一、jmeter html报告优化

**1. 邮件发送html报告有中文时，显示乱码：**

修改encoding为“UTF-8”

<xsl:output method="html" indent="yes" encoding="UTF-8" doctype-public="-//W3C//DTD HTML 4.01 Transitional//EN" />

**2. Summary中的只标红Failures数：**

* 屏蔽Summary中class属性
* <!--
* <xsl:attribute name="class">
* <xsl:choose>
* <xsl:when test="$allFailureCount &gt; 0">Failure</xsl:when>
* </xsl:choose>
* </xsl:attribute>

-->

* 修改allFailureCount
* <td align="center">
* <xsl:value-of select="$allSuccessCount" />
* </td>
* <xsl:choose>
* <xsl:when test="$allFailureCount &gt; 0">
* <td align="center" style="font-weight:bold">
* <font color="red">
* <xsl:value-of select="$allFailureCount" />
* </font>
* </td>
* </xsl:when>
* <xsl:otherwise>
* <td align="center">
* <xsl:value-of select="$allFailureCount" />
* </td>
* </xsl:otherwise>

</xsl:choose>

**3. Pages页面按Average Time倒序排序：**

在Pagelist模板中for-each下添加

<xsl:for-each select="/testResults/\*[not(@lb = preceding::\*/@lb)]" >

<!-- 按平均时间排序 -->

<xsl:sort select="sum(../\*[@lb = current()/@lb]/@t) div count(../\*[@lb = current()/@lb])" data-type="number" order="descending"/>

**4. 接口Average Time超过2s标黄显示：**

* 添加LongTime css

.Failure { font-weight:bold; color:red;

} .LongTime { font-weight:bold; color:#ff9900;

}

* Pagelist 模块中针对错误和超长时间接口标色显示

<tr valign="top">

<xsl:choose>

<!-- 失败用例标红显示 -->

<xsl:when test="$failureCount &gt; 0">

<xsl:attribute name="class">

<xsl:choose>

<xsl:when test="$failureCount &gt; 0">Failure</xsl:when>

</xsl:choose>

</xsl:attribute>

</xsl:when>

<!-- 平均时间超过2s，标色显示 -->

<xsl:when test="$averageTime &gt; 2000">

<xsl:attribute name="class">

<xsl:choose>

<xsl:when test="$averageTime &gt; 2000">LongTime</xsl:when>

</xsl:choose>

</xsl:attribute>

</xsl:when>

</xsl:choose>

**5. 添加90% Line和QPS：**

* 添加90 %lineTime模板

<xsl:template name="max">

<xsl:param name="nodes" select="/.." />

<xsl:choose>

<xsl:when test="not($nodes)">NaN</xsl:when>

<xsl:otherwise>

<xsl:for-each select="$nodes">

<xsl:sort data-type="number" order="descending" />

<xsl:if test="position() = 1">

<xsl:value-of select="number(.)" />

</xsl:if>

</xsl:for-each>

</xsl:otherwise>

</xsl:choose></xsl:template>

<!-- 90% line time -->

<xsl:template name="lineTime">

<xsl:param name="nodes" select="/.." />

<xsl:choose>

<xsl:when test="not($nodes)">NaN</xsl:when>

<xsl:otherwise>

<xsl:for-each select="$nodes">

<xsl:sort data-type="number" />

<!-- last() 返回当前上下文中的最后一个节点位置数 -->

<!-- ceiling(number) 返回大于number的最小整数 -->

<!-- floor(number) 返回不大于number的最大整数 -->

<!-- position() 返回当前节点位置的数字 -->

<!-- number(object) 使对象转换成数字 -->

<xsl:choose>

<!-- 当只有一个节点时，向上取整 -->

<xsl:when test="last() = 1">

<xsl:if test="position() = ceiling(last()\*0.9)">

<xsl:value-of select="number(.)" />

</xsl:if>

</xsl:when>

<xsl:otherwise>

<xsl:if test="position() = floor(last()\*0.9)">

<xsl:value-of select="number(.)" />

</xsl:if>

</xsl:otherwise>

</xsl:choose>

</xsl:for-each>

</xsl:otherwise>

</xsl:choose>

</xsl:template>

* Sunmary中添加标题

<tr valign="top">

<th># Samples</th>

<th>Success</th>

<th>Failures</th>

<th>Success Rate</th>

<th>Average Time</th>

<th>Min Time</th>

<th>Max Time</th>

<th>90% Line</th>

<th>QPS</th>

</tr>

* Summary中添加allLineTime和qps变量

<xsl:variable name="allMaxTime">

<xsl:call-template name="max">

<xsl:with-param name="nodes" select="/testResults/\*/@t" />

</xsl:call-template>

</xsl:variable>

<!-- New add 90% line -->

<xsl:variable name="allLineTime">

<xsl:call-template name="lineTime">

<xsl:with-param name="nodes" select="/testResults/\*/@t" />

</xsl:call-template>

</xsl:variable>

<!-- 将毫秒转换成秒 -->

<xsl:variable name="qps" select="$allCount \* 1000 div $allTotalTime"/>

* Summary中调用allLineTime和qps变量

<td align="center">

<xsl:call-template name="display-time">

<xsl:with-param name="value" select="$allMaxTime" />

</xsl:call-template>

</td>

<td align="center">

<xsl:call-template name="display-time">

<xsl:with-param name="value" select="$allLineTime" />

</xsl:call-template>

</td>

<td align="center">

<xsl:call-template name="display-qps">

<xsl:with-param name="value" select="$qps" />

</xsl:call-template>

* pagelist中添加标题

<xsl:template name="pagelist">

<h2>Pages</h2>

<table align="center" class="details" border="0" cellpadding="5" cellspacing="2" width="95%">

<tr valign="top">

<th>URL</th>

<th># Samples</th>

<th>Success</th>

<th>Failures</th>

<th>Success Rate</th>

<th>Average Time</th>

<th>Min Time</th>

<th>Max Time</th>

<th>90% Line</th>

<th>QPS</th>

<th></th>

</tr>

* pagelist中添加allLineTime和qps变量

<xsl:variable name="maxTime">

<xsl:call-template name="max">

<xsl:with-param name="nodes" select="../\*[@lb = current()/@lb]/@t" />

</xsl:call-template>

</xsl:variable>

<!-- new add 90% line time -->

<xsl:variable name="nintyTime">

<xsl:call-template name="lineTime">

<xsl:with-param name="nodes" select="../\*[@lb = current()/@lb]/@t" />

</xsl:call-template>

</xsl:variable>

<xsl:variable name="qpsTime" select="$count \* 1000 div $totalTime"/>

* pagelist中调用allLineTime和qps变量

<td align="center">

<xsl:call-template name="display-time">

<xsl:with-param name="value" select="$maxTime" />

</xsl:call-template>

</td>

<!-- Page页面添加90% LineTime -->

<td align="center">

<xsl:call-template name="display-time">

<xsl:with-param name="value" select="$nintyTime" />

</xsl:call-template>

</td>

<td align="center">

<xsl:call-template name="display-qps">

<xsl:with-param name="value" select="$qpsTime" />

</xsl:call-template>

</td>

**6.Failure Detail模块显示Response Data：**

* 设置showData为‘y’

<!-- Defined parameters (overrideable) -->

<xsl:param name="showData" select="'y'"/>

<xsl:param name="titleReport" select="'Interface Test Results'"/>

<xsl:param name="dateReport" select="'date not defined'"/>

* 替换内容

<table class="details" border="0" cellpadding="5" cellspacing="2" width="95%">

<tr valign="top">

<th align="center">Response</th>

<th align="center">Failure Message</th>

<xsl:if test="$showData = 'y'">

<th align="left">Response Data</th>

</xsl:if>

</tr>

<xsl:for-each select="/testResults/\*[@lb = current()/@lb][attribute::s='false']">

<tr>

<td><xsl:value-of select="@rc | @rs" /> - <xsl:value-of select="@rm" /></td>

<td><xsl:value-of select="assertionResult/failureMessage" /></td>

<xsl:if test="$showData = 'y'">

<td><xsl:value-of select="responseData" /></td>

</xsl:if>

</tr>

</xsl:for-each>

**7.添加Host**

<xsl:template name="pageHeader">

<h1><xsl:value-of select="$titleReport" /></h1>

<table width="100%">

<tr>

<!-- 获取requestHeader数据 -->

<xsl:variable name="req" select="/testResults/httpSample/requestHeader" />

<!-- 从获取的URL中截取数据，以Host：分割，取后半部分 -->

<xsl:variable name="remaining" select="substring-after($req,'Host:')" />

<!-- 已换行符来截取，before代表换行符之前的数据 -->

<xsl:variable name="host" select="substring-before($remaining, '&#xA;')" />

<td align="left">Date report: <xsl:value-of select="$dateReport" /></td>

<td align="center">Host: <xsl:value-of select="$host" /></td>

<td align="right"><a href="./TestLog.html">测试日志</a></td>

</tr>

</table>

<hr size="1" /></xsl:template>

### 二、build.xml和表格样式文件[jmeter-results-detail-report\_30.xsl](https://pan.baidu.com/s/1gfvIVNt)

传参问题

**修改url**

1.打开样式的jmeter-results-detail-report\_21.xsl文件，在文件头部 定义变量：ReportUrl，给了一个默认值

<xsl:param name="ReportUrl" select="'http://10.32.1.235:8080/'"/>



2.修改显示的内容，增加超链接

<td align="right"><a><xsl:attribute name="href"><xsl:value-of select="$ReportUrl" /></xsl:attribute>

详细报告查看地址地址

</a>

</td>



3.修改bulid.xml文件

增加变量ReportUrlName，这里的内容来源Jenkins的传递



5.Jenkins配置上

