



链滴

# SpringMVC 实现上传 Excel 文件并读取至数据库

作者: [QuinnTian](#)

原文链接: <https://ld246.com/article/1532060371742>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 1.添加依赖

```
<!--文件上传-->
<!-- https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.3</version>
</dependency>
<!-- https://mvnrepository.com/artifact/commons-io/commons-io -->
<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.6</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>3.17</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.xmlbeans/xmlbeans -->
<dependency>
    <groupId>org.apache.xmlbeans</groupId>
    <artifactId>xmlbeans</artifactId>
    <version>2.6.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml-schemas -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml-schemas</artifactId>
    <version>3.17</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.17</version>
</dependency>
```

## 2.添加WDWUtil.java工具类

用于判断Excel版本

```
package tk.mybatis.web.util.fileutil;
```

```
public class WDWUtil {
    // @描述：是否是2003的excel，返回true是2003
    public static boolean isExcel2003(String filePath) {
        return filePath.matches("^.+\\.?(?i)(xls)$");
    }
}
```

```

//@描述: 是否是2007的excel, 返回true是2007
public static boolean isExcel2007(String filePath) {
    return filePath.matches("^.+\\.?(?i)(xlsx)$");
}
}

```

### 3.创建Read.java文件

用于读取Excel,实体类根据实际情况自己替换

```

package tk.mybatis.web.util.fileutil;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.multipart.MultipartFile;
import org.springframework.web.multipart.commons.CommonsMultipartFile;
import tk.mybatis.web.model.Message;

public class ReadExcel {
    //总行数
    private int totalRows = 0;
    //总条数
    private int totalCells = 0;
    //错误信息接收器
    private String errorMsg;
    //构造方法
    public ReadExcel(){}
    //获取总行数
    public int getTotalRows() { return totalRows;}
    //获取总列数
    public int getTotalCells() { return totalCells;}
    //获取错误信息
    public String getErrorInfo() { return errorMsg; }

    /**
     * 验证EXCEL文件

```

```

* @param filePath
* @return
*/
public boolean validateExcel(String filePath){
    if (filePath == null || !(WDWUtil.isExcel2003(filePath) || WDWUtil.isExcel2007(filePath))){
        errorMsg = "文件名不是excel格式";
        return false;
    }
    return true;
}

/**
 * 读EXCEL文件，获取客户信息集合
 * @param
 * @return
 */
public List<Message> getExcelInfo(String fileName, MultipartFile Mfile){

    //把spring文件上传的MultipartFile转换成CommonsMultipartFile类型
    CommonsMultipartFile cf= (CommonsMultipartFile)Mfile; //获取本地存储路径
    File file = new File("D:\\fileupload");
    //创建一个目录 （它的路径名由当前 File 对象指定，包括任一必须的父路径。）
    if (!file.exists()) file.mkdirs();
    //新建一个文件
    File file1 = new File("D:\\fileupload\\" + new Date().getTime() + ".xlsx");
    //将上传的文件写入新建的文件中
    try {
        cf.getFileItem().write(file1);
    } catch (Exception e) {
        e.printStackTrace();
    }

    //初始化客户信息的集合
    List<Message> customerList=new ArrayList<Message>();
    //初始化输入流
    InputStream is = null;
    try{
        //验证文件名是否合格
        if(!validateExcel(fileName)){
            return null;
        }
        //根据文件名判断文件是2003版本还是2007版本
        boolean isExcel2003 = true;
        if(WDWUtil.isExcel2007(fileName)){
            isExcel2003 = false;
        }
        //根据新建的文件实例化输入流
        is = new FileInputStream(file1);
        //根据excel里面的内容读取客户信息
        customerList = getExcelInfo(is, isExcel2003);
        is.close();
    }catch(Exception e){
        e.printStackTrace();
    } finally{

```

```

        if(is !=null)
        {
            try{
                is.close();
            }catch(IOException e){
                is = null;
                e.printStackTrace();
            }
        }
    }
    return customerList;
}
/**
 * 根据excel里面的内容读取客户信息
 * @param is 输入流
 * @param isExcel2003 excel是2003还是2007版本
 * @return
 * @throws IOException
 */
public List<Message> getExcelInfo(InputStream is,boolean isExcel2003){
    List<Message> customerList=null;
    try{
        /** 根据版本选择创建Workbook的方式 */
        Workbook wb = null;
        //当excel是2003时
        if(isExcel2003){
            wb = new HSSFWorkbook(is);
        }
        else{//当excel是2007时
            wb = new XSSFWorkbook(is);
        }
        //读取Excel里面客户的信息
        customerList=readExcelValue(wb);
    }
    catch (IOException e) {
        e.printStackTrace();
    }
    return customerList;
}
/**
 * 读取Excel里面客户的信息
 * @param wb
 * @return
 */
private List<Message> readExcelValue(Workbook wb){
    //得到第一个sheet
    Sheet sheet=wb.getSheetAt(0);

    //得到Excel的行数
    this.totalRows=sheet.getPhysicalNumberOfRows();

    //得到Excel的列数(前提是有行数)
    if(totalRows>=1 && sheet.getRow(0) != null){
        this.totalCells=sheet.getRow(0).getPhysicalNumberOfCells();
    }
}

```

```

    }

    List<Message> customerList=new ArrayList<Message>();
    Message customer;
    //循环Excel行数,从第二行开始。标题不入库
    for(int r=1;r<totalRows;r++){
        Row row = sheet.getRow(r);
        if (row == null) continue;
        customer = new Message();

        //循环Excel的列
        for(int c = 0; c <=this.totalCells; c++){
            Cell cell = row.getCell(c);
            if (null != cell){
                if(c==0){//第一列不读
                }else if(c==1){
                    System.out.println(cell.getStringCellValue());
                    customer.setMsgType(cell.getStringCellValue());//客户名称
                }else if (c==2){
                    System.out.println(cell.getStringCellValue());
                    customer.setContent(cell.getStringCellValue());//客户名称
                }
            }
        }
        //添加客户
        customerList.add(customer);
    }
    return customerList;
}
}
}

```

## 4.创建服务层

数据库层根据实际情况编写

```

package tk.mybatis.web.service.impl;

import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;
import tk.mybatis.web.model.Message;
import tk.mybatis.web.util.fileutil.ReadExcel;

import java.util.List;

@Service
public class UploadService {
    //批量导入客户
    public boolean batchImport(String name,MultipartFile file){
        boolean b = false;
        //创建处理EXCEL
        ReadExcel readExcel=new ReadExcel();
        //解析excel，获取客户信息集合。
    }
}

```

```

        List<Message> customerList = readExcel.getExcelInfo(name ,file);
        System.out.println(customerList);
        if(customerList != null){
            b = true;
        }

        //迭代添加客户信息（注：实际上这里也可以直接将customerList集合作为参数，在Mybatis的
        应映射文件中使用foreach标签进行批量添加。）
        for(Message customer:customerList){
            System.out.println(customer.toString());
            //Dao层根据实际情况编写
        }
        return b;
    }
}

```

## 5.创建控制层

```

package tk.mybatis.web.controller;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.multipart.MultipartFile;
import tk.mybatis.web.service.impl.UploadService;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@Controller
@RequestMapping("addController/")
public class FileUploadController {

    private static Log log = LogFactory.getLog(FileUploadController.class);
    @Autowired
    private UploadService customerService;

    @RequestMapping(value = "batchimport", method = RequestMethod.POST)
    public String batchimport(@RequestParam(value="filename") MultipartFile file,
                              HttpServletRequest request, HttpServletResponse response) throws IOExce
tion {
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");
        log.info("AddController ..batchimport() start");
        //判断文件是否为空
        if(file==null) return null;
        //获取文件名
        String name=file.getOriginalFilename();

```

```

//进一步判断文件是否为空（即判断其大小是否为0或其名称是否为null）
long size=file.getSize();
if(name==null || "".equals(name) && size==0) return null;

//批量导入。参数：文件名，文件。
boolean b = customerService.batchImport(name,file);
if(b){
    String Msg ="import success! ";
    request.getSession().setAttribute("msg",Msg);
}else{
    String Msg ="import failed! ";
    request.getSession().setAttribute("msg",Msg);
}
return "addfile";
}
}

```

## 6.前端页面

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

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

<%
    String importMsg="";
    if(request.getSession().getAttribute("msg")!=null){
        importMsg=request.getSession().getAttribute("msg").toString();
    }
    request.getSession().setAttribute("msg", "");
%>
<head>
    <title>批量导入</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<body>
<div><font color="bule">批量导入</font></div>
<form action="addController/batchimport" method="post" enctype="multipart/form-data"
nsubmit="return check();">
    <div style="margin: 30px;"><input id="excel_file" type="file" name="filename" accept="xl
x" size="80"/>
        <input id="excel_button" type="submit" value="导入Excel"/></div>
        <font id="importMsg" color="red"><%=importMsg%></font><input type="hidden"/>
</form>
</body>
<script type="text/javascript">
    function check() {
        var excel_file = $("#excel_file").val();
        if (excel_file == "" || excel_file.length == 0) {
            alert("请选择文件路径!");
            return false;
        }
    }

```



```
        } else {
            return true;
        }
    }

    $(document).ready(function () {
        var msg="";
        if($("#importMsg").text()!=null){
            msg=$("#importMsg").text();
        }
        if(msg!=""){
            alert(msg);
        }
    });
</script>

</html>
```