

Java poi operation word

原创 HDXiazai Posted on 2020-04-09 13:58:47 Views: 635 Collection 2 Likes 1

Category Column: Development Article Tags: java poi



Development This column includes this content

4 Subscribe 59 articles

There is too much information on the Internet. It is basically useless to follow their articles. I wrote it myself combining their methods.

I don't guarantee how good the code is, but I guarantee that you can run it in your project and see the effect.

Read in template: test.docx

报告日期: \${data}

报告内容:

苹果	\${apple}
香蕉	\${banana}
合计	<u>\${aaa}</u>

java:

```
package com.example.wordpoi.poi;

import org.apache.poi.POIXMLProperties;
import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.xwpf.extractor.XWPFWordExtractor;
import org.apache.poi.xwpf.usermodel.*;
import org.apache.xmlbeans.XmlException;
import org.apache.xmlbeans.XmlToken;
import org.openxmlformats.schemas.drawingml.x2006.main.CTNonVisualDrawingProps;
import org.openxmlformats.schemas.drawingml.x2006.main.CTPositiveSize2D;
import org.openxmlformats.schemas.drawingml.x2006.wordprocessingDrawing.CTInline;
import org.openxmlformats.schemas.wordprocessingml.x2006.main.CTTblPr;
import org.openxmlformats.schemas.wordprocessingml.x2006.main.CTTblWidth;
import org.openxmlformats.schemas.wordprocessingml.x2006.main.CTTcPr;
import org.openxmlformats.schemas.wordprocessingml.x2006.main.STVerticalJc;

import java.io.*;
import java.math.BigInteger;
import java.util.*;

/**
 * Created By: hdx
 * Date: 2020-04-09 09:13
 */
public class XwpfTest {

    protected static List<XWPFPictureData> pictures = new ArrayList<XWPFPictureData>();

    /**
     * Access the contents of XWPFDocument through XWPFWordExtractor
     * @throws Exception
     */
    public static void main(String[] args) throws Exception{
//testReadByExtractor();
        Map<String, Object> param = new HashMap<String, Object>();
        param.put("${apple}", "College 1");
    }
}
```

```

param.put("${data}", "2020-04-09");
param.put("${banana}", "Professional 1");
Map<String, Object> header = new HashMap<String, Object>();
header.put("width", 100);
header.put("height", 50);
header.put("type", "png");
header.put("content", "F:\\123.png");
param.put("${aaa}", header);
XWPFDocument doc = testReadByDoc2(param);
FileOutputStream fopts = new FileOutputStream("F:\\1111\\2.docx");
doc.write(fopts);
fopts.close();
//testReadByDoc();
//testSimpleWrite();
//testWriteTable();
}

public static XWPFDocument testReadByDoc2(Map<String, Object> param) throws Exception {
    InputStream is = new FileInputStream("F:\\1111\\test.docx");
    XWPFDocument doc = new XWPFDocument(is);
    try {
        if (param != null && param.size() > 0) {

            //Processing paragraphs
            List<XWPFFParagraph> paragraphList = doc.getParagraphs();
            processParagraphs(paragraphList, param, doc);

            //Process the table
            Iterator<XWPFTable> it = doc.getTablesIterator();
            while (it.hasNext()) {
                XWPFTable table = it.next();
                List<XWPFTableRow> rows = table.getRows();
                for (XWPFTableRow row : rows) {
                    List<XWPFTableCell> cells = row.getTableCells();
                    for (XWPFTableCell cell : cells) {
                        List<XWPFFParagraph> paragraphListTable = cell.getParagraphs();
                        processParagraphs(paragraphListTable, param, doc);
                    }
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return doc;
}

/**
 * Processing paragraphs
 * @param paragraphList
 * @throws FileNotFoundException
 * @throws InvalidFormatException
 */
public static void processParagraphs(List<XWPFFParagraph> paragraphList, Map<String, Object> param, XWPFDocument doc) throws InvalidFormatException, FileNotFoundException{
    if(paragraphList != null && paragraphList.size() > 0){
        for(XWPFFParagraph paragraph:paragraphList){
            List<XWPFFRun> runs = paragraph.getRuns();
            for (XWPFFRun run : runs) {
                String text = run.getText(0);
                System.out.println("text=="+text);
                if(text != null){

```

```

        boolean isSetText = false;
        for (Map.Entry<String, Object> entry : param.entrySet()) {
            String key = entry.getKey();
            if(text.indexOf(key) != -1){
                isSetText = true;
                Object value = entry.getValue();
                if (value instanceof String) {//text replacement
                    System.out.println("key=="+key);
                    text = text.replace(key, value.toString());
                } else if (value instanceof Map) { //Image replacement
                    text = text.replace(key, "");
                    Map pic = (Map)value;
                    int width = Integer.parseInt(pic.get("width").toString());
                    int height = Integer.parseInt(pic.get("height").toString());
                    int picType = getPictureType(pic.get("type").toString());
                    String byteArray = (String) pic.get("content");

                    System.out.println("==" + byteArray);
                    //ByteArrayInputStream byteInputStream = new ByteArrayInputStream(byteArray);

                    //int ind = doc.getAllPictures().size() - 1;
                    //doc.createPicture(ind, width, height,paragraph);
                    CTInline inline = run.getCTR().addNewDrawing().addNewInline();
                    insertPicture(doc,byteArray,inline, width, height);

                }
            }
            if(isSetText){
                run.setText(text,0);
            }
        }
    }
}

/**
 * insert Picture
 * @param document
 * @param filePath
 * @param inline
 * @param width
 * @param height
 * @throws InvalidFormatException
 * @throws FileNotFoundException
 */
private static void insertPicture(XWPFDocument document, String filePath,
                                 CTInline inline, int width,
                                 int height) throws InvalidFormatException,
                                 FileNotFoundException {
    document.addPictureData(new FileInputStream(filePath),XWPFDocument.PICTURE_TYPE_PNG);
    int id = document.getAllPictures().size() - 1;
    final int EMU = 9525;
    width *= EMU;
    height *= EMU;
    String blipId =
        document.getAllPictures().get(id).getPackageRelationship().getId();
    String picXml = getPicXml(blipId, width, height);
    XmlToken xmlToken = null;
    try {
        xmlToken = XmlToken.Factory.parse(picXml);
    } catch (XmlException xe) {
        xe.printStackTrace();
    }
}

```

```

    inline.set(xmlToken);
    inline.setDistT(0);
    inline.setDistB(0);
    inline.setDistL(0);
    inline.setDistR(0);
    CTPositiveSize2D extent = inline.addNewExtent();
    extent.setCx(width);
    extent.setCy(height);
    CTNonVisualDrawingProps docPr = inline.addNewDocPr();
    docPr.setId(id);
    docPr.setName("IMG_" + id);
    docPr.setDescr("IMG_" + id);
}

/**
 * get the xml of the picture
 * @param blipId
 * @param width
 * @param height
 * @return
 */
private static String getPicXml(String blipId, int width, int height) {
    String picXml =
        "" + "<a:graphic xmlns:a=\"http://schemas.openxmlformats.org/drawingml/2006/main\">" +
        " <a:graphicData uri=\"http://schemas.openxmlformats.org/drawingml/2006/picture\">" +
        " <pic:pic xmlns:pic=\"http://schemas.openxmlformats.org/drawingml/2006/picture\">" +
        " <pic:nvPicPr>" + " <pic:cNvPr id=\"\" + 0 +
        " name=\"Generated\"/>" + " <pic:cNvPicPr/>" +
        " </pic:nvPicPr>" + " <pic:blipFill>" +
        " <a:blip r:embed=\"\" + blipId +
        " xmlns:r=\"http://schemas.openxmlformats.org/officeDocument/2006/relationships\"/>" +
        " <a:stretch>" + " <a:fillRect/>" +
        " </a:stretch>" + " </pic:blipFill>" +
        " <pic:spPr>" + " <a:xfrm>" +
        " <a:off x=\"0\" y=\"0\"/>" +
        " <a:ext cx=\"\" + width + " cy=\"\" + height +
        "\"/>" + " </a:xfrm>" +
        " <a:prstGeom prst=\"rect\"/>" +
        " <a:avLst/>" + " </a:prstGeom>" +
        " </pic:spPr>" + " </pic:pic>" +
        " </a:graphicData>" + "</a:graphic>";
    return picXml;
}

public static List<XWPFPictureData> getAllPictures() {
    return Collections.unmodifiableList(pictures);
}
/**
 * @param id
 * @param width width
 * @param height
 * @param paragraph paragraph
 */
public static void createPicture(int id, int width, int height,XWPFPParagraph paragraph) {
    final int EMU = 9525;
    width *= EMU;
    height *= EMU;
    String blipId = getAllPictures().get(id).getPackageRelationship().getId();
    CTInline inline = paragraph.createRun().getCTR().addNewDrawing().addNewInline();

    System.out.println(blipId+":"+inline);

    String picXml = ""
        + "<a:graphic xmlns:a=\"http://schemas.openxmlformats.org/drawingml/2006/main\">" +
        " <a:graphicData uri=\"http://schemas.openxmlformats.org/drawingml/2006/picture\">" +
        " <pic:pic xmlns:pic=\"http://schemas.openxmlformats.org/drawingml/2006/picture\">"

```

```

+ " <pic:nvPicPr>" + " <pic:cNvPr id="""
+ id
+ "\" name=\"Generated\"/>"
+ " <pic:cNvPicPr/>"
+ " </pic:nvPicPr>"
+ " <pic:blipFill>"
+ " <a:blip r:embed="""
+ blipId
+ "\" xmlns:r=\"http://schemas.openxmlformats.org/officeDocument/2006/relationships\"/>"
+ " <a:stretch>"
+ " <a:fillRect/>"
+ " </a:stretch>"
+ " </pic:blipFill>"
+ " <pic:spPr>"
+ " <a:xfrm>"
+ " <a:off x=\"0\" y=\"0\"/>"
+ " <a:ext cx="""
+ width
+ " cy="""
+ height
+ " />"
+ " </a:xfrm>"
+ " <a:prstGeom prst=\"rect\"/>"
+ " <a:avLst/>"
+ " </a:prstGeom>"
+ " </pic:spPr>"
+ " </pic:pic>"
+ " </a:graphicData>" + "</a:graphic>",

inline.addNewGraphic().addNewGraphicData();
XmlToken xmlToken = null;
try {
    xmlToken = XmlToken.Factory.parse(picXml);
} catch (XmlException xe) {
    xe.printStackTrace();
}
inline.set(xmlToken);

inline.setDistT(0);
inline.setDistB(0);
inline.setDistL(0);
inline.setDistR(0);

CTPositiveSize2D extent = inline.addNewExtent();
extent.setCx(width);
extent.setCy(height);

CTNonVisualDrawingProps docPr = inline.addNewDocPr();
docPr.setId(id);
docPr.setName("picture" + id);
docPr.setDescr("test");
}

/**
 * According to the image type, get the corresponding image type code
 * @param picType
 * @return int
 */
private static int getPictureType(String picType){
    int res = XWPFDocument.PICTURE_TYPE_PICT;
    if(picType != null){
        if(picType.equalsIgnoreCase("png")){
            res = XWPFDocument.PICTURE_TYPE_PNG;
        }else if(picType.equalsIgnoreCase("dib")){
            res = XWPFDocument.PICTURE_TYPE_DIB;
        }else if(picType.equalsIgnoreCase("emf")){

```

```

        res = XWPFDocument.PICTURE_TYPE_EMF;
    }else if(picType.equalsIgnoreCase("jpg") || picType.equalsIgnoreCase("jpeg")){
        res = XWPFDocument.PICTURE_TYPE_JPEG;
    }else if(picType.equalsIgnoreCase("wmf")){
        res = XWPFDocument.PICTURE_TYPE_WMF;
    }
}
return res;
}
/***
 * Match the incoming information set with the template
 * @param value The area of the template that needs to be replaced
 * @param textMap incoming information collection
 * @return The template needs to replace the corresponding value of the regional information set
 */
public static Object changeValue(String value, Map<String, Object> textMap){
    Set<Map.Entry<String, Object>> textSets = textMap.entrySet();
    Object valu = "";
    for (Map.Entry<String, Object> textSet : textSets) {
        //Match template and replacement value format ${key}
        String key = textSet.getKey();
        if (value.indexOf(key)!= -1){
            valu = textSet.getValue();
        }
    }
    return valu;
}

/***
 * Check if the text contains $
 * @param text text
 * @return includes return true, does not include return false
 */
public static boolean checkText(String text){
    boolean check = false;
    if(text.indexOf("$")!= -1){
        check = true;
    }
    return check;
}

// Read the file
public static void testReadByExtractor() throws Exception {

    InputStream is = new FileInputStream("F:\\1111\\test.docx");
    XWPFDocument doc = new XWPFDocument(is);
    XWPFWordExtractor extractor = new XWPFWordExtractor(doc);
    String text = extractor.getText();
    System.out.println(text);
    POIXMLProperties.CoreProperties coreProps = extractor.getCoreProperties();
    printCoreProperties1(coreProps);
    close1(is);
}

/***
 * Output CoreProperties information
 * @param coreProps
 */
private static void printCoreProperties1(POIXMLProperties.CoreProperties coreProps) {
    System.out.println(coreProps.getCategory()); //Category
    System.out.println(coreProps.getCreator()); //Creator
    System.out.println(coreProps.getCreated()); //Creation time
    System.out.println(coreProps.getTitle()); //Title
}

```

```

/**
 * Close the input stream
 * @param is
 */
private static void close1(InputStream is) {
    if (is != null) {
        try {
            is.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

/**
 * Access the content through XWPFDocument. For XWPF documents, this is the best way to read.
 * @throws Exception
 */
public static void testReadByDoc() throws Exception {
    InputStream is = new FileInputStream("F:\\1111\\test.docx");
    XWPFDocument doc = new XWPFDocument(is);
    List<XWPFFParagraph> paras = doc.getParagraphs();
    for (XWPFFParagraph para : paras) {
        // Properties of the current paragraph
// CTPPr pr = para.getCTP().getPPr();
        System.out.println(para.getText());
    }
    //Get all the tables in the document
    List<XWPFTable> tables = doc.getTables();
    List<XWPFTableRow> rows;
    List<XWPFTableCell> cells;
    for (XWPFTable table : tables) {
        //Table properties
// CTTblPr pr = table.getCTTbl().getTblPr();
        //Get the corresponding row of the table
        rows = table.getRows();
        for (XWPFTableRow row : rows) {
            //Get the cell corresponding to the row
            cells = row.getTableCells();
            for (XWPFTableCell cell : cells) {
                System.out.println(cell.getText());
            }
        }
    }
    close1(is);
}

/**
 * Basic write operations
 * @throws Exception
 */
public static void testSimpleWrite() throws Exception {
    //Create a new document
    XWPFDocument doc = new XWPFDocument();
    //Create a paragraph
    XWPFFParagraph para = doc.createParagraph();

    //An XWPFFRun represents an area with the same properties.
    XWPFFRun run = para.createRun();
    run.setBold(true); // bold
    run.setText("bold content");
    run = para.createRun();
    run.setColor("FF0000");
    run.setText("Red words.");
    OutputStream os = new FileOutputStream("F:\\1111\\simpleWrite.docx");
}

```

```

// Output doc to the output stream
doc.write(os);
close(os);
}

/**
 * Write a table
 * @throws Exception
 */
public static void testWriteTable() throws Exception {
    XWPFDocument doc = new XWPFDocument();
    //Create a table with 5 rows and 5 columns
    XWPFTable table = doc.createTable(5, 5);
    //The 5 rows originally created by the added columns here cannot be obtained when using the getTableCells() method, but can be obtained
rows.
// table.addNewCol(); //Add a column to the table, making it 6 columns
    table.createRow(); //Add a new row to the table, making it 6 rows
    List<XWPFTableRow> rows = table.getRows();
    //Table properties
    CTTblPr tablePr = table.getCTTbl().addNewTblPr();
    //Table width
    CTTblWidth width = tablePr.addNewTblW();
    width.setW(BigInteger.valueOf(8000));
    XWPFTableRow row;
    List<XWPFTableCell> cells;
    XWPFTableCell cell;
    int rowSize = rows.size();
    int cellSize;
    for (int i=0; i<rowSize; i++) {
        row = rows.get(i);
        //Add new cell
        row.addNewTableCell();
        //Set the row height
        row.setHeight(500);
        // Row attributes
// CTTPr rowPr = row.getCTRow().addNewTrPr();
        //This method can get the newly added cell.
// List<CTTc> list = row.getCTRow().getTcList();
        cells = row.getTableCells();
        cellSize = cells.size();
        for (int j=0; j<cellSize; j++) {
            cell = cells.get(j);
            if ((i+j)%2==0) {
                //Set the color of the cell
                cell.setColor("ff0000"); //red
            } else {
                cell.setColor("0000ff"); //blue
            }
            //Cell properties
            CTTcPr cellPr = cell.getCTTc().addNewTcPr();
            cellPr.addNewVAlign().setVal(STVerticalJc.CENTER);
            if (j == 3) {
                //Set the width
                cellPr.addNewTcW().setW(BigInteger.valueOf(3000));
            }
            cell.setText(i + ", " + j);
        }
    }
    //If the file does not exist, it will be created automatically
    OutputStream os = new FileOutputStream("F:\\1111\\table.docx");
    //Write to file
    doc.write(os);
    close(os);
}
/** 
 * Close the output stream

```

```
* @param os
*/
private static void close(OutputStream os) {
    if (os != null) {
        try {
            os.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.2.6.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>wordpoi</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>wordpoi</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <!--<repositories>-->
  <!--<repository>-->
    <!--<id>com.e-iceblue</id>-->
    <!--<name>e-iceblue</name>-->
    <!--<url>http://repo.e-iceblue.com/nexus/content/groups/public/</url>-->
  <!--</repository>-->
<!--</repositories>-->

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
      <exclusions>
        <exclusion>
          <groupId>org.junit.vintage</groupId>
          <artifactId>junit-vintage-engine</artifactId>
        </exclusion>
      </exclusions>
    </dependency>

    <!--<dependency>-->
    <!--<groupId>org.apache.poi</groupId>-->
    <!--<artifactId>poi-ooxml</artifactId>-->
    <!--<version>3.14</version>-->
  <!--</dependency>-->
```

```

<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-excelant</artifactId>
    <version>3.14</version>
</dependency>

<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-examples</artifactId>
    <version>4.0.0</version>
</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml-schemas</artifactId>
    <version>3.10-FINAL</version>
</dependency>

<!--<dependency>-->
<!--<groupId>org.apache.poi</groupId>-->
<!--<artifactId>poi-ooxml</artifactId>-->
<!--<version>4.0.0</version>-->
<!--</dependency>-->

<!--<!-Apache poi inserts tables, pictures, etc. into tables in word and starts->-->
<!--<dependency>-->
<!--<groupId>org.apache.poi</groupId>-->
<!--<artifactId>poi</artifactId>-->
<!--<version>4.0.0</version>-->
<!--</dependency>-->

<!--<dependency>-->
<!--<groupId>org.apache.poi</groupId>-->
<!--<artifactId>poi</artifactId>-->
<!--<version>3.11</version>-->
<!--</dependency>-->

<!--<dependency>-->
<!--<groupId>org.apache.poi</groupId>-->
<!--<artifactId>poi-ooxml</artifactId>-->
<!--<version>3.9</version>-->
<!--</dependency>-->

<!--<dependency>-->
<!--<groupId>org.apache.xmlbeans</groupId>-->
<!--<artifactId>xmlbeans</artifactId>-->
<!--<version>3.1.0</version>-->
<!--</dependency>-->

<dependency>
    <groupId>com.deeppoove</groupId>
    <artifactId>poi-tl</artifactId>
    <version>1.6.0-beta1</version>
</dependency>
<!--END-->

<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml-schemas</artifactId>
    <version>3.9</version>
</dependency>

<!--<dependency>-->

```

```

<!--<groupId>org.apache.poi</groupId>-->
<!--<artifactId>poi-ooxml</artifactId>-->
<!--<version>3.7</version>-->
<!--</dependency>-->

<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.9</version>
</dependency>

<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>3.9</version>
</dependency>

<dependency>
    <groupId>org.apache.xmlbeans</groupId>
    <artifactId>xmlbeans</artifactId>
    <version>2.3.0</version>
</dependency>

<!--<dependency>-->
<!--<groupId>org.apache.xmlbeans</groupId>-->
<!--<artifactId>xmlbeans</artifactId>-->
<!--<version>2.6.0</version>-->
<!--</dependency>-->
<dependency>
    <groupId>org.junit.platform</groupId>
    <artifactId>junit-platform-commons</artifactId>
    <version>RELEASE</version>
    <scope>compile</scope>
</dependency>

<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.4</version>
</dependency>

<dependency>
    <groupId>joda-time</groupId>
    <artifactId>joda-time</artifactId>
    <version>2.9.4</version>
</dependency>

<dependency>
    <groupId>jacob</groupId>
    <artifactId>jacob</artifactId>
    <scope>system</scope>
    <systemPath>${basedir}/lib/jacob.jar</systemPath>
</dependency>

<dependency>
    <groupId>ooxml-schemas</groupId>
    <artifactId>ooxml-schemas</artifactId>
    <scope>system</scope>
    <systemPath>${basedir}/lib/ooxml-schemas-1.1.jar</systemPath>
</dependency>

<dependency>
    <groupId>spire.doc.free-2.7.3</groupId>
    <artifactId>spire.doc.free-2.7.3</artifactId>
    <scope>system</scope>
    <systemPath>${basedir}/lib/spire.doc.free-2.7.3.jar</systemPath>

```

```

</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.9</version>
</dependency>

</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>

```

Output effect:



苹果	学院 1
香蕉	专业 1
合计	

<https://blog.csdn.net/mDXiazai>

The three methods in the code can also run `testReadByDoc(); testSimpleWrite(); testWriteTable();`

Java poi export word

Java POI library is an open source project developed by Apache Software Foundation, which is specially used to read and write files in Microsoft Office format, such as `Word` , `Excel` , `PPT` .

poi realizes word electronic signature_ poi -tl 1.12 export signature image

1. Electronic signature approval by uploading signature images: Parse template: `//1.getResourceAsStream()&templateInputStream=resourceAsStream=null;resourceAsStream=`

java springboot poi -tl generates multiple signatures to word documents, floating above the text

`java springboot poi -tl` generates multiple signatures to `word` documents, suspended above the text 1. There are a large number of different `word` files and different online signat

Error: `java.lang.NoClassDefFoundError: org/apache/xmlbeans/XmlException`

laterstage's blog

Error: `java.lang.NoClassDefFoundError: org/apache/xmlbeans/XmlException`

[Java] Solved: `org.apache.poi.openxml4j.exceptions.InvalidFormatException`

[Latest Releases](#)

Yu Xiaoxia's knowledge blog

The Apache POI library is a widely used toolkit for document processing in Java . However, when processing Office documents such as Excel and Word , developers may enco

Java word digital signature, unable to verify the digital signature on the word document.

Hi kaniganti, >> I have added the Digital Signature on a `word` document using my own .pfx file in `java` with Apache POI API. What do you mean with this? Do you create .pfx file

Java implements the function of opening word documents online and adding seals/stamps/signatures_ Java adds electronic signatures in word ...

`Java` implements the function of opening a `Word` document online and adding a seal/stamp/signature This article introduces how to use the PageOffice plug-in in `Java` to implen

Solve the problem that `Java poi` 3.8 and other versions `cannot` insert pictures into `word`

Stars

Solve the problem that `Java poi` 3.8 and other versions `cannot` insert pictures into `word`. Problem: Recently, there is a demand to convert data in Excel to `word` , including text ar

Error importing Excel file

seanchen1990's Colum

org.apache.poi .POI XMLException: org.apache.poi .openxml4j.exceptions.InvalidFormatException : Package should contain a content type part [M1.13]

Java fills in the preset parameters of the word document and generates (including pictures) _ java word template fills in data-C...

(3) Java fills in the word template (4) Notepad++ XML Tools plug-in installation I need to develop a contract signing function. The login function on the applet will electronically si

[java - electronic signature function implementation]_ java electronic signature

WordUtils import lombok.extern.slf4j.Slf4j; import org.apache.commons.lang3.StringUtils; import org.apache.poi.xwpf.usermodel.XWPFDocument; import org.apache.poi.xwpf.u

Maven缺少依赖: java.lang.ClassNotFoundException: org.apache.poi.openxml4j.exceptions.InvalidFormatException 热门推荐

小蚯蚓的博客

1.报错描述 运行IpHelper类的时候，报错： Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/poi/openxml4j/exceptions/InvalidFormatException at com.g

java poi操作word模版文件生成表单和修改

在提供的压缩包文件中，“改进wordDemo-master.zip”和“java poi template .zip”可能包含了示例代码和模板文件，用于演示如何使用Java POI操作Word模板生成表单和进行修改。

Java 给 Word 添加certificate数字签名_ java word 使用 使用数字证书...

数字签名是一种应用于数字数据的电子加密身份验证戳记，主要用于确认来自签名者的信息以及文档内容未被篡改。本文，将介绍如何使用 Spire.Doc for Java 以编程方式向 Word 文档添加数字签名。

java word 数字签名_ java 加密和数字签名4 数字签名

import java.security.Signature; import java.security.KeyPairGenerator; import java.security.KeyPair; import java.security.SignatureException; /** * 此例子是数字签名的例子,使用F

java使用poi操作.doc word模板替换，循环插入表格

总之，Apache POI为Java开发者提供了强大而灵活的工具，可以方便地对Word文档进行各种操作，包括模板替换和表格插入。通过熟练掌握这些技术，你可以构建自动化报告生

java POI 操作 WORD 包

Java POI 操作 Word 是一个高效且广泛使用的技术，它允许开发者在 Java 应用程序中创建、修改和读取 Microsoft Word 文档。相对于其他如 Jacob 的解决方案，POI 提供了更

java 制作签名版电子合同_ java 电子合同demo

Java 制作带签名的电子合同 根据项目需求,需要生成一个带电子签名的合同,即客户在手机进行签名后,将签名图片合成到合同中。目前方案有两个,使用docx4j插件根据书签生成w

Poi实现根据word模板导出-文本段落篇

迷途小书童的博客

之前说过poi本质就是把word解析成xml，那我们直接把word导出为xml看一下，用office可以快速把word导出为xml格式，这里我们把我们的模板导出为xml看下同样的占位符有啥：

java使用poi、itextpdf将word、ppt转为pdf文件，并对pdf文件加盖签章

xuwenze的博客

转pdf文件

关于不可变列表unmodifiableList

进击的豌豆的博客

场景：在使用poi操作docx时，想对runs进行删除和追加操作，发现抛出异常：Exception in thread “main” java.lang.UnsupportedOperationException。 经过断点调试发现runs是-

使用 XMLBeans 进行编程

weixin_34202952的博客

使用 XMLBeans 进行编程 ...

2021-06-11 java POI导出word文档用到的工具类，包括饼状图，柱状图，以及表格的跨行跨列

qq_36388711的博客

package com.joyin.idc.entity.util; import java.awt.Point; import java.awt.Rectangle; import java.io.ByteArrayInputStream; import java.io.IOException; import java.io.InputStream; i

使用poi根据模版生成word文档，支持插入图片，复制表格，插入、循环插入数据，继承模版大部分格式及样式（优化版）

kanglong129的博客

一、制作word模版，\${xxxx}是一会要替换的内容，最下面的表格是要插入数据，根据是否以\$开头来判断是需要替换还是插入数据，注意如果是需要循环插入数据，制作的表格模

解决poi导出Excel异常org.openxmlformats.schemas.spreadsheetspreadshe

diyu8056的博客

未捕获到 servlet OUTEXCEL 的其中一个服务方法中抛出的异常。抛出的异常：java.lang.NoClassDefFoundError: org.openxmlformats.schemas.spreadsheetml.x2006.main.CTF

java poi操作word

Apache POI library can be used in Java to operate Word documents . The following is a simple example code: ```` java import java.io.File ; import java.io.FileInputStream ; impor