

全景网模版标签

TAG V1.0

撰写人: 张阳

MAIL : sdzy890819@126.com

电话: 18210550495

QQ: 466291570

公司: 潍坊华盛信息科技有限公司

概述

全景网模版语言基于目前比较主流的模版语言 Velocity 改造而成, 配合全景网 CMS, 实现数据的聚合数据, 除了 Velocity 基础的语法可以使用以外。新增了 TAG 数据聚合标签。[TAGLIST, TAGDETAIL]。TAGLIST 标签用于聚合列表类、分页类列表数据。TAGDETAIL 标签主要用于聚合散列类数据。

文档目的

方便开发人员快速入手, 并基于 Velocity +TAG 语法 能做到快速迭代开发的目的。

本文档会使用快入简出 的方式介绍 TAG 的使用方法、包含 TAG 参数的传递使用说明、数据返回后的结构化数据的回调等。

使用用途

1. TAG 支持, 第二模版类文件提取数据使用
2. TAG 支持, 模版类文件提取数据使用。
3. TAG 支持, 专题内文件提取数据使用。

专心、专注、专业

里程碑

2017-01-01

全景网模版语言 TAG 开发测试, 并测试通过。

2016-12-26

全景网模版语言 TAG - TAGDETAIL 标签语言出生, 用于单条数据的聚合。

2016-12-20

全景网模版语言 TAG - TAGLIST 标签语言出生, 用于列表页数据的聚合。。

TAGLIST 标签

1) TAGLIST 标签说明

基于 Velocity 模版语言开发而成，属于 BLOCK（块）类型模版标签。用来取多条数据的列表。

2) TAGSLIST 标签参数规范

```
#TAGLIST(String resultObjName, Integer type, String content, Integer count ,Integer size)
```

```
#end
```

A. 参数 resultObjName: 文本类型，自定义的结构化数据返回的对象引用。在使用的时候通过使用

`${resultObjName}` 来使用数据信息

B. 参数 type: 数字类型，可以使用以下 1-5 之间的数字来选择提取的数据方式类型。Type 选择的类型

决定了 content 的内容。类型如下：

- a) 1: 通过新闻栏目 ID 提取新闻列表。
- b) 2: 通过系列专题栏目 ID 提取专题列表。
- c) 3: 通过专题分类 ID 提取专题列表。
- d) 4: 通过推荐栏目 ID 提取推荐栏目列表。

- e) 5: 通过 SQL 提取数据列表。(禁止使用*号提取数据, 建议自定义字段返回引用 来操作 比如

select file_name as filename ,file_path as filePath from abc)

C. 参数 content: 文本类型, 根据参数 type 决定填写,

- a) type=1 ,content 填写新闻栏目 ID 提取新闻列表
- b) type=2 ,content 填写系列专题栏目 ID 提取专题列表
- c) type=3 ,content 填写专题分类 ID 提取专题列表
- d) type=4 ,content 填写推荐栏目 ID 提取推荐栏目列表
- e) type=5 ,content 填写 SQL 提取数据列表

D. 参数 count: 数字类型。用来表示需要提取的数量。所有的类型都受 count 影响。

必须填写。默认是 100 条。可与参数 size 来标示分页。比如 count 为 200 size 为 20

这样的话就代表要分 10 页(注: 一个页面只可以有一个分页列表。)

E. 参数 size: 数字类型。每页条数。必须填写如果不分页的话保持和 count 一致即可。

如果分页的话 如上述所说 操作即可。页面分页后得到: index.html,index-2.html,....

依次类推

3) TAGLIST 数据返回

根据以上规定的类型主要分为: 新闻、新闻推荐信息、专题、SQL 4 种数据格式。根据这 4 种

数据类型详细介绍每种数据返回的字段

A. 新闻字段:

id, lastModifyUserId,title,subtitle,keyword,description,source,author,buildTime,writeTime,channelId
,columnId,url,publish,field1,field2,field3,field4,field5,categoryId,sort....详细参考 PO 类 News

B. 新闻推荐信息:

id, recommendTitle,recommendDescription, recommendImages, recommendColumnId,

recommendUserId,url, sort.....详细参考 PO 类 NewsRecommen

C. 专题:

id,lastModifyUserId,topicTitle,topicClassifyId,categoryId,channelId,releaseTime,keyword,description

,topicColumnId,topicUrl,buildUserId,buildTime.....详细参考 PO 类 Topic

D. SQL:

根据查询的字段使用。

4) TAGLIST 分页

TAGSLIST 当使用了分页的时候。在代码块内可以使用内置对象,page。进行获取总页数、页大小。总条数。等信息。下面介绍一下 page 对象。使用方式: `${page.pageCount}`

A: pageSize:每页数量。跟 TAG 参数中第五个参数定义的数字一致, 调用方式`${page.pageSize}`。

B: count: 总条数。跟 TAG 参数中第四个参数定义的数字一致。如果实际数量达不到定义的数字。则会根据实际数量返回。如果总条数多于定义的数字。显示定义的数字。调用方式: `${page.count}`。

C: pageCount: 总页数。根据 count、pageSize 计算出的总页数。调用方式: `${page.pageCount}`

D: page: 当前页。调用方式: `${page.page}`

5) TAGLIST 实例

A. 实例一 内置类型.

```
#TAGLIST("result",1,"10001",5,5)
```

```
#foreach($item in $result)
```

```
  ${item.title}
```

```
#end
```

```
#end
```

B. 实例二 SQL 类型 不分页.

```
#TAGLIST("result",5,"select title as TITLE from news where column_id=10001",10,10)
```

```
#foreach($item in $result)
```

```
${item.title}
```

```
#end
```

```
#end
```

C. 实例三 SQL 类型 分页（建议 10 页即可 生成规则是 index.html, index-2.html....）

```
#TAGLIST("result",5,"select title as TITLE from news where column_id=10001",100,10)
```

```
#foreach($item in $result)
```

```
${item.title}
```

```
#end
```

```
#end
```

TAGDETAIL 标签

1) TAGDETAIL 标签说明

基于 Velocity 模版语言开发而成，属于 BLOCK（块）类型模版标签。

2) TAGDETAIL 标签参数规范

```
#TAGDETAIL(String resultObjName, Integer type, String content)
```

```
#end
```

A. resultObjName: 文本类型，自定义的结构化数据返回的对象引用。在使用的时候通过使用

`${resultObjName}` 来使用数据信息

B. **type**: 数字类型，可以使用以下 1-9 之间的数字来选择提取的数据方式类型。**Type** 选择的类型

决定了 **content** 的内容，类型如下：

- a) **type**=1, 通过 新闻 ID，提取对应的详情信息
- b) **type**=2, 通过 新闻栏目 ID，提取对应的详情信息
- c) **type**=3, 通过 频道 ID，提取对应的详情信息
- d) **type**=4, 通过 部门分类 ID，提取对应的详情信息
- e) **type**=5, 通过 专题 ID，提取对应的详情信息
- f) **type**=6, 通过 专题分类 ID，提取对应的详情信息
- g) **type**=7, 通过 系列专题 ID，提取对应的详情信息
- h) **type**=8, 通过 碎片 ID，提取对应的详情信息
- i) **type**=9, 通过 SQL，提取对应的详情信息

C. **content**: 文本类型，根据参数 **type** 决定填写：

- a) **type**=1, **content** 需要填写 新闻 ID，提取对应的 新闻信息
- b) **type**=2, **content** 需要填写 新闻栏目 ID，提取对应的 新闻栏目信息
- c) **type**=3, **content** 需要填写 频道 ID，提取对应的 频道信息
- d) **type**=4, **content** 需要填写 部门分类 ID，提取对应的 部门分类信息
- e) **type**=5, **content** 需要填写 专题 ID，提取对应的 专题信息
- f) **type**=6, **content** 需要填写 专题分类 ID，提取对应的 专题分类信息
- g) **type**=7, **content** 需要填写 系列专题 ID，提取对应的 系列专题信息
- h) **type**=8, **content** 需要填写 碎片 ID，提取对应的 碎片信息

- i) type=9, content 需要填写 SQL，提取对应的 SQL 执行数据

3) TAGDETAIL 数据返回

根据以上规定的类型 涵盖了多种数据格式。根据这些数据格式详细介绍每种数据返回的字段

A. type=1，返回的新闻信息：

id,title, subtitle, keyword, description,source, author,newsDetail.content,.....详细的参考 PO 包的 News

B. type=2，返回的新闻栏目信息：

id,columnName,channelId,listTemplate2Id,detailTemplate2Id,keywords,详细的参考 PO 包的 NewsColumn

C. type=3，返回的频道信息：

id,channelName,channelUrl,channelPath,...详细的参考 PO 包的 Channel 类

D. type=4，返回的部门分类信息：

id,categoryName,categoryDesc.....详细的参考 PO 包的 Category 类

E. type=5，返回的专题信息：

id,topicTitle,topicPath,keyword,description,.....详细的参考 PO 包的 Topic 类

F. type=6，返回的专题分类信息：

id,classifyName.....详细的参考 PO 类的 TopicClassify 类

G. type=7，返回的系列专题信息：

id,columnName.....详细的参考 PO 类的 TopicColumn 类

H. type=8，返回的碎片信息：

id,fragmentClassifyId,fragmentContent,fragmentName,fragmentModel...详细的参考 PO 包

Fragment 类

- I. type=9, 返回的 SQL 信息:
sql 查询的字段.

4) TAGDETAIL 实例

- A. 实例一 内置类。

```
#TAGDETAIL("result", 1, "10001")
```

```
${result.title}
```

```
#end
```

- B. 实例二 SQL 类

```
#TAGDETAIL("result", 9, "select title as TITLE from news where title like '%ABC%'")
```

```
${result.TITLE}
```

```
#end
```

详情页内置分页对象

1) 分页说明

详情页当使用了分页。在模版内可以使用内置对象,pageDetail。进行获取总页数、当前页码。等信息。下面介绍一下 pageDetail 对象。使用方式: `${ pageDetail.pageCount}`

A: pageSize:每页数量 1 只会等于 1。调用方式`${ pageDetail.pageSize}`。

B: count: 总条数 根据分页标记计算得到。调用方式: `${ pageDetail.count}`。

C: pageCount: 总页数。根据 count、pageSize 计算出的总页数。调用方式: `${ pageDetail.pageCount}`

D: page: 当前页码。调用方式: `${ pageDetail.page}`

工具类

自定义内置的操作类，在模版内可以使用指定的工具类定义的引用字段调用 函数方法来做一些动作类的操作。比如格式化数据、字符串处理等等。遵循 velocity 语法

1) dateTools 内置工具类

引用字段: dateTools

使用方法: `$dateTools.format(参数 1,参数 2)`

例如: `!{$dateTools.format('yyyy-MM-dd',!{item.createTime})}`

含义: 格式化 item.createTime 为 yyyy-MM-dd 格式。

详细的 API 参考:

<http://velocity.apache.org/tools/2.0/apidocs/index.html>

2) stringTools 内置工具类

引用字段: stringTools

使用方法: `$stringTools.sub(参数 1, 参数 2)`

例如: `!{$stringTools.sub(!{item.title},4)}`

含义: 截取 item.title 的前 4 个字符。

详细的 API 参考:

cms-model 项目内的: com.cn.cms.utils.StringUtils