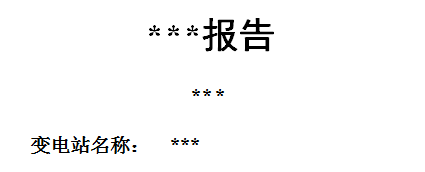
1. **制作文件模板**
2. **创建word文档**

删掉重复的部分，重复的部分会在xml模板里进行循环输出

1. **将活动的数据用\*\*\*来替代**

用\*\*\*替代，可以避免数据在导成模板时，被拆开，难以查询



1. **另存为文档为.xml格式**

一定要用office另存为xml2003格式，不要用插件,否则office无法识别导出的word

1. **格式化.xml文件**

安装firstobject XML Editor，格式化.xml文件

1. **将文件后缀改为.ftl**



1. **替换\*\*\***

将\*\*\*替换为自己的表达式，一定要与java代码中的key相匹配

1. **添加循环**

检查.ftl中自己需要循环输出的地方，添加循环







**注：**循环内的内容，需要用list . 出key，代码值，如上图

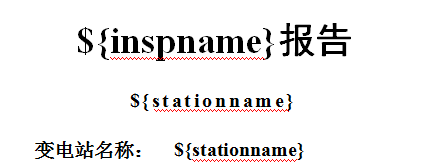
root存放的值可以直接调用，如下图



||



**注：**建议存一份原文的，将表达式一一对应起来，方便后期填值，如下图



1. **后台操作**
2. **导入包**

导入freemarker.jar包,用于生成word

java自带的java.util.zip.ZipOutputStream类实现对多个文件进行压缩

1. **具体代码（直接打包为压缩文件）**

/\*\*

\* 下载报告

\* **@return**

\*/

**public** String downLoadWordAction(){

//设置STRUTS 2不去响应

ActionContext.*getContext*().getActionInvocation().getProxy().setExecuteResult(**false**);

**this**.getResponse().setContentType("text/html;charset=utf-8");

Map<String, Object> map = **new** HashMap<String, Object>();

//文档名称

String wordName = "测试报告";

//获取文档保存地址

String URL = BaseUtils.*getProperties*("URL");

String aa = URL+"/TEST";

String aaa = URL+"/报告.zip";

//创建AA

File f = **new** File(aa);

**if**(!f.exists()){

f.mkdirs();

}**else**{

*deleteDir*(f);

f.mkdirs();

}

aa+="/";

**try** {

File sffile = **new** File(URL);

**if** (!sffile.exists()) {

sffile.mkdirs();

}

Template template = **null**;

Map<String, Object> root = **null**;

Writer out = **null**;

//模板位置

Configuration configuration = **new** Configuration();

configuration.setDirectoryForTemplateLoading(**new** File(URL + "BDE/报告模板/"));

configuration.setDefaultEncoding("utf-8");

configuration.setObjectWrapper(**new** DefaultObjectWrapper());

template = configuration.getTemplate("T.ftl");

root = **new** HashMap<String, Object>();

out = **new** OutputStreamWriter(**new** FileOutputStream(aa+wordName+".doc"), "UTF-8");

//业务逻辑代码

List proinfoList=**new** ArrayList();

root.put("insptype","普测（巡检）");

root.put("proinfoList", proinfoList);

template.process(root, out);// 写入目标文件

out.flush();

out.close();

CompressedUtil compressed = **new** CompressedUtil();

compressed.zip(aa,aaa);// 你要压缩的文件夹

System.*out*.println(aa+ "压缩完。。。"+aaa);

File updatefile = **new** File(aaa);

// 取得文件名。

String filename = updatefile.getName();

// 以流的形式下载文件。

InputStream fis = **new** BufferedInputStream(**new** FileInputStream(aaa));

**byte**[] buffer = **new** **byte**[fis.available()];

fis.read(buffer);

fis.close();

// 清空response

getResponse().reset();

// 设置response的Header

getResponse().addHeader("Content-Disposition", "attachment;filename=" +URLEncoder.*encode*("报告.zip", "UTF-8"));

getResponse().addHeader("Content-Length", "" + updatefile.length());

OutputStream toClient = **new** BufferedOutputStream(getResponse().getOutputStream());

getResponse().setContentType("application/octet-stream");

toClient.write(buffer);

toClient.flush();

toClient.close();

} **catch** (Exception e) {

e.printStackTrace();

map.put("state", "error");

}

**return** **null**;

}

/\*\*

\* 删除目录及其子文件

\* **@param** file

\*/

**public** **static** **void** deleteDir(File file) {

**if** (file.exists()) {

**if** (file.isFile()) {

file.delete();

} **else** **if** (file.isDirectory()) {

File[] files = file.listFiles();

**for** (**int** i = 0;i < files.length;i ++) {

*deleteDir*(files[i]);

}

file.delete();

}

}

}

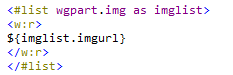
1. **封装好的代码**

文件/图片代码 ：FileUtil.java

压缩代码：CompressedUtil.java

1. **导出带图片的word**
2. **模板**

如果有多张图片，则应创建list循环导入图片



1. **后台代码**

//创建一个list，存放图片转码后的数据

List url=**new** ArrayList();

//获取图片地址的集合

List imgurl=(List) map3.get("G");

//如果图片地址的集合为空，则不显示图片

**if**(imgurl.size()>0){

**for** (**int** l = 0; l < imgurl.size(); l++) {

//全局变量imgnum,防止不同图片的wordml和src相同

**int** aaaa=++imgnum;

Map imgmapp=**new** HashMap();

//获取单张图片地址（电子中的地址）

String urls=imgurl.get(l)==**null**?"":imgurl.get(l).toString();

//如果地址为空，则不显示图片

**if**(urls.equals("")){

imgmapp.put("imgurl", "");

}**else**{

//将图片转码后存入集合中

String str="<w:pict>" +"<v:shape id='图片 3' o:spid='\_x0000\_i1026' type='#\_x0000\_t75'

style='width:210pt;" + "height:138.75pt;mso-position-horizontal-relative:page;mso -position-vertical-relative:page'>" +

"<v:fill o:detectmouseclick='t'/>" +

"<w:binData w:name='wordml://"+aaaa+".jpg'

xml:space='preserve'>" +ImagetoEncode(URL+urls)+

"</w:binData><v:imagedata src='wordml://"+aaaa+".jpg' o:title=''/>" +"</v:shape></w:pict>";

imgmapp.put("imgurl", str);

}

url.add(imgmapp);

}

}**else**{

Map imgmapp=**new** HashMap();

imgmapp.put("imgurl","");

url.add(imgmapp);

}

wgpart.put("img",url);

**3.图片转码**

//图片转码

**public** String ImagetoEncode(String path) {

InputStream in = **null**;

**byte**[] data = **null**;

**try** {

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

**if**(!file.exists()){

**return** " ";

}

in = **new** FileInputStream(path);

data = **new** **byte**[in.available()];

in.read(data);

in.close();

} **catch** (IOException e) {

e.printStackTrace();

}

BASE64Encoder encoder = **new** BASE64Encoder();

**return** encoder.encode(data);

}

注意：不同图片的w:name和src不能一样，否则图片无法显示

