1. txt、excel、doc、ppt、pdf等格式文档的在线预览
2. 首先是将上述文档格式转成pdf格式：

下载openoffice.org，然后安装，安装到C：program Files目录下：C:\Program Files\OpenOffice.org 3。安装好，启动服务：有两种方法启动服务，一种是通过命令的形式启动，另一种是通过安装(待补充……)。

1. 命令形式启动openoffiec服务：首先定位到该程序的program文件夹下C:\Program Files\OpenOffice.org 3\program，然后输入启动命令：

Soffice -headless -accept="socket,host=127.0.0.1,port=8100;urp;" –nofirststartwizard；其中配置的端口号8100必须与工具类中的端口号一致。命令形式启动的缺点是每次服务器重启（电脑重启）需要重输入一次启动命令。

下载jodconverter，将该文件lib下的jar包拷贝到工程中，建工具类DOC2PDFUtil，代码如下：

**public** **class** DOC2PDFUtil **extends** java.lang.Thread{

**private** File inputFile;//需要转换的文件

**private** File outputFile;// 输出的文件

**private** String inputFileNewName;//对于txt文档需要进行重新编码，定义一个新名称，以便操作

**public** DOC2PDFUtil(File inputFile, File outputFile,String inputFileNewName) {

**this**.inputFile = inputFile;

**this**.outputFile = outputFile;

**this**.inputFileNewName=inputFileNewName;

}

**public** DOC2PDFUtil(){}

**public** **void** docToPdf() {

**synchronized** (**this**) {//线程同步

OpenOfficeConnection connection = **new** SocketOpenOfficeConnection(8100);

**try** {

connection.connect();

DocumentConverter converter = **new** OpenOfficeDocumentConverter(connection);

converter.convert(inputFile, outputFile);

} **catch** (ConnectException cex) {

cex.printStackTrace();

} **finally** {

// close the connection

**if** (connection != **null**) {

connection.disconnect();

connection = **null**;

}

}

**if**(inputFileNewName!=**null**){

File file=**new** File(inputFileNewName);

file.delete();

}

}

}

/\*\*

\*

由于服务是线程不安全的，所以„„需要启动线程

\*/

**public** **void** run() {

**this**.docToPdf();

}

/\*\*

\* 外部调用方法

\* **@throws** FileNotFoundException

\* **@throws** UnsupportedEncodingException

\* \*/

**public** **void** toPDF(String inputFileName,String outputFileName) **throws** Exception{

String fileType=inputFileName.substring(inputFileName.lastIndexOf("."), inputFileName.length());

**if**(fileType.indexOf("txt")!=-1){

String inputFileNameNew=inputFileName.substring(0,inputFileName.lastIndexOf("."));

InputStreamReader in = **new** InputStreamReader( **new** FileInputStream(inputFileName),"GB2312");

inputFileNewName=inputFileName=inputFileNameNew+"(1)."+fileType;

OutputStreamWriter out = **new** OutputStreamWriter(**new** FileOutputStream(inputFileName),"utf-8");

**char**[] cbuf = **new** **char**[1024];

**int** n;

**while**((n=in.read(cbuf))!=-1){

out.write(cbuf, 0, n);

}

in.close();

out.close();

}

File inputFile = **new** File(inputFileName);

File outputFile = **new** File(outputFileName);

DOC2PDFUtil dp=**new** DOC2PDFUtil(inputFile,outputFile,inputFileNewName);

dp.start();

}

**public** File getInputFile() {

**return** inputFile;

}

**public** **void** setInputFile(File inputFile) {

**this**.inputFile = inputFile;

}

**public** File getOutputFile() {

**return** outputFile;

}

**public** **void** setOutputFile(File outputFile) {

**this**.outputFile = outputFile;

}

**public** String getInputFileNewName() {

**return** inputFileNewName;

}

**public** **void** setInputFileNewName(String inputFileNewName) {

**this**.inputFileNewName = inputFileNewName;

}

/\*\*

\* 测试main方法

\* **@throws** ConnectException

\* **@paramargs**

\*/

**public** **static** **void** main(String[] args) **throws** Exception {

**new** DOC2PDFUtil().toPDF("d://temp//test2.doc","d://temp//333.pdf");

}

}

调用工具类DOC2PDFUtil的toPDF方法即可将(txt,doc,excel,ppt)文件格式转成pdf格式，参数是需要转换的文件全称以及路径和转换后的文件名全称和路径(绝对路径)。如：

String inputFileName=”d://test.doc”;

String outputFileName=”d://test.pdf”;

New DOC2PDFUtil().toPDF(inputFileName, outputFileName);

由于该服务是线程不安全的，所以需启动线程。且当所需转换的文件格式是txt时，由于文件格式编码差异会出现乱码，所以需要将txt文件格式的文件重新以utf8格式编码。重新编译txt文件是在原文件的目录下生成一个副本，然后将副本文件转为pdf格式，再将副本文件删除（因为是多线程，所以需线程同步）。

2、将pdf文件格式转为swf文件格式

（1）下载swftools，安装好，本次测试安装在C:\Program Files\SWFTools

建一个工具类PDF2SWFUtil，代码如下：

**public** **class** PDF2SWFUtil {

/\*\*

\*利用SWFTools工具将pdf转换成swf，转换完后的swf文件与pdf同名

\* **@author** lwz

\* **@paramfileDir** PDF文件存放路径（包括文件名）

\* **@paramexePath** 转换器安装路径

\* **@throws** IOException

\*/

**public** **synchronized** **void** pdf2swf(String fileDir, String exePath) **throws** IOException {

String filePath = fileDir.substring(0, fileDir.lastIndexOf("/")); //文件路径

String fileName = fileDir.substring((filePath.length() + 1), fileDir.lastIndexOf("."));//文件名，不带后缀

Process pro = **null**;

**if** (isWindowsSystem()) { //如果是windows系统

String cmd = exePath + " \"" + fileDir + "\" -o \"" + filePath + "/" + fileName + ".swf\" -s " +

" languagedir=D:\\xpdf\\xpdf-chinese-simplified"; //命令行命令

pro = Runtime.*getRuntime*().exec(cmd); //Runtime执行后返回创建的进程对象

} **else** { //如果是linux系统,路径不能有空格，而且一定不能用双引号，否则无法创建进程

String[] cmd = **new** String[3];

cmd[0] = exePath;

cmd[1] = fileDir;

cmd[2] = filePath + "/" + fileName + ".swf";

//Runtime执行后返回创建的进程对象

pro = Runtime.*getRuntime*().exec(cmd);

}

// 非要读取一遍 cmd 的输出，要不不会flush生成文件（多线程）

**new** DoOutput(pro.getInputStream()).start();

**new** DoOutput(pro.getErrorStream()).start();

**try** {

// 调用 waitFor 方法，是为了阻塞当前进程，直到 cmd 执行完

pro.waitFor();

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

/\*\*

\*

判断是否是

windows

操作系统

\* **@author** lwz

\* **@return**

\*/

**private** **boolean** isWindowsSystem() {

String p = System.*getProperty*("os.name");

**return** p.toLowerCase().indexOf("windows") >= 0 ? **true** : **false**;

}

/\*\*

\* 多线程内部类

\* \* 读取转换时cmd进程的标准输出流和错误输出流，这样做是因为如果不读取流，进程将死锁

\* **@author** lwz

\*/

**private** **static** **class** DoOutput **extends** Thread {

**public** InputStream is;

//构造方法

**public** DoOutput(InputStream is) {

**this**.is = is;

}

**public** **void** run() {

BufferedReader br = **new** BufferedReader(**new** InputStreamReader(**this**.is));

String str = **null**;

**try** {

//这里并没有对流的内容进行处理，只是读了一遍

**while** ((str = br.readLine()) != **null**);

} **catch** (IOException e) {

e.printStackTrace();

} **finally** {

**if** (br != **null**) {

**try** {

br.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

}

}

/\*\*

\* 测试main方法

\* **@paramargs**

\*/

**public** **static** **void** main(String[] args) {

//转换器安装路径

String exePath = "c:/Program Files/SWFTools/pdf2swf.exe";

**try** {

**new** PDF2SWFUtil().pdf2swf("d:/temp/test.pdf", exePath);

} **catch** (IOException e) {

System.*err*.println("转换出错！");

e.printStackTrace();

}

}

}

调用工具类PDF2SWFUtil的pdf2swf方法即可将Pdf文件转成swf格式文件。pdf2swf方法参数是

String inputFilePath=” d:/temp/test.pdf”;//需要转换格式文件的路径

String exePath=

”c:/Program Files/SWFTools/pdf2swf.exe”;//swftools的安装路径

new PDF2SWFUtil().pdf2swf(inputFilePath,exePath)即可。(注：windows和linux操作系统都支持)

1. 生成好swf文件，在html中使用embed标签即可，如下：

<embed width="100%" height="180%" wmode="transparent" play="false" loop="true" quality="high" hidden="false" menu="false" autostart="false" allowscriptaccess="never"

allowfullscreen="true" src="D:/temp/333.swf"

pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-flash" class="edui-faked-video">

</embed>