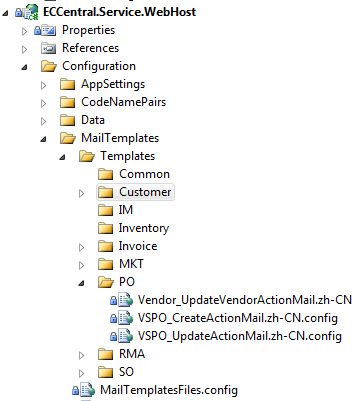
# 邮件模板配置

对于大多数内容格式比较固定（通常是在一个邮件内容中，变动的部分只是其中某几个固定的区域，其它地方不会改变，比如称呼，日期等。）的邮件，就会做成一个可配置的模板，放到一个集中的地方，使用的时候调相关的API获取即可，也便于以后的修改和维护。

## 模板Config文件结构及配置说明

所有关于邮件模板和发送相关的配置都在**Service.WebHost\Configuration\MailTemplates**这个文件夹里：



|  |  |
| --- | --- |
| **文件/文件夹名称** | **说明** |
| Templates | 这个子文件夹里面存放的是系统所定义的所有的邮件模板配置，每当新增一个邮件模板时，就在这个文件夹里添加config即可，这里需要说明的是，**除了新增一个模板config之外,还需要在MailTemplatesFiles.config里面，根据语言增加一条关于此模板的一行路径配置**.（MailTemplatesFiles.config会在下面说明.） |
| MailTemplatesFiles.config | 根据语言编码，定义邮件模板配置文件目录的位置(支持绝对路径)，  每一个templateList都有一个LanguageCode，它的所有子元素，就是对应这个语言编码的所有邮件模板的一个集合.),例如:  <templateList languageCode="zh-CN">  <template path="Templates/Customer/CustomerGift\_Notify. zh-CN.config"/>  <template path="Templates/Customer/CustomerGift\_Remind. zh-CN.config"/>  </templateList>  <templateList languageCode="en-US">  <template path="Templates/Customer/CustomerGift\_Notify. en-US.config"/>  <template path="Templates/Customer/CustomerGift\_Remind. en-US.config"/>  </templateList> |

## 模板Config文件内容格式

一个邮件模板的配置格式如下:

<?xml version="1.0"?>

<mailTemplate>

<id>模板ID（必须唯一）</id>

<sender>发件人地址</sender>

<senderDisplayName>发件人显示名称</senderDisplayName>

<to>收件人地址</to>

<isHtmlType>是否是html格式</isHtmlType>

<mailPriority>邮件重要级别</mailPriority>

<subject>

<![CDATA[

邮件标题

]]>

</subject>

<body>

<template reference="shared/header.htm"/>

<template>

<![CDATA[

邮件内容

]]>

</template>

<template reference="shared/footer.htm"/>

</body>

</mailTemplate>

|  |  |
| --- | --- |
| **节点** | **说明** |
| **id** | 模板ID（必须唯一） |
| **sender** | 发件人地址，如:xxx@newegg.com,(若不配置此节点，则默认读取SMTP邮件配置(Mail.config)中的SenderAddress) |
| **senderDisplayName** | 发件人显示名称.(若不配置此节点，则默认读取SMTP邮件配置(Mail.config)中的SenderDisplayName) |
| **to** | 收件人email地址，不是必填；该节点主要针对那些收件地址都是固定的内部邮件；如果to节点有填email地址，同时在程序里也传入了收件人的email地址，那么最终的收件人地址将是 :模板配置的To节点的email地址 + 传入的email地址. |
| **isHtmlType** | 是否html格式，默认配置为1(是) |
| **mailPriority** | 邮件重要级别(普通=0，低=1，重要=2,默认配置为0) |
| **subject** | 邮件标题 |
| **body/template** | 邮件内容，支持2种方式：1.直接把模板html文本内容放到<template>的innerHtml中；2.用<template>的attribute “reference”来指定包含模板文本内容的文件地址，支持绝对路径和相对该config文件的相对路径 |

## 模板邮件Body中的 Table配置

可参见文档《邮件模板及打印模板的统一规则.docx》

对于要循环输出某个固定样式的Table Row，可以在模板里面进行样式设置，并且在MailInfoMessage实体中传入KeyTableList参数，即可按照配置的样式进行HTML输出. 而不必自己在代码里面拼凑html代码.

配置语法为:

*…*

*<%TABLE\_ROWS\_BEGIN ID=*TableID*%>*

*<tr><td><%=id%></td><td><%=name%></td>……</tr>*

*<%TABLE\_ROWS\_END ID=*TableID *%>*

*…*

**注意: 必须以 <%TABLE\_ROWS\_BEGIN ID=xxxxx%> 开头，以 <%TABLE\_ROWS\_END ID=xxxxx%> 结尾，中间包含想要输出的row的html代码，才能代表一个Table的RowList.** 其中，TableID为KeyTableVariables的Key值, 对于*<%=id%>，<%=name%>*，其中的id, name就是KeyTableVariables的Value（DataTable类型）中的列名称. (KeyTableVariables的构建请参考下面的章节.),

如果在模板中配置了上述代码，则会根据传入的KeyTableVariables，自动构建Table样式的输出,

支持一个模板有多个Tables的配置，但必须保证TableID的唯一性,例如:

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1">

</head>

<body style="background: #FFF;font-size: 9pt;">

<table>

<tr>

<td>标题1</td>

<td>标题2</td>

<td>标题3</td>

</tr>

*<%TABLE\_ROWS\_BEGIN ID=tbl\_1%>*

*<tr>*

*<td style="font-size:12px;color:red"><%=id%></td>*

*<td style="font-size:12px;color:red"><%=name%></td>*

*<td style="font-size:12px;color:blue"><%=sex%></td>*

*</tr>*

*<%TABLE\_ROWS\_END ID=tbl\_1%>*

</table>

<table>

<tr>

<td>标题1\_1</td>

<td>标题2\_1</td>

<td>标题3\_1</td>

<td>标题4\_1</td>

</tr>

*<%TABLE\_ROWS\_BEGIN ID=tbl\_2%>*

*<tr>*

*<td style="font-size:12px;color:red"><%=title%></td>*

*<td style="font-size:12px;color:red"><%=content%></td>*

*<td style="font-size:12px;color:blue"><%=datetime%></td>*

*<td style="font-size:12px;color:blue"><%=username%></td>*

*</tr>*

*<%TABLE\_ROWS\_END ID=tbl\_2%>*

</table>

</body>

</html>

## 模板邮件Body中的 IF配置

可参见文档《邮件模板及打印模板的统一规则.docx》

# 模板邮件的标准数据结构:

## 构建Key-Value 替换变量

KeyValueVariables keyValueList= **new** KeyValueVariables**();**

keyValueList.AddKeyValue**(**"#替换符1#"**,** "替换内容1"**);**

keyValueList.AddKeyValue**(**"#替换符2#"**,** "替换内容2"**);**

**……**

**……**

**……**

## 构建Key-Table 替换变量

KeyTableVariables keyTableList= **new** KeyTableListVariables**();**

tableList = new DataTable();  
tableList.Columns.AddRange(new DataColumn[]

{

new DataColumn("ProductID"),

new DataColumn("ProductName"),

new DataColumn("Price"),

new DataColumn("Quantity"),

new DataColumn("Amount"),

});

tableList.Rows.Add("Item177292", "XXX电视机", 1270.00, 1, 127.00);

keyTableList.Add("ProductList", tableList);

# 邮件发送接口

Service端，发送邮件的API为 ：ECCentral.Service.Utility.EmailHelper

只有一个静态方法：public static void SendEmailByTemplate

该方法具有9个重载形式，根据需要选择调用：

