**Kettle Web组件实现说明书**

**By liuhaiyong whose githubid is wind1373290**

## 约定

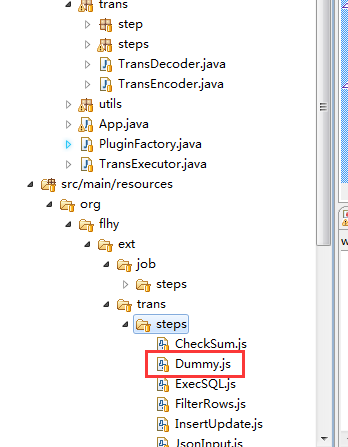
优先级由1开始，分5级，1为最高级，5为最低级。优先级越高的优先实现

二级目录（比如文件管理、脚本等）如果分配了人员，则二级目录下的所有组件归该人员开发，如果没有则需要制定三级目录的优先级和人员

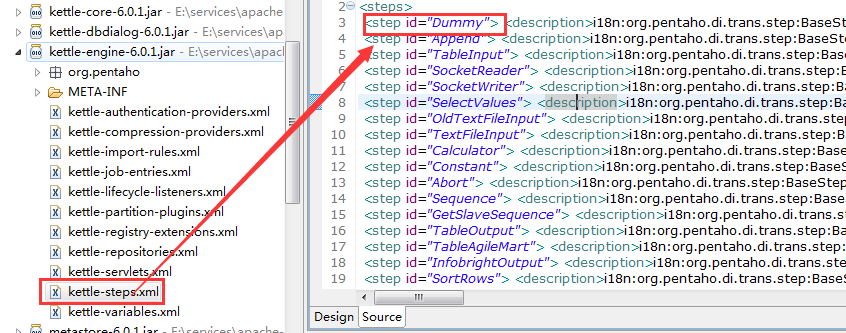
## 实现思路

组件的实现思路主要有写两部分的代码，一个是继承自org.flhy.ext.trans.step.AbstractStep（如果是Job的话，继承org.flhy.ext.job.step.AbstractJobEntry）的Java类，还有一个是前端展示的js类

Java类需要加上两个注解，其中Component注解的名字和JS文件保持一致，如下图，但是名字取名有讲究，不能随便取，如果是转换组件



的话，它的名字必须和kettle-steps.xml（如果是Job，就请打开kettle-job-entries.xml文件）文件中step节点的id保持一致，至于怎么定位你要实现的在文件中是哪个step，想点办法吧。



下面说一下具体的实现，以下我都以转换组件CheckSum为例，先实现encode在实现decode；encode方法代表从kettle组件中读取信息包装给前端，第一个参数必须强转类型，具体类型以开发的组件为准，强转之后，把以下代码复制过去吧

Document doc = mxUtils.*createDocument*();

Element e = doc.createElement("Step");

// TODO

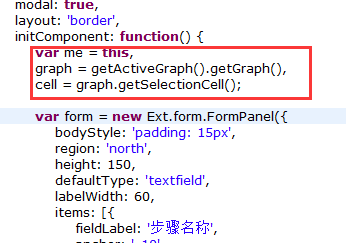
**return** e;

Element对象其实就是XML的一个节点，名字叫Step，输出就是<Step></Step>，我们要做的就是把配置信息以节点属性的方式组织上去，我觉得有必要对org.pentaho.di.trans.steps.checksum.CheckSumMeta（我怎么知道是这个类？，请查看kettle-steps.xml）类的getXml方法多读几遍，这样你就知道该往Element存入哪些配置信息了；decode方法代表了将前端配置转换成kettle对象的方法，包含四个参数；第一个参数应该熟读org.pentaho.di.trans.steps.checksum.CheckSumMeta的代码，只有搞清楚了loadXml方法才能知道需要往CheckSumMeta中设置哪些数据；第二个参数就是前端传过来的前端信息，它的具体内容在encode方法中找。

在来说说JS文件，JS文件是EXTJS代码，不会的话赶紧了解一下。JS代码会在界面上右键“编辑步骤”的时候调用，这个时候会弹出对话框。代码直接参考CheckSum.js吧，我说几个重点部分，

1、标红框的代码必须和文件名一致

2、下面截图中的红框部分最后的cell其实就是decode方法的第二个参数



3、其他问题在群里面交流哈

## 1、作业组件

### 1.1、文件管理(优先级1)

1.1.1、写入组件

1.1.2、创建文件

1.1.3、复制、移动结果文件

### 1.2、脚本(优先级2)

1.2.1、SQL

## 2、任务组件

### 2.1、文件传输(优先级2)

2.1.1、FTP上传

2.1.2、FTP下载

3、人员分派

|  |  |
| --- | --- |
| **组件** | **负责人员** |
| 1.1 | XXX |
| 2.1.1 | YYY |
| 2.2.2 | ZZZ |
|  |  |
|  |  |