

```

package com.haiyisoft.common.poi;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;

import org.apache.poi.POIXMLDocument;
import org.apache.poi.openxml4j.exceptions.InvalidFormatException;
import org.apache.poi.xwpf.usermodel.ParagraphAlignment;
import org.apache.poi.xwpf.usermodel.XWPFDocument;
import org.apache.poi.xwpf.usermodel.XWPFParagraph;
import org.apache.poi.xwpf.usermodel.XWPFRun;
import org.apache.poi.xwpf.usermodel.XWPFTable;
import org.apache.poi.xwpf.usermodel.XWPFTableCell;
import org.apache.poi.xwpf.usermodel.XWPFTableRow;
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;

```

```

public class PoiWordTemplate {

    private XWPFDocument document;

    public XWPFDocument getDocument() {
        return document;
    }
    public void setDocument(String templatePath) {
        try {
            this.document = new XWPFDocument(
                POIXMLDocument.openPackage(templatePath));
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

```

        //this.document = document;
    }
    /**
     * 往模板中插入图片
     * @param document
     * @param mapImage
     * @param width
     * @param height
     */
    public void replaceTextToImage(Map<String,String> mapImage,int width,int height){
        List<XWPFParagraph> listParagraphs = this.document.getParagraphs();
        for(int i=0;i<listParagraphs.size();i++){
            for (Entry<String,String> entry : mapImage.entrySet()) {
                if (listParagraphs.get(i).getText().trim().indexOf(entry.getKey()) != -1) {
                    CTInline inline =
listParagraphs.get(i).createRun().getCTR().addNewDrawing().addNewInline();

                    try {
                        insertPicture(entry.getValue(),inline,width,height);
                    } catch (InvalidFormatException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    } catch (FileNotFoundException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }
            }
        }
    }

    /**
     * 替换 word 中的文字
     * @param document
     * @param mapValue    * @param newText
     */
    public void replaceTextToText(Map<String,String> mapValue){
        List<XWPFRun> listRun;
        List<XWPFParagraph> listParagraphs = this.document.getParagraphs();
        for(int i=0;i<listParagraphs.size();i++){

            listRun = listParagraphs.get(i).getRuns();
            for(int j=0;j<listRun.size();j++){

```

```

                if(mapValue.get(listRun.get(j).getText(0)) != null ||
                "null".equals(mapValue.get(listRun.get(j).getText(0)))){
                    listRun.get(j).setText(mapValue.get(listRun.get(j).getText(0)),0);
                }
            }
        }
    }
}

```

```

/**
 * 往表格中插入数据
 * @param document
 * @param list
 * @param position
 */
public void setTablesValue(List<Map<String, Object>> list,int position,int columnNum) {
    Iterator<XWPFTable> it = this.document.getTablesIterator();
    int count = 0;
    Map<String,String> columnMap = new HashMap<String,String>();
    while (it.hasNext()) {
        XWPFTable table = (XWPFTable) it.next();
        if(count == position){
            XWPFTableRow firstRow = table.getRow(1);
            List<XWPFTableCell> rowCell = firstRow.getTableCells();
            for(int i=0;i<rowCell.size();i++){
                columnMap.put(i+"", rowCell.get(i).getText().trim());
            }

            table.removeRow(1);//删除第二行

            for(int i=0;i<list.size();i++){
                XWPFTableRow row = table.createRow();//创建新的一行时创建了第
                一个 cell

                row.setHeight(30);
                /* for(int j=0;j<=columnNum-2;j++){
                    row.addNewTableCell();
                }*/
            }

            int rcount = table.getNumberOfRows();
            for (int i = 1; i < rcount; i++) {
                XWPFTableRow newRow = table.getRow(i);
                List<XWPFTableCell> cells = newRow.getTableCells();
                for(int j=0;j<cells.size();j++){

```

```

        cells.get(j).setText(list.get(i-1).get(columnMap.get(j+""))+"");// 标题栏不用改变单元格的值
    }

    }

    }
    count++;
}

}

/**
 * 设置用电建议书里的基本信息 第一个表格
 */
public void setOfferBorkBasicInfo(Map<String, Object> map,List<Map<String, Object>>
list,int position){
    Iterator<XWPFTable> it = this.document.getTablesIterator();
    int count = 0;
    Map<String,String> columnMap = new HashMap<String,String>();
    while (it.hasNext()) {
        XWPFTable table = (XWPFTable) it.next();
        if(count == position){
            int rcount = table.getNumberOfRows();
            for (int i = 0; i < rcount; i++) {
                if(i<=4){
                    XWPFTableRow newRow = table.getRow(i);
                    List<XWPFTableCell> cells = newRow.getTableCells();
                    for(int j=0;j<cells.size();j++){

                        if(!"null".equals(map.get(cells.get(j).getText().trim()+""))){
                            String temp = cells.get(j).getText().trim()+"";
                            cells.get(j).removeParagraph(0);
                            cells.get(j).setText(map.get(temp)+"");
                        }
                    }
                }
            }
        }

        XWPFTableRow sixRow = table.getRow(6);
        List<XWPFTableCell> rowCell = sixRow.getTableCells();
        for(int i=0;i<rowCell.size();i++){
            columnMap.put(i+"", rowCell.get(i).getText().trim());
        }

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

```

一个 cell

```
XWPFTableRow row = table.createRow();//创建新的一行时创建了第
一个 cell

    row.setHeight(35);
    for(int j=0;j<=6;j++){
        row.addNewTableCell();
    }

}
table.removeRow(6);//删除模板中的字段行

int rowCount = table.getNumberOfRows();
int k= 0;
for (int i = 6; i < rowCount; i++) {
    XWPFTableRow tempRow = table.getRow(i);
    List<XWPFTableCell> cells = tempRow.getTableCells();
    for(int j=0;j<cells.size();j++){
        // cells.get(j).removeParagraph(0);
        cells.get(j).setText(list.get(k).get(columnMap.get(j+""))+"");
    }
    k++;

}

}
count++;
}
}
/**
 * 生成用电分析与优化用电建议
 * @param list
 * @param position
 */
public void setAnanlyResultTbl(List<String> list,int position){
    Iterator<XWPFTable> it = this.document.getTablesIterator();
    int count = 0;

    while (it.hasNext()) {
        XWPFTable table = (XWPFTable) it.next();

        if(count == position){
            XWPFTableRow tempRow = table.getRow(1);
            List<XWPFTableCell> cellList = tempRow.getTableCells();
            XWPFTableCell cell = cellList.get(0);
            List<XWPFPParagraph> listGraph = cell.getParagraphs();
```

```

        for(int i=0;i<list.size();i++){
            listGraph.get(0).setAlignment(ParagraphAlignment.LEFT);
            XWPFRun r2 = listGraph.get(0).createRun();
            // cell.setText(list.get(i));
            // cell.
            r2.setText(list.get(i));
            r2.setFontSize(12);
            r2.addBreak();
        }
    }
    count++;
}
}
/**
 * 生成预期节能收益
 */
public void setExpectedBenefits(List<String> reslut, Map<String,String> saveMoney,int
position){
    Iterator<XWPFTable> it = this.document.getTablesIterator();
    int count = 0;

    while (it.hasNext()) {
        XWPFTable table = (XWPFTable) it.next();

        if(count == position){
            XWPFTableRow tempRow = table.getRow(1);
            List<XWPFTableCell> cellList = tempRow.getTableCells();
            XWPFTableCell cell = cellList.get(0);
            List<XWPFPParagraph> listGraph = cell.getParagraphs();

            for(int i=0;i<reslut.size();i++){
                listGraph.get(0).setAlignment(ParagraphAlignment.LEFT);
                XWPFRun r2 = listGraph.get(0).createRun();
                r2.setText(reslut.get(i));
                r2.setFontSize(12);
                r2.addBreak();
            }
            /*
            cell.setText(reslut.get(i));
            r2.addBreak();*/
        }

        XWPFTableRow fourRow = table.getRow(3);
    }
}

```

```

        List<XWPFTableCell> cells = fourRow.getTableCells();
        for(int j=0;j<cells.size();j++){

            if(!"null".equals(saveMoney.get(cells.get(j).getText().trim()+""))){
                String temp = cells.get(j).getText().trim()+"";
                cells.get(j).removeParagraph(0);
                cells.get(j).setText(saveMoney.get(temp)+"");
            }
        }

        count++;
    }
}

/* public void setAnanlyResultTbl(Map<String, String> map,int position){
    Iterator<XWPFTable> it = this.document.getTablesIterator();
    int count = 0;

    while (it.hasNext()) {
        XWPFTable table = (XWPFTable) it.next();

        if(count == position){
            int rowCount = table.getNumberOfRows();
            for (int i = 0; i < rowCount; i++) {
                XWPFTableRow tempRow = table.getRow(i);
                List<XWPFTableCell> cells = tempRow.getTableCells();
                for(int j=0;j<cells.size();j++){
                    List<XWPFPParagraph> listParagraphs = cells.get(j).getParagraphs();
                    for(int m=0;m<listParagraphs.size();m++){

                        List<XWPFRun> listRun = listParagraphs.get(m).getRuns();
                        for(int n=0;n<listRun.size();n++){
                            if(map.get(listRun.get(n).getText(0)) != null
|| !"null".equals(map.get(listRun.get(n).getText(0)))){

                                listRun.get(n).setText(map.get(listRun.get(n).getText(0)),0);
                                listRun.get(n).setBold(true);
                            }
                        }
                        for (Entry<String,String> entry : map.entrySet()) {
                            if(listRun.get(n).getText(0).equals(entry.getKey())){

                                }

```

```

    }
    }
    }
    }
    }
    count++;
}

}*/

```

```

public void insertPicture(String filePath,CTInline inline,int width, int height) throws
InvalidFormatException, FileNotFoundException{

```

```

    try {
        String ind = this.document.addPictureData(new FileInputStream(filePath), 5);
    } catch (InvalidFormatException e) {
        System.out.println("exception");
    }

```

```

    int id = this.document.getAllPictures().size()-1;
    final int EMU = 9525;
    width *= EMU;
    height *= EMU;
    String blipId = this.document.getAllPictures().get(id).getPackageRelationship()
        .getId();

```

```

        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("IMG_" + id);
docPr.setDescr("IMG_" + id);
}

```

```

/*    public    static    void    replaceTextToImage(String    templatePath,String
pastePath,Map<String,String> mapImage,int width,int height){
    XWPFDocument document;

```

```

try {
    document = new XWPFDocument(
        POIXMLDocument.openPackage(templatePath));
    replaceTextToImage(document, mapImage, 400, 300);

} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}*/
public void writeNewWord(String pastePath){
    File newFile = new File(pastePath);
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(newFile);
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    try {
        this.document.write(fos);
        fos.flush();
        fos.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

}
public static void main(String args[]) throws Exception{
    XWPFDocument document = new XWPFDocument(
        POIXMLDocument.openPackage("D:\\workspaceHaiYi\\TestPoi\\test.docx"));
    //          CTInline          inline          =
    document.createParagraph().createRun().getCTR().addNewDrawing().addNewInline();
    // List<XWPFParagraph> listParagraphs = document.getParagraphs();
    //          CTInline          inline          =
    listParagraphs.get(1).createRun().getCTR().addNewDrawing().addNewInline();
    //          CTInline          inline          =
    listParagraphs.get(1).getRuns().get(0).getCTR().addNewDrawing().addNewInline();
    //System.out.println("first:"+inline);

```

```

//
insertPicture(document,"D:\\workspaceHaiYi\\TestPoi\\IMG_20130112_165522.jpg",inline,300,2
00);
// test(document,"D:\\workspaceHaiYi\\TestPoi\\IMG_20130112_165522.jpg",inline);

Map<String ,String> map = new HashMap<String,String>();
map.put("图片", "D:\\workspaceHaiYi\\TestPoi\\IMG_20130112_165522.jpg");

//replaceTextTolmage(map,400,300);
//replaceTextToText(document,map,"sssssssss");
List<Map<String,Object>> list = new ArrayList<Map<String,Object>>();
Map<String,Object> dataMap = new HashMap <String,Object>();
dataMap.put("A", "XIAO WANG");
dataMap.put("B", "LIN");
dataMap.put("C", "汪义");
list.add(dataMap);
//setTablesValue(document,list,0);

}
}

```