### JAVA第二阶段—DAY18-JAVA作业答案

1. 什么是会话管理技术?

会话：一次会话中包含多次请求和响应。

一次会话：浏览器第一次给服务器资源发送请求，会话建立，直到有一方断开为止

功能：在一次会话的范围内的多次请求间，共享数据

方式：客户端会话技术：Cookie、服务器端会话技术：Session

1. cookie和session的区别?

session与Cookie的区别：

1. session存储数据在服务器端，Cookie在客户端

2. session没有数据大小限制，Cookie有

3. session数据安全，Cookie相对于不安全

1. 讲解JSP中的四种作用域？

pageContext 当期页面 代理其他域对象

Request 一次请求 数据只是做显示的，看完了就没用了 （请求转发， Servlet 产生的处理结果(数据) 交给jsp显示）

Session 一次会话 保存用户特有数据 或 数据给用户看完了，一会还要用，会话结束了就没用了

application与服务器共存亡 数据给一个用户用完了，别人还要用，保存所有用户共享的数据

1. 编程题：使用 cookie技术 统计每个用户访问网站的次数

需求：

访问一个jsp页面，每访问一次统计用户的访问次数在页面输出 当前访问次数为n次

分析：

1. 可以采用Cookie来完成

2. 在服务器中的JSP判断是否有一个名为count的cookie

有：

1. 从该cookie中取出 上一次的访问次数 ，并 +1

2. 写回Cookie

没有：是第一次访问

1. 创建一个名为count的cookie ,并将值设置为 1

2. 写回Cookie

* 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day03\_demo项目

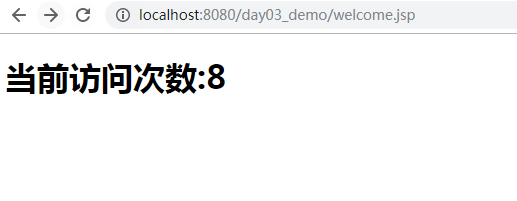
步骤二：定义JSP页面

在web根目录WebContent 中创建welcome.jsp 页面，书写功能实现代码

|  |
| --- |
| <%@page import=*"java.net.URLDecoder"*%>  <%@page import=*"java.net.URLEncoder"*%>  <%@page import=*"java.util.Date"*%>  <%@page import=*"java.text.SimpleDateFormat"*%>  <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>欢迎页面</title>  </head>  <body>  <%  //创建计数器 记录访问次数  **int** count = 0;  //标记是否有 名为 count的cookie存在  **boolean** flag = **false**; //默认不存在  //获取所有Cookie  Cookie[] cookies = request.getCookies();  //遍历Cookie数组  **if** (cookies != **null**&&cookies.length > 0) {  **for** (Cookie c : cookies) {  //获取Cookie的名称  String name = c.getName();  //判定名称是否是 count  **if** ("count".equals(name)) {  //如果是count 获取count值  count = Integer.parseInt(c.getValue());  //记录一次  count++;  //更新该cookie中保持的数据  c.setValue(count+"");  //设置生命周期  c.setMaxAge(60\*60);//保持一小时  //保存到浏览器  response.addCookie(c);  //设置为存在  flag=**true**;  }  }  }    //如果是第一次访问  **if**(cookies == **null**||cookies.length ==0 || flag==**false**){  //访问一次记录一次  count++;  //创建cookie  Cookie cookie = **new** Cookie("count", count + "");  //将次数保存在cookie  cookie.setMaxAge(60\*60);//保持一小时  response.addCookie(cookie);  }    %>  <h1>当前访问次数:<%=count%></h1>  </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/welcome.jsp



1. 编程题：list集合数据展示到jsp页面的表格table中

思路：

在request域中有一个存有User对象的List集合

需要使用jstl+el将list集合数据展示到jsp页面的表格table中

* 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day03\_demo项目

步骤二：定义JSP页面

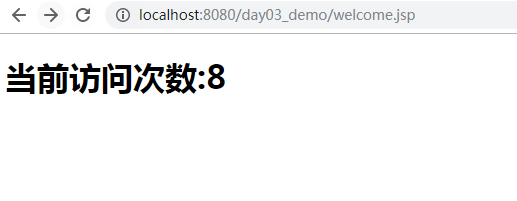
在web根目录WebContent 中创建目录jstl,在该目录中创建jstl\_test.jsp 页面

代码如下

|  |
| --- |
| <%@page import=*"com.bailiban.domain.User"*%>  <%@ page import=*"java.util.List"* %>  <%@ page import=*"java.util.ArrayList"* %>  <%@ page import=*"java.util.Date"* %>  <%@ page contentType=*"text/html;charset=UTF-8"* language=*"java"* %>  <%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>  <html>  <head>  <title>test</title>  </head>  <body>  <%  List list = **new** ArrayList();  list.add(**new** User("如花",18,**new** Date()));  list.add(**new** User("小强",28,**new** Date()));  list.add(**new** User("旺财",18,**new** Date()));  request.setAttribute("list",list);  %>  <table border=*"1"* width=*"500"* align=*"center"*>  <tr>  <th>编号</th>  <th>姓名</th>  <th>年龄</th>  <th>生日</th>  </tr>  <%--数据行--%>  <c:forEach items=*"*${list}*"* var=*"user"* varStatus=*"s"*>  <c:if test=*"*${s.count % 2 != 0}*"*>  <tr bgcolor=*"green"*>  <td>${s.count}</td>  <td>${user.name}</td>  <td>${user.age}</td>  <td>${user.birthday}</td>  </tr>  </c:if>  <c:if test=*"*${s.count % 2 == 0}*"*>  <tr bgcolor=*"yellow"*>  <td>${s.count}</td>  <td>${user.name}</td>  <td>${user.age}</td>  <td>${user.birthday}</td>  </tr>  </c:if>  </c:forEach>  </table>  </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/welcome.jsp



1. 编程题 ：完成图书的修改功能

思路：

修改功能分为2个步骤

1. 用户点击修改按钮，跳转到修改页面，回显图书以前的数据
2. 点击修改页面的修改按钮，完成修改 回到图书列表页面

步骤一：编写图书 dao层

在BookDao中 编写 根据id查询图书和修改图书 二个方法

|  |
| --- |
|  |
| //根据id查询图书  @Override  **public** Book findById(**int** id) {  //定义SQL语句  String sql ="SELECT \* FROM t\_book WHERE id = ? ";  Book book = template.queryForObject(sql, **new** BeanPropertyRowMapper<Book>(Book.**class**), id);  **return** book;  }  //修改图书  **public** **void** update (Book book) {    //定义添加sql语句  String sql = "UPDATE t\_book SET NAME= ? ,price=?,author=? ,image =? ,publish\_date=? WHERE id = ?";  //定义添加参数  Object[] args = {  book.getName(),  book.getPrice(),  book.getAuthor(),  book.getImage(),  book.getPublishDate(),  book.getId()    };  template.update(sql,args );    } |

步骤二：编写图书 service层

在BookService中 编写 根据id查询图书和修改图书 二个方法

|  |
| --- |
| //根据id查询一个图书  @Override  **public** Book findById(**int** id) {  **return** bookDao.findById(id);  }  //修改图书  @Override  **public** **void** update(Book book) {  bookDao.update(book);  } |

步骤三：创建显示修改页面的Servlet和完成修改的servlet

显示修改页面的BookUpdateUIServlet

|  |
| --- |
| **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** com.bailiban.domian.Book;  **import** com.bailiban.service.BookService;  **import** com.bailiban.service.impl.BookServiceImpl;  /\*\*  \* Servlet implementation class BookUpdateUIServlet  \*/  @WebServlet("/book/updateUI")  **public** **class** BookUpdateUIServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **public** BookUpdateUIServlet() {  **super**();  // **TODO** Auto-generated constructor stub  }  **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //获取书籍id  **int** id = Integer.*parseInt*(request.getParameter("id"));  //创建service  BookService bookService = **new** BookServiceImpl();    //根据id查询一个书籍  Book book = bookService.findById(id);  //将数据放入request域  request.setAttribute("book", book);  //转发到book-update.jsp页面显示  request.getRequestDispatcher("/book-update.jsp").forward(request, response);  }    **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

完成修改功能的BookUpdateServlet

|  |
| --- |
| **package** com.bailiban.servlet;  **import** java.io.IOException;  **import** java.lang.reflect.InvocationTargetException;  **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** org.apache.commons.beanutils.BeanUtils;  **import** com.bailiban.domian.Book;  **import** com.bailiban.service.BookService;  **import** com.bailiban.service.impl.BookServiceImpl;  @WebServlet("/book/update")  **public** **class** BookUpateServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;      **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    //设置请求编码  request.setCharacterEncoding("utf-8");  //获取表单提交所有参数的map集合  Map<String, String[]> map = request.getParameterMap();  //创建图书对象  Book book = **new** Book();  //将参数封装进对象  **try** {  BeanUtils.*populate*(book, map);  } **catch** (IllegalAccessException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (InvocationTargetException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  //创建service  BookService bookService = **new** BookServiceImpl();  //完成修改  bookService.update(book);  //添加成功跳转到图书列表页面  response.sendRedirect(request.getContextPath()+"/book/list");      }  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  // **TODO** Auto-generated method stub  doGet(request, response);  }  } |

步骤五：创建修改页面 book-update.jsp

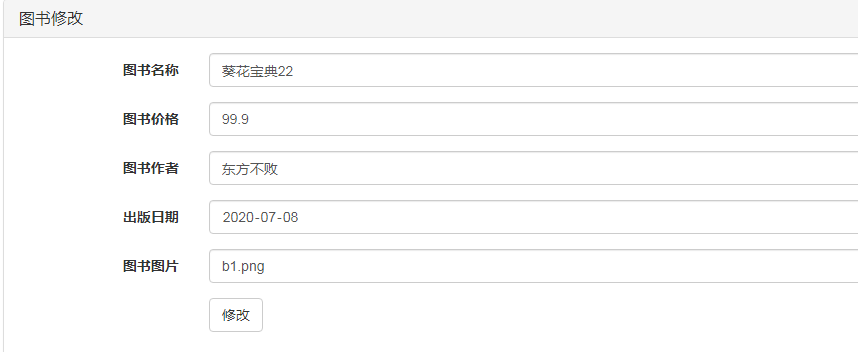
|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>图书添加</title>  <!--导入框架样式文件-->  <link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/bootstrap.min.css"* />  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/jquery1.12.4.min.js"*></script>  <script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/bootstrap.min.js"*></script>  <script type=*"text/javascript"*>  $(**function**(){  //处理时间回显问题  **var** publishDate = "${book.publishDate}";  publishDate = publishDate.substring(publishDate,publishDate.indexOf(" "));  $("input[name='publishDate']").val(publishDate);    })  </script>  </head>  <body>  <div class=*"container"*>  <div class=*"panel panel-default"*>  <div class=*"panel-heading"*>  <h3 class=*"panel-title"*>图书修改</h3>  </div>  <div class=*"panel-body"*>  <form class=*"form-horizontal"* method=*"post"* action=*"*${pageContext.request.contextPath}*/book/update"*>  <!--隐藏域 -->  <input type=*"hidden"* name=*"id"* value=*"*${book.id }*"* />  <div class=*"form-group"*>  <label for=*"name"* class=*"col-sm-2 control-label"*>图书名称</label>  <div class=*"col-sm-10"*>  <input type=*"text"* value=*"*${book.name}*"* class=*"form-control"* name=*"name"* placeholder=*"请输入图书名称"* required >  </div>  </div>  <div class=*"form-group"*>  <label for=*"inputEmail3"* class=*"col-sm-2 control-label"*>图书价格</label>  <div class=*"col-sm-10"*>  <input type=*"text"* value=*"*${book.price}*"* class=*"form-control"* name=*"price"* placeholder=*"请输入图书价格"* required >  </div>  </div>  <div class=*"form-group"*>  <label for=*"inputEmail3"* class=*"col-sm-2 control-label"*>图书作者</label>  <div class=*"col-sm-10"*>  <input type=*"text"* value=*"*${book.author}*"* class=*"form-control"* name=*"author"* placeholder=*"请输入图书作者"* required >  </div>  </div>  <div class=*"form-group"*>  <label for=*"inputEmail3"* class=*"col-sm-2 control-label"*>出版日期</label>  <div class=*"col-sm-10"*>  <input type=*"date"* value=*"*${book.publishDate}*"* class=*"form-control"* name=*"publishDate"* placeholder=*"请输入出版日期"* required>  </div>  </div>  <div class=*"form-group"*>  <label for=*"inputEmail3"* class=*"col-sm-2 control-label"*>图书图片</label>  <div class=*"col-sm-10"*>  <input type=*"text"* value=*"*${book.image}*"* class=*"form-control"* name=*"image"* placeholder=*"请输入图书图片"* required>  </div>  </div>    <div class=*"form-group"*>  <div class=*"col-sm-offset-2 col-sm-10"*>  <button type=*"submit"* class=*"btn btn-default"*>修改</button>  </div>  </div>  </form>  </div>  </div>  </div>  </body>  </html> |

步骤六：设置图书列表页面book-list.jsp的修改按钮跳转地址

|  |
| --- |
| <a href=*"*${pageContext.request.contextPath}*/book/updateUI?id=*${book.id}*"*>修改</a> |

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)



点击修改

