

WebService 接口方案

1.1数据接口实现说明

1.1.1 服务端信息

服务端各 Webservice 接口发布信息如下：

IP 地址	服务名称	端口
待定	待定	待定

1.2 系统接口设计方案

1.2.1 监控接口

推送设备监控数据接口：**pushMonitor**

函数名称	pushMonitor
描述	设备监控数据传递：函数入参数为 1 个，设备监控数据接口函数为 pushMonitor (String pushData)。pushData 是推送的设备监控信息的 JSON 字符串（入参），为全量数据推入（下一次调用推送会把前一次清除）；函数返回值为 JSON 字符串（出参）。
入参	<p><pushData>标准 JSON 数据格式</p> <p>事例：pushData= {</p> <pre> "_KEY_" : "BMC" , // 监控标识（用于区分不同监控系统）, 相同的 "_KEY_" 会被下次传来的监控数据替换 "_DATA_" : { "2000028" : { // 监控设备ID "湿度" : { // 监控关键字</pre>

	<pre> "group" : "", //分组名称 值可以为空（主要是3D内显示时 候区分不同的类别用） "type" : "数值", //数据类型 必选项(数值、状态)二选一， 比如温度，属于数值，比如漏水通讯状态，属于状态,主要用于在3D页面内显示用 "unit" : "%", //变量单位 可以传入空字符（数值的单位， 比如%，v，数据类型为状态不用填） "value" : "11" // 变量值 必选项 3D内显示用 }, "漏水通讯状态" : { "group" : "", "type" : "状态", "unit" : "", "value" : "1" // 3D界面用三色灯显示显示:0 灰色灯,1 绿 色灯,2 红色灯 }, //对于架式设备、机柜或独立设备的实际监控功率必须用【功率】监控关键字，其 将用于机房或机柜已使用功率统计。 "功率" : { "group" : "", "type" : "数值", "unit" : "kw", //单位必须是kw或w （不区分大小写） "value" : "2.0" } ////当前设备有多少项监控属性就显示多少个键值对 } ////当前要监控的所有设备 } };</pre>
出参	返回数据集:

	<p>数据 Push 成功后, 返回: {"success":true,"data":""}, data 信息。</p> <p>数据 Push 失败后, 返回: {"success":false,"data":""}, data 中有相应的失败信息。</p>
备注	

推送设备监控数据接口：pushMonitor

函数名称	pushMonitor
描述	<p>设备监控数据传递：函数入参数为 2 个，设备监控数据接口函数为 pushMonitor (String pushData,boolean rule)。pushData 是推送的设备监控信息的 JSON 字符串（入参），rule 数据追加标识,true 为追加数据,false 为覆盖原有数据(入参);函数返回值为 JSON 字符串(出参)。</p>
入参	<p><pushData>标准 JSON 数据格式</p> <p>事例：pushData= {</p> <pre> "_KEY_" : "BMC", // 监控标识（用于区分不同监控系统）,相同的"_KEY_" 会被下次传来的监控数据替换 "_DATA_" : { "2000028" : { // 监控设备ID "湿度" : { // 监控关键字 "group" : "", //分组名称 值可以为空（主要是3D内显示时 候区分不同的类别用） "type" : "数值", //数据类型 必选项(数值、状态)二选一， 比如温度，属于数值，比如漏水通讯状态，属于状态,主要用于在3D页面内显示用 "unit" : "%", //变量单位 可以传入空字符（数值的单位，</pre>

	<p>比如%, v, 数据类型为状态不用填)</p> <pre> "value" : "11" // 变量值 必选项 3D内显示用 }, "漏水通讯状态" : { "group" : " ", "type" : "状态", "unit" : " ", "value" : "1" // 3D界面用三色灯显示显示:0 灰色灯,1 绿色灯,2 红色灯 }, //对于架式设备、机柜或独立设备的实际监控功率必须用【功率】监控关键字, 其将用于机房或机柜已使用功率统计。 "功率" : { "group" : " ", "type" : "数值", "unit" : "kw", //单位必须是kw或w (不区分大小写) "value" : "2.0" } ////当前设备有多少项监控属性就显示多少个键值对 } ////当前要监控的所有设备 } };</pre>
--	---

关闭设备监控数据接口：closeMonitor

函数名称	closeMonitor
描述	设备监控数据传递:函数入参数为 1 个,设备监控数据接口函数为 close (String closeData)。pushData 是推送的设备监控信息的 JSON 字符

	串（入参）；函数返回值为 JSON 字符串（出参）。
入参	<p><closeData>标准 JSON 数据格式</p> <p>事例：closeData= {</p> <pre> "_KEY_" : "BMC", // 监控标识（用于区分不同监控系统），相同的 "_KEY_" 会被下次传来的监控数据替换 "_DATA_" : { "2000028" : { // 监控设备ID "湿度" : { // 监控关键字 "group" : "", // 分组名称 值可以为空（主要是3D内显示时 候区分不同的类别用） "type" : "数值", // 数据类型 必选项（数值、状态）二选一， 比如温度，属于数值，比如漏水通讯状态，属于状态，主要用于在3D页面内显示用 "unit" : "%", // 变量单位 可以传入空字符（数值的单位， 比如%，v，数据类型为状态不用填） "value" : "11" // 变量值 必选项 3D内显示用 }, "漏水通讯状态" : { "group" : "", "type" : "状态", "unit" : "", "value" : "1" // 3D界面用三色灯显示显示：0 灰色灯,1 绿 色灯,2 红色灯 }, }, // 对于架式设备、机柜或独立设备的实际监控功率必须用【功率】监控关键字，其 将用于机房或机柜已使用功率统计。 "功率" : { "group" : "", "type" : "数值", "unit" : "kw", // 单位必须是kw或w （不区分大小写）</pre>

	<pre> "value" : "20.0" } ////当前设备有多少项监控属性就显示多少个键值对 } ////当前要监控的所有设备 } };</pre>
出参	<p>返回数据集:</p> <p>数据 Push 成功后, 返回: {"success":true,"data":""}, data 信息。</p> <p>数据 Push 失败后, 返回: {"success":false,"data":""}, data 中有相应的失败信息。</p>

1.2.2 报警接口

推送报警信息接口：pushAlarm

函数名称	pushAlarm
描述	设备报警数据传递：函数入参数为 1 个，设备报警数据接口函数为 pushAlarm (String pushData)。pushData 是推送的设备报警信息的 JSON 字符串（入参），为全量数据推入（下一次调用推送会把前一次清除）；函数返回值为 JSON 字符串（出参）。
入参	<p><pushData>标准 JSON 数据格式</p> <p>事例：pushData={</p> <pre>"_KEY_" : "BMC", // 告警标识（用于区分不同监控系统） "_DATA_" : { "2000028" : { // 报警设备ID "湿度" : { // 告警关键字 "报警级别" : "critical", // *必传项 项目提供 "报警时间" : "2011-09-05 19:01:48", // *非必传项 "报警信息" : "温度过高..." // *非必传项 }, "温度" : { "报警级别" : " critical", //项目提供 "报警时间" : "2011-09-05 19:01:48", "报警信息" : "..." } } ////当前设备有多少项属性报警就显示多少个键值对 }, ////当前要报警的所有设备 "漏水线A" : { /**如果漏水报警，增加 "报警位置"必传项 **//</pre>

	<pre> "漏水": { "报警级别": "MINOR", // *必传项 项目提供 "报警信息": "漏水", // *非必传项 "报警时间": "2012/05/16 16:12:09", // *非必传项 "报警位置": 10 // *必传项 } } } };</pre>
出参	<p>返回数据集:</p> <p>数据 Push 成功后, 返回: {"success":true,"data":""}, data 信息。</p> <p>数据 Push 失败后, 返回: {"success":false,"data":""}, data 中有相应的失败信息。</p>
备注	

推送报警信息接口：pushAlarm

函数名称	pushAlarm
描述	设备报警数据传递：函数入参数为 2 个，设备报警数据接口函数为 pushAlarm (String pushData,boolean rule)。pushData 是推送的设备报警信息的 JSON 字符串（入参），rule 数据追加标识,true 为追加数据,false 为覆盖原有数据(入参);函数返回值为 JSON 字符串(出参)。
入参	<pushData>标准 JSON 数据格式

	<p>事例：pushData={</p> <pre>"_KEY_" : "BMC", // 告警标识（用于区分不同监控系统） "_DATA_" : { "2000028" : { // 报警设备ID "湿度" : { // 告警关键字 "报警级别" : "critical", // *必传项 项目提供 "报警时间" : "2011-09-05 19:01:48", // *非必传项 "报警信息" : "温度过高..." // *非必传项 }, "温度" : { "报警级别" : " critical", //项目提供 "报警时间" : "2011-09-05 19:01:48", "报警信息" : "..." } ////当前设备有多少项属性报警就显示多少个键值对 }, ////当前要报警的所有设备 "漏水线A" : { //**如果漏水报警，增加 “报警位置”必传项 **// "漏水" : { "报警级别" : "MINOR", // *必传项 项目提供 "报警信息" : "漏水", // *非必传项 "报警时间" : "2012/05/16 16:12:09", // *非必传项 "报警位置" : 10 // *必传项 } } } };</pre>
出参	返回数据集:

	<p>数据 Push 成功后, 返回: {"success":true,"data":""}, data 信息。</p> <p>数据 Push 失败后, 返回: {"success":false,"data":""}, data 中有相应的失败信息。</p>
备注	

关闭报警信息接口: **closeAlarm**

函数名称	closeAlarm
描述	设备监控数据传递: 函数入参数为 1 个, 设备监控数据接口函数为 closeAlarm (String closeData)。pushData 是推送的设备监控信息的 JSON 字符串 (入参); 函数返回值为 JSON 字符串 (出参)。
入参	<p><closeData>标准 JSON 数据格式</p> <p>事例: closeData= {</p> <pre> "_KEY_" : "BMC", // 告警标识 (用于区分不同监控系统) "_DATA_" : { "2000028" : { // 报警设备ID "湿度" : { // 告警关键字 "报警级别" : "critical", // 项目提供 "报警时间" : "2011-09-05 19:01:48", // "报警信息" : "温度过高..." // }, "温度" : { "报警级别" : " critical", //项目提供 "报警时间" : "2011-09-05 19:01:48", "报警信息" : "... " } } ////当前设备有多少项属性报警就显示多少个键值对 } </pre>

	<pre>////当前要报警的所有设备 "漏水线A" : {/**如果漏水报警，增加 "报警位置"必传项 **// "漏水" : { "报警级别" : "MINOR",// *必传项 项目提供 "报警信息" : "漏水",// *非必传项 "报警时间" : "2012/05/16 16:12:09",// *非必传项 "报警位置" : 10 // *必传项 } } };</pre>
出参	<p>返回数据集:</p> <p>数据 Push 成功后, 返回: {"success":true,"data":""}, data 信息。</p> <p>数据 Push 失败后, 返回: {"success":false,"data":""}, data 中有相应的失败信息。</p>