综合项目

Comprehensive project

Author：Gavin

Version：9.0.2

一、引言

1.1 场景

> - 在项目中，文件的上传和下载是常见的功能。很多程序或者软件中都经常使用到文件的上传和下载。
> - 邮箱中有附件的上传和下载
> - OA办公系统中有附件材料的上传

二、 文件上传

2.1 概念

> 当用户在前端页面点击文件上传后，用户上传的文件数据提交给服务器端，实现保存。

2.2 文件上传实现步骤

2.2.1 提交方式

> - 提供form表单,method必须是post。因为post请求无数据限制。

```
<form method="post"> </form>
```

2.2.2 提交数据格式

> - 表单的enctype属性值必须为multipart/form-data。
> - 以多段的形式进行拼接提交。以二进制流的方式来处理表单数据，会把指定的文件内容封装进请求参数中。

```
<form enctype="multipart/form-data" method="post"></form>
```

2.2.3 提供组件

> - 提供file表单组件，提供给用户上传文件。

```
<form enctype="multipart/form-data" method="post">
        上传用户：<input type="text" name="username"><br/>
        上传文件1：<input type="file" name="file1"><br/>
        <input type="submit" value="提交">
    </form>
```

2.2.4 Controller编写

> 在 Servlet 3.0 及其以上版本的容器中进行服务器端文件上传的编程，是围绕着注解类型 MultipartConfig 和 javax. servlet.http.Part 接口进行的。处理已上传文件的 Servlet 必须以@MultipartConfig 进行注解。

```
package com.qf.servlet;

import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;
import java.io.File;
import java.io.IOException;
import java.util.Collection;

@WebServlet(name = "UploadController",value = "/uploadController")
@MultipartConfig(maxFileSize = 1024*1024*100,maxRequestSize = 1024*1024*200)
public class UploadController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doGet(request,response);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        //设置乱码
        request.setCharacterEncoding("UTF-8");
        response.setContentType("text/html;charset=utf-8");
        //通过getPart方法获取文件组件
        Part file1 = request.getPart("file1");
        //获取上传文件保存路径（真实路径）
        String uploadPath = request.getServletContext().getRealPath("/WEB-INF/upload");
        System.out.println(uploadPath);
        //创建文件夹对象
        File file = new File(uploadPath);
        //如果文件夹不存在，新建
        if(!file.exists()){
            file.mkdirs();
        }
        //如果上传的文件不是空，进行保存
        if(file1!=null){
            file1.write(uploadPath+File.separator+file1.getSubmittedFileName());
        }
        response.getWriter().println("上传成功！ "+file1.getSubmittedFileName());
    }
}
```

2.3 文件上传细节注意

> 上述的代码虽然可以成功将文件上传到服务器的指定目录当中，但是文件上传功能有许多需要注意的小细节问题。

2.3.1 安全问题

> 为保证服务器安全，上传文件应该放在外界无法直接访问的目录下，比如放于WEB-INF目录下。

```
String filepath = request.getServletContext().getRealPath("/WEB-INF/upload");
```

2.3.2 文件覆盖

> 当上传重名的文件时，为防止文件覆盖的现象发生，要为上传文件产生一个唯一的文件名。

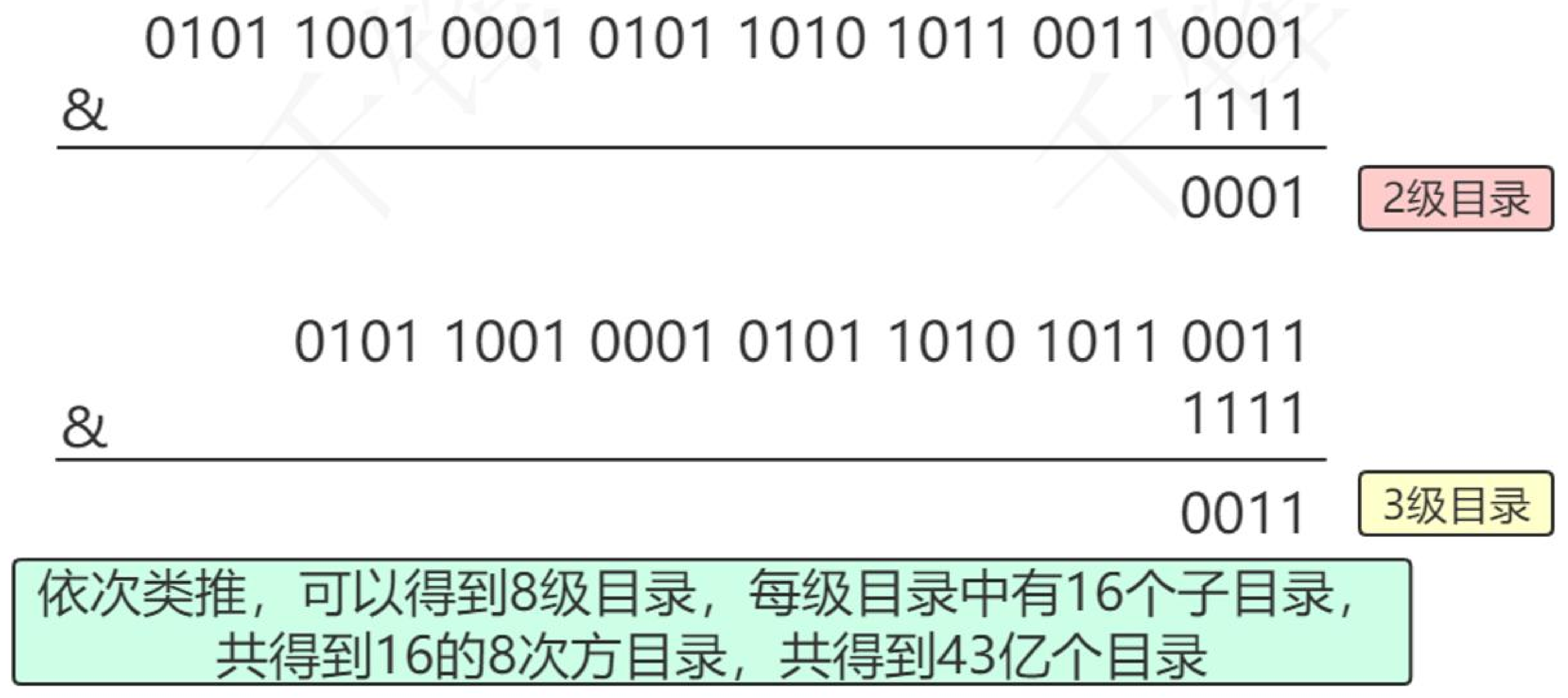
```
public class UploadUtils{
    //使用UUID生成唯一标识码，拼接上图片的名称。
  public static String NewFileName(String filename){
        return  UUID.randomUUID().toString().replaceAll("-","")+"_"+filename;
    }
}
```

2.3.3 散列存储

> 为防止一个目录下面出现太多文件，要使用hash算法生成二级、三级目录，散列存储上传的文件。

```
public static String NewFilePath(String basePath,String filename){
        int hashcode = filename.hashCode();
        int path1 = hashcode&15;//与运算  0~15 二级
        int path2 = (hashcode>>4)&15;//与运算 0~15  三级
        String dir = basePath+"\\"+path1+"\\"+path2;//与一级目录拼接一起
        File file = new File(dir);//创建文件夹对象
        if(!file.exists()){//不存在则新建
            file.mkdirs();
        }
        return dir;//返回新路径
    }
```

目录分离算法



依次类推，可以得到8级目录，每级目录中有16个子目录，共得到16的8次方目录，共得到43亿个目录

2.3.4 文件类型限制

> 要限制上传文件的类型，在收到上传文件名时，判断后缀名是否合法。

```
//创建一个集合存放允许上传的文件的类型（后缀名）
//判断所上传的文件在当前集合当中是否包含。
List<String> nameList = new ArrayList<String>();
       nameList.add(".jpg");
       nameList.add(".bmp");
       nameList.add(".png");
       String extName = filename.substring(filename.lastIndexOf("."));
       if(!nameList.contains(extName)){
         System.out.println("上传失败");
       return;
}
```

2.4 多文件上传

```
package com.qf.servlet;

import java.io.IOException;
import java.util.Collection;

import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;

import org.apache.commons.io.IOUtils;

import com.qf.utils.FileUploadUtils;

@WebServlet(value = "/upLoadfile2")
@MultipartConfig(fileSizeThreshold=1024*100,maxFileSize=1024*1024*2,maxRequestSize=1024*1024*20)
public class UploadFileServlet2 extends HttpServlet {
  private static final long serialVersionUID = 1L;

    public UploadFileServlet2() {
        super();
    }

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

    request.setCharacterEncoding("utf-8");
    response.setContentType("text/html;charset=utf-8");
        //得到上传文件的保存目录，将上传的文件存放于WEB-INF目录下，不允许外界直接访问，保证上传文件的安全
    String basepath=request.getServletContext().getRealPath("/WEB-INF/upload");
        File dir=new File(savepath);
        if(!dir.exists()){
            dir.mkdirs();
        }
        //获取多段数据的集合
    Collection<Part> parts = request.getParts();

        if(parts!=null) {
      for (Part part : parts) {
                //获取文件提交的名字
        String filename=part.getSubmittedFileName();
        if(filename!=null) {//文件
          if(filename.trim().equals("")) {
            continue;
          }
                    //获得包含UUID的文件名
          String newFilename=FileUploadUtils.getNewFileName(filename);
                    //获取分散后的路径
          String newpath=FileUploadUtils.getNewPath(basepath, filename);
                    //存储
          part.write(newpath+"\\"+newFilename);

          response.getWriter().write(filename+"上传成功");

        }else {//不是文件
          String name=part.getName();
          String value=request.getParameter(name);
          System.out.println(name+"...."+value);
        }
      }
    }

  }

  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
  }
}
```

三、文件下载

3.1 概念

> 我们要将Web应用系统中的文件资源提供给用户进行下载，首先我们要有一个页面列出上传文件目录下的所有文件，当用户点击文件下载超链接时就进行下载操作。

3.2 获取文件列表

3.2.1 DownLoadUtils

```
    public static void getFileList(File file, HashMap<String,String> filenames){
        //获取当前文件对象下的所有内容（文件、文件夹）
        File[] files = file.listFiles();
        //如果数组不为空，证明有文件、文件夹
        if(files!=null){
            //每次拿到文件对象（文件、文件夹   ）
            for (File file1 : files) {
                if(file1.isDirectory()){
                    getFileList(file1,filenames);
                }else{
                    //获得文件的名称
                    String filename = file1.getName();
                    //获取第一个_的下标
                    int i = filename.indexOf("_");
                    //获取源文件名称（可能包含_）
                    String realName= filename.substring(i + 1);
                    //UUID键   源文件名 值
                    filenames.put(filename, realName);
                }
            }
        }
    }
```

3.2.2 FileListController

```
package com.qf.web.servlet;

import com.qf.utils.UpLoadUtils;

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 java.io.File;
import java.io.IOException;
import java.util.HashMap;


@WebServlet(name = "FileListServlet",value = "/fileListController")
public class FileListController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doGet(request, response);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        //1乱码
        request.setCharacterEncoding("utf-8");//用户名的乱码
        response.setContentType("text/html;charset=utf-8");
        //2获取文件列表
        HashMap<String, String> filemap=new HashMap<>();
        String savepath=request.getServletContext().getRealPath("/WEB-INF/upload");
        UpLoadUtils.fileList(new File(savepath), filemap);
        //3转发
        request.setAttribute("map", filemap);
        request.getRequestDispatcher("/list.jsp").forward(request, response);
    }
}
```

3.3 下载

3.3.1 fileList.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
    <title>下载文件列表</title>
</head>
<body>
    <h2>下载文件列表</h2>
    <table>
        <tr>
            <th>文件名</th>
            <th>操作</th>
        </tr>

        <c:forEach items="${map}" var="entry">
            <tr>
                <td>
                    ${entry.value}
                </td>
                <td>
                    <a href="${pageContext.request.contextPath}/down?filename=${entry.key}">下载</a>
                </td>
            </tr>

        </c:forEach>
    </table>
</body>
</html>
```

3.3.2 DownLoadController

```
package com.qf.web.servlet;

import com.qf.utils.UpLoadUtils;

import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.FileInputStream;
import java.io.IOException;
import java.net.URLEncoder;


@WebServlet(name = "DownServlet",value ="/down")
public class DownServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doGet(request, response);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        request.setCharacterEncoding("utf-8");//用户名的乱码
        response.setContentType("text/html;charset=utf-8");


        String savepath=request.getServletContext().getRealPath("/WEB-INF/upload");
        //1获取文件名
        String uuidfilename = request.getParameter("filename");
        //拿到UUID的名字拆分。_之后的是原文件名
        String filename=uuidfilename.split("_")[1];
        //通过原文件名得到分散后的路径就是要下载的路径
        String realpath=UpLoadUtils.createNewPath(savepath, filename);
        //设置响应头,告诉浏览器如何处理流，附件下载
        response.setHeader("content-disposition", "attachment;filename="+ URLEncoder.encode(filename,"utf-8" ));

        //2使用流读取
        FileInputStream fis=new FileInputStream(realpath+"/"+uuidfilename);
        ServletOutputStream os = response.getOutputStream();
        byte[] buf=new byte[1024*4];
        int len=0;
        while((len=fis.read(buf))!=-1){
            os.write(buf,0,len);
        }
        //3关闭
        os.close();
        fis.close();

    }
}
```

四、EMS综合项目

4.1 项目需求

> - 将提供好的HTML页面，修改为JSP。
> - 实现管理员登录功能，并应用权限验证。
> - 对员工实现增删改查。
> - 查询员工使用分页查询，并提供首页、尾页、上一页、下一页。

4.2 项目搭建

4.2.1 项目目录搭建

> - com.qf.ems.utils 工具包
> - com.qf.ems.entity 实体类
> - com.qf.ems.dao 数据访问接口
> - com.qf.ems.dao.impl 数据访问实现类
> - com.qf.ems.service 业务逻辑接口
> - com.qf.ems.service.impl 业务逻辑实现类
> - com.qf.ems.controller 控制器
> - com.qf.ems.filter 过滤器

4.2.2 项目资源引入

> 创建EMS项目，导入相关jar文件。
>
> - commons-dbutils-1.7.jar
> - druid-1.1.5.jar
> - jstl.jar
> - standard.jar
> - ValidateCode.jar
> - mysql-connector-java-5.1.25-bin.jar

4.2.3 数据库创建

```sql
#员工信息表
CREATE TABLE emp(
  id INT PRIMARY KEY AUTO_INCREMENT,
  NAME VARCHAR(20) NOT NULL,
  salary DOUBLE NOT NULL,
  age INT NOT NULL
)CHARSET=utf8;


#管理员表
CREATE TABLE empManager(
  username VARCHAR(20) NOT NULL,
  PASSWORD VARCHAR(20) NOT NULL
)CHARSET=utf8;
```

4.3 项目开发

> 以下仅展示controller代码。页面、entity、DAO、Service省略

4.3.1 登录功能

```java
package com.qf.ems.controller;

import com.qf.ems.entity.EmpManager;
import com.qf.ems.service.EmpManagerService;
import com.qf.ems.service.impl.EmpManagerServiceImpl;

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 java.io.IOException;

@WebServlet(name = "EmpManagerLoginController", value = "/manager/empManagerLogin")
public class EmpManagerLoginController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String inputVcode = request.getParameter("inputVcode");

        String codes = (String) request.getSession().getAttribute("codes");
        if(!inputVcode.isEmpty() && inputVcode.equalsIgnoreCase(codes)){

            EmpManagerService empManagerService = new EmpManagerServiceImpl();
            EmpManager empManager = empManagerService.login(username, password);
            if (empManager != null) {
                HttpSession session = request.getSession();
                session.setAttribute("empManager", empManager);

                response.sendRedirect(request.getContextPath() + "/manager/safe/showAllEmp");
            } else {
                response.sendRedirect(request.getContextPath() + "/login.jsp");
            }

        }else{
            response.sendRedirect(request.getContextPath() + "/login.jsp");
        }



    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doPost(request, response);
    }
}
```

4.3.2 验证码

```
package com.qf.ems.controller;

import cn.dsna.util.images.ValidateCode;

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 java.io.IOException;

@WebServlet(name = "CreateCodeController",value = "/createCode")
public class CreateCodeController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        ValidateCode validateCode = new ValidateCode(200,30,4,10);
        String codes = validateCode.getCode();
        HttpSession session = request.getSession();
        session.setAttribute("codes",codes);

        validateCode.write(response.getOutputStream());
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doPost(request, response);
    }
}
```

4.3.3 分页查询(Controller)

```
package com.qf.ems.controller;

import com.qf.ems.entity.Emp;
import com.qf.ems.entity.Page;
import com.qf.ems.service.EmpService;
import com.qf.ems.service.impl.EmpServiceImpl;

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 java.io.IOException;
import java.util.List;

@WebServlet(name = "ShowAllEmpController",value = "/manager/safe/showAllEmp")
public class ShowAllEmpController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        String pageIndex = request.getParameter("pageIndex");
        if(pageIndex == null){
            pageIndex="1";
        }

        Page page = new Page(Integer.valueOf(pageIndex));

        EmpService empService = new EmpServiceImpl();
        List<Emp> emps = empService.showAllEmpByPage(page);

        request.setAttribute("emps",emps);
        request.setAttribute("page",page);

        request.getRequestDispatcher("/emplist.jsp").forward(request,response);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doPost(request, response);
    }
}
```

4.3.4 分页查询(JSP)

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
    <title>查询所有员工</title>
    <link rel="stylesheet" type="text/css" href="${pageContext.request.contextPath}/css/style.css"/>
</head>
<body>
<div id="wrap">
    <div id="top_content">
        <div id="header">
            <div id="rightheader">
                <p>
                    2009/11/20
                    <br/>
                </p>
            </div>
            <div id="topheader">
                <h1 id="title">
                    <a href="#">main</a>
                </h1>
            </div>
            <div id="navigation">
            </div>
        </div>
        <div id="content">
            <p id="whereami">
            </p>
            <h1>
                Welcome!
            </h1>
            <table class="table">
                <tr class="table_header">
                    <td>
                        ID
                    </td>
                    <td>
                        Name
                    </td>
                    <td>
                        Salary
                    </td>
                    <td>
                        Age
                    </td>
                    <td>
                        Operation
                    </td>
                </tr>
                <c:forEach items="${emps}" var="emp" varStatus="e">
                    <c:if test="${e.count % 2  !=0}">
                        <tr class="row1">
                    </c:if>
                    <c:if test="${e.count % 2  ==0}">
                        <tr class="row2">
                    </c:if>
                    <td>
                            ${emp.id}
                    </td>
                    <td>
```

```
                            ${emp.name}
                    </td>
                    <td>
                            ${emp.salary}
                    </td>
                    <td>
                            ${emp.age}
                    </td>
                    <td>
                        <a href="<c:url  context='${pageContext.request.contextPath}' value='/manager/safe/deleteEmp?id=${emp.id}'/>">delete emp</a>&nbsp;<a href="<c:url context='${pageContext.request.contextPath}' value='/manager/safe/showEmp?id=${emp.id}'/>">update emp</a>
                    </td>
                    </tr>
                </c:forEach>
                <tr>
                    <td colspan="5" align="center">
                        <a href="<c:url context='${pageContext.request.contextPath}' value='/manager/safe/showAllEmp?pageIndex=1'/>">首页</a>

                        <c:if test="${page.pageIndex > 1}">
                            <a href="<c:url context='${pageContext.request.contextPath}' value='/manager/safe/showAllEmp?pageIndex=${page.pageIndex - 1}' />">上一页</a>
                        </c:if>
                        <c:if test="${page.pageIndex==1}">
                            <a>上一页</a>
                        </c:if>

                        <c:if test="${page.pageIndex < page.totalPages}">
                            <a href="<c:url context='${pageContext.request.contextPath}' value='/manager/safe/showAllEmp?pageIndex=${page.pageIndex + 1}' />">下一页</a>
                        </c:if>
                        <c:if test="${page.pageIndex == page.totalPages}">
                            <a>下一页</a>
                        </c:if>

                        <a href="<c:url context='${pageContext.request.contextPath}' value='/manager/safe/showAllEmp?pageIndex=${page.totalPages}'/>">尾页</a>
                    </td>
                </tr>
            </table>
            <p>
                <input type="button" class="button" value="Add Employee" onclick="location='${pageContext.request.contextPath}/addEmp.jsp'"/>
            </p>
        </div>
    </div>
    <div id="footer">
        <div id="footer_bg">
            ABC@126.com
        </div>
    </div>
</div>
</body>
</html>
```

4.3.5 新增

```
package com.qf.ems.controller;

import com.qf.ems.entity.Emp;
import com.qf.ems.service.EmpService;
import com.qf.ems.service.impl.EmpServiceImpl;

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 java.io.IOException;

@WebServlet(name = "InsertEmpController",value = "/manager/safe/insertEmp")
public class InsertEmpController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String name = request.getParameter("name");
        Double salary = Double.valueOf(request.getParameter("salary"));
        Integer age = Integer.valueOf(request.getParameter("age"));

        Emp emp = new Emp(name,salary,age);

        EmpService empService = new EmpServiceImpl();
        empService.addEmp(emp);

        response.sendRedirect(request.getContextPath()+"/manager/safe/showAllEmp");
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doPost(request, response);
    }
}
```

4.3.6 修改

```
package com.qf.ems.controller;

import com.qf.ems.entity.Emp;
import com.qf.ems.service.EmpService;
import com.qf.ems.service.impl.EmpServiceImpl;

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 java.io.IOException;

@WebServlet(name = "UpdateEmpController",value = "/manager/safe/updateEmp")
public class UpdateEmpController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        Integer id = Integer.valueOf(request.getParameter("id"));
        String name = request.getParameter("name");
        Double salary = Double.valueOf(request.getParameter("salary"));
        Integer age = Integer.valueOf(request.getParameter("age"));

        Emp emp = new Emp(id,name,salary,age);

        EmpService empService = new EmpServiceImpl();
        empService.modifyEmp(emp);

        response.sendRedirect(request.getContextPath()+"/manager/safe/showAllEmp");
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doPost(request, response);
    }
}
```

4.3.7 删除

```
package com.qf.ems.controller;

import com.qf.ems.service.EmpService;
import com.qf.ems.service.impl.EmpServiceImpl;

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 java.io.IOException;

@WebServlet(name = "DeleteEmpController",value = "/manager/safe/deleteEmp")
public class DeleteEmpController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        Integer id = Integer.valueOf(request.getParameter("id"));

        EmpService empService = new EmpServiceImpl();
        empService.deleteEmp(id);

        response.sendRedirect(request.getContextPath()+"/manager/safe/showAllEmp");
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        doPost(request, response);
    }
}
```

五、Web开发总结

5.1 开发流程

> 在Web开发流程中，遵守以下开发顺序
>
> - DAO
>   - table
>   - entity
>   - DAO接口
>   - DAO实现
> - Service
>   - Service接口
>   - Service实现(调用DAO实现类，并控制事务)
> - Controller(处理请求的Servlet)
>   - [收集请求中的数据]
>   - 调用业务功能(Service实现类)
>   - [在相应合适的作用域中存储数据]
>   - 流程跳转(forward | sendRedirect) --> *.jsp
> - JSP
>   - [在作用域中获取数据]
>   - 使用EL+JSTL将数据嵌套在HTML标签中
> - Filter
>   - EncodingFilter
>   - CheckFilter