  <%--  foreach:相当于java代码的for语句  1. 完成重复的操作  for(int i = 0; i < 10; i ++){    }  \* 属性：  begin：开始值  end：结束值  var：临时变量  step：步长  varStatus:循环状态对象  index:容器中元素的索引，从0开始  count:循环次数，从1开始  2. 遍历集合  List<User> list;  for(User user : list){    }    \* 属性：  items:容器对象  var:容器中元素的临时变量  varStatus:循环状态对象  index:容器中元素的索引，从0开始  count:循环次数，从1开始  --%>  <c:forEach begin=*"1"* end=*"10"* var=*"i"* step=*"2"* varStatus=*"s"*>  ${i} <h3>${s.index}<h3> <h4> ${s.count} </h4><br>  </c:forEach>  <hr>  <%  List list = **new** ArrayList();  list.add("aaa");  list.add("bbb");  list.add("ccc");  request.setAttribute("list",list);  %>  <c:forEach items=*"*${list}*"* var=*"str"* varStatus=*"s"*>  ${s.index} ${s.count} ${str}<br>  </c:forEach>  </body>  </html> |

步骤三：浏览器访问 查看效果

在浏览器地址输入：http://localhost:8080/day03\_demo/jstl/jstl3.jsp



1. 优化登录和图书管理功能

需求：

1. 使用EL和JSTL替换JSP页面的所有JAVA代码,并动态获取项目名
2. 给用户登录增加会话跟踪技术，用户登录成功将用户信息放入session 跟踪其登录状态,在图书列表页面显示欢迎信息
3. 访问图书列表加入权限判断，如果用户没有登录成功，禁止访问，强制跳转到登录页面提示用户登录
4. 登录页面添加记住用户名功能,提高用户体验

* 参考答案

实现此案例需要按照如下步骤进行。

步骤一：创建JavaEE项目

在eclipse中复制项目day02\_book 命名为day03\_book

步骤二：使用EL和JSTL替换JSP页面的所有JAVA代码

导入JSTL相关jar包

javax.servlet.jsp.jstl.jar

jstl-impl.jar

使用taglib指令引入标签库

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

替换login.jsp页面java代码

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!-- 使用taglib 指令导入JSTL标签库 -->  <%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>管理员登录</title>  <!--导入框架样式文件-->  <link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/bootstrap.min.css"* />  <link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/base.css"* />  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/jquery1.12.4.min.js"*></script>  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/bootstrap.min.js"*></script>  </head>  <body>  <div class=*"container"*>  <h2>管理员登录</h2>  <form method=*"post"* action=*"*${pageContext.request.contextPath}*/login"*>  <div class=*"form-group"*>  <label class=*"control-label"*>用户名</label>  <input type=*"text"* class=*"form-control"* placeholder=*"请输入用户名"* name=*"username"* required />  </div>  <div class=*"form-group"*>  <label class=*"control-label"*>密码</label>  <input type=*"password"* class=*"form-control"* placeholder=*"请输入密码"* name=*"password"* required />  </div>  <button type=*"submit"* class=*"btn btn-success btn-block"*>登录</button>  <button type=*"submit"* class=*"btn btn-danger btn-block"*>取消</button>  </form>  <!-- 使用小脚本 从request域中 取出 错误信息-->  <c:if test=*"*${msg!=null}*"*>  <div class=*"alert alert-warning alert-dismissible"* role=*"alert"*>  <button type=*"button"* class=*"close"* data-dismiss=*"alert"* aria-label=*"Close"*>  <span aria-hidden=*"true"*>&times;</span>  </button>  ${msg}  </div>  </c:if>    </div>  </body>  </html> |

替换book-list.jsp页面java代码

|  |
| --- |
| <%@page import=*"com.bailiban.domian.Book"*%>  <%@page import=*"java.util.List"*%>  <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!-- 使用taglib 指令导入JSTL标签库 -->  <%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>图书列表</title>  <!--导入框架样式文件-->  <link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/bootstrap.min.css"* />  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/jquery1.12.4.min.js"*></script>  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/bootstrap.min.js"*></script>  </head>  <body>  <div class=*"container"*>  <div class=*"panel panel-default"*>  <div class=*"panel-heading clearfix"*>  <a href=*"/day02\_book/book-add.jsp"* target=*"main"* class=*"btn btn-primary pull-right"* >添加书籍</a>  </div>    <table class=*"table table-bordered table-hover"*>  <tr class=*"success"*>  <th>编号</th>  <th>书名</th>  <th>图片</th>  <th>价格</th>  <th>作者</th>  <th>出版日期</th>  <th>操作</th>  </tr>  <!-- 使用foreach标签 遍历 -->  <c:forEach items=*"*${list}*"* var=*"book"*>  <tr>  <td>${book.id}</td>  <td>${book.name}</td>  <td ><img src=*"*${pageContext.request.contextPath}*/img/*${book.image}*"* width=*"50"* /></td>  <td>${book.price}</td>  <td>${book.author}</td>  <td>${book.publishDate}</td>  <td>  <a class=*"btn btn-default btn-sm"* href=*""*>修改</a>&nbsp;  <a class=*"btn btn-default btn-sm"* href=*"/day02\_book/book/delete?id=*${book.id}*"*>删除</a>  </td>  </tr>  </c:forEach>    </table>  </div>  </div>  </body>  </html> |

步骤三：给用户登录增加会话跟踪技术

用户登录成功将用户信息放入session 跟踪其登录状态

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** java.util.Map;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  **import** org.apache.commons.beanutils.BeanUtils;  **import** com.bailiban.domian.User;  **import** com.bailiban.service.UserService;  **import** com.bailiban.service.impl.UserServiceImpl;  /\*\*  \* Servlet implementation class LoginServlet  \*/  @WebServlet("/login")  **public** **class** LoginServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** LoginServlet() {  **super**();  }    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //使用request对象解决 post 提交中文乱码  request.setCharacterEncoding("utf-8");  //使用BeanUtils优化参数的封装  //获取所有请求参数  Map<String, String[]> map = request.getParameterMap();  //将参数数据封装进实体类  User loginUser = **new** User();  **try** {  BeanUtils.*populate*(loginUser, map);  } **catch** (Exception e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  //实例化UserService对象  UserService userService = **new** UserServiceImpl();  //调用登录方法 得到从数据库中查询的用户对象  User user = userService.login(loginUser);  //判断用户是否登录成功  **if**(user!=**null**) {    //用户登录成功， 跟踪用户登录状态 将用户对象保存到session中  HttpSession session = request.getSession();  session.setAttribute("u", user);  //重定向到 图书列表页面  response.sendRedirect(request.getContextPath()+"/book/list");    }**else** {  //出现错误 向request域中放错误信息  request.setAttribute("msg", "用户登录失败!!!");  //转发到login.jsp 显示错误提示  request.getRequestDispatcher("/login.jsp").forward(request, response);    }    }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  //无论是get还是post请求 都执行doGet方法 避免写重复的代码  doGet(request, response);  }  } |

步骤四：图书列表页面显示欢迎信息

在book-list.jsp页面使用EL表达式从session域中取出用户名

|  |
| --- |
| <%@page import=*"com.bailiban.domian.Book"*%>  <%@page import=*"java.util.List"*%>  <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!-- 使用taglib 指令导入JSTL标签库 -->  <%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>图书列表</title>  <!--导入框架样式文件-->  <link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/bootstrap.min.css"* />  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/jquery1.12.4.min.js"*></script>  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/bootstrap.min.js"*></script>  </head>  <body>  <div class=*"container"*>  <!-- 从session取出用户信息 显示欢迎信息 -->  <h1>欢迎您:${u.username}</h1>  <div class=*"panel panel-default"*>  <div class=*"panel-heading clearfix"*>  <a href=*"/day02\_book/book-add.jsp"* target=*"main"* class=*"btn btn-primary pull-right"* >添加书籍</a>  </div>    <table class=*"table table-bordered table-hover"*>  <tr class=*"success"*>  <th>编号</th>  <th>书名</th>  <th>图片</th>  <th>价格</th>  <th>作者</th>  <th>出版日期</th>  <th>操作</th>  </tr>  <!-- 使用foreach标签 遍历 -->  <c:forEach items=*"*${list}*"* var=*"book"*>  <tr>  <td>${book.id}</td>  <td>${book.name}</td>  <td ><img src=*"*${pageContext.request.contextPath}*/img/*${book.image}*"* width=*"50"* /></td>  <td>${book.price}</td>  <td>${book.author}</td>  <td>${book.publishDate}</td>  <td>  <a class=*"btn btn-default btn-sm"* href=*""*>修改</a>&nbsp;  <a class=*"btn btn-default btn-sm"* href=*"/day02\_book/book/delete?id=*${book.id}*"*>删除</a>  </td>  </tr>  </c:forEach>    </table>  </div>  </div>  </body>  </html> |

步骤五：访问图书列表加入权限判断

如果用户没有登录成功，禁止访问，强制跳转到登录页面提示用户登录

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** java.util.List;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  **import** com.bailiban.domian.Book;  **import** com.bailiban.service.BookService;  **import** com.bailiban.service.impl.BookServiceImpl;  /\*\*  \* Servlet implementation class BookListServlet  \*/  @WebServlet("/book/list")  **public** **class** BookListServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //从session中取出用户对象  HttpSession session = request.getSession();  Object user = session.getAttribute("u");  //如果用户对象为空  **if**(user==**null**) {  //回显登录页面 提示用户登录  //向request域中放入错误信息提示  request.setAttribute("msg", "亲！请先登录再操作!!!");  //转发到 login.jsp页面  request.getRequestDispatcher("/login.jsp").forward(request, response);  **return**;  }  //创建service  BookService bookService = **new** BookServiceImpl();    //查询所有书籍  List<Book> list = bookService.selecAll();    //将数据放入request域  request.setAttribute("list", list);  //转发到 book-list.jsp页面显示  request.getRequestDispatcher("/book-list.jsp").forward(request, response);      }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

步骤六：登录页面添加记住用户名功能,提高用户体验

如果用户勾选复选框，登录成功 记住用户名，将用户名添加到Cookie

如果没有勾选，登录成功 删除保存用户名的cookie

LoginServlet 代码如下

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** java.util.Map;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.Cookie;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  **import** org.apache.commons.beanutils.BeanUtils;  **import** com.bailiban.domian.User;  **import** com.bailiban.service.UserService;  **import** com.bailiban.service.impl.UserServiceImpl;  /\*\*  \* Servlet implementation class LoginServlet  \*/  @WebServlet("/login")  **public** **class** LoginServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** LoginServlet() {  **super**();  }    **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //使用request对象解决 post 提交中文乱码  request.setCharacterEncoding("utf-8");  //使用BeanUtils优化参数的封装  //获取所有请求参数  Map<String, String[]> map = request.getParameterMap();  //将参数数据封装进实体类  User loginUser = **new** User();  **try** {  BeanUtils.*populate*(loginUser, map);  } **catch** (Exception e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  //实例化UserService对象  UserService userService = **new** UserServiceImpl();  //调用登录方法 得到从数据库中查询的用户对象  User user = userService.login(loginUser);  //判断用户是否登录成功  **if**(user!=**null**) {  //添加记住用户名功能  //获取复选框的值  String remember = request.getParameter("remember");  //如果勾选了  **if**(remember!=**null**){    //创建Cookie保存用户名  Cookie cookie = **new** Cookie("username",user.getUsername());  //设置保存时间  cookie.setMaxAge(60\*60\*24\*30);//30天  //发送cookie到浏览器  response.addCookie(cookie);    }**else** {  //如果没有勾选  //获取cookie  Cookie[] cookies = request.getCookies();  **if**(cookies!=**null**) {    **for** (Cookie c : cookies) {    //找到name为username的cookie  **if**("username".equals(c.getName())) {    //删除该cookie  c.setMaxAge(0);  //发送到浏览器  response.addCookie(c);  **break**;  }    }    }    }    //用户登录成功， 跟踪用户登录状态 将用户对象保存到session中  HttpSession session = request.getSession();  session.setAttribute("u", user);  //重定向到 图书列表页面  response.sendRedirect(request.getContextPath()+"/book/list");    }**else** {  //出现错误 向request域中放错误信息  request.setAttribute("msg", "用户登录失败!!!");  //转发到login.jsp 显示错误提示  request.getRequestDispatcher("/login.jsp").forward(request, response);    }    }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  //无论是get还是post请求 都执行doGet方法 避免写重复的代码  doGet(request, response);  }  } |

login.jsp页面

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!-- 使用taglib 指令导入JSTL标签库 -->  <%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>管理员登录</title>  <!--导入框架样式文件-->  <link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/bootstrap.min.css"* />  <link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/base.css"* />  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/jquery1.12.4.min.js"*></script>  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/bootstrap.min.js"*></script>  </head>  <body>  <div class=*"container"*>  <h2>管理员登录</h2>  <form method=*"post"* action=*"*${pageContext.request.contextPath}*/login"*>  <div class=*"form-group"*>  <label class=*"control-label"*>用户名</label>  <input type=*"text"* class=*"form-control"* value=*"*${cookie.username.value}*"* placeholder=*"请输入用户名"* name=*"username"* required />  </div>  <div class=*"form-group"*>  <label class=*"control-label"*>密码</label>  <input type=*"password"* class=*"form-control"* placeholder=*"请输入密码"* name=*"password"* required />  </div>  <div class=*"checkbox"*>  <label>  <input type=*"checkbox"* ${cookie.username.value!=null?"checked='checked'":''} value=*"remember"* name=*"remember"*> 记住用户名  </label>  </div>  <button type=*"submit"* class=*"btn btn-success btn-block"*>登录</button>  <button type=*"submit"* class=*"btn btn-danger btn-block"*>取消</button>  </form>  <!-- 使用小脚本 从request域中 取出 错误信息-->  <c:if test=*"*${msg!=null}*"*>  <div class=*"alert alert-warning alert-dismissible"* role=*"alert"*>  <button type=*"button"* class=*"close"* data-dismiss=*"alert"* aria-label=*"Close"*>  <span aria-hidden=*"true"*>&times;</span>  </button>  ${msg}  </div>  </c:if>    </div>  </body>  </html> |

步骤七：浏览器访问

在浏览器地址输入：[http://localhost:8080/day03\_book/login.jsp](http://localhost:8080/day02_book/login.jsp)



登录成功 ，勾选记住用户名

登录成功 显示图书列表 欢迎信息



下次再访问登录界面



已经将用户名记住

如果没有登录就访问图书列表

