**修改记录：**

|  |  |  |
| --- | --- | --- |
| **时间** | **修改内容** | **版本** |
| 2015-03-27 | 1. 修改了排名取值的内容，showType=”text”,first移到value里作为对value数据的进一步描述; 2. 排序增加了random关键字，格式random(3)，任取3行; 3. 增加了地图的描述，新加了map\_pts定义 | 1.0.1 |
| 2015-04-01 | 1、根据上周5讨论的结果，对标准进行修改，包括：  ·统一并简化D标签的属性设置  2、加入新的功能，包括：  ·地图处理的标签  ·一个画布多个图形的功能  3、在修饰中，加入了percent(num) | 1.1.1 |
| 2015-06-19 | 1、调整了table标签，在param中加入了过滤列显示的内容 | 1.1.2 |
| 2015-07-13 | 1、加入针对html的style功能  2、加入了全局参数，应对分类过多的问题 | 1.1.2 |

**备忘：**

**1-下一个版本把“图形”的过滤完善；**

**2-目前的地图只有连点的功能，还没有画线和画面的功能，以后再说；**

**3-目前规划的4个图形显示的功能还比较死板，可通过扩充decorateView来实现，这个下一个版本争取规定好。**

**备忘：**

**1-本文档中，若出现删除线(一般在表格中)表示此功能已经实现（至少实现了一个版本）；**

**一、源起**

分析平台中，用来显示数据结果的结构（也可称为报告），此结构暂命名为：report。

也可叫做分析报告结果报告。

**二、思路**

1-用json格式来描述报告，里面混杂html；

2-各种图的显示，在这里定义，但不在这里实现，定义包括，但不限于——

a)图的种类：饼图柱图等；

b)图的操作：攥取——这个本版本不处理

3-根据这个报告生成实际的显示内容。

4-这个报告是异步的，这里的异步指的是报告中用到的数据是异步的

5-报告要有结构，类似word的目录结构。

**三、流程图**

处理流程如下：



1-分析过程生成report，这个文件属于REPORT-file范畴

2-analReportJson总体框架一次性生成(或异步?)

**四、报告展现**

对report的展现，可以有多种形式，目前看有两种：

a)html5，用于在界面展示——目前主要是PC

b)word，用于导出内容

将来可能还有

d)移动端展示——肯能与html5类似，或以此为基础

e)flash等

**注意：**

1、实现展示时要考虑数据是否准备完成

2、本期先不实现这些报告展现功能，主要就实现html的样式

**五、报告结构规定**

**1-数据格式规定**

1-带下划线的属性，是本标准的默认属性，其他属性是用户定义的（建议用户自定义属性不带有下划线）；

2-由\_HEAD属性、\_DATA属性和\_REPORT属性组成主体结构；

3-分割符定义，若一个属性内，有不同意义的域的定义，用”::”分开，这种分割符比较好区分（注意：与C++中的域分割符相同），在本规范中value,decorateView中都会用到，在value中是数据与过滤函数之间的区分，在decoreateView中是串替换部分和扩展参数部分之间的区分；

4-新括号：当括号嵌套级别多后，””和’’不够了，用^代替双引号，用~代替单引号，如：

|  |
| --- |
| content:"<div style=‘font-height:bold; font-size:18px;’>3、人员地图分析</div><br/>  <d did=‘2’ showType=‘map\_pts’  param=‘{^X^:^coordX^,^Y^:^coordY^,^Z^:^coordZ^,^mapType^:^BAIDU^}’ value=‘quotas[0]’/>" |

请参看样例。

**2-主体结构解释**

|  |  |  |  |
| --- | --- | --- | --- |
| **标签** | **分类** | **格式说明** | **补充** |
| \_HEAD | **标准：必须** | 头信息，包括如下内容：报告名称、报告编码、创建时间，描述等 | ~~本期使用~~ |
| \_DLIST | **标准：可选** | 报告需要的所有数据，是一个数组，数组中的元素为数据来源。可以没有，则这个报告就是一篇文章 | ~~本期使用~~ |
| \_REPORT | **标准：必须** | 报告内容部分由树形结构组成，叶结点是seg(段)类型 | ~~本期使用~~ |

**a)-** **"\_HEAD"：**

这个是头信息，包括如下内容：报告名称、报告编码、创建时间，描述。

***例子：***

|  |
| --- |
| "\_HEAD":{  "\_id":"abcd-efg-higklmn",  "\_reportType":"test",  "\_reportName":"test",  "\_code":"T.TEST::0002",  "\_cTime":"-480::1421744444796"  "\_desc":"测试报告"  "\_param":"{'representFilter':{ 'maxCatogorys': '10'}}"  } |

***解释：***

|  |  |  |  |
| --- | --- | --- | --- |
| **标签** | **分类** | **格式说明** | **补充** |
| \_id | **标准：必须** | 36位标准UUID  此编码是实体码，类似于类的实例 | ~~本期使用~~ |
| \_reportName | **标准：必须** | 报告名称 | ~~本期使用~~ |
| \_reportType | **标准：可选** | 报告分类，目前没有明确的定义 | ~~本期使用~~ |
| \_code | **标准：必须** | 报告，此编码分为大类编码：大写的以.隔开的名称，<::>后是小类型编号，采用数字，也可以是UUID  此编码是类型码，类似于类 | ~~本期使用~~ |
| \_cTime | **标准：必选** | 数据创建时间 | ~~本期使用~~ |
| \_desc | **标准：可选** | 数据信息的描述。 | ~~本期使用~~ |
| \_param | **标准：可选** | 全局参数 | 本期使用 |

***关于\_param：***

全局参数用来控制整个报告的解析过程，这是一个json串，以便转换为对象。

目前仅想到maxCatogorys，展示时对过多的分类进行处理。

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **分类** | **说明** | **补充** |
| representFilter | **标准：可选** | 一级参数：表现层过滤 | 本期使用 |
| └maxCatogorys | **标准：可选** | 最大分类数，默认值是5 | 本期使用 |

另外：是否可以考虑，没一个D标签都有这个参数，这样能够使得控制更灵活。等实现时再做具体的处理。

**b)-"\_DLIST"：**

此报告需要的所有数据，是一个数组，数组中的元素为数据来源。

包括数据的编号、获取数据的url、若数据是jsonD格式，指明jsonD格式的编码。可选

***例子：***

|  |
| --- |
| "\_DLIST":[  {"\_id":"0", "\_ url":"getJsonD.do?datafile=/??/.jsond"," \_ jsonDCode":"SP.TEAM-00007", "\_jsonDId":""},  {"\_id":"1"," \_url":"getJsonD.do?datafile=/??/anal\_quota\_716-446655440000.jsond"," \_ jsonDCode":"SP.TEAM-00009", "\_jsonDId":""}  ] |

***数据来源标签解释：***

|  |  |  |  |
| --- | --- | --- | --- |
| **标签** | **分类** | **格式说明** | **补充** |
| \_id | **标准：必须** | 为数据编号，报告中的<d>标签会用到，只在本报告中使用的id，是整数，从0开始 | ~~本期使用~~ |
| \_url | **标准：必须** | 指明数据的来源，可以是一个服务过程的结果，也可以直接是一个json文件的地址。  目前系统提供的是 | ~~本期使用~~ |
| \_jsonDCode | **标准：可选** | jsonD格式的编码，若数据不是jsonD格式，也可以不设置此属性。 | ~~本期使用~~ |

**c)-"\_REPORT"：**

报告内容部分由树形结构组成，叶结点是seg(段)类型。

***例子：***

|  |
| --- |
| \_REPORT:[  {  id:"seg1",  name:"上传数据CCC",  title:"<div style="font-height:bold;font-size:24px;’>上传数据￥￥￥￥</div>",  subSegs:[  {  id:"seg1\_1",  name:"结构分析",  content:"<div style=‘font-height:bold; font-size:18px;’>1、结构分析</div><br/>  <d did=‘1’ showType="table" value=‘excelMdmArray[0]’/><br/>"  },{  id:"seg1\_2\_1",  name:"单项指标分析",  content:"<div style=‘font-height:bold; font-size:18px;’>2、单项指标分析</div><br/>  <div style=‘font-height:bold;’>“<d did=‘2’ showType=‘value’ value=‘mdName’/>”[<d did=‘2’ showType=‘value’ value=‘quotas[0].titleName’/>]</div>指标分析：<br/>  “<d did=‘2’ showType=‘value’ value=‘quotas[0].titleName’/>”中  大多为{<d did=‘2’ showType=‘value’ value=‘quotas[0].tableData.tableBody[0].category’/>}占<d did=‘2’ showType=‘value’ value=‘quotas[0].tableData.tableBody[0].percent’/>%，  {<d did=‘2’ showType=‘value’ value=‘quotas[0].tableData.tableBody[1].category’/>}占<d did=‘2’ showType=‘value’ value=‘quotas[0].tableData.tableBody[1].percent’/>%，具体分析数据如下：<br/>  <d did=‘2’ showType=‘table’ value=‘quotas[0]’ titleDcrt=‘quotas[0].titles’/>  <d did=‘2’ showType=‘pie’ param=‘{‘xAxis’:’category’, ‘yAxis’:’num’}’ value=‘quotas[0]’ decorateView=‘#category#, #percent(num)#’/>  <br/>  <div style=‘font-height:bold; ‘>“<d did=‘2’ showType=‘value’ value=‘mdName’/>”[<d did=‘2’ showType=‘value’ value=‘quotas[1].titleName’/>]</div>指标分析：<br/>  新加测试showType=“line”：<br/><d did=‘2’ showType=‘line’ param=‘{‘xAxis’:’category’, ‘yAxis’:’num’}’ value=‘quotas[0]’ decorateView=‘#category#, #percent(num)#’/>  新加测试showType=“bar”：<br/><d did=‘2’ showType=‘bar’ param=‘{‘xAxis’:’category’, ‘yAxis’:’num’}’ value=‘quotas[0]’ decorateView=‘#category#, #percent(num)#’/>  测试降序：<d did=‘2’ showType=‘text’ value=‘quotas[1]::first(3|num)’ decorateView=‘{#category#}占#percent#(num)%::{suffix:^、^}’/>  测试升序：<d did=‘2’ showType=‘text ‘ value=‘quotas[1]::!first(3|num)’ decorateView=‘{#category#}占#percent#(num)%’/>"  }, {/\*地图上以点状显示人员坐标信息\*/  id:"seg1\_3",  name:"人员地图分析",  content:"<div style=‘font-height:bold; font-size:18px;’>3、人员地图分析</div><br/>  /\*人员 坐标分布 mapType:使用哪家地图，AGIS|BAIDU|GAODE\*/  <d did=‘2’ showType=‘map\_pts’ param=‘{^X^:^coordX^,^Y^:^coordY^,^Z^:^coordZ^,^mapType^:^BAIDU^}’ value=‘quotas[0]’/>"  }  ]  }  ] |

**注意，Content中的内容html中的内容都用单引号。**

***seg(段)内元素标签解释：***

|  |  |  |  |
| --- | --- | --- | --- |
| **标签** | **分类** | **格式说明** | **补充** |
| Id | **标准：必须** | 段id，注意此id仅作为区分标记，不带有任何树结构信息 | ~~本期使用~~ |
| Name | **标准：可选**  必须任选其一 | 段名称，可用此生成树的名称 | ~~本期使用~~ |
| title | 段显示名称，可用此生成树的名称，也用此生成实际报告显示时的段标题。若有Name树名称以Name为准。  **此属性内容是html代码** | ~~本期使用~~ |
| 以上两个属性必须要有一个被指定：  ·若只指定了name，则实际显示中，段标题也按照name设置；  ·若只指定了title，则实际显示中，树结点名称也按照title设置； | | | |
| content | **标准：可选** | 段内容，其中包括d标签，一般叶结点此标签都要设置内容  **此属性内容是html代码** | ~~本期使用~~ |
| subSegs | **标准：可选** | 下级段数组 | ~~本期使用~~ |

**3-<d>标签的详细解释**

用于显示数据内容的标签，与\_DATA中的数据配合，目前包括：

表格、图形(饼图、柱图、雷达图、折线图)、文字内容(简单取值，排名取值)、地图(点图、密度图、其他专题图)。

***例子：***

|  |
| --- |
| <d did="1" showType="value" value="excelMdmArray[0].sheetInfo.name"/>  <d did="2" showType="text" value="quotas[1]::!first(3|num)" decorateView="{#category#}占#percent(num)#%::{suffix:’、’}"/>  <d did="2" showType="text" value="quotas[1]::first(3|num)" decorateView="{#category#}占#percent(num)#%"/>  <d did="1" showType="table" value="excelMdmArray[0] "/>  <d did="2" showType="pie" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]"  decorateView="#category#, #percent(num)#"/>  <d did="2" showType="bar" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]"  decorateView="#category#, #percent(num)#"/>  <d did="2" showType="line" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]"  decorateView="#num#"/>  <d did="2" showType="radar" param="{‘xAxis’:’category’ , ‘yAxis’:’num’ , ‘max’:’100’}" value="quotas[0]"  decorateView="#num#"/>  <d did="2" showType="map\_pts" param="{‘X’:’coordX’, ‘Y’:’coordY’ ,’Z’:’coordZ’,’mapType’:’BAIDU’}"  value="quotas[0]" /> |

***解释：***

|  |  |  |  |
| --- | --- | --- | --- |
| **标签** | **分类** | **格式说明** | **补充** |
| did | **标准：必须** | 与\_DATA标签中的“数据来源”中的\_id相对应 | ~~本期使用~~ |
| showType | **标准：必须** | 数据显示形式，目前支持：value-值;table-表格;pie-饼图;bar-柱图;line-折线;text-文本;map-地图(map\_pts- 点状;) | ~~本期使用~~ |
| param | **标准：可选** | 对showType进行具体的说明，可包括：  图元素和数据的对应关系，  数据在显示时的模式——如地图的格式  Param中应该是一个json对象 | ~~本期使用~~ |
| value | **标准：必须** | 获取的数据及数据的过滤过则，获取数据分两种情况：1-直接数据；2-表数据，对表数据来说才有过滤规则。过滤与数据之间用“：：”分割。 | ~~本期使用~~ |
| decorateView | **标准：可选** | 对显示的修饰说明 | ~~本期使用~~ |
| htmlExt | **标准：可选** | 对html类型展现端的扩展 | 本期使用 |

**注意：**

**1-标签的顺序最好也按照上表进行处理**

**2- lable/data，若图进行90度翻转，怎样定义？应该定义在param中**

**3-每类标签的说明，请参见后面的附录**

**\*4-删除了radom的过滤值：若引入此过滤条件，会造成两次报告的显示结果不同，这不应是一个预期的结果。**

**a.1)-简单取值(value)：**

***例子：***

|  |
| --- |
| <d did=“1” showType=“value” value=“excelMdmArray[0].sheetInfo.name”/> |

***解释：***

|  |  |
| --- | --- |
| did=1 | \_DLIST数据列表中\_id=1的jsonD数据 |
| showType="value" | 显示类型—“简单取值” |
| value  ="excelMdmArray[0].sheetInfo.name" | 取得jsonD中\_DATA对象中  excelMdmArray[0].sheetInfo.name的值  (excelMdmArray数组第0个记录sheetinfo对象中name属性的值) |
| decorateView  ="<span style=‘color:red’>#value#</span>" | 红色显示数值结果，  可选 |
| htmlExt="{'style':'width:100; '}" | 主要设置容器div的style可选 |

***数据类型：***

任何数据

***说明：***

1-此类标签在显示中，直接显示数据的值。

2-此标签没有param属性。

3-数据获取无过滤

**a.2)-文本块(text)：**

***例子：***

|  |
| --- |
| <d did="2" showType="text" value="quotas[1]::!first(3|num)" decorateView="{#category#}占#percent(num)#%"/>  <d did=“2” showType="text" value=“quotas[1]::first(3|num )" decorateView="{#category#}占#percent(num)#%"/>  <d did=“2” showType="text" value=“quotas[1]::first(3|num )" decorateView="{#category#}占#percent(num)#%::{suffix:’、’}"/> |

***解释：***

|  |  |
| --- | --- |
| did=2 | \_DLIST数据列表中\_id=2的jsonD数据 |
| showType="text" | 显示类型—“文本块” |
| value  ="quotas[1]"  ="quotas[1]::first(3|num)"  ="quotas[1]::!first(3|num)" | 取得jsonD中\_DATA对象中quotas[1]的数据表  (quotas数组第1个记录，此记录是表格数据)  1)若无过滤，取所有数据  2)first-对数据中的num列，从大到小排序，取前3列;  3)!first-对数据中的num列，从小到大排序，取前3列; |
| decorateView  ="{#category#}占#percent(num)#%"  ="{#category#}占#percent(num)#%::{‘suffix’:’、’}" | 显示修饰：按照修饰定义，循环显示数据。  ##之间是列名，其他字符按照原样显示，如排序后的数据为：   |  | | --- | | {category:"北京", num:"42", percent(num):"21"},  {category:"四川", num:"8", percent(num):"4"},  {category:"其他", num:"2", percent(num):"1"}, |   **则first(3|num)显示为：**   |  | | --- | | {北京}占21%、{四川}占8%、{其他}占2% | |

***数据类型：***

表数据

***说明：***

1-此类标签在显示中，按照Value的设置获取数据，并按照decorateView的方式进行显示。

2-此标签没有param属性。

3-数据获取必须有过滤

另：则通过复杂的decorateView可以实现表格。

**问题：**

1-排序取值默认确定是从大到小排序的，请参考“对jsonD格式中表格数据的规定”；

2-若列表中没有百分比列，则解析时自动计算百分比（按照获取数据过滤列定义），而且当且仅当过滤的函数是first时；

**b)-列表取值(table)：**

***例子：***

|  |
| --- |
| <d did="1" showType="table" value="excelMdmArray[0]" decorateView=”::”/> |

***解释：***

|  |  |
| --- | --- |
| did=1 | \_DLIST数据列表中\_id=1的jsonD数据 |
| showType="table" | 显示类型—“表格” |
| value="excelMdmArray[0]" | 取得jsonD中\_DATA对象中excelMdmArray[0]的数据表  (excelMdmArray数组第0个记录，此记录是表格数据) |
| param="{‘分类’:’category’, ‘数量’:’count’}" | 列过滤，这里指出的列用于显示，注意，这里的key-value的映射关系与表数据中的titles是相反的，但与其它类型标签的param是一致的 |
| htmlExt="{'style':'width:100; '}" | 主要设置容器div的style可选 |

***数据类型：***

表数据

***说明：***

1-此标签没有decorateView属性；

2-数据获取可以有过滤的函数first和!first，但不建议使用；

**c.1)-饼图(pie)：**

***例子：***

|  |
| --- |
| <d did="2" showType="pie" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]"  decorateView="#category#, #percent(num)#"/> |

***解释：***

|  |  |
| --- | --- |
| did=2 | \_DLIST数据列表中\_id=2的jsonD数据 |
| showType="pie" | 显示类型—“饼图” |
| value="quotas[0]" | 取得jsonD中\_DATA对象中quotas[0]的数据表  (quotas数组第0个记录，此记录是表格数据) |
| param=  "{'xAxis': 'category', 'yAxis': 'num'}" | ·xAxis-横坐标/分类维度：对应quotas[1]的数据表中dataList中category列，一般这样的列是离散的，表现为字符串或时间点，从意义上看是分类和时间；  ·yAxis-纵坐标/数值维度：对应quotas[0]的数据表中dataList中num列，一般这样的列是数值的；  **每扇形，规定死：在扇形上(或外区域)显示xAxis的值，并用括号括起显示yAxis的值。** |
| decorateView  ="#category#, #percent(num)#" | 怎样显示饼图中鼠标滑过每个扇形上的提示  显示表格中的"category"列，中间逗号分隔，之后显示百分比列percent(num) |
| htmlExt="{'style':'width:100; '}" | 主要设置容器div的style可选 |

***数据类型：***

表数据

***说明：***

1-decorateView在这里用于规定，鼠标滑过时显示的内容，这里先规定死，若没有定义此属性，则不显示任何内容。

**c.2)-柱图(bar)：**

***例子：***

|  |
| --- |
| <d did="2" showType="bar" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]"  decorateView="#category#, #percent(num)#"/> |

***解释：***

|  |  |
| --- | --- |
| did=2 | \_DLIST数据列表中\_id=2的jsonD数据 |
| showType="bar" | 显示类型—“柱图” |
| value="quotas[0]" | 取得jsonD中\_DATA对象中quotas[0]的数据表  (quotas数组第0个记录，此记录是表格数据) |
| param=  "{‘xAxis:’category’, ‘yAxis’:’num’}" | ·xAxis-横坐标/分类维度：对应quotas[1]的数据表中dataList中category列，一般这样的列是离散的，表现为字符串或时间点，从意义上看是分类和时间；  ·yAxis-纵坐标/数值维度：对应quotas[0]的数据表中dataList中num列，一般这样的列是数值的；  **每个柱，规定死：在横坐标对应位置显示xAxis的值，在柱的上面显示yAxis的值。** |
| decorateView  ="#category#, #percent(num)#" | 怎样显示柱图中鼠标滑过每个柱时的提示  显示表格中的"category"列，中间逗号分隔，之后显示百分比列percent(num) |
| htmlExt="{'style':'width:100; '}" | 主要设置容器div的style可选 |

***数据类型：***

表数据

***说明：***

1-decorateView在这里用于规定，鼠标滑过时显示的内容，这里先规定死，若没有定义此属性，则不显示任何内容。

**c.3)-折线图(line)：**

***例子：***

|  |
| --- |
| <d did="2" showType="line" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]" decorateView="#num#"/> |

***解释：***

|  |  |
| --- | --- |
| did=2 | \_DLIST数据列表中\_id=2的jsonD数据 |
| showType="line" | 显示类型—“折线图” |
| value="quotas[0]" | 取得jsonD中\_DATA对象中quotas[0]的数据表  (quotas数组第0个记录，此记录是表格数据) |
| param=  "{‘xAxis:’category’, ‘yAxis’:’num’ }" | ·xAxis-横坐标/分类维度：对应quotas[1]的数据表中dataList中category列，一般这样的列是离散的，表现为字符串或时间点，从意义上看是分类和时间；  ·yAxis-纵坐标/数值维度：对应quotas[0]的数据表中dataList中num列，一般这样的列是数值的；  **每个点，规定死：在横坐标对应位置显示xAxis的值，在点的上面显示yAxis的值。** |
| decorateView  ="#num#" | 怎样显示折线图中鼠标滑过每个点时的提示  显示表格中的"num"列的值 |
| htmlExt="{'style':'width:100; '}" | 主要设置容器div的style可选 |

***数据类型：***

表数据

***说明：***

1-decorateView在这里用于规定，鼠标滑过时显示的内容，这里先规定死，若没有定义此属性，则不显示任何内容。

**c.4)-雷达图(radar)：**

***例子：***

|  |
| --- |
| <d did="2" showType="radar" param="{‘xAxis’:’category’, ‘yAxis’:’num’ ,’max’:’100’}"  value="quotas[0]" decorateView="#num#"/> |

***解释：***

|  |  |
| --- | --- |
| did=2 | \_DLIST数据列表中\_id=2的jsonD数据 |
| showType="radar" | 显示类型—“折线图” |
| value="quotas[0]" | 取得jsonD中\_DATA对象中quotas[0]的数据表  (quotas数组第0个记录，此记录是表格数据) |
| param=  "{‘xAxis:’category’, ‘yAxis’:’num’,’max’:’100’}" | ·xAxis-维度坐标：每辐轴半径对应的解释文字，对应quotas[1]的数据表中dataList中category列，一般这样的列是离散的，表现为字符串，从意义上看是分类；  ·yAxis-数值维度：辐轴半径上的数值点，不同维度对应的图形区域的顶点。对应quotas[0]的数据表中dataList中num列，一般这样的列是数值的；  ·max-雷达图辐轴半径最大值：辐轴半径最大值；  **每辐轴半径，规定死：雷达辐轴半径显示xAxis的值，辐轴半径的数值点上面显示yAxis的值。** |
| decorateView  ="#num#" | 怎样显示雷达图中鼠标滑过每个点时的提示  显示表格中的"num"列的值 |
| htmlExt="{'style':'width:100; '}" | 主要设置容器div的style可选 |

***数据类型：***

表数据

***说明：***

1-decorateView在这里用于规定，鼠标滑过时显示的内容，这里先规定死，若没有定义此属性，则不显示任何内容。

2-对于雷达图，param定义得更复杂些，其中定义了辐轴半径的最大值max。**目前不支持雷达图中每辐轴半径最大值不同，雷达辐轴半径最大值必须相同。**

**3-若数据维度(分类/行数)小于等于2，则无法显示雷达图；**

**d.1)-地图\_画点(map\_pts)：**

***例子：***

|  |
| --- |
| <d did="2" showType="map\_pts" param="{‘X’:’coordX’, ‘Y’:’coordY’ ,’Z’:’coordZ’,’mapType’:’BAIDU’}"  value="quotas[0]"/> |

***解释：***

|  |  |
| --- | --- |
| did=2 | \_DLIST数据列表中\_id=2的jsonD数据 |
| showType="map\_pts" | 显示类型—“地图-画点” |
| value="quotas[0]" | 取得jsonD中\_DATA对象中quotas[0]的数据表  (quotas数组第0个记录，此记录是表格数据) |
| param=  "{‘X’:’coordX’, ‘Y’:’coordY’ ,  ‘Z’:’coordZ’,’mapType’:’BAIDU’}" | ·mapType-地图类型：主要是指地图数据的类型，根据类型不同，在显示时可能需要进行不同的调整；  ·X-经度：根据mapType的不同，处理可能不同。  对应quotas[0]的数据表中dataList中coordX列是经度；  ·Y-纬度：根据mapType的不同，处理可能不同。  对应quotas[0]的数据表中dataList中coordY列是纬度；  ·Z-高程：根据mapType的不同，处理可能不同，**对于地图，此项可能不需要。** |
| decorateView  ="<tr>  <td>姓名</td><td>#xm#</td>  <td>身份证</td><td>#sfz#</td>  <td>性别</td><td>#sb#</td>  <td>城市</td><td>#city#</td>  </tr>::{‘envelopeType’:’table’}" | 怎样显示地图中鼠标滑过每个点时的提示  显示表格中的"num"列的值 |
| htmlExt="{'style':'width:100; '}" | 主要设置容器div的style可选 |

***数据类型：***

表数据

***说明：***

1-decorateView在这里用于规定，鼠标滑过时显示的内容，其中::tab说明按照table显示这个内容，若没有定义此属性，则不显示任何内容。

**2-param中可能要加一个扩展：defineUrl，指向一个文件，此文件为json格式，用于详细定义图的不同元素，此文件中的元素应该也能够定义在param中。这个文件中的格式今后再定义。**

**附：1-value说明：**

***a)-value分类：***

value用于定义显示的数据，包括两个部分：

1-jsonD中的数据标识：按照json的结构，用“. ”的方式，直接定位到数据，这里的数据又可以分为两种方式：

·值；

·json对象：若是json对象则一定是表结构数据，见“JsonD中表格数据的规定”；

2-数据过滤函数

***b)-数据过滤函数***

只对表结构数据进行过滤，过滤情况见下表：

|  |  |  |
| --- | --- | --- |
| **过滤函数** | **意义** | **参数说明** |
| ::first(3|num) | 按照num正序(从大到小)取3行数据 | 用|进行分割  3——必须，指定取几行数据；  num——以表中的num列为排序指标 |
| ::!first(3|num) | 按照num倒序(从小到大)取3行数据 |

**说明：**

1-若表中没有num列，则按照原表的顺序取值，first(3)前3列，!first(3)后三列；

2-若表中的sort属性所指定的列于num相同，则无需排序，取表中数据即可，若不同，则需要在生成report时，按照num列对表进行排序，之后取值。

3-只有text/table可以有过滤，其他图形不建议有过滤，**若“图形”的4个标签有过滤，则除按序取得行数据外，其他行数据都归入”其他分类”，否则会造成数据不完整。（下一个版本把过滤的功能都加入）**

***c)-过滤函数和标签类对应关系***

|  |  |  |  |
| --- | --- | --- | --- |
| **参数**  **showType** | | ::first(3|num) | ::!first(3|num) |
| 按num正序(从大到小)取3行数据 | 按num倒序(从小到大)取3行数据 |
| **a.文本** | **1)value** | X | X |
| **2)text** | √可选 | √可选 |
| **b.表格** | **1)table** | √可选 | √可选 |
| **c.图形** | **1)pie** | 不建议 | 不建议 |
| **2)bar** | 不建议 | 不建议 |
| **3)line** | 不建议 | 不建议 |
| **4)radar** | 不建议 | 不建议 |
| **d.地图** | **1)map\_pts** | X | X |

**附：2-param说明：**

此属性是showType的一个附属说明，其指明：

1-与图相关的基础参数，记为关系A；

2-图中标准元素和数据之间的对应关系，记为关系B；

***a)-param和标签类对应关系***

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Param** | **说明** |
| **a.文本** | **1)value** | X |  |
| **2)text** | X |  |
| **b.表格** | **1)table** | 可选 | 列筛选，例：param="{‘籍贯’:’category’, ‘数量’:’num’}"  筛选出要显示的列，若不设置，显示数据中的所有列 |
| **c.图形** | **1)pie** | √ | 例：param="{‘xAxis’:’category’, ‘yAxis’:’num’}"  ·xAxis-横坐标/分类维度；  ·yAxis-纵坐标/数值维度；  **每扇形，规定死：在扇形上(或外区域)显示xAxis的值，并用括号括起显示yAxis的值。** |
| **2)bar** | 例：param="{‘xAxis’:’category’, ‘yAxis’:’num’}"  ·xAxis-横坐标/分类维度；  ·yAxis-纵坐标/数值维度；  **每个柱，规定死：在横坐标对应位置显示xAxis的值，在柱的上面显示yAxis的值。** |
| **3)line** | 例：param="{‘xAxis’:’category’, ‘yAxis’:’num’}"  ·xAxis-横坐标/分类维度；  ·yAxis-纵坐标/数值维度；  **每个点，规定死：在横坐标对应位置显示xAxis的值，在点的上面显示yAxis的值。** |
| **4)radar** | 例：param="{‘xAxis’:’category’, ‘yAxis’:’num’ ,’max’:’100’}"  ·xAxis-维度坐标：每辐轴半径对应的解释文字；  ·yAxis-数值维度：辐轴半径上的数据点，不同维度对应的图形区域的顶点；  ·max-雷达图辐轴半径最大值；  **每辐轴半径，规定死：雷达辐轴半径边缘显示xAxis的值，辐轴半径点的上面显示yAxis的值，雷达辐轴半径的最大值为。** |
| **d.地图** | **1)map\_pts** | √ | 例：param="{‘X’:’coordX’, ‘Y’:’coordY’ ,’Z’:’coordZ’,’mapType’:’BAIDU’}  ·mapType-地图类型；  ·X-经度；  ·Y-纬度；  ·Z-高程； |

***b)-param中各参数说明***

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **意义** | **值范围** | **可能的标签类** |
| xAxis | 横坐标：其值应是离散的，是分类列，不同标签意义也不尽相同 | 表数据中的离散列，如category列 | bar、line、pie、radar |
| yAxis | 横坐标：其值大多是数值，不同标签意义也不尽相同 | 表数据中的离散列，如num列 | bar、line、pie、radar |
| max | 雷达图辐轴半径最大值 | 一般取yAxis中最大值的110%，来可能更美观 | radar |
| X | 经度 | 表数据中的经度列 | map\_pts |
| Y | 纬度 | 表数据中的纬度列 | map\_pts |
| Z | 高程 | 表数据中的纬度列，可选，即可以没有高程(本版本无) | map\_pts |
| mapType | 地图坐标类型 | BAIDU——百度坐标  **请原总补齐** | map\_pts |

**注：**百分比列——比如为percent(num)则可能需要计算，有可能出现在yAxis参数中，计算规则如下：

·若数据中有为percent(num)的列，则直接取值；

·否则，看列表中是否有num列，若有，计算其百分比，并显示计算后的结果。

这里需注意：计算时总数应是sum(num)的值，其中num为value中定义的数据表中的num列的和，而不管——

A)value中是否有过滤函数；（即使为value：~~::first(3|num)，也不是只计算3行数据的合，而是整个数据表中所有列的合）

B)原始数据中的总和；(如果原始数据为100行，而JsonD中dlist的数据是10行，这里也按照10行计算，因为到这一步，report只和jsonD打交道，不知道原始数据是什么情况)

***c)-param中各参数和标签类对应关系***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **参数**  **showType** | | xAxis  横坐标 | yAxis  纵坐标 | max  最大值 | X  经度 | Y  纬度 | Z  高程 | mapType  地图类型 |
| 关系B | 关系B | 关系A | 关系B | 关系B | 关系B | 关系A |
| **a.文本** | **1)value** | X | X | X | X | X | X | X |
| **2)text** | X | X | X | X | X | X | X |
| **b.表格** | **1)table** | X | X | X | X | X | X | X |
| **c.图形** | **1)pie** | √ | √ | X | X | X | X | X |
| **2)bar** | √ | √ | X | X | X | X | X |
| **3)line** | √ | √ | X | X | X | X | X |
| **4)radar** | √ | √ | √ | X | X | X | X |
| **d.地图** | **1)map\_pts** | X | X | X | √ | √ | 可选 | 可选 |

**param中可以加一个扩展：defineUrl，指向一个文件，此文件为json格式，用于详细定义图的不同元素，此文件中的元素应该也能够定义在param中。这个文件中的格式今后再定义。这个主要为复杂定义预留空间，目前看地图最有可能用到。**

**附：3-decorateView说明：**

用于定义数据的显示方式。

***a)-两部分：***

可把decorateView分为两部分：

A)串替换部分：此部分将按照原样显示，并按照规则，替换##中括起来的部分，此部分若对应的是表数据，将根据数据的条数进行循环输出(对网页，是输出到html串中)。

B)扩展参数部分：定义更复杂的情况，采用json方式进行定义

这两部分之间用“::”分割开来

***b)-显示效果：***

可分为两类：

A)对于value/text，直接文本显示；

B)对于图形和地图，是tooltip的内容。

***c)-对于串替换部分的#percent(num)#说明：***

·若数据中有为percent(num)的列，则直接取值；

·否则，看列表中是否有num列，若有，计算其百分比，并显示计算后的结果。

这里需注意：计算时总数应是sum(num)的值，其中num为value中定义的数据表中的num列的和，而不管——

A)value中是否有过滤函数；（即使为value：~~::first(3|num)，也不是只计算3行数据的合，而是整个数据表中所有列的合）

B)原始数据中的总和；(如果原始数据为100行，而JsonD中dlist的数据是10行，这里也按照10行计算，因为到这一步，report只和jsonD打交道，不知道原始数据是什么情况)

***d)-decorateView中扩展参数说明：***

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | | **意义** | **可能的标签类** |
| 行循环 | suffix | 一行数据后的字符串，若没有，在text中默认为“；” | text |
| prefix | 一行数据前的字符串 | Text |
| 数据块 | envelopeType | 封装类型，目前有：  ·table，用table标签封装；  ·div，用div标签封装； | 图/Map |
| segSuffix | 数据块前缀（本版本不实现） | 图/Map |
| segPrefix | 数据块后缀（本版本不实现） | 图/Map |

***e)-decorateViewg各部分和标签类对应关系***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **串**  **替换** | **扩展参数** | | | | | **显示**  **效果** |
| 行循环 | | 数据块循环 | | |
| 前缀 | 后缀 | 块封装类型 | 块前缀 | 块后缀 |
| **Suffix** | **prefix** | **envelopeType** | **segSuffix** | **segPrefix** |
| **a.文本** | **1)value** | √ | 可选 | 可选 | X | X | X | 直显 |
| **2)text** | √ | 可选 | 可选 | 可选 | 可选 | 可选 | 直显 |
| **b.表格** | **1)table** | X | X | X | X | X | X | tooltip |
| **c.图形** | **1)pie** | √ | 可选 | 可选 | 可选 | 可选 | 可选 | tooltip |
| **2)bar** | √ | 可选 | 可选 | 可选 | 可选 | 可选 | tooltip |
| **3)line** | √ | 可选 | 可选 | 可选 | 可选 | 可选 | tooltip |
| **4)radar** | √ | 可选 | 可选 | 可选 | 可选 | 可选 | tooltip |
| **d.地图** | **1)map\_pts** | √ | 可选 | 可选 | 可选 | 可选 | 可选 | tooltip |

***f)-举例说明***

用decorateView实现表格功能：

decorateView="<tr><td>#xm#</td><td>#sfz#</td><td>#sb#</td><td>#city#</td></tr>

::{‘segPerfix’:’<table><tr><th>姓名</th><th>身份证</th><th>性别</th><th>城市</th></tr>‘

, ‘segSuffix’:’</table>‘}"

**附：4-对不同展现的特殊规定**

不同展现适用于不同的客户端(浏览程序)，目前主要为：html/word/pdf。针对不同的客户端程序制定相适应的规则，以更好的展现数据报告。

***a)HTML格式***

在每个d标签中扩充属性，此类属性，只有用html解析时才有意义。

用htmlExt属性描述，格式采用json串，便于扩展与转换。

**附：5-标签类型及标签属性对照表：**

（本表中√项是必须有的内容）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **标签**  **showType** | | 指向dlist中  数据的Id | 显示类型 | 类型参数 | 数据获取 | | 显示修饰 | **不同显示类型参数**  **(目前仅html)** | **说明** | **数据类型** |
| **Did** | **showType** | **param** | **value** | **是否过滤** | **decorateView** |
| **a.文本** | **1)value** | √ | √  (value) | X | √ | X | 直接显示数据值 | 可选 | 直接显示数据值 | 任意类型数据 |
| **2)text** | √ | √  (text) | X | √ | first  !first | 显示文本段 | 可选 | 显示文本段 | 表数据 |
| **b.表格** | **1)table** | √ | √  (table) | 可选 | √ | first  !first | 按表格显示数据 | 可选 | 按表格显示数据 | 表数据 |
| **c.图形** | **1)pie** | √ | √  (pie) | √ | √ | X? | 显示饼图 | 可选 | 显示饼图 | 表数据 |
| **2)bar** | √ | √  (bar) | √ | √ | X? | 显示柱图 | 可选 | 显示柱图 | 表数据 |
| **3)line** | √ | √  (line) | √ | √ | X? | 显示折线图 | 可选 | 显示折线图 | 表数据 |
| **4)radar** | √ | √  (radar) | √ | √ | X? | 显示雷达图 | 可选 | 显示雷达图 | 表数据 |
| **d.地图** | **1)map\_pts** | √ | √  (map\_pts) | 可选 | √ |  |  |  |  | 表数据 |

**4-<ds>标签的详细解释(本期不实现)**

图形组：本标签用来实现在一个画布上同时显示多个图形，图形可以是同类型的，也可以是不同类型的，它们的合法组合方式。

目前如下（按类正则表达式方式给出）：

1-line(1..n)|bar(1..n)。

如：

|  |
| --- |
| <ds>  <d did="1" showType="bar" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]" decorateView="#num#"/>  <d did="2" showType="line" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]" decorateView="#num#"/>  <d did="2" showType="line" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[1]" decorateView="#num#"/>  </ds> |

目前规定，这类图形的横坐标一定单位要相同，纵坐标可以不同。

解释：

此图是3个图形的叠加，分别是：

·柱图，从quotas[0]中取出的分类和数值之间关系的图；

·折线图，从quotas[0]中取出的分类和数值之间关系的图；（也就是一个纬度用两个图形展示）；

·折线图，从quotas[1]中取出的分类和数值之间关系的图；

2-radar(1..n)。

如：

|  |
| --- |
| <ds>  <d did="1" showType="radar" param="{‘xAxis’:’category’, ‘yAxis’:’num’,’max’:’100’}"  value="quotas[0]" decorateView="#num#"/>  <d did="1" showType="radar" param="{‘xAxis’:’category’, ‘yAxis’:’num’,’max’:’100’}"  value="quotas[1]" decorateView="#num#"/>  </ds> |

解释：

此图是2个雷达图叠加，分别是：

·从quotas[0]中取出分类和数值信息并组成第一个雷达图

·从quotas[1]中取出分类和数值信息并组成第二个雷达图

目前规定，同雷达图的辐轴半径维度相同，辐轴半径最大值相同。

**同画布图形，可能对画布还有具体的定义，这个功能后续版本再处理，本版本不处理，但一定是定义在<ds>中的。**

**六、JsonD中表格数据的规定**

为配合report中对二维表的展示，规定——有表数据的jsonD必须按如下方式定义：

**例子：**

|  |
| --- |
| "tableDataDemo":{  "columnName":"col\_4",  "titleName":"籍贯",  "**tableData**"**:{**  **"tableName":**"籍贯数量分布情况",  **"sort":**{"sortCol":"num","shotType":"1"},  **"titles":**{"category":"籍贯","num":"数量","percent(num) ":"百分比"},  **"dataList":[**  {"category":"其他", "num":"2", "percent(num)":"1"},  {"category":"北京", "num":"42", "percent(num)":"21"},  {"category":"四川", "num":"8", "percent(num)":"4"},  {"category":"江苏", "num":"6", "percent(num)":"3"},  {"category":"河北", "num":"36", "percent(num)":"18"},  {"category":"天津", "num":"27", "percent(num)":"13.5"},  {"category":"上海", "num":"25", "percent(num)":"12.5"},  {"category":"河南", "num":"28", "percent(num)":"14"},  {"category":"辽宁", "num":"12", "percent(num)":"6"},  {"category":"浙江", "num":"8", "percent(num)":"4"},  {"category":"黑龙江", "num":"6", "percent(num)":"3"}  ]  }  } |

**说明：**

1-所有表数据的内容应该是json中的对象，其中要包括tableData属性，此属性也是一个json对象，这个对象中的dataList属性是真正的二维表数据。

2-百分比列要说明是那列的百分比，这个规定是通过生成report.json的程序(或人)来遵守的。

**解释：**

|  |  |  |  |
| --- | --- | --- | --- |
| **标签** | **分类** | **格式说明** | **补充** |
| tableData | **标准：必须** | 所有二维表数据都要有这个属性 | 本期使用 |
| ┝tableName  │ | **标准：可选** | 表名称，一般是中文名称  是tableData的子属性 | 本期使用 |
| ┝sort  │  │  │ | **标准：可选** | 排序字段说明：  ·sortCol说明表按照那列进行排序  ·sortType说明排序的方式=1大到小;=2小到大  是tableData的子属性 | 本期使用 |
| ┝titles  │  │ | **标准：必须** | 对列数据进行说明，目前还只有名称，后续可考虑把数据类型加入  是tableData的子属性 | 本期使用 |
| └dataList | **标准：必须** | 实际的二维表数据，是一个json数组，  是tableData的子属性 | 本期使用 |

**应用举例：**

***1、first标签：***

<d did=“2” showType=“!first(3|num)” value=“quotas[1]” decorateView=“{#category#}占#percent#%”/>

value=“quotas[1]”：数据必须符合“表格数据”

!first(3|num)：倒排序，按照num列，此时显示数据时，就要看“表格数据”是否有sort属性，并根据属性确定是否需要前端排序(若列表是不排序列表)，和取数据的方式

***2、bar标签：***

<d did="2" showType="bar" param="{‘xAxis’:’category’, ‘yAxis’:’num’}" value="quotas[0]"

decorateView="#category#, #percent(num)#"/>

param="{‘xAxis’:’category’, ‘yAxis’:’num’}"：“表格数据”中dataList数组中元素的category属性作为横坐标，“表格数据”中dataList的num列作为纵坐标；

value="quotas[0] "：数据必须符合“表格数据”格式

decorateView="#category#, #percent(num)#"：提示中显示“分类，num的百分比”；

**七、样例**

|  |
| --- |
| {  \_HEAD:{  reportName:"“XXXSSS”分析报告"  },  \_DATA:[/\*data问题，一定要注重规范\*/  {\_id:1, \_url:"http://localhost:8080/sa/jsonD/getJsonD.do?uri=/demo/templetDemo/metedataInfo550E8400-E29B-11D4-A716-446655440000.jsond", \_jsonD\_code:"SP.TEAM-00007"},  {\_id:2, \_url:"http://localhost:8080/sa/jsonD/getJsonD.do?uri=/demo/templetDemo/anal\_quota\_716-446655440000.jsond"}  ],  \_REPORT:[  {  id:"seg1",  name:"上传数据CCC",  title:"<div style="font-height:bold;font-size:24px;’>上传数据￥￥￥￥</div>",  subSegs:[  {  id:"seg1\_1",  name:"结构分析",  content:"<div style=‘font-height:bold; font-size:18px;’>1、结构分析</div><br/>  <div style=‘font-height:bold;’>“<d did=‘1’ showType=‘value’ value=‘excelMdmArray[0].sheetInfo.name’/>”页签(sheet<d did=‘1’ showType=‘value’ value=‘excelMdmArray[0].sheetInfo.index’/>)</div>为新增结构，元数据结构分析结果如下：<br/>  <d did=‘1’ showType="table" value=‘excelMdmArray[0]’/><br/>  <div style=‘font-height:bold;’>“<d did=‘1’ showType=‘value’ value=‘excelMdmArray[1].sheetInfo.name’/>”页签(sheet<d did=‘1’ showType=‘value’ value=‘excelMdmArray[1].sheetInfo.index’/>)</div>为原有结构，元数据用原有结构。"  },{  id:"seg1\_2\_1",  name:"单项指标分析",  content:"<div style=‘font-height:bold; font-size:18px;’>2、单项指标分析</div><br/>  <div style=‘font-height:bold;’>“<d did=‘2’ showType=‘value’ value=‘mdName’/>”[<d did=‘2’ showType=‘value’ value=‘quotas[0].titleName’/>]</div>指标分析：<br/>  “<d did=‘2’ showType=‘value’ value=‘quotas[0].titleName’/>”中  大多为{<d did=‘2’ showType=‘value’ value=‘quotas[0].tableData.tableBody[0].category’/>}占<d did=‘2’ showType=‘value’ value=‘quotas[0].tableData.tableBody[0].percent’/>%，  {<d did=‘2’ showType=‘value’ value=‘quotas[0].tableData.tableBody[1].category’/>}占<d did=‘2’ showType=‘value’ value=‘quotas[0].tableData.tableBody[1].percent’/>%，具体分析数据如下：<br/>  <d did=‘2’ showType=‘table’ value=‘quotas[0]’ titleDcrt=‘quotas[0].titles’/>  <d did=‘2’ showType=‘pie’ param=‘{‘xAxis’:’category’, ‘yAxis’:’num’}’ value=‘quotas[0]’ decorateView=‘#category#, #percent(num)#’/>  <br/>  <div style=‘font-height:bold; ‘>“<d did=‘2’ showType=‘value’ value=‘mdName’/>”[<d did=‘2’ showType=‘value’ value=‘quotas[1].titleName’/>]</div>指标分析：<br/>  新加测试showType=“line”：<br/><d did=‘2’ showType=‘line’ param=‘{‘xAxis’:’category’, ‘yAxis’:’num’}’ value=‘quotas[0]’ decorateView=‘#category#, #percent(num)#’/>  新加测试showType=“bar”：<br/><d did=‘2’ showType=‘bar’ param=‘{‘xAxis’:’category’, ‘yAxis’:’num’}’ value=‘quotas[0]’ decorateView=‘#category#, #percent(num)#’/>  测试降序：<d did=‘2’ showType=‘text’ value=‘quotas[1]::first(3|num)’ decorateView=‘{#category#}占#percent#(num)%::{suffix:^、^}’/>  测试升序：<d did=‘2’ showType=‘text ‘ value=‘quotas[1]::!first(3|num)’ decorateView=‘{#category#}占#percent#(num)%’/>"  }, {/\*地图上以点状显示人员坐标信息\*/  id:"seg1\_3",  name:"人员地图分析",  content:"<div style=‘font-height:bold; font-size:18px;’>3、人员地图分析</div><br/>  /\*人员 坐标分布 mapType:使用哪家地图，AGIS|BAIDU|GAODE\*/  <d did=‘2’ showType=‘map\_pts’ param=‘{^X^:^coordX^,^Y^:^coordY^,^Z^:^coordZ^,^mapType^:^BAIDU^}’ value=‘quotas[0]’/>"  }  ]  }  ]  } |

**八、实现总结**

这里总结各种图元素的实现的使用的技术

**1-对于Html展现**

·report.json，用html元素展现；

·饼图、柱图、折线图：用jqPlot；

·表格：用EasyUi；

·雷达图、地图：用百度EChart

**留存：**

***解释：***

|  |  |
| --- | --- |
| did=2 | \_DLIST数据列表中\_id=2的jsonD数据 |
| showType="map\_pts" | 地图上画点 |
| value="quotas[2]" | 取得jsonD中\_DATA对象中quotas[2]的数据表  (quotas数组第2个记录，此记录是表格数据)，任意取100条记录 |
| mapType | 地图类型：BAIDU-百度地图；GAODE-高德地图；TIANDITU-天地图 |
| lable="name", | tooltip标签显示的name,默认格式：”name:value”  对应quotas[2]的数据表中dataList中name列 |
| data="value", | tooltip标签显示的value,默认格式：”name:value”  对应quotas[2]的数据表中dataList中value列，这列必须是数值的 |
| decorateView="姓名:#xm#,身份证:#sfz#,性别:#sb#,colCN(city):#city#" | 怎样显示地图中的标签。  当鼠标移动到点上时，tooltip显示的内容 |

**问题：**

1-如果按tooltip来定义的话，decorateView和label、data会产生冲突，所以是否把decorateView定义为弹出框显示简项描述信息（如果需要弹框显示简项描述信息，那么是否还使用decorateView关键字，或者换成simpInfo）？？？

decorateView和tooltip的关系是有些重复，我在这次整理本文档中已经感觉到了(decorateView和param有重复)，从减少定义的角度考虑还是要保留。

相关的问题，我整理完这个文档，你看看，是否就解决了

1-lable

**注意：！！！**

1-目前的定义，不是很统一，可能还需要重新定义各标签；

2-目前只能定义一种图，如果要实现图的叠加，则不能实现，这个在以后的定义中要实现；

3-是否有这样的图，不是数值而是点，看密度分布，如地图？？？

**扩展：**

showType="pie"

value="excelMdmArray[0]"

decorateView—可以考虑废弃掉

param={paramSoucre:”c.json”,mapType:baidu|gaode|google,parseFun:}

以下标签已经合并到param标签中

|  |  |  |  |
| --- | --- | --- | --- |
| ~~lable~~ | ~~标准：可选~~ | ~~只有饼、柱、折线图有意义，X轴信息~~ | ~~本期使用~~ |
| ~~data~~ | ~~标准：可选~~ | ~~只有饼、柱、折线图有意义，Y轴信息~~  ~~一般是数值~~ | ~~本期使用~~ |
| ~~mapType~~ | ~~标准：可选~~ | ~~只对地图类型有效，指明使用的哪种地图(百度、高德、天地图)，目前支持：BAIDU~~ | ~~本期使用~~ |
| ~~Coord~~ | ~~标准：可选~~ | ~~只对地图类型有效，格式”X,Y”，以逗号隔开，指明使用的地图经度所对应的列、维度所对应的列~~ | ~~本期使用~~ |