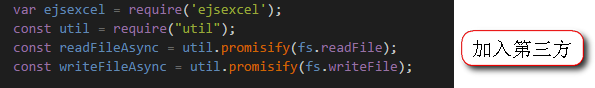
# 使用ejsExcel导出Excel&&压缩文件夹

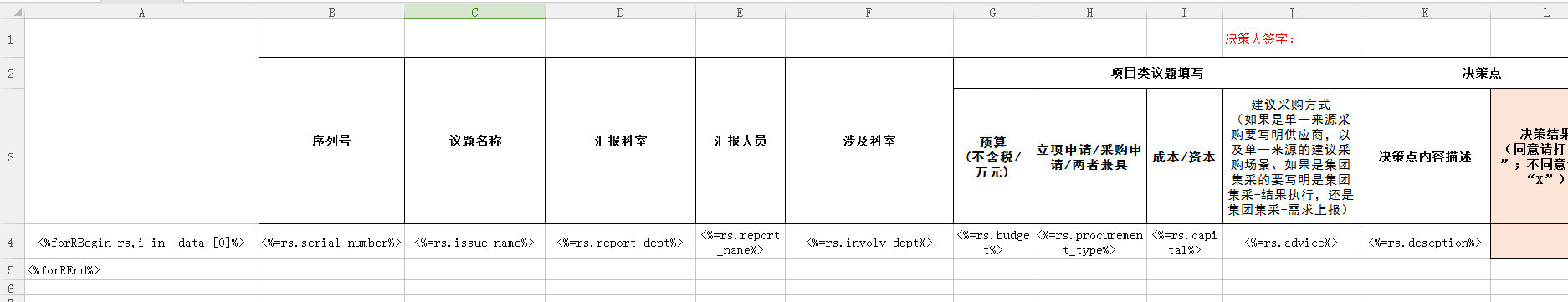
下载：npm install esjexcel;







模板样式：



<%forRBegin rs,i in \_data\_[0]%>：开始循环；因为数据封装为

[

[{},{},{},{}.....]

]；所以通过data[0]，即可取出所有数据，数组中套用数组;

<%forREnd%>：结束循环；

<%=rs.issue\_name%>：通过你封装的key来取值；

## 压缩文件夹：

1. 引入**archiver：**

**npm install archiver;**

**从数据库中获取数据，循环加入打包的文件：**

**列如result.rows[i].fileInfo[0].filePath =**

**public\\uploadFile\\meetFile\\upload\_b59da502a074638ea721ff16ef1e83f0.xlsx**

var archiver = require('archiver');

issueService.getIssueInfo(1, 100000, meet\_name, function (result) {

if (result.rows.length < 1) {

res.send("0");

} else {

//被打包文件

var files = [];

for (var i = 0; i < result.rows.length; i++) {

files.push(result.rows[i].fileInfo[0].filePath);

}

var zipPath = './public/uploadFile/temporaryFile/' + meet\_name + '.zip';

//创建一最终打包文件的输出流

var output = fs.createWriteStream(zipPath);

//生成archiver对象，打包类型为zip

var zipArchiver = archiver('zip');

//将打包对象与输出流关联

zipArchiver.pipe(output);

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

console.log(files[i]);

//将被打包文件的流添加进archiver对象中

zipArchiver.append(fs.createReadStream(files[i]), { 'name': files[i] });

}

//打包

zipArchiver.finalize();

res.send('/uploadFile/temporaryFile/' + meet\_name + '.zip');

}

})

1. zip-local
   * 1. npm install zip-local
     2. 学习地址：<https://www.npmjs.com/package/zip-local>