

JAVA操作Office

江西软件职业技术大学

区块链分院

江海



家 救命 组件API 参与进来

Search the site with google 搜索

上次发布时间: 2019/09/25 07:26:00

总览

家
下载
变更日志
Java文档
文字提取
加密支持
实例探究
相关项目
法律
阿帕奇宽

Apache POI-下载发布工件

可用下载

本页提供有关如何下载和验证Apache POI发行工件的说明。根据代码的稳定性，可以使用不同的版本。

- 最新的稳定版本是[Apache POI 4.1.0](#)
- [每晚/CI版本也可用](#)
- [所有先前版本的档案](#)

Apache POI发行版可通过[Apache License 2.0版获得](#)。有关适用的版权归属声明，请参阅每个发行工件中包含的NOTICE文件。

为了确保您下载了真实版本，您应该使用此页面上可用的签名和校验和来[验证文件的完整性](#)。

2019年4月9日-提供POI 4.1.0

Apache POI小组很高兴宣布4.1.0版本。精选了一些新功能，并修复了许多错误。

[发行说明中](#)提供了更改摘要。[更改日志中](#)提供了完整的更改列表。有兴趣的人还应该遵循[开发人员列表](#)来跟踪进度。

下面列出了POI源代码版本以及预构建的二进制部署程序包。所有[POI组件](#)的预构建版本都可以在中央Maven存储库中以组ID“org.apache.poi”和“4.1.0”获得。

二元分布



09 April 2019 - POI 4.1.0 available

The Apache POI team is pleased to announce the release of 4.1.0. Featured are a handful of new areas of functionality and numerous bug fixes. A summary of changes is available in the [Release Notes](#). A full list of changes is available in the [change log](#). People interested should also [dev list](#) to track progress.

The POI source release as well as the pre-built binary deployment packages are listed below. Pre-built versions of all [POI components](#) are in the central Maven repository under Group ID "org.apache.poi" and Version "4.1.0".

Binary Distribution

- [poi-bin-4.1.0-20190412.tar.gz](#) (27.32 MB, [signature \(.asc\)](#), checksum: [SHA-256](#), [SHA-512](#))
- [poi-bin-4.1.0-20190412.zip](#) (37.23 MB, [signature \(.asc\)](#), checksum: [SHA-256](#), [SHA-512](#))

Source Distribution

- [poi-src-4.1.0-20190412.tar.gz](#) (96.70 MB, [signature \(.asc\)](#), checksum: [SHA-256](#), [SHA-512](#))
- [poi-src-4.1.0-20190412.zip](#) (100.96 MB, [signature \(.asc\)](#), checksum: [SHA-256](#), [SHA-512](#))

下载

Nightly Builds

The POI nightly builds are run on the [Jenkins](#) continuous integration server.

These builds should not be used in production: they are mostly intended for use by developers to help with resolving bugs and evaluating features or users who want to try out the latest version.

- [Last Successful Jenkins build for POI-trunk](#)



CELEBRATING 20 YEARS OF COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

我们建议你从这里下，看看url，清华的镜像

[Community ▾](#)[License ▾](#)[Sponsors ▾](#)

We suggest the following mirror site for your download:

<http://mirrors.tuna.tsinghua.edu.cn/apache/poi/release/bin/poi-bin-4.1.0-20190412.zip>

Other mirror sites are suggested below.

It is essential that you [verify the integrity](#) of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

HTTP

<http://mirrors.tuna.tsinghua.edu.cn/apache/poi/release/bin/poi-bin-4.1.0-20190412.zip>

BACKUP SITES

备份下载地址

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

<https://www-eu.apache.org/dist/poi/release/bin/poi-bin-4.1.0-20190412.zip>

<https://www-us.apache.org/dist/poi/release/bin/poi-bin-4.1.0-20190412.zip>

<http://poi.apache.org/text-extraction.html>



heavy modifications, then the POI provided text extractors documented below might be a better fit for your needs.

Common functionality

All of the POI text extractors extend from *org.apache.poi.extractor.POITextExtractor*. This provides a common method across all extractors, *getText()*. For many cases, the text returned will be all you need. However, many extractors do provide more targeted text extraction methods, so you may wish to use these in some cases.

All POIFS / OLE 2 based text extractors also extend from *org.apache.poi.extractor.POIOLE2TextExtractor*. This additionally provides common methods to get at the [HPFS document metadata](#).

All OOXML based text extractors also extend from *org.apache.poi.POIOOXMLTextExtractor*. This additionally provides common methods to get at the OOXML metadata.

Text Extractor Factory

POI provides a common class to select the appropriate text extractor for you. You simply pass it an *InputFile* and it figures out the correct text extractor for you, and returns it.

这个必须掌握

For complete detection and text extractor auto-selection, users are strongly encouraged to investigate [Apache Tika](#).

Excel

操作Excel

For .xls files, there is *org.apache.poi.extractor.ExcelExtractor*, which will return text, optionally with formulas instead of their contents. Similarly, for .xlsx files there is *org.apache.poi.xssf.extractor.XSSFExcelExtractor*, which provides the same functionality.

For those working in constrained memory footprints, there are two more Excel text extractors available. For .xls files, it's *org.apache.poi.hssf.extractor.EventBasedExcelExtractor*, based on the streaming EventUserModel code, and will generally deliver a lower memory footprint for extraction. However, it will have problems correctly outputting more complex formulas, as it works with records as they pass, and so doesn't have access to all parts of complex and shared formulas. For .xlsx files the equivalent is *org.apache.poi.xssf.extractor.EventBasedXSSFExtractor*, which is based on the XSSF SAX Event codebase.

Word

操作Word

For .doc files from Word 97-2003, in scratchpad there is *org.apache.poi.hwpf.extractor.WordExtractor*, which will return text for your document.

Those using POI 3.7 can also extract simple textual content from older Word 6 and Word 95 files, using the scratchpad class *org.apache.poi.hwpf.extractor.Word6Extractor*.

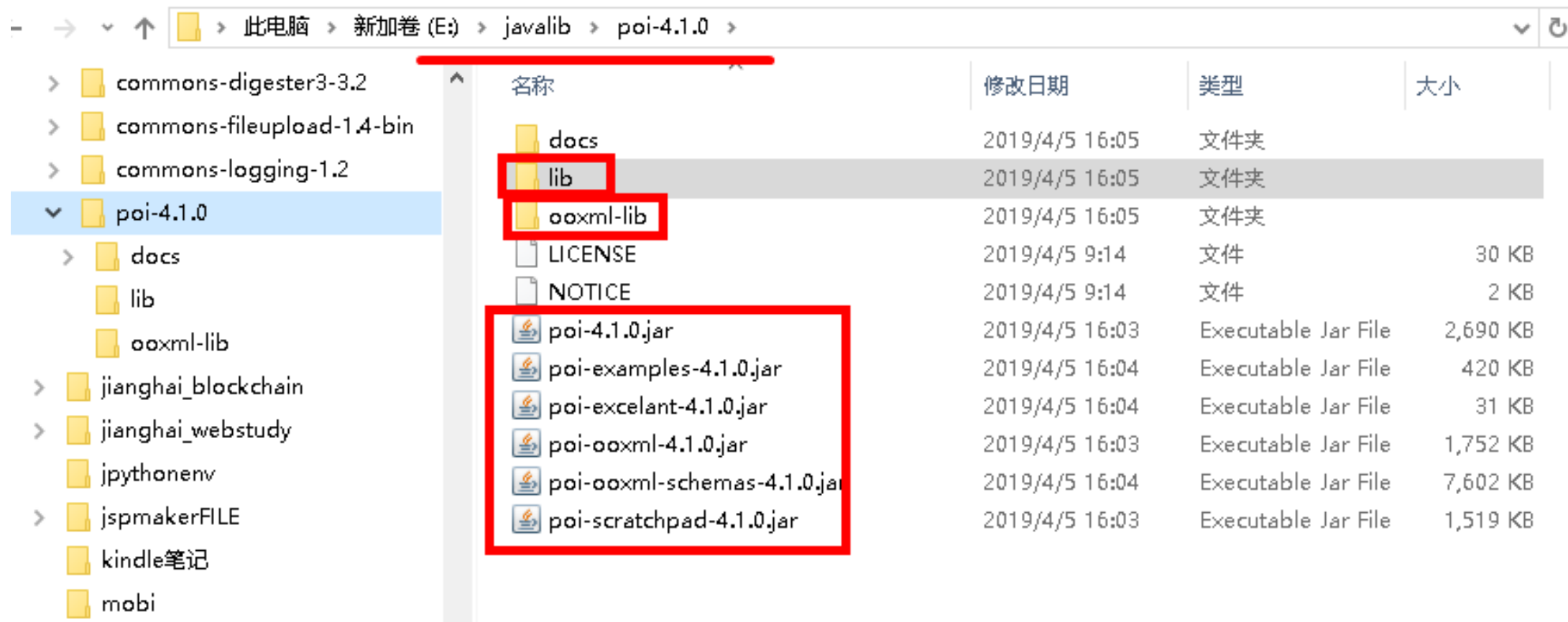
For .docx files, the relevant class is *org.apache.poi.xwpf.extractor.XPFFWordExtractor*

PowerPoint

操作Powerpoint

For .ppt and .pptx files, there is *org.apache.poi.presentation.extractor.SlideShowExtractor.SlideShowExtractor*, which will return text for your slideshow, optionally restricted to just slides text or notes text. For .ppt you need to add the poi-scratchpad.jar and for .pptx the poi-ooxml.jar and its dependencies are needed

请注意 有相当多的jar文件，随时可能会用到





全英文啊！你们怎么可能
看得懂呢？

方法1：请胡老师帮忙

方法2：放弃算了

方法3：百度！demo代码

[网页](#) [资讯](#) [视频](#) [图片](#) [知道](#) [文库](#) [贴吧](#) [采购](#) [地图](#) [更多»](#)

百度为您找到相关结果约9,220,000个

搜索工具

[java读写excel_简单易用方案选PageOffice](#)

PageOffice提供Java平台Excel专业控件,动态生成报表,导出excel到本地/服务器,提供简洁高效的纯Java类调用接口,让您摆脱VBA/Jacob/POI/JXL的编程痛苦.

[www.zhuozhengsoft.com](#) 2019-10  - 评价 广告

[excel java编辑器 为您私人定制的在线Excel - SpreadJS](#)

SpreadJS在线表格编辑器,类似在线Excel功能和外观的表格编辑程序,也是SpreadJS桌面设计器的在线版本,提供源代码,自定义,任扩展,离线/在线均可进行excel表格编辑,数据绑定.

[www.grapacity.com.cn](#) 2019-10  - 评价 广告

[Java POI 操作Excel\(读取/写入\) - PinBo - 博客园](#)

2018年2月5日 - Java POI 操作Excel(读取/写入) pom.xml依赖: <dependency>
<groupId>org.apache.poi</groupId> <artifactId>poi</artifactId> <version>3.17</version>...

[www.cnblogs.com...](#)  - 百度快照

[Java 利用poi生成excel表格 - 程序员,贵在坚持 - 博客园](#)

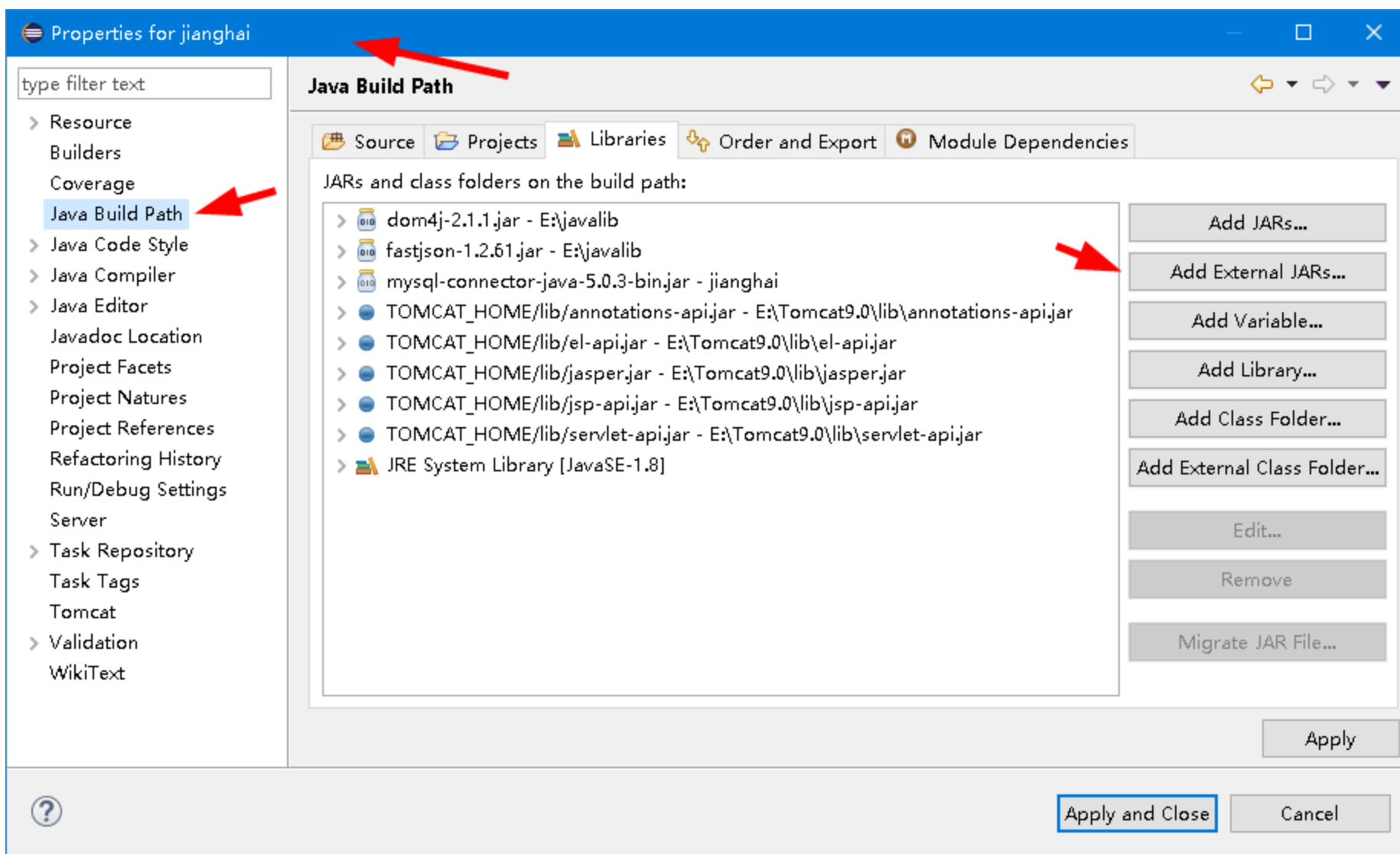
2018年05月16日 - import java.util.List; import org.apache.poi.hssf.usermodel.H...

[Java使用POI读取和写入Excel指南 - Dreamer-1 - 博客园](#)

2019年03月04日 - Java使用POI读取和写入Excel指南做项目时经常有通过程序读取Excel...

[更多同站结果>>](#)

总有一款
适合你!



type filter text

- > Resource
- Builders
- Coverage
- Java Build Path
- > Java Code Style
- > Java Compiler
- > Java Editor
- Javadoc Location
- Project Facets
- Project Natures
- Project References
- Refactoring History
- Run/Debug Settings
- Server
- > Task Repository
- Task Tags
- Tomcat
- > Validation
- WikiText

Java Build Path

Source Projects Libraries Order and Export Module Dependencies

JARs and class folders on the build path:

- > dom4j-2.1.1.jar - E:\javalib
- > fastjson-1.2.61.jar - E:\javalib
- > mysql-connector-java-5.0.3-bin.jar - jianghai
- > poi-4.1.0.jar - E:\javalib\poi-4.1.0
- > poi-excelant-4.1.0.jar - E:\javalib\poi-4.1.0
- > poi-ooxml-4.1.0.jar - E:\javalib\poi-4.1.0
- > poi-ooxml-schemas-4.1.0.jar - E:\javalib\poi-4.1.0
- > poi-scratchpad-4.1.0.jar - E:\javalib\poi-4.1.0
- > TOMCAT_HOME/lib/annotations-api.jar - E:\Tomcat9.0\lib\annotations-api.jar
- > TOMCAT_HOME/lib/el-api.jar - E:\Tomcat9.0\lib\el-api.jar
- > TOMCAT_HOME/lib/jasper.jar - E:\Tomcat9.0\lib\jasper.jar
- > TOMCAT_HOME/lib/jsp-api.jar - E:\Tomcat9.0\lib\jsp-api.jar
- > TOMCAT_HOME/lib/servlet-api.jar - E:\Tomcat9.0\lib\servlet-api.jar
- > JRE System Library [JavaSE-1.8]

Add JARs...

Add External JARs...

Add Variable...

Add Library...

Add Class Folder...

Add External Class Folder...

Apply

Apply and Close

Cancel

先加4个，看看效果

https://blog.csdn.net/weixin_38842707/article/details/80638819

上面实现的就是拿到当前第一页的对象，然后给第一页的每一行设置最小宽度，4300并不是随便写的，其实他并不是很大，具体这个需要你们去测试，匹配适合自己的值，第一个参数i可能已经被猜到了吧，我用了一个循环而已，去循环每一行！


【具体实现流程】

```
import org.apache.poi.hssf.usermodel.HSSFCellStyle;
import org.apache.poi.hssf.util.HSSFColor;
import org.apache.poi.ss.usermodel.*;
import org.apache.poi.ss.util.CellRangeAddress;
import org.apache.poi.xssf.usermodel.XSSFFont;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import java.io.*;
import java.util.ArrayList;
import java.util.List;

public class ZxExportExcel {

    /**
     * 生成Excel
     */
    public void zxExprotExcelXLSX() {
        XSSFWorkbook wb = new XSSFWorkbook();
        Sheet sheet = wb.createSheet("0");
        for (int i = 0; i < 9; i++) {
            sheet.setColumnWidth(i, 4300);
        }
    }
}
```

```
33 cellStyle.setTopBorderColor(HSSFColor.BLACK.index);
34 cellStyle.setBottomBorderColor(HSSFColor.BLACK.index);
35 cellStyle.setLeftBorderColor(HSSFColor.BLACK.index);
36 cellStyle.setRightBorderColor(HSSFColor.BLACK.index);
37 cellStyle.setAlignment(HSSFCellStyle.); // 水平居中
38 cellStyle.setVerticalAlignment(HSSFCellStyle.VERTICAL_CENTER); // 上下居中
39
40 /**
41  * 标题样式 样式
42  */
43 XSSFFont titleFont = wb.createFont();
44 titleFont.setFontHeight(24);
45 titleFont.setBold(true);
46 CellStyle titleCellStyle = wb.createCellStyle();
47 titleCellStyle.setBorderTop(BorderStyle.THIN);
48 titleCellStyle.setBorderBottom(BorderStyle.THIN);
49 titleCellStyle.setBorderLeft(BorderStyle.THIN);
50 titleCellStyle.setBorderRight(BorderStyle.THIN);
51 titleCellStyle.setTopBorderColor(HSSFColor.BLACK.index);
52 titleCellStyle.setBottomBorderColor(HSSFColor.BLACK.index);
53 titleCellStyle.setLeftBorderColor(HSSFColor.BLACK.index);
54 titleCellStyle.setRightBorderColor(HSSFColor.BLACK.index);
55 titleCellStyle.setAlignment(HSSFCellStyle.ALIGN_CENTER); // 水平居中
56 titleCellStyle.setVerticalAlignment(HSSFCellStyle.VERTICAL_CENTER); // 上下居中
57 titleCellStyle.setFont(titleFont);
58
```



居中报错，
不是关键，
把它注释掉

```
107      */
108      setExcelFooterName("测试", 0, wb);
109
110      /**
111       * 进行导出
112       */
113      exportOutPutExcel("d:\\jinaghai.xlsx", wb);
114
115  }
116
117  /**
118   * 设置Excel页脚
119   */
120  public void setExcelFoot
```

改成你找到的目录，c盘如果权限错误，就换一个目录

> chapter3. p82
v excel
 > ZxExportExcel.java
 > httpClient
 > jdbc
 > jianghai
 > jianghaiUrlDemo
 > oracle.docs
 > Text.util
Referenced Libraries
work
mysql-connector-
Servers

Properties		
el.ZxExportExcel.java - jiang		
Source	Property	Value
v Info		
	derived	false
	editable	true
	last modified	2023/10/10 14:44
	linked	false
	location	E:\work\chapter3
	name	ZxExportExcel.java
	path	/
	size	4,104 bytes

New

Open Type Hierarchy F4

Show In Alt+Shift+W

Open F3

Open With

Copy Ctrl+C

Copy Qualified Name

Paste Ctrl+V

Delete Delete

Remove from Context Ctrl+Alt+Shift+Down

Build Path

Source Alt+Shift+S

Refactor Alt+Shift+T

Import...

Export...

Refresh F5

References

Declarations

Coverage As

Run As

Debug As

Profile As

Restore from Local History...

Team

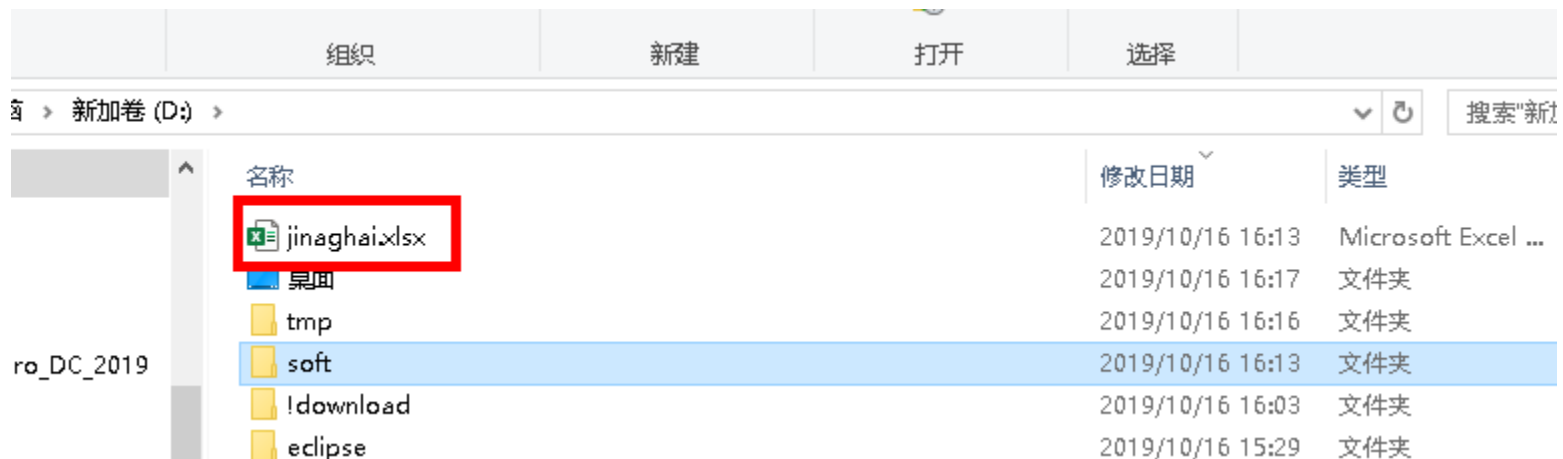
Compare With

Replace With

112

113

```
*/  
exportOutPutExcel("d:\\jinag  
  
Excel页脚  
  
void setExcelFooterName(S  
b.setSheetName(setExcelFoot  
  
流 导出Excel到桌面  
  
void exportOutPutExcel(St  
ry {  
    File file = new File(exp  
    FileOutputStream fileOut  
    wb.write(fileOutputStre  
    fileOutputStream.close()  
    sheet.setFormulaCell(s  
get  
  
1 Run on Server Alt+Shift+X, R  
2 Java Application Alt+Shift+X, J  
Run Configurations...  
  
static void main(String[]  
new ZxExportExcel().zxExprot
```



剪贴板		字体	对齐方式		数字	样式			
A1		x	✓	fx	2018年度能源科技进步奖				
	A	B	C	D	E	F	G	H	
1	2018年度能源科技进步奖								
2									
3									
4	测试标题列【1】	测试标题列【2】	测试标题列【3】	测试标题列【4】	测试标题列【5】	测试标题列【6】	测试标题列【7】	测试标题列【8】	测试标题列【9】
5	测试 - 01	测试 - 01	测试 - 01	测试 - 01	测试 - 01	测试 - 01	测试 - 01	测试 - 01	测试 - 01
6	测试 - 02	测试 - 02	测试 - 02	测试 - 02	测试 - 02	测试 - 02	测试 - 02	测试 - 02	测试 - 02
7	测试 - 03	测试 - 03	测试 - 03	测试 - 03	测试 - 03	测试 - 03	测试 - 03	测试 - 03	测试 - 03
8	测试 - 04	测试 - 04	测试 - 04	测试 - 04	测试 - 04	测试 - 04	测试 - 04	测试 - 04	测试 - 04
9	测试 - 05	测试 - 05	测试 - 05	测试 - 05	测试 - 05	测试 - 05	测试 - 05	测试 - 05	测试 - 05
10	测试 - 06	测试 - 06	测试 - 06	测试 - 06	测试 - 06	测试 - 06	测试 - 06	测试 - 06	测试 - 06
11	测试 - 07	测试 - 07	测试 - 07	测试 - 07	测试 - 07	测试 - 07	测试 - 07	测试 - 07	测试 - 07
12	测试 - 08	测试 - 08	测试 - 08	测试 - 08	测试 - 08	测试 - 08	测试 - 08	测试 - 08	测试 - 08
13	测试 - 09	测试 - 09	测试 - 09	测试 - 09	测试 - 09	测试 - 09	测试 - 09	测试 - 09	测试 - 09
14	测试 - 010	测试 - 010	测试 - 010	测试 - 010	测试 - 010	测试 - 010	测试 - 010	测试 - 010	测试 - 010
15	测试 - 011	测试 - 011	测试 - 011	测试 - 011	测试 - 011	测试 - 011	测试 - 011	测试 - 011	测试 - 011
16	测试 - 012	测试 - 012	测试 - 012	测试 - 012	测试 - 012	测试 - 012	测试 - 012	测试 - 012	测试 - 012
17	测试 - 013	测试 - 013	测试 - 013	测试 - 013	测试 - 013	测试 - 013	测试 - 013	测试 - 013	测试 - 013
18	测试 - 014	测试 - 014	测试 - 014	测试 - 014	测试 - 014	测试 - 014	测试 - 014	测试 - 014	测试 - 014
19	测试 - 015	测试 - 015	测试 - 015	测试 - 015	测试 - 015	测试 - 015	测试 - 015	测试 - 015	测试 - 015
20	测试 - 016	测试 - 016	测试 - 016	测试 - 016	测试 - 016	测试 - 016	测试 - 016	测试 - 016	测试 - 016
21	测试 - 017	测试 - 017	测试 - 017	测试 - 017	测试 - 017	测试 - 017	测试 - 017	测试 - 017	测试 - 017
22	测试 - 018	测试 - 018	测试 - 018	测试 - 018	测试 - 018	测试 - 018	测试 - 018	测试 - 018	测试 - 018
23	测试 - 019	测试 - 019	测试 - 019	测试 - 019	测试 - 019	测试 - 019	测试 - 019	测试 - 019	测试 - 019
24	测试 - 020	测试 - 020	测试 - 020	测试 - 020	测试 - 020	测试 - 020	测试 - 020	测试 - 020	测试 - 020
25	测试 - 021	测试 - 021	测试 - 021	测试 - 021	测试 - 021	测试 - 021	测试 - 021	测试 - 021	测试 - 021

完美!
然后修修改改