

模版机制说明书

技术类文档

Version 1.0

2017-01-22

撰写人

张阳

公司

潍坊华盛信息科技有限公司

电话

18210550495

QQ

466291570

模版机制说明书

一、概述

模版是整个 CMS 的核心部门。CMS 分为三块核心内容 “模版（把一个页面或拆分多个页面 使用模版语言编写过以后建立到服务器上即为模版，）” + “映射（把数据和模版之间交互的桥梁，我们认为这是映射。比如栏目下的列表页模版和详情页模版选定(这属于第二套模版体系)、模版列表的关联关系(这属于模版体系), 这两个是目前系统中两套映射）” + “数据(新闻、碎片、专题、推荐、等等。所有的可维护的内容都可以理解为数据)”。

下面我们会把模版、第二模版融合到这一个文档中说明。在本文档中，也会相应的介绍一下 新版模版体系相比目前使用的 TRS 的模版发布体系的优势。

模版机制说明书

二、 模版

模版概述

什么是模版？顾名思义。一个模型。一个版式。我们把一些常用、拥有相同内容或版式的一些文件、页面、页面片、等等通过特殊的编写、修改 制作成一个模块、一个模型。通过数据、映射、模型来生成我们想要的文件、页面、页面片。我们把这个模块、模型称之为模版。

目前 **java** 中有很多成熟的模版框架、语言去实现模版。在本系统中。我们使用 **velocity** 做为模版语言去编写模版这一体系。模版语言这块不在本文档中提及。下面我们介绍一下本系统中的模版。

模版创建

模版根据以下几点详细说明 模版：

1. 模版保存地址及模版名

- a) 在咱们系统中，模版的根目录维护在频道中。一个模版必须要归属到一个频道下。频道维护的时候需要填写频道下模版的根目录。与模版相互辉映。
- b) 模版可以有相对路径。比如根目录/**/data/template/** 那么模版下我想通过相对路径来标示模版的功能。比如列表页我统一放到/**/list/**下。所以有了相对路径的维护
- c) 模版需要有一个指定的文件名、这个文件名也是发布的时候使用的文件名(详情页除外 特例)。比如：**index.html**。
- d) 那么结合起来就是/**/data/template/list/index.html**。这是模版存储的路径。普及：频道内存储了模版根目录、发布地址、发布路径等。集合起来就可以确定要发布的地址和模版位置

2. 模版类型

本系统内把模版分为 **4** 类、方便区分每个模版的类型和功能。

模版机制说明书

- a) 首页类型、首页是一个聚合页。我们通过 SSI 把一个页面拆分成多个小页面。首页是为保存一些聚合页。
- b) 列表页类型、列表页类的可以保存一些列表类、分页列表的页面。
- c) 碎片页类型、碎片页类的可以保存一些碎片片段类的页面。
- d) 详情页类型、详情页只保存新闻详情页面。建立详情页不能使用定时生成。因为详情页的规则和其他的不一样。
- e) 前三种都是逻辑分类。机制都是一样的。只有详情页类型不一样。详情页的自带 data 信息(我们可以理解为自带 Buffer),其他的则不具备

3. 模版生成方式

- a) 触发生成，如果选择触发生成。基本上属于小延迟性生成。不过选择触发生成的话。必须要 关联 相应的数据。目前可以关联、新闻栏目、推荐栏目、专题分类、碎片 ID。
- b) 定时生成，如果选择定时生成。则会在每整分钟的时候触发生成。会按照模版的排序值进行生成。不过咱们采用的是多线程生成机制、所以顺序相差并不大(顺序取决于线程抢夺执行权机制)。切记、详情页不可以使用定时生成(如果熟读了系统代码和模版语言及模版的机制。可以适当的使用)。

4. 模版维护方式

- a) 模版文件通过在模版列表页上使用上传按钮上传。如果需要修改可以通过下载模版进行下载。仅支持文本类型的模版。

5. 模版关联

- a) 模版关联关系。在模版是触发生成的时候可以使用。如果不是触发生成。设置后无效果。关联关系可以建立以下 4 种模版关联
- b) 新闻栏目关联：当新闻发布以后，发布的新闻所属栏目有关联关系存在，则会触发相关模版生成
- c) 专题分类关联：当专题发布以后，发布的专题所属分类有关联关系存在，则会触发相关模版生成

模版机制说明书

- d) 碎片 ID 关联：当碎片发布以后，发布的碎片如果有关联关系存在，则会触发相关模版生成。
- e) 新闻推荐栏目关联：当新闻发布以后，当有新闻被推荐以后，如果推荐的栏目有关联关系存在，则会触发相关模版生成。
- f) 关联关系如果被创建，即时生效。

6. 模版内置对象

- a) **template** 对象。存储了模版信息。详情字段可以参考 PO 类 **Template**
- b) **data** 对象。存储了当前触发模版生成的新闻 | 专题 | 碎片的信息。专题不包含内容。只存在于详情页
- c) **pageDetail** 对象。存储了分页情况。只存在于使用了分页的新闻页详情页调用。使用方式为
 - i. 当前页：`${pageDetail.page}`
 - ii. 总页数：`${pageDetail.pageCount}`
 - iii. 总条数：`${pageDetail.count}` .不常用
 - iv. 每页条数`${pageDetail.pageSize}`.不常用
- d) **currentFileName**。显示当前模版的文件名。比如 `test-2.html` 文件。`${currentFileName}`输出为：`test`
- e) **currentFileSuffix**。显示当前模版的文件名后缀。比如 `test-2.html` 文件。`${currentFileSuffix}`输出为：`html`

总结：模版使用过程中。如果使用 SQL 类型比较多的情况下。建议在做 SQL 的时候加一些索引。在使用 SQL 的时候可以先在客户端里通过 **EXPLAIN** 查看执行计划 里查看一下执行效率。和索引使用情况。。

模版机制说明书

三、 第二模版

第二模版概述

第二模版和第一套模版有一些细节区别。也同样支持 TAG1.0。velocity 语法。一些细节区别会在下面。一一介绍。

第二模版创建

第二模版从以下几点说明第二模版的机制：

1. 第二模版保存地址及模版名

- a) 第二模版的根目录使用第二模版基础信息维护。不和模版走一套体系。单独列出。以免混淆，发布还是走频道地址发布。
- b) 第二模版页可以有相对路径。比如根目录/data/template/ 那么第二模版下我想通过相对路径来标示模版的功能。比如列表页我统一放到/list/下。所以有了相对路径的维护
- c) 第二模版页需要有一个指定的文件名、这个文件名也是发布的时候使用的文件名(详情页除外 特例)。比如：index.html。
- d) 那么结合起来就是/data/template/list/index.html。这是第二模版存储的路径。普及：频道内存储了发布地址、发布路径等。集合起来就可以确定要发布的地址和模版位置

2. 第二模版类型

本系统内把第二模版分为 2 类，每一类都对应一类功能。

- a) 列表页类型、列表页类保存一些公共的分页列表类型的页面。支持 TAG1.0、velocity 语法
- b) 详情页类型、保存新闻详情页面。支持 TAG1.0、velocity 语法

模版机制说明书

- c) 和模版不一样的是。这个是指定类的。模版只作用于新闻。和模版可以共存。如果有一样的详情页面。或者列表页名一样的列表页则。则采用覆盖机制。同一个频道下的模版生成一样名字的页面时。以最后生成页面为准。

3. 第二模版生成方式

- a) 第二模版生成方式只有一种就是触发机制。必须有新闻栏目内指定后才会触发生成。生成会以新闻栏目的频道路径为生成目录。（详情页和列表页一样。）

4. 第二模版维护方式

- a) 第二模版文件通过在第二模版列表页上使用上传按钮上传。如果需要修改可以通过下载模版文件进行下载。仅支持文本类型的模版文件。

5. 第二模版关联

- a) 第二模版关联和模版区别很大。第二模版沿用了 TRS 指定机制。维护新闻栏目指定列表第二模版、和详情页第二模版。来触发第二模版。

6. 第二模版内置对象

- a) **template** 对象。存储了第二模版信息。详情字段可以参考 PO 类 **Template2**
- b) **data** 对象。存储了当前触发模版生成的新闻的信息。详情页有效。
- c) **column** 对象。存储了当前新闻所属栏目信息。详细参考 PO 类的 **column**
- d) **pageDetail** 对象。存储了分页情况。只存在于使用了分页的新闻页详情页调用。使用方式为
 - i. 当前页: `${pageDetail.page}`
 - ii. 总页数: `${pageDetail.pageCount}`
 - iii. 总条数: `${pageDetail.count}` .不常用
 - iv. 每页条数`${pageDetail.pageSize}`.不常用

模版机制说明书

- e) `currentFileName`。显示当前模版的文件名。比如 `test-2.html` 文件。`${currentFileName}`输出为: `test`
- f) `currentFileSuffix`。显示当前模版的文件名后缀。比如 `test-2.html` 文件。`${currentFileSuffix}`输出为: `html`

总结：模版使用过程中。如果使用 `SQL` 类型比较多的情况下。建议在做 `SQL` 的时候加一些索引。在使用 `SQL` 的时候可以先在客户端里 通过 `EXPLAIN` 查看执行计划 里查看一下执行效率。和索引使用情况。。