集点网络

文件使用文档

[文档副标题]

guchongfeng

2017-11-23

目录

1.content.pattern.xml 文件说明 2
1.1 pattern 标签 2
1.2 column 标签 3
1.3 Matcher 标签 4
1.4 Replace 标签 6
1.5 Matcher 匹配例子 7
1.6 教程：有德招标（www.youde.net）招标公告抓取 10
2.packages.tablepattern.xml 文件说明 14
2.1 tablepattern 标签 15
2.2 tablematchers 标签、tableexpect 标签、singlematchers 标签和 singleexpect 标签 ... 15
2.3 tablecolumn 标签 16
2.4 headermatchers 标签和 valuematchers 标签 16
2.5 包组表格处理过程和采购货物表格处理过程 17
2.6 .project.pattern.xml 文件说明 19
2.7 教程:抓取有德招标的中标公告的包组 19
3 filter.xml 文件说明 23

1.content.pattern.xml 文件说明

类型:.content.pattern.xml 文件

作用:

用于抓取正文各个字段,(Tender,Bid,Corrections 实体的各个字段)

标签及属性说明:

1.1 pattern 标签

一个.content.pattern.xml 可以包含一个或者多个 pattern 标签，一个 pattern 标签对应一个实体，如图 1-1，<pattern id=”26”>对应的是有德招标（www.youde.net）的 Tender 实体（怎么判断稍后再做解释）,<pattern id=”399”>对应的是有德中标的 Bid 实体。

```
<?xml version="1.0" encoding="UTF-8"?>
<patterns>

    <!--广州有德招标-->
    <pattern id="26">

    <!--广州有德中标-->
    <pattern id="399">

</patterns>
```

图 1-1 文件结构

pattern 标签的属性列表

属性	说明	取值类型	必选	默认值	例子
id	用于标识 pattern，应该与数据库中的 gian_pattern 的 id 保持一致	整数	是		id=“26” id=”399”
filter	使用的过滤器列表	一个或者多个过滤器连接而成的字符串	否	none	filter=”tag-filter” filter=”tag-filter,trans-filter”

表 1-1 pattern 的属性列表

ps:关于过滤器使用，后面会有专门的说明。

1.2 column 标签

一个 pattern 标签下面会包含多个 column 标签，对应实体的某个字段，如图 1-2，pattern 标签包含了 7 个 column 标签，分别对应 Tender 实体的 AnnouncementDate，OpenDate，RegistrationDate，RegistrationEndDate，Price，BuyerName，BuyerMobile 等字段。

```
<!--广州有德招标-->
<pattern id="26">

    <!--发布时间-->
    <column name="AnnouncementDate" type="date">

    <!--开标时间-->
    <column name="OpenDate" type="date">

    <!--报名时间-->
    <column name="RegistrationDate" type="date">

    <!--报名截至时间-->
    <column name="RegistrationEndDate" type="date">

    <!--采购预算-->
    <column name="Price">

    <!--采购人名称-->
    <column name="BuyerName" filter="">

    <!--采购人联系人-->
    <column name="BuyerMobile">

</pattern>
```

图 1-2 Column 标签

Column 标签的属性列表

属性	说明	取值类型	必选	默认值	例子
name	字段名称，如 Tender 实体有个发布时间字段，对应的 setter 方法为 setAnnouncementDate()，那么这个字段对应的名称就是 set 后面的字符串 AnnouncementDate	字符串，实体字段名称	是		开标时间 name=” OpenDate” 项目金额 name=” Price” 采购人名称 name=” BuyerName”
type	字段类型，目前只需对日期类型的字段进行标注	“date”	否	不做标注	type=“date”

formatNumber	是否为格式化数字，如：金额，实体中的数据类型是 String，实际上是个格式化的数字如“100,000.00”，这个属性为“true”时，会将“100000.00”转换为“100,000.00”	“true”“false”	否	“false”	formatNumber=”true”
filter	使用的过滤器列表	一个或者多个过滤器连接而成的字符串	否	none	filter=”tag-filter” filter=”tag-filter,trans-filter”

表 1-2 column 的属性列表

ps:关于过滤器使用，后面会有专门的说明。

1.3 Matcher 标签

一个 column 标签下面包含多个 matcher 标签，一个 matcher 对应该字段下的一个匹配规则，如图 1-3，OpenDate 字段下有个 Matcher 标签，也就是两个匹配规则

```
<!--开标时间-->
<column name="OpenDate" type="date">
    <matcher regular="开标时间[^\d]{0,6}\d{4}年\d{2}月\d{2}日" format="yyyy年MM月dd日">
        <replace from="开标时间.*?(?=\d)"></replace>
    </matcher>

    <matcher regular="磋商时间：\s{0,2}\d{4}\s{0,4}年\s{0,4}\d{1,2}\s{0,4}月\s{0,4}\d{1,2}
        <replace from="磋商时间：\s{0,2}"></replace>
        <replace from="\s"></replace>
    </matcher>
</column>
```

图 1-3 Matcher 标签

Matcher 标签属性列表

属性	说明	取值类	必选	默认值	例子

		型			
regular	匹配内容的正则表达式	字符串(正则表达式)	是		regular="开标时间[^\d]{0,6}\d{4}年\d{2}月\d{2}日" regular="采购预算[\s\S]{0,100}?[0-9,\.]+元”
format	日期格式	字符串（日期格式）	对于 date 类型的字段是必选属性，对于非 date 类型的字段是无效属性	“yyyy-MM-dd”	format=”yyyy 年 MM 月 dd 日”
max	最大有效长度，用于检查匹配到的字符串是否有效	正整数	否	Integer.MAX_VALUE(整型数据的最大的数值)	max=”30”
min	最小有效长度，用于检查匹配到的字符串是否有效	整数	否	0	min=“5”
scale	对匹配到的数值进行缩放，如匹配到 300，scale=”100”，则最后输出是 300*100=30000	小数双精度类型	否	1	scale=”10000” scale=”0.01”
filter	使用的过滤器列表	一个或者多个过滤器连接而成的字符串	否	none	filter=”tag-filter” filter=”tag-filter,trans-filter”

表 1-3 Matcher 的属性列表

ps:关于过滤器使用，后面会有专门的说明。

1.4 Replace 标签

一个 matcher 标签下包含零个至多个 replace 标签，replace 标签对应一个替换规则，matcher 匹配到内容后，需要一般需要经过多次替换才能转换成我们想要的格式，例如，在抓取采购预算的时候，matcher 匹配出的字符串是”采购预算:1000 元”，我们最终想要的是“1000”,那么就将其他的字符替换成“”。

```
<!--采购预算xxx元-->
<matcher regular="采购预算[\s\S]{0,100}?[0-9,\.]+元" notag="false">
    <replace from="&lt;.+?&gt;"></replace>
    <replace from="[^0-9\.]"></replace>
</matcher>
```

图 1-4 replace 标签

Replace 标签属性列表

属性	说明	取值类型	必选	默认值	例子
from	被替换的字符串（正则表达式）	字符串（正则表达式）	是		替换非数字字符 from=”[^0-9\.]”
to	替换成为的字符串	字符串	否	空字符串“”	to=“-”

表 1-4 Replace 的属性列表

```
<!--广州有德招标-->
<pattern id="26">

    <!--发布时间-->
    <column name="AnnouncementDate" type="date">

    <!--开标时间-->
    <column name="OpenDate" type="date">
        <matcher regular="开标时间[^\d]{0,6}\d{4}年\d{2}月\d{2}日" format="yyyy年MM月dd日">
            <replace from="开标时间.*?(?=\d)"></replace>
        </matcher>

        <matcher regular="磋商时间：\s{0,2}\d{4}\s{0,4}年\s{0,4}\d{1,2}\s{0,4}月\s{0,4}\d{1,
            <replace from="磋商时间：\s{0,2}"></replace>
            <replace from="\s"></replace>
        </matcher>
    </column>

    <!--报名时间-->
    <column name="RegistrationDate" type="date">

    <!--报名截至时间-->
    <column name="RegistrationEndDate" type="date">

    <!--采购预算-->
    <column name="Price">

    <!--采购人名称-->
    <column name="BuyerName" filter="">

    <!--采购人联系人-->
    <column name="BuyerMobile">

</pattern>
```

图 1-5 一个完整的 Pattern

1.5 Matcher 匹配例子

Example 1:

使用 Matcher 找出人民币的金额

Matcher：

```
<matcher regular="人民币[\d,.]+元">
                <replace from=","></replace>
                <replace from="人民币"></replace>
                <replace from="元"></replace>
</matcher>
```

输入数据:

```
输入
项目金额人民币100,100.00元
```

匹配过程:

```
使用: matcher[ regular="人民币[\d,.]+元" ]
找到: [人民币100,100.00元]

使用: replace[ from="," to="" ]
原文: [人民币100,100.00元]
结果: [人民币100100.00元]

使用: replace[ from="人民币" to="" ]
原文: [人民币100100.00元]
结果: [100100.00元]

使用: replace[ from="元" to="" ]
原文: [100100.00元]
结果: [100100.00]

检测: 字符串长度为9,有效字符串，要求[0~2147483647]
输出: [100100.00]
```

Example2：

使用 Matcher 找出发布时间,匹配后的格式:MMddyyyy

Matcher:

```
<matcher regular="发布时间:[a-zA-Z]+ [a-zA-Z]{3} \d\d \d\d:\d\d:\d\d [a-zA-Z]+ \d{4}" format="MMddyyyy">
                    <replace from="Jan" to="01"></replace>
                    <replace from="Feb" to="02"></replace>
                    <replace from="Mar" to="03"></replace>
                    <replace from="Apr" to="04"></replace>
                    <replace from="May" to="05"></replace>
                    <replace from="Jun" to="06"></replace>
                    <replace from="Jul" to="07"></replace>
                    <replace from="Aug" to="08"></replace>
                    <replace from="Sep" to="09"></replace>
                    <replace from="Oct" to="10"></replace>
                    <replace from="Nov" to="11"></replace>
                    <replace from="Dec" to="12"></replace>
                    <replace from="发布时间:"></replace>
                    <replace from="[a-zA-Z]"></replace>
                    <replace from="\d\d:\d\d:\d\d"></replace>
                    <replace from=" "></replace>
</matcher>
```

输入数据:

```
输入
<div class="pub_note">
    【信息来源:广东省政府采购中心   发布时间:Mon Nov 13 10:47:36 CST 2017 】    访问次数:
   <span id="viewId"></span>   
   <a class="pub_note" onclick="window.print()">【我要打印】</ a>
   </div>
```

匹配过程:

使用: matcher[regular="发布时间:[a-zA-Z]+ [a-zA-Z]{3} \d\d \d\d:\d\d:\d\d [a-zA-Z]+ \d{4}"]
找到: [发布时间:Mon Nov 13 10:47:36 CST 2017]

使用: replace[from="Jan" to="01"]
原文: [发布时间:Mon Nov 13 10:47:36 CST 2017]
结果: [发布时间:Mon Nov 13 10:47:36 CST 2017]

使用: replace[from="Feb" to="02"]
原文: [发布时间:Mon Nov 13 10:47:36 CST 2017]
结果: [发布时间:Mon Nov 13 10:47:36 CST 2017]

使用: replace[from="Mar" to="03"]
原文: [发布时间:Mon Nov 13 10:47:36 CST 2017]
结果: [发布时间:Mon Nov 13 10:47:36 CST 2017]

.....

使用: replace[from="Oct" to="10"]
原文: [发布时间:Mon Nov 13 10:47:36 CST 2017]
结果: [发布时间:Mon Nov 13 10:47:36 CST 2017]

使用: replace[from="Nov" to="11"]
原文: [发布时间:Mon Nov 13 10:47:36 CST 2017]
结果: [发布时间:Mon 11 13 10:47:36 CST 2017]

使用: replace[from="Dec" to="12"]
原文: [发布时间:Mon 11 13 10:47:36 CST 2017]
结果: [发布时间:Mon 11 13 10:47:36 CST 2017]

使用: replace[from="发布时间:" to=""]
原文: [发布时间:Mon 11 13 10:47:36 CST 2017]
结果: [Mon 11 13 10:47:36 CST 2017]

使用: replace[from="[a-zA-Z]" to=""]
原文: [Mon 11 13 10:47:36 CST 2017]
结果: [11 13 10:47:36 2017]

使用: replace[from="\d\d:\d\d:\d\d" to=""]
原文: [11 13 10:47:36 2017]
结果: [11 13 2017]

使用: replace[from=" " to=""]
原文: [11 13 2017]
结果: [11132017]

检测: 字符串长度为8,有效字符串，要求[0~2147483647]
输出: [11132017]

1.6 教程：有德招标（www.youde.net）招标公告抓取

step1:添加对应 gain_pattern

在数据库添加对应的 gain_pattern，添加 gain_pattern 过程本文不作说明，请另找文档，添加完 gain_pattern 后，数据库就可以找到对应的 gain_pattern 的 id。如图 1-6，有德招标招标公告的 gain_pattern 的 id 为 26。知道对应的 id 我们就可以添加对应 pattern 了。关于 pattern 写在哪个文件上，.content.pattern.xml 类型的文件都放在 content 目录下，该目录下以.content.pattern.xml 结尾的文件都会被加载，为了方便区分，可以为一个网站新建一个文件，如图 1-7，网站的域名作为文件名，把这个网站的 pattern 都写到这个文件上。我们新建一个文件，然后写入如图 1-8 内容，pattern 的 id=26,说明跟 id=26 的 gain_pattern 关联，因此通过 id 去查 gain_pattern 表就可以知道 pattern 的实体类型和处理哪个网站，但是为了方便查看，请做好相关注释。

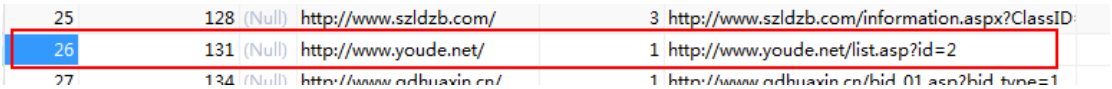

图 1-6 数据库中 gain_pattern

图 1-7 .content.pattern.xml 文件列表

```
<?xml version="1.0" encoding="UTF-8"?>
<patterns>
    <!--广州有德招标-->
    <pattern id="26">

    </pattern>

</patterns>
```

图 1-8 www.youde.net.content.pattern.xml 文件内容

Step2:进入后台测试界面尝试抓取数据

运行项目，然后在浏览器进入 http://localhost:8080/crawler/app/test.html，就可以进入抓取页面，输入抓取的正文链接，点击 go，可以查看页面内容，然后类型选择为招标，方案填 26（就是我们刚刚新建的 pattern）然后点击全部抓取，就会显示出各个字段的抓取结果

（注意:为什么 pattern 里面没有填任何规则，但却有很多字段抓取到了？答案：这个是在原有的代码上重新抓取的，那些字段都是原有的代码抓取到的），我们可以看到 announcementDate 这个字段没有抓取到，接下来将会通过在文件中新建 Matcher 来抓取这个字段。

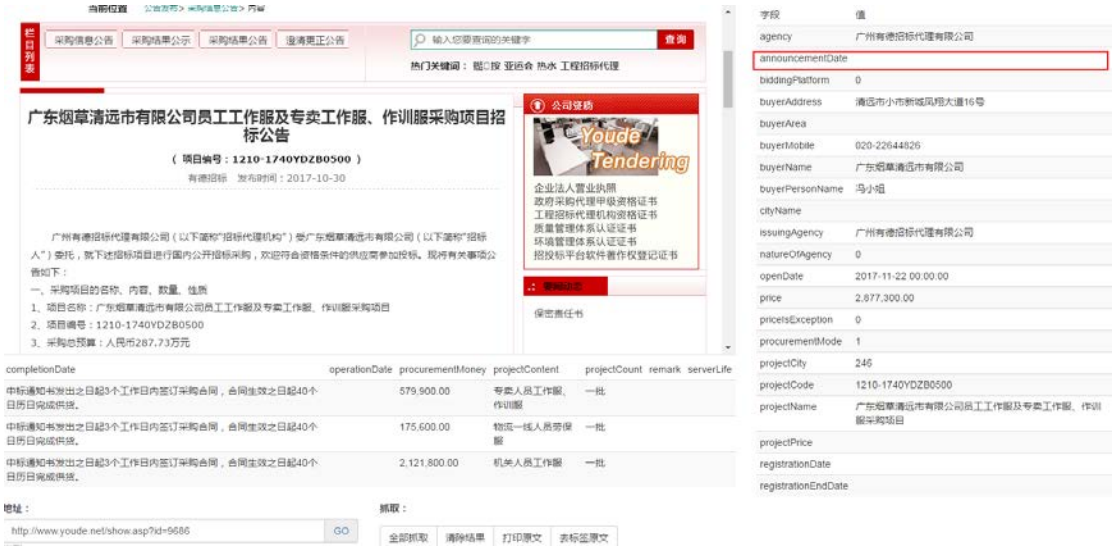

图 1-9 抓取界面

Step3:设计 Matcher

Step2 中发现 announcementDate 字段没有抓取到，首先通过打印原文按钮，查看我们截取到的未处理的文本内容，找到 announcementDate 在文本中的位置，根据文本设计出 Matcher，打开 MatcherTest 工具，测试 Matcehr 是否能够正确匹配，如图看到最终输出结果“2017-10-30”,说明正确匹配，Matcher 设计完成。

```
开始下载正文url:http://www.youde.net/show.asp?id=9686
<h1>广东烟草清远市有限公司员工工作服及专卖工作服、作训服采购项目招标公告</h1>
<center>
 <b> (  项目编号：1210-1740YDZB0500  ) </b>
</center>
<div class="info">
 有德招标   发布时间：2017-10-30
</div>
<div class="common">
 <p> </p>
 <p><br />      广州有德招标代理有限公司（以下简称“招标代理
```

图 1-10 原文内容

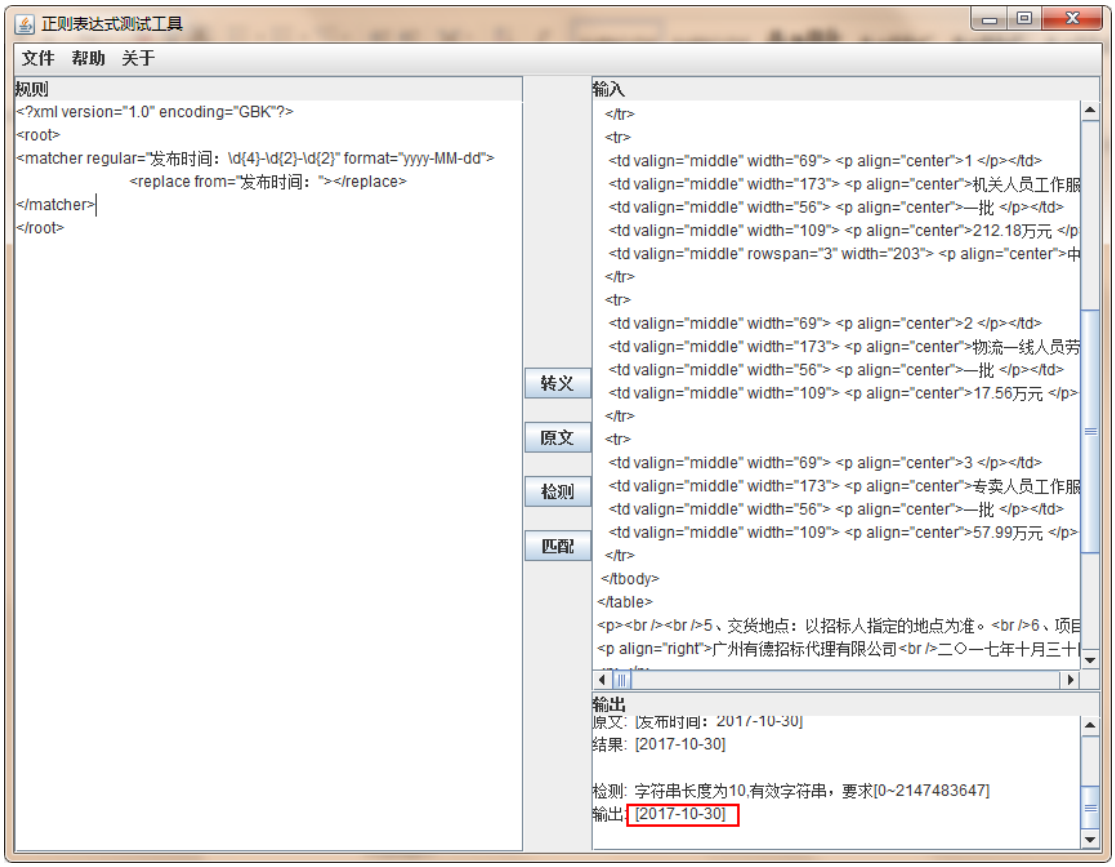

图 1-11 测试 Matcher

Step4：添加 Mather 到 pattern 中

Matcher 测试通过后，就可以将 Matcher 加入到文件中了，找到 id=26 的 pattern 标签，现在要抓取是 announcementDate 字段，找到 name=“AnnouncementDate”(注意:首个字母是大写的)的 column 标签,如果没有找到就新建一个 column 标签(注意：由于字段是日期类型，需要声明 type=”date”)。然后把刚刚设计好的 Matcher 复制到 column 标签下（”注意:要添加 format 属性到 Matcher 中”），记得按一下 Cltr+S 保存文件。

```
<!--广州有德招标-->
<pattern id="26">

    <!--发布时间-->
    <column name="AnnouncementDate" type="date">
        <matcher regular="发布时间：\d{4}-\d{2}-\d{2}" format="yyyy-MM-dd">
            <replace from="发布时间："></replace>
        </matcher>
    </column>

</pattern>
```

图 1-12 添加 Matcher

Step5:再次抓取测试 Matcher 是否生效

添加完 Mather 后，回到我们的抓取界面，首先点击重新加载文件按钮（要重新把文件加载内存中，不然文件没办法生效，当然重启整个项目也是可以的），加载完成后，点击全部抓取后就可以看到抓取结果，如图-13，我们准确抓取到发布时间 2017-10-30。

字段	值
agency	广州有德招标代理有限公司
announcementDate	2017-10-30 00:00:00
biddingPlatform	0
buyerAddress	清远市小市新城凤翔大道16号
buyerArea	
buyerMobile	020-22644826
buyerName	广东烟草清远市有限公司
buyerPersonName	冯小姐
cityName	

图 1-13 重新抓取的结果

2.packages.tablepattern.xml 文件说明

类型:.packages.tablepattern.xml

作用:用于抓取表格数据，主要是用于抓取包组表格和采购货物表格

```
<tablepatterns>
    <!--有德招标-->
    <tablepattern id="26" maxrow="25">

    <!--有德中标-->
    <tablepattern id="399" maxrow="25">

        <!--包组表格匹配条件-->
        <tablematchers>

        <!--排除条件-->
        <tableexpect>

        <!--采购内容表格匹配条件-->
        <singlematchers>

        <!--排除条件-->
        <singleexpect>
        <!--索引列-->
        <tablecolumn name="Index" type="string">

        <!--包名称-->
        <tablecolumn name="ProjectName" type="string">

        <!--中标公司名称-->
        <tablecolumn name="BidCompanyName" type="string">

        <!--中标公司地址-->
        <tablecolumn name="BidCompanyAddress" type="string">

        <!--包金额-->
        <tablecolumn name="BidMoney" type="string" formatNumber="true">
            <headermatchers>
                <matcher regular="成交金额（元）" max="10" min="0"></matcher>
                <matcher regular="中标金额（元）" max="10" min="0"></matcher>
                <matcher regular="中标金额" max="10" min="0"></matcher>
                <matcher regular="成交金额" max="10" min="0"></matcher>
            </headermatchers>

            <valuematchers>
                <matcher regular="[\d\.,]+" max="30" min="0">
                    <replace from="," ></replace>
                </matcher>
            </valuematchers>
        </tablecolumn>

    </tablepattern>

</tablepatterns>
```

图 2-1 一个典型.packages.tablepattern.xml 文件的结构

2.1 tablepattern 标签

tablepattern 标签对应一个网站一个实体，如图 2-1 中，id=26 的 tablepattern 是对应有德的招标的表格处理，实体类型是 TenderProject,id=399 的 tablepattern 是对应有德的中标的表格处理，实体类型是 BidProject。

tablepattern 的属性列表

属性	说明	取值类型	必选	默认值	例子
id	用于标识 tablepattern，应该与 gian_pattern 的 id 对应	整数	是		id=”26” id=”399”
maxrow	表格的最大行数，引入表格行数是为了处理整篇正文就是一个表格的情况	正整数	否	Integer.MAX_VALUE	maxrow=”25”
minrow	表格的最小行数	正整数	否	0	minrow=”2”

2.2 tablematchers 标签、tableexpect 标签、singlematchers 标签和 singleexpect 标签

tablematchers 标签、tableexpect 标签、singlematchers 标签和 singleexpect 标签分别用确定于包组表格、排除包组表格、确定采购货物表格和排除采购货物表格。目前这四个标签均没有属性，每个标签下都可以包含多个不带 Repalce 标签的 Matcher 标签(关于 Matcher 标签请查看第 1.3 节和 1.6 节),为什么不带 Repalce 标签，因为我们只需要匹配到表头内容就能判定这个表格是不是我们需要处理的表格，并不需要对匹配到内容进行额外的处理。Tablematchers 和 tableexpect 的区别是，tablematchers 匹配到就是确定表格是包组表格，而 tableeapect 则是反过来匹配到内容就确定他不是包组表格，这个两个标签共同作用，只有被 tablematchers 匹配到且没被 tableexcept 匹配到才能判定这个表格是包组表格。Singlematchers 和 singleexpect 类似，不一样是的这两个标签是匹配采购货物表格。

```
<!--包组表格匹配条件-->
<tablematchers>
    <matcher regular="包组"></matcher>
    <matcher regular="中标"></matcher>
</tablematchers>
```

图 2-2 中标包组表格匹配条件

2.3 tablecolumn 标签

tablecolumn 对应实体的字段，例如图 2-1 中，id=399 的 tablepattern 中有一个 name=”BidCompanyName”的 tablecolumn，id=”399”的 gain_pattern（在数据库的 gian_pattern 表中）的处理有德中标公告，那么这个 tablecolumn 就是对应 BidProject（中标的包组实体）的 bidCompanyName 字段。

属性	说明	取值类型	必选	默认值	例子
name	字段名称，命名规则可以参考第 1.2 的 column 标签	字符串(实体字段)	是		name=“BidCompanyName”
type	字段类型，目前只针对 date 类型的字段有效(参考第 1.2 节)	“date”	对于 date 类型的字段必选		type=“date”
formatNumber	是否为格式化数字，如：金额，实体中的数据类型是 String，实际上是个格式化的数字如“100,000.00”, 这个属性为“true”时，会将“100000.00”转换为“100,000.00”	“true” “false”	否	“false”	formatNumber=”true”

2.4 headermatchers 标签和 valuematchers 标签

一个 tablecolumn 标签下会包含一个 headermatchers 标签和一个 valuematchers 标签，这两个标签下面都包含多个 Matcher(关于 Matcher，请参考第 1.3~1.5 节)，headmatchers 下的 Matcer 用于匹配表头，valuematchers 下的 Matcher 用于处理表格单元内容，如果

valuematchers 下不包含 Matcher，则表示不对表格单元数据进行处理。另外要说明的是 scale 属性，headmatchers 下的 Matcher 的 scale 属性会对整列数据进行缩放，例如，表头是通过 scale=10000 的 matcher 匹配到的，那么在这种情况，这一列数据都会被乘以 10000，而 valuematchers 下的 Matcher 的 scale 属性只会对匹配到的单元格，进行缩放。

2.5 包组表格处理过程和采购货物表格处理过程

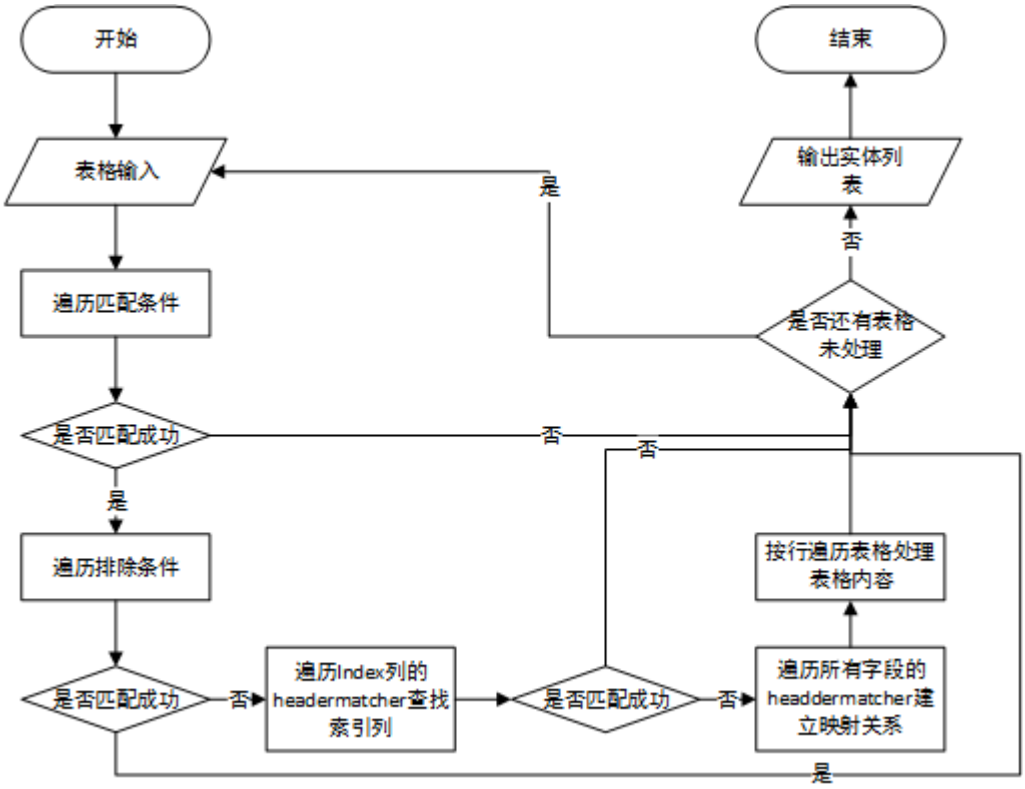

图 2-3 包组表格匹配流程图

遍历匹配条件：在这个阶段就是遍历 tablematchers 下的 matcher 去匹配表头那一行数据，如果能够匹配成功，则进入下一阶段，否则，认为这个表格不是包组表格，直接跳过，处理下一个表格。

遍历排除条件：在这个阶段遍历 tableexpect 下的 matcher 去匹配表头那一行数据，如果匹配成功，则表示这个表格应该被排除，直接跳过，处理下一个表格，否则，进入下一个阶段。

查找索引列：前面两个步骤已经可以确定这个表格是包组，在这个步骤中，将在当前处理表格中，找出表格的索引列，一般这一列是包组，包组号列或者类似的列，索引列用于确定，表格行与包组之间的关系，也就是说，一个包组可能包含多行内容，通过索引列表格单元的

rowspan 属性可以确定包组的行数，另外索引列还用于多个表格之间的关联，可能包组内容会分布在多个表格中，例如，表格 1 包含包组 1 包组 2 包组 3 的金额，表格 2 包含包组 1 包组 2 包组 3 的中标供应商名称，在程序中会每个包组需要建立一个实体，只要能够准确找出索引列和序号，程序就能将多个表格的数据存储到对应的实体中。

建立字段与表格列之间的映射关系：找到索引列之后，之后开始建立字段与列之间的映射关系了，为什么在这之前一定要先找到索引列？答案：因为没有找出索引列就没有办法确定，哪些行属于哪个包组，也没有办法处理多个包组表格的合并，所以准确地找到索引列非常重要，如果你写的规则没有办法找到，你就要想办法优化了（PS：会不会没有索引列的情况？只要你能看出哪些行属于哪个包组的，就存在索引列，找索引可以参考你是靠什么去分辨行是属于哪个包组的）。建立实体字段与表格列之间的映射，例如，表格有一列表头是“中标金额”，说明这一列的内容是包组的金额，你要把这一列跟实体的 Price 字段起来，通过遍历 Price 字段的 headermatcher 匹配到“中标金额”，就能建立起映射关系，通过遍历所有字段的 headermatcher 就可以获得一张字段跟表格列之间的映射表。

处理表格内容：经过前面两个步骤，程序已经知道，表格的哪一列是索引列，以及表格列跟实体哪个字段对应，可以开始处理表格内容了，首先通过 Index 列的 valuematcher 可以从索引列中处理出索引，第一遇到这个索引号的时候就新建一个实体，如果以前已经遇到过这个索引号，则取出之前建立好的实体，然后把这一行的内容写入到对应字段，有些内容时需要做一些处理才能写入到实体中的，如金额，表格内容是”人民币 300,000 元”,我们需要通过 valuematcher 将”人名币””元”和逗号去掉，如果表格内容是”人民币 30 万元”，我还需要声明 scale=”10000”，将 30 转换为 300000。最后是关于同一个包组行合并策略，对于字符类型，多行内容会以逗号作为分隔符进行连接，如果是数值类型的将会相加，其他类型取第一个值。

关于，采购货物表格的处理过程，如果你已经理解了包组表格的处理过程，那么理解采购货物表格的处理过程就很容易了，我们只要把所有表格都当做是属于同一个包组就行了,这是索引列是无效的，也不需要找索引列，就是匹配到表格中的行都写入到同一个实体中（包组中是按照索引号，写入到不同的实体）。

2.6 .project.pattern.xml 文件说明

类型:.project.pattern.xml

作用：这个文件的跟.content.pettern.xml 完成是一样的，只不过这个文件用于在全文范围抓取包组的内容，在只有一个包组的是时候会尝试全文去抓取包组实体的各个字段，如果没有匹配到表格，那么包组实体的内容全部通过全文抓取获得，因此是不会出现没有包组的情况。

2.7 教程:抓取有德招标的中标公告的包组

正文链接: http://www.youde.net/show.asp?id=9632

图 2-4 中两个表格都包含了包组内容，所以这两个表格都要处理，处理出来的实体个数应该是 2。这里我们要抓取的列包括中标供应商，地址，主要中标名称，单价（元），服务期，而且要能正确把两个表格关联起来。

四、采购方式：公开招标

五、中标供应商

包组号	中标供应商	法定代表人	地址
1	广东和平国际旅行社有限公司	李元生	广州市经济开发区青年路98号507室
2	广州凤凰国际旅行社有限公司	李明	广州市越秀区新河浦86号

六、报价明细

包组号	主要中标标的名称	规格型号	数量	单价（元）	服务期
1	小学生外出社会实践活动	/	1	¥334,669.00	合同签订至合同完成
2	中学生拓展活动	/	1	¥295,542.00	/

图 2-4 要抓取包组表格

Step1:添加包组表格匹配条件

为了能够找到这个两个表格，我们可以用”中标”去确定这两个表格，然后排除条件不写，当然，这样写去匹配其他页面时肯定会出现很多判断错误的情况，这里只当做测试用。通过添加一个 Matcher 到 tablematchers 中添加匹配条件。

Step2:创建索引列匹配规则

这里选包组号那一列作为索引列，表头匹配规则为“包组号”,处理规则“\d*”，也就是提取数字，如图 2-5。

```
<!--索引列-->
<tablecolumn name="Index">
    <headermatchers>
        <matcher regular="包组号"></matcher>
    </headermatchers>
    <valuematchers>
        <matcher regular="\d+"></matcher>
    </valuematchers>
</tablecolumn>
```

图 2-5 索引列

Step3:其他列的匹配规则

在这里除了金额那一列需要对表格单元进行额外的处理外，其他的列都不需要进行额外处理。

```
<!--地址-->
<tablecolumn name="BidCompanyAddress">
    <headermatchers>
        <matcher regular="地址"></matcher>
    </headermatchers>
    <valuematchers>
    </valuematchers>
</tablecolumn>
<!--公司名称-->
<tablecolumn name="BidCompanyName">
    <headermatchers>
        <matcher regular="中标供应商"></matcher>
    </headermatchers>
    <valuematchers>
    </valuematchers>
</tablecolumn>
<!--金额-->
<tablecolumn name="BidMoney" formatNumber="true">
    <headermatchers>
        <matcher regular="单价"></matcher>
    </headermatchers>
    <valuematchers>
        <matcher regular="[\d\.,]+">
            <replace from=","></replace>
        </matcher>
    </valuematchers>
</tablecolumn>
<!--名称-->
<tablecolumn name="ProjectName">
    <headermatchers>
        <matcher regular="主要中标标的名称"></matcher>
    </headermatchers>
    <valuematchers>
    </valuematchers>
</tablecolumn>
<!--服务期限-->
<tablecolumn name="ServerLife">
    <headermatchers>
        <matcher regular="服务期"></matcher>
    </headermatchers>
    <valuematchers>
    </valuematchers>
</tablecolumn>
```

图 2-6 tablecolumn

Step4:测试

打开测试界面，重新加载文件，然后点击全部抓取，查看抓取结果。

五、中标供应商

包组号	中标供应商	法定代表人	地址
1	广东和平国际旅行社有限公司	李元生	广州市经济开发区青年路98号507室
2	广州凤凰国际旅行社有限公司	李明	广州市越秀区新河浦86号

六、报价明细

包组号	主要中标标的名称	规格型号	数量	单价（元）	服务期
1	小学生外出社会实践活动	/	1	¥334,669.00	合同签订至合同完成
2	中学生拓展活动	/	1	¥295,542.00	/

保密确认书

推荐内容

- 广东省财政厅预算绩效管理信息系...
- 广东烟草清远市有限公司员工工作...
- 广东省公安厅2017-120项目成交公.
- 广东省公安厅2017-130项目成交公.
- 广州铁路公安局2017-5项目中标公..
- 广东烟草韶关市有限公司南雄市分...
- 广东出入境检验检疫局技术中心试...
- 广东工贸职业技术学院平安校园监...

友情链接

bidCompany	bidCompanyAddress	bidCompanyMobile	bidCompanyName	bidMoney	operationDate	projectName	serverLife	success
	广州市越秀区新河浦86号		广州凤凰国际旅行社有限公司	295,542.00		中学生拓展活动	/	1
	广州市经济开发区青年路98号507室		广东和平国际旅行社有限公司	334,669.00		小学生外出社会实践活动	合同签订至合同完成	1

图 2-7 抓取结果

3 filter.xml 文件说明

作用：用于在匹配正文前对正文进行过滤，去除标签、转义字符替换、连续空白字符合并等工作都可以放到过滤器里面做。

图 3-1 中 tag-filter 就是标签过滤器，而 trans-filter 则是转义过滤器。图 3-2 时使用 tag-filter,trans-filter,space-filter 过滤后的正文，多个过滤器可以组合使用，会按照字符串的顺序一次调用过滤器，进行过滤，注意，过滤器的顺序不同，可能导致过滤的结果不同，例如，自定义一个 atag-filter 用于过滤掉正文中所有链接，那么 atag-filter 应该放在 tag-filter 之前，否则，标签都已经被过滤掉了，无法过滤 a 标签。

了解过滤器的效果之后，我们来看一下怎么把过滤器应用到 pattern 文件中，pattern 标签，column 标签，matcher 标签都有一个 filter 属性，这个属性指明了要使用的过滤，如 filter=“tag-filter,trans-filter”，在匹配正文之前会首先依次用 tag-filter,trans-filter 对正文进行过滤。pattern 标签的作用范围时整个 pattern,即所有字段匹配之前都会先过滤，依次类推，column 中的 filter 的作用范围是这个字段，matcher 是用这个 matcher 去匹配的时候。那么三者之间关系是怎样的呢？优先级 matcher>column>pattern，优先级高的会覆盖掉优先级低的，就是说 pattern 声明使用 filter=“tag-filter”，而 matcher 上的 filter=”trans-filter”，则在用这个 matcher 去匹配正文之前只会调用 trans-filter，而对于其他没有声明 filter 的 Matcher 则会调用 tag-filter。假设，pattern 上用了 tag-filter，但是在某个 matcher 中需要匹配标签，可以声明 filter=“none”,把 tag-filter 覆盖,none 这个 filter 是不做任何过滤操作的。

```xml
<?xml version="1.0" encoding="UTF-8"?>
<filters>

    <filter name="none"></filter>

    <filter name="tag-filter">
        <replace from="&lt;br/\s*?&gt;" to=" "></replace>
        <replace from="&lt;.*?&gt;" to=""></replace>
    </filter>

    <filter name="trans-filter">
        <replace from="&amp;quot;" to="&quot;"></replace>
        <replace from="&amp;amp;" to="&amp;"></replace>
        <replace from="&amp;lt;" to="&lt;"></replace>
        <replace from="&amp;gt;" to="&gt;"></replace>
        <replace from="&amp;nbsp;" to=" "></replace>
    </filter>

    <filter name="spaceline-filter">

    <filter name="atag-filter">

    <filter name="space-filter">

</filters>
```

图 3-1 过滤器

图 3-2 使用过滤器之后的文本