数据获取问题：  
由报表rpx脚本生成对应的数据传递给flash解析成flash文件。  
**ExportFlashData.java文件生成flash需要的数据**

包位置：package com.raqsoft.report.view.svg;

package com.raqsoft.report.view.svg;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import com.raqsoft.common.Logger;

import com.raqsoft.dm.Sequence;

import com.raqsoft.report.model.ReportDefine;

import com.raqsoft.report.usermodel.Context;

import com.raqsoft.report.usermodel.Engine;

import com.raqsoft.report.usermodel.IReport;

import com.raqsoft.report.usermodel.PrintSetup;

import com.raqsoft.report.util.ReportUtils;

import com.raqsoft.app.config.ConfigUtil;

import com.raqsoft.app.config.RaqsoftConfig;

import com.raqsoft.parallel.UnitContext;

public class ExportFlashData {

IReport m\_report = null;

FileOutputStream m\_fos;

static RaqsoftConfig m\_rc = null;

public static InputStream getConfigIS(String configFile) throws Exception {

return UnitContext.getUnitInputStream(configFile);

}

public static RaqsoftConfig loadRaqsoftConfig() throws Exception {

InputStream inputStream = getConfigIS("../web/webapps/demo/WEB-INF/raqsoftConfig.xml");

RaqsoftConfig raqsoftConfig = ConfigUtil.load(inputStream,ConfigUtil.FROM\_ESPROC);//ConfigUtil.load(inputStream, true);

inputStream.close();

return raqsoftConfig;

}

private static void loadFunction(){

try {

m\_rc = loadRaqsoftConfig();

ConfigUtil.loadExtLibs("demo",m\_rc);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void main(String args[]) {

loadFunction();

ExportFlashData cls = new ExportFlashData();

cls.doExport();

System.out.println("");

}

public void doExport(){

try {

String fileLic = "D:/backup/OneDrive/2020/report.xml";

String fileSrc = "D:/works/shell/flash/5\_font\_flash\_print.rpx";

// fileSrc = "D:/works/shell/flash/sub/DIQ\_AR\_RAQRPT\_ADJ\_D.rpx";

// fileSrc = "D:/works/shell/flash/1112.rpx";

// //fileSrc = "D:/works/shell/flash/0808FDGZD1.rpx";

//fileSrc = "D:/works/shell/flash/report\_grid.rpx";

fileLic = "d:/temp/report.xml";

fileSrc = "d:/temp/report\_nodata.rpx";

String outFile = "d:/temp/reportServlet1002.dat";

//fileSrc = "D:/works/shell/flash/1.rpx";

Sequence.readLicense(Sequence.P\_RPT, fileLic);// 2.rpx");//

m\_report = ReportUtils.read(fileSrc);// tmp/jingfen1.rpx");//

Engine engine = new Engine((ReportDefine) m\_report, new Context());

m\_report = engine.calc();

// String fileName = "d:/1.html";

// HtmlReport hr = new HtmlReport(m\_report, fileName);

// String syntax = hr.generateHtml();

// m\_fos = new FileOm\_fosutStream(fileName);

// m\_fos.write(syntax.getBytes());

// m\_fos.close();

Logger.setLevel("debug");

SvgReport sr = new SvgReport(m\_report);

if (1==1){

Logger.debug("1");

m\_fos = new FileOutputStream(outFile);

doFlashInfo();

m\_fos.close();

}

// String

if (1==2){

outFile = "C:/Users/Administrator/Downloads/reportServlet301.svg";

Logger.debug("2");

// FileOutputStream

m\_fos = new FileOutputStream(outFile);

// sr.setClearColor(true);

// sr.setClearImageLayer(true);

sr.exportSvg(m\_fos);

m\_fos.close();

}

// ArrayList<byte[]> pageSvgs = sr.exportPaged();

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

// byte[] svg = pageSvgs.get(i);

// FileOutputStream m\_fos = new

// FileOutputStream("d:/svg\_"+(i+1)+".svg");

// m\_fos.write(svg);

// m\_fos.close();

// }

//

// int totalPage = sr.getReportTotalPage();

// for (int i = 1; i <= totalPage; i++) {

// byte[] svg = sr.getReportPage(i);

// FileOutputStream m\_fos = new FileOutputStream("d:/svg\_" + i+

// ".svg");

// m\_fos.write(svg);

// m\_fos.close();

// }

Logger.debug("OK");

} catch (Throwable x) {

x.printStackTrace();

}

}

private void doFlashInfo(){

try{

SvgReport sr = new SvgReport(m\_report);

boolean isExportCmd = true;

boolean ischrome = false;

int totalPage = sr.getReportTotalPage();

writeInt(totalPage);

for (int i = 1; i <= totalPage; i++) {

byte[] content = sr.getReportPage(i, isExportCmd);

if (i == 1) {

int pageW = sr.getPaperWidth();

int pageH = sr.getPaperHeight();

writeInt(pageW);

writeInt(pageH);

int left = sr.getLeftMargin();

int right = sr.getRightMargin();

int top = sr.getTopMargin();

int bottom = sr.getBottomMargin();

writeInt(left);

writeInt(right);

writeInt(top);

writeInt(bottom);

//String srt2=new String(content,"gbk");

//System.out.println(srt2);

}

int size = content.length;

writeInt(size);

m\_fos.write(content);

//sr.exportCmd(m\_fos);

}

PrintSetup ps = m\_report.getPrintSetup();

byte orient0= ps.getOrientation();

writeInt((int)(orient0));

int imagew=sr.getImageableWidth();

int imageh=sr.getImageableHeight();

writeInt(imagew);

writeInt(imageh);

writeInt(sr.getshrink());

if(ischrome){

writeInt(1);

}

else{

writeInt(0);

}

Logger.debug("After export svg");

}catch( Throwable et){

et.printStackTrace();

}

}

public final void writeInt(int v) throws IOException {

OutputStream out = this.m\_fos;

out.write((v >>> 24) & 0xFF);

out.write((v >>> 16) & 0xFF);

out.write((v >>> 8) & 0xFF);

out.write((v >>> 0) & 0xFF);

}

}

由rpx脚本生成xxx.dat文件给flash解析.

Flash主要四个模块：

1. 主打印
2. 直接打印
3. 字体输出