1. JSP开发要点

**1. JSP开发要点**

**1.1. 转发**

**1.1.1. 什么是转发**

在Web服务器端处理用户请求的时候，会有需要多个Web组件配合才能完成的情况。一个Web组件（Servlet/JSP）将未完成的处理通过容器转交给另外一个Web组件继续完成，这个转交的过程叫做转发。

常见情况是Servlet负责获取数据，然后将数据转交给JSP进行展现。

**1.1.2. 如何实现转发**

实现转发过程可遵循以下三个步骤：

**步骤一、绑定数据到request对象**

在转交的过程中一定会有数据的传递，并且涉及到的Web组件都是针对同一个请求，所以利用request来保存共同处理的数据不仅仅能让数据共享，也能够随着请求响应的结束而销毁，不会占用服务器更多的资源。使用如下代码可以实现数据的绑定:

1. request**.**setAttribute**(**String name**,**Object obj**);**

setAttribute()方法实现数据绑定，与其对应的还有getAttribute（）方法获取绑定的数据，removeAttribute（）方法移除绑定的数据。

**步骤二、获得转发器**

使用如下代码可以获取到转发器，用于说明转交的下一个组件的路径:

1. RequestDispatcher rd **=** request**.**getRequestDispatcher**(**String uri**);**

**步骤三、实现转发**

使用转发器完成转发的动作，因下一个Web组件要针对同一个请求和响应继续完成后续的工作，所以在转发时要将本次的请求和响应对象作为参数传给下一个Web组件。实现代码如下所示：

1. rd**.**forward**(**request**,**response**);**

其中步骤二和步骤三可以合并为一行代码：

1. request**.**getRequestDispatcher**(**String uri**).**forward**(**request**,**response**);**

**1.1.3. 转发的原理**

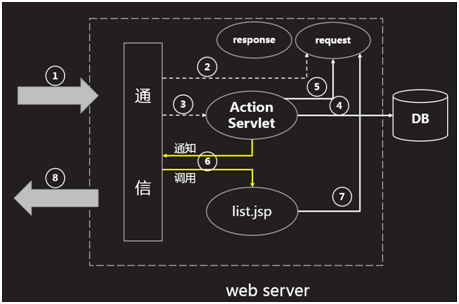


图 – 1

如图- 1所示，是客户端向服务器请求查看员工列表信息页面时服务器端的处理过程。

1.请求到达服务器(2.根据请求信息创建request和response对象(3.根据请求资源路径找到对应的Servlet执行处理(4.Servlet在处理过程中从数据库获取到结果信息(5.Servlet将结果信息绑定在request对象中(6.Servlet通知容器将request和response对象转交给list.jsp页面继续执行对请求的响应(7. list.jsp页面被调用并执行时从传递过来的request对象中获取绑定的数据生成结果页面(8.服务器将list.jsp页面的执行结果返回给客户端。

在整个的处理过程中，从ActionServlet到list.jsp的这个转换过程就是转发，也就是图示中的6这个步骤。控制这个过程的是服务器，同时ActionServlet和list.jsp共享了同一组请求和响应，两个Web组件共同协作完成了客户端的一次请求。对于客户端来讲，只发了一次请求，并且客户端浏览器也并不知道在服务器端是由两个组件配合提供的响应。

**1.1.4. 转发的特点**

转发过程发生在服务器端，客户端只发送了一个请求，虽然请求到达服务器的指定位置后被容器控制着传到了第二个组件继续完成工作，但浏览器并不知道这个过程，所以转发之后地址栏地址不会发生变化。

转发的目的地必须是同一个应用内部的某个地址，决不能跳出应用。毕竟这个转交过程由容器实现，容器只有访问本应用的权限，而不能控制请求到达应用以外的位置。

转发过程中涉及到的所有Web组件共享同一个request对象和response对象，数据的传递和共享就依赖request对象。

***注意：在forward之后的其他语句还是会继续执行完的，只要不报异常。***

**1.1.5. 转发和重定向的区别**

重定向：浏览器发送请求到容器访问A，A可以发送一个状态码302和一个Location消息头到浏览器，于是浏览器会立即向Location发新的请求。

转发：浏览器发送请求到容器访问A，A可以通知容器去调用B。转发所涉及的各个Web组件会共享同一个request和response对象；而重定向不行。

说明：当请求到达容器，容器会创建request对象和response对象。当响应发送完毕，容器会立即删除request对象和response对象。即，request对象和response对象的生存时间是一次请求与响应期间。

转发之后，浏览器地址栏的地址不变，重定向会变。转发的地址必须是同一个应用内部某个地址，而重定向没有这个限制。转发是一件事情未做完，调用另外一个组件继续做；而重定向是一件事情已经做完，再做另外一件事情。

转发和重定向的执行过程如图- 2所示：

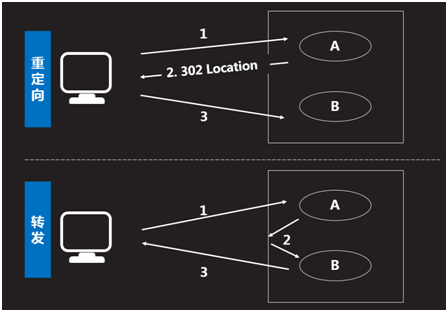


图 - 2

**JAVA SERVLETJSP**

1. 员工管理——使用转发、JSP实现完整的增删改查

**1 员工管理——使用转发、JSP实现完整的增删改查**

**1.1 问题**

将美工制作的页面转变成JSP页面，使用转发、重定向等技术实现完整的员工管理功能

**1.2 方案**

功能的完成顺序为查询(增加(删除(修改。在查询功能中，JSP页面负责展示，数据来源使用转发的技术从request对象中获取。请求的地址映射规则为后缀匹配模式，由一个ActionServlet处理分发请求。

**1.3 步骤**

**1.4 完整代码**

DBUtil.java和EmployeeDAO.java和Employee.java和ActionServlet.java和lisetEmp1.jsp和addEmp.jsp和updateEmp.jsp和web.xml文件代码参考。

EmployeeDao.java代码如下：

package dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import entity.Employee;

public class EmployeeDao {

public List<Employee> findAll() throws ClassNotFoundException, SQLException{

ArrayList<Employee> emps = new ArrayList<Employee>();

Connection con=null;

PreparedStatement ps=null;

ResultSet rs=null;

con=DBUtil.getConnection();

ps=con.prepareStatement("select \* from t\_emp");

rs=ps.executeQuery();

while(rs.next()){

Employee emp=new Employee(rs.getInt("id"),rs.getString("name"),rs.getDouble("salary"),rs.getInt("age"));

emps.add(emp);

}

DBUtil.close(con);

return emps;

}

//删除员工

public void delete(int id) throws ClassNotFoundException, SQLException{

Connection con=null;

PreparedStatement ps=null;

con=DBUtil.getConnection();

ps=con.prepareStatement("delete from t\_emp where id=?");

ps.setInt(1, id);

ps.executeUpdate();

DBUtil.close(con);

}

//增加员工

public void save(Employee emp) throws SQLException, ClassNotFoundException{

Connection con=null;

PreparedStatement ps=null;

con=DBUtil.getConnection();

ps=con.prepareStatement("insert into t\_emp(name,salary,age) values(?,?,?)");

ps.setString(1, emp.getName());

ps.setDouble(2, emp.getSalary());

ps.setInt(3, emp.getAge());

ps.executeUpdate();

DBUtil.close(con);

}

//根据id查询员工

public Employee findById(int id) throws ClassNotFoundException, SQLException{

Connection con=null;

PreparedStatement ps=null;

ResultSet rs=null;

Employee emp=null;

con=DBUtil.getConnection();

ps=con.prepareStatement("select \* from t\_emp where id=?");

ps.setInt(1, id);

rs=ps.executeQuery();

if(rs.next()){

emp = new Employee(rs.getInt("id"),rs.getString("name"),rs.getDouble("salary"),rs.getInt("age"));

}

DBUtil.close(con);

return emp;

}

//修改员工信息

public void modify(Employee emp) throws ClassNotFoundException, SQLException{

Connection con=null;

PreparedStatement ps=null;

con=DBUtil.getConnection();

ps=con.prepareStatement("update t\_emp set name=?,salary=?,age=? where id=?");

ps.setString(1, emp.getName());

ps.setDouble(2, emp.getSalary());

ps.setInt(3, emp.getAge());

ps.setInt(4, emp.getId());

ps.executeUpdate();

DBUtil.close(con);

}

}

ActionServlet.java代码如下：

   package web;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import dao.EmployeeDao;

import entity.Employee;

public class ActionServlet extends HttpServlet {

@Override

protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

String uri = request.getRequestURI();

String action = uri.substring(uri.lastIndexOf("/")+1, uri.lastIndexOf("."));

EmployeeDao dao=new EmployeeDao();

if(action.equals("list")){

try {

List<Employee> emps = dao.findAll();

request.setAttribute("emps", emps);

request.getRequestDispatcher("listEmp1.jsp").forward(request, response);

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}else if(action.equals("add")){

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

double salary=Double.valueOf(request.getParameter("salary"));

int age=Integer.valueOf(request.getParameter("age"));

Employee emp=new Employee();

emp.setName(name);;

emp.setSalary(salary);

emp.setAge(age);

try {

dao.save(emp);

response.sendRedirect("list.do");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}else if(action.equals("delete")){

int id=Integer.valueOf(request.getParameter("id"));

try {

dao.delete(id);

response.sendRedirect("list.do");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}else if(action.equals("load")){

int id=Integer.valueOf(request.getParameter("id"));

try {

Employee emp=dao.findById(id);

request.setAttribute("emp", emp);

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

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}else if(action.equals("update")){

int id=Integer.valueOf(request.getParameter("id"));

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

double salary=Double.valueOf(request.getParameter("salary"));

int age=Integer.valueOf(request.getParameter("age"));

Employee emp=new Employee();

emp.setId(id);

emp.setName(name);

emp.setSalary(salary);

emp.setAge(age);

try {

dao.modify(emp);

response.sendRedirect("list.do");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

listEmp1.jsp文件代码如下：

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@page import="entity.\*,dao.\*,java.util.\*" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

<style type="text/css">

.row1{

background-color: red;

}

.row2{

background-color: yellow;

}

</style>

</head>

<body>

<table border="1px">

<caption>员工信息</caption>

<tr>

<td>编号</td>

<td>姓名</td>

<td>薪水</td>

<td>年龄</td>

</tr>

<%

EmployeeDao dao=new EmployeeDao();

List<Employee> emps=dao.findAll();

for(int i=0;i<emps.size();i++){

Employee emp=emps.get(i);

%>

<tr class="row<%=i%2+1%>">

<td><%=emp.getId()%></td>

<td><%=emp.getName()%></td>

<td><%=emp.getSalary()%></td>

<td><%=emp.getAge()%></td>

<td><a href="delete.do?id=<%=emp.getId()%>" onclick="return confirm('是否确认删除<%=emp.getName()%>信息？')">删除</a>

&nbsp;

<a href="load.do?id=<%=emp.getId()%>">修改</a>

</td>

</tr>

<%} %>

</table>

<p>

<input type="button" value="增加员工" onclick="location='addEmp.jsp'">

</p>

</body>

</html>

addEmp.jsp文件代码如下：

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

<form action="add.do">

姓名：<input type="text" name="name"><br>

薪水：<input type="text" name="salary"><br>

年龄：<input type="text" name="age"><br>

<input type="submit" value="增加">

</form>

</body>

</html>

updateEmp.jsp文件代码如下：<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8" import="entity.\*"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

<%

Employee emp=(Employee)request.getAttribute("emp");

%>

<form action="update.do?id=<%=emp.getId()%>" method="post">

<table border="1px">

<tr>

<td>编号</td>

<td><input type="text" name="id" value="<%=emp.getId()%>"></td>

</tr>

<tr>

<td>姓名</td>

<td><input type="text" name="name" value="<%=emp.getName()%>"></td>

</tr>

<tr>

<td>薪水</td>

<td><input type="text" name="salary" value="<%=emp.getSalary()%>"></td>

</tr>

<tr>

<td>年龄</td>

<td><input type="text" name="age" value="<%=emp.getAge()%>"></td>

</tr>

</table>

<p>

<input type="submit" value="修改">

</p>

</form>

</body>

</html>

web.xml代码如下：

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd" id="WebApp\_ID" version="3.1">

<servlet>

<servlet-name>actionServlet</servlet-name>

<servlet-class>web.ActionServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>actionServlet</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

</web-app>