**­­**

**产线现场集中管理通信协议**

|  |  |
| --- | --- |
| **Version:** | **1.0.1** |
| **Author:** | **DSM TTDD** |
| **Date:** | **2021.04.20** |

**变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SEQ** | **Version** | **Date** | **Author** | **Description** | **Remark** |
| 1 | 1.0.0 | 2021.04.16 | FF.LIU | TTDD相关人员第一次会议后定稿 |  |
| 2 | 1.0.1 | 2021.04.20 | FF.LIU | TTDD相关人员第二次会议后定稿 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1.整体架构 4](#_Toc69733878)

[2.各层API概述 5](#_Toc69733880)

[2.1 HTTP API 5](#_Toc69733881)

[2.2 函数式接口 6](#_Toc69733882)

[1）发送数据函数 6](#_Toc69733883)

[2）发送换线指令函数 6](#_Toc69733884)

[2.3 Lora接口---待完善 7](#_Toc69733885)

[3. API详情 8](#_Toc69733886)

[1）编号1: 端---接收换线指令 8](#_Toc69733887)

[2）编号2: PI---接收端登录消息 9](#_Toc69733888)

[3）编号3: PI---接收端退出登录消息 10](#_Toc69733889)

[4）编号4: PI---接收测试/制程/产线状态数据 11](#_Toc69733890)

[5）编号5: PI---接收换线结果 12](#_Toc69733891)

[6）编号6: PI---接收心跳信号 13](#_Toc69733892)

[7）编号7: PI---接收换线指令 14](#_Toc69733893)

[8）编号8: PI---网页实时显示数据->待完善 15](#_Toc69733894)

[9）编号9: CCS---接收模组登录消息 16](#_Toc69733895)

[10）编号10: CCS---接收模组退出登录消息 17](#_Toc69733896)

[11）编号11: CCS-接收测试/制程/产线状态数据 18](#_Toc69733897)

[12）编号12: CCS---接收换线结果 19](#_Toc69733898)

[13）编号13: CCS---接收心跳信号 20](#_Toc69733899)

[14）编号14: CCS---接收当站状态变化 21](#_Toc69733900)

[15）编号15: CCS---网页实时显示数据->待完善 22](#_Toc69733901)

[4.HTTP加密方式 23](#_Toc69733902)

# 1.整体架构

**一条线体**

**产线状态**

(静电/离子风机/接地/温湿度/用电)

**测试站**

**(**全功能/热阻)

GPIO/USB

TCP

传感器

**制程站**

(治工具/人工/老旧设备)

Modbus

GPIO/USB

RS232/485

TCP

PLC

传感器

串口

扫描枪

PC

**对端接口**

**PI**

**函数式接口**

**函数式接口**

**PI**

**(TMM)**

**PI**

**(PMM)**

**HTTP API**

**对边接口**

**HTTP API**

**HTTP API**

**HTTP API**

LORA

IOT

IOT

**IOT**

**LORA**

**队长指挥系统**

**CCS**

**(监控+换线)**

注：队长指挥系统, Captain Command System, 简称CCS

# 2.各层API概述

## 2.1 HTTP API

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **层** | **编号** | **API定义** | **API名称** | **请求方式** | **API参数** |
| **端**  (支持HTTP) | 1 | [接收换线指令](#_1）编号1:_端---接收换线指令) | /ChangeLine | POST | ①设备ID ②工单 ③机种名 |
|  | | | | | |
| **PI**  注：  \* PI有线IP固定,供端访问;  \* 用户需配置CCS的IP，端口号,设备ID | 2 | [接收端登录消息](#_2）编号2:_PI---接收端登录消息) | /UserLogin | POST | ①端IP ②端口号  ③当站基本信息(工单机种线段组站) |
| 3 | [接收端退出登录消息](#_3）编号3:_PI---接收端退出登录消息) | /UserExit | POST | ①设备ID |
| 4 | [接收测试/制程/产线状态数据](#_4）编号4:_PI-接收测试/制程/产线状态数据) | /ProducedDatas |  | ①当站基本信息 ②实时数据 |
| 5 | [接收换线结果](#_5）编号5:_PI---接收换线结果) | /ChangeLineResult | POST | ①”OK”/”FAIL”/”Waiting”/”Changing”换线成功/失败/等待换线/正在换线 |
| 6 | [接收心跳信号](#_6）编号6:_PI---接收心跳信号) | /HeartbeatSignal | POST | ①设备ID ②”It’s OK” |
| 7 | [接收换线指令（CCS到PI）](#_8）编号8:_PI---接收换线指令) | /ChangeLine | POST | ①设备ID ②工单 ③机种名 |
| 8 | [网页实时显示数据](#_9）编号9:_PI---网页实时显示数据->待完善)  (SocketIO) | monitor\_server\_getdata | 向网页推送实时数据 | |
| monitor\_server\_message | 向网页推送模组当前状态,网页弹框提示用户 | |
|  | | | | | |
| **CCS** | 9 | [接收模组登录消息](#_10）编号10:_MLCM---接收模组登录消息) | /LoginData | POST | ①设备ID ②端IP ③端口号  ④当站信息 |
| 10 | [接收模组退出登录消息](#_11）编号11:_MLCM---接收模组退出登录消息) | /ExitData | POST | ①设备ID |
| 11 | [接收测试/制程/产线状态数据](#_12）编号12:_MLCM-接收测试/制程/产线状态数据) | /ProducedDatas | POST | ①设备ID ②当站基本信息  ③实时数据 |
| 12 | [接收换线结果](#_13）编号5:_MLCM---接收换线结果) | /ChangeLineResult | POST | ①设备ID ②”OK”/”FAIL” ③结果描述 |
| 13 | [接收心跳信号](#_14）编号14:_MLCM---接收心跳信号) | /HeartbeatSignal | POST | ①设备ID ②”It’s OK” |
| 14 | [接收当站状态变化](#_15）编号15:_MLCM---接收当站状态变化) | /StateChange | POST | ①设备ID  ②状态：设备离线, 设备自检FAIL，设备在线,等待换线,正在换线,模组掉线（该状态由CCS根据心跳信号判断） |
| 15 | [网页实时显示数据(SocketIO)](#_16）编号16:_MLCM---网页实时显示数据->待完善) |  |  | |
|  | | | | | |

## 2.2 函数式接口

设备接口放在interface文件夹里，每个设备接口用设备型号\*作为\*.py的名称，如AT-11.py，\*.py里包含发送数据函数，接收换线指令函数，同时上传数据到MES/PQM.

### ****1）发送数据函数****

* **函数名：**get\_productdata (argsnum, uploadMode, modelName)
* **函数功能：**产生产品的制程数据和上传数据至MES/PQM
* **传递参数：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| argsnum | List | 用户选择参数的标号, 形如：[100,102,…] |
| uploadMode | List | 用户选择的上传数据方式, "1"表示MES, "2"表示PQM, 形如：["1"] |
| modelName | String | 当前机种名 |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| productData | List | 'machine\_state':设备状态’ON’/’OFF’; 'data': [[标号1,值1]，[标号2,值2],…]  'ErrorMes':[1, '']或[5, '保护状态']  示例：  productData =  {  'machine\_state': 'ON',  'data': [[100, 350.19], [101, 14.6]],  'ErrorMes': [1, '']  } |
| processDatas | List | 过程数据 |

### ****2）发送换线指令函数****

* **函数名：send\_**changeline(new\_mode\_name)
* **函数功能：向设备发送换线指令**
* **传递参数：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| new\_model\_name | string | 新机种名称 |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| current\_state | string | 当前设备换线状态->"11":换线成功, "12":换线失败, "13":重复换线  示例：  current\_state="11" |

## 2.3 Lora接口---待完善

# 3. API详情

## ****1）编号1: 端---接收换线指令****

* **简要描述：** 接收PI发来的换线指令
* **请求URL：**  ‘ http:// ‘ + 端IP + ’:5000/ ChangeLine’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |
| MO\_NUMBER | String | 新工单 |
| MODEL\_NAME | String | 新机种 |

* **传递参数示例：**

|  |
| --- |
| {  "deviceID ": "S1234567",  " MO\_NUMBER ": "20210408015",  " MODEL\_NAME ": "DPS-400AB-31 A"  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## **2）编号**2: PI---接收端登录消息****

* **简要描述：** 接收端发来的登录信息
* **请求URL：**  ‘ http:// ‘ + 模组IP + ’:5000/UserLogin’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| userIP | String | 端的IP |
| userPort | String | 换线指令用到的端口号 |
| baseInfor | String | 端传输当前站基本信息. "工单}机种}线别}段别}组别}站别}"，用右大括号隔开 |

* **传递参数示例：**

|  |
| --- |
| {  " userIP ": "169.234.27.1",  " userPort ": “5000",  " baseInfor ": "20120408123}DPS-400AB-31 A}S03}2次站}第一段}焊接站}"  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "deviceID" //将用户配置的设备ID发给端  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## ****3）**编号3**: PI---接收端退出登录消息****

* **简要描述：** 接收端发来的退出登录信息
* **请求URL：**  ‘ http:// ‘ + 模组IP + ’:5000/ UserExit’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |

* **传递参数示例：**

|  |
| --- |
| {  "deviceID ": "S1234567"  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因, 一般FAIL原因有解析出错或无权限访问 |

## 4）****编号4: PI---接收测试/制程/产线状态数据****

* **简要描述：** 接收端传来的数据(测试/制程/产线状态)
* **请求URL：** ‘ http:// ‘ + 模组IP + ’:5000/ProducedDatas’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| baseInfor | String | PC端传输当前站基本信息. "工单}机种}线别}段别}组别}站别}"，用右大括号隔开 |
| lineData | Array | 产线具体数据，参数格式如下：  [ [SN, 测试/采集时间, 测试/采集条件, 参数类型, 参数名1, 单位, 参数上限值, 参数下限值, 参数值, 判定结果, 备注], [……],……]  **SN** :没有时为”NA”；某栏位没有为空””;  **测试/采集条件:**分隔符用右大括号 ‘}’ ，自检数据最前面加C}，测试数据最前面加T}；  **参数类型**：当实际数据为过程值时参数类型=”1”，参数值均为一个数组，反之参数类型=“0”，参数值为一个字符串数值;  **参数名**：当出现参数一致且规格不同时应以参数名+通道为最终参数名 |

* **传递参数示例：**

|  |
| --- |
| {  " baseInfor ": "20120408123}DPS-400AB-31 A}S03}2次站}第一段}焊接站}",  " lineData ": [ ["123455534", "2012-04-08 11:04:23", "T}电压100v}电流200A}","0", "电流", "A","200", "100", "150","PASS", ""], ,…… ]  } |

* **返回示例：**

|  |  |
| --- | --- |
| Case A:  {  " result ": "OK",  "description ": "…"  } | Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## 5）****编号5: PI---接收换线结果****

* **简要描述：** 接收端发来的换线结果
* **请求URL：**  ‘ http:// ‘ + 模组IP + ’:5000/ChangeLineResult’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| changeLineResult | String | 换线结果”OK”/”FAIL”/”Waiting”/”Changing” |
| description | String | 结果具体描述说明,OK为空，FAIL时为换线失败原因 |

* **传递参数示例：**

|  |
| --- |
| Case A:  {  "changeLineResult ": "OK",  "description ": "…"  }  Case B:  {  "changeLineResult ": "FAIL",  "description ": "设备故障"  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## 6）****编号6: PI---接收心跳信号****

* **简要描述：** 接收端发来的心跳信号, 定时频率为10s
* **请求URL：**  ‘ http:// ‘ + 模组IP + ’:5000/HeartbeatSignal’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| heartSignal | String | 心跳信号 "It’s OK" |

* **传递参数示例：**

|  |
| --- |
| {  " heartSignal ": "It‘s OK "  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## 7）****编号7: PI---接收换线指令****

* **简要描述：** 接收CCS发来的换线指令
* **请求URL：**  ‘ http:// ‘ + 模组IP + ’:5000/ ChangeLine’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |
| MO\_NUMBER | String | 新工单 |
| MODEL\_NAME | String | 新机种 |

* **传递参数示例：**

|  |
| --- |
| {  " deviceID ": "S1234567",  " MO\_NUMBER ": "20210408015",  " MODEL\_NAME ": "DPS-400AB-31 A"  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## ****8）编号8: PI---网页实时显示数据->待完善****

* **命名空间**: namespace='/test'，
* **自带事件：**
* 连接事件: socketio.on("connect", namespace='/test')；
* 断开事件: socketio.on("disconnect", namespace='/test')；
* 错误事件: socketio.on("error", namespace='/test').
* **自定义事件：**
* 获取监控界面数据事件: "monitor\_server\_getdata";
* 消息弹框显示事件: "monitor\_server\_showmessage".

## **9）编号9**: CCS---接收模组登录消息****

* **简要描述：** 接收模组发来的登录信息
* **请求URL：**  ‘ http:// ‘ + CCS的IP + ’:5000/LoginData’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |
| userIP | String | 模组IP, 换线指令会用到 |
| userPort | String | 换线指令用到的端口号 |
| baseInfor | String | 端传输当前站基本信息. "工单}机种}线别}段别}组别}站别}"，用右大括号隔开  有内容照常填写，无内容为空”” |

* **传递参数示例：**

|  |
| --- |
| {  " deviceID ": "S1234567",  " userIP ": "169.234.27.1",  " userPort ": “5000",  " baseInfor ": "12156630}213}S21} } } }"  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## **10）编号10**: CCS---接收模组退出登录消息****

* **简要描述：** 接收模组发来的退出登录信息
* **请求URL：**  ‘ http:// ‘ + CCS的IP + ’:5000/ExitData’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |

* **传递参数示例：**

|  |
| --- |
| {  "deviceID ": "S1234567",  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## 11）****编号11: CCS-接收测试/制程/产线状态数据****

* **简要描述：** 接收模组传来的数据(测试/制程/产线状态)
* **请求URL：** ‘ http:// ‘ + CCS的IP + ’:5000/ProducedDatas’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |
| baseInfor | String | PC端传输当前站基本信息. "工单}机种}线别}段别}组别}站别}"，用右大括号隔开 |
| lineData | Array | 产线具体数据，参数格式如下：  [ [SN, 测试/采集时间, 测试/采集条件, 参数类型, 参数名1, 单位, 参数上限值, 参数下限值, 参数值, 判定结果, 备注], [……],……]  **SN** :没有时为”NA”；某栏位没有为空””;  **测试/采集条件:**分隔符用右大括号 ‘}’ ，自检数据最前面加C}，测试数据最前面加T}；  **参数类型**：当实际数据为过程值时参数类型=”1”，参数值均为一个数组，反之参数类型=“0”，参数值为一个字符串数值; 参数类型=”2”时产线状态数据，如直通率，达成率，此时只有时间，参数名和参数值,SN可能会有；  **参数名**：当出现参数一致且规格不同时应以参数名+通道为最终参数名 |

* **传递参数示例：**

|  |
| --- |
| {  " deviceID ": "S1234567",  " baseInfor ": "20120408123}DPS-400AB-31 A}S03}2次站}第一段}焊接站}",  " lineData ": [ ["123455534", "2012-04-08 11:04:23", "T}电压100v}电流200A}","0", "电流", "A","200", "100", "150","PASS", ""] ,…… ]  } |

* **返回示例：**

|  |  |
| --- | --- |
| Case A:  {  " result ": "OK",  "description ": "…"  } | Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## ****12）编号12: CCS---接收换线结果****

* **简要描述：** 接收模组发来的换线结果
* **请求URL：**  ‘ http:// ‘ + CCS的IP + ’:5000/ChangeLineResult’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |
| changeLineResult | String | 换线结果”OK”/”FAIL” |
| description | String | 结果具体描述说明,OK为空，FAIL时为换线失败原因 |

* **传递参数示例：**

|  |
| --- |
| Case A:  {  " deviceID ": "S1234567",  "changeLineResult ": "OK",  "description ": "…"  }  Case B:  {  " deviceID ": "S1234567",  "changeLineResult ": "FAIL",  "description ": "设备故障"  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## 13）****编号13: CCS---接收心跳信号****

* **简要描述：** 接收模组发来的心跳信号, 定时频率为10s
* **请求URL：**  ‘ http:// ‘ + CCS的IP + ’:5000/HeartbeatSignal’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |
| heartSignal | String | 心跳信号 "It’s OK" |

* **传递参数示例：**

|  |
| --- |
| {  " deviceID ": "S1234567",  " heartSignal ": "It‘s OK "  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## 14）****编号1****4****: CCS---接收当站状态变化****

* **简要描述：** 接收模组发来的当站状态变化
* **请求URL：**  ‘ http:// ‘ + CCS的IP + ’:5000/StateChange’
* **请求方式：** POST
* **传递参数:**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| deviceID | String | 设备ID |
| stateChange | String | 状态变化，包括online，offline，waiting, changing  分别对应设备在线，设备离线，设备自检FAIL，等待换线，正在换线，  另外根据心跳信号CCS判断模组是否掉线，可能原因没网/模组没开机）  即共5种状态 |

* **传递参数示例：**

|  |
| --- |
| {  " deviceID ": "S1234567",  " stateChange ": "offline "  } |

* **返回示例：**

|  |
| --- |
| Case A:  {  " result ": "OK",  "description ": "…"  }  Case B:  {  " result ": "FAIL",  "description ": "Fail Message"  } |

* **返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| result | String | "OK"：表示无错误; "FAIL"：表示有错误 |
| description | String | 结果具体描述说明,OK为空，FAIL时为FAIL原因 |

## 15）****编号1****5****: CCS---网页实时显示数据->待完善****

* **命名空间**: namespace='/CCS'，
* **自带事件：**
* 连接事件: socketio.on("connect", namespace='/CCS')；
* 断开事件: socketio.on("disconnect", namespace='/CCS')；
* 错误事件: socketio.on("error", namespace='/CCS').
* **自定义事件：**
* 获取监控界面数据事件: "monitor\_server\_getdata";
* 消息弹框显示事件: "monitor\_server\_showmessage".

# **4.HTTP加密方式**

采用HTTP头部增加’token’形式进行加密, 具体token计算方式如下：

1. 由登录接口所有参数字符串依次相连（连接顺序按照接口定义的顺序即可）;
2. 在第一步最前面加上secertkey（该密钥转为二进制存入本地配置文件方便后期更改）;
3. 将第二步的一长串字符串进行MD5加密,并转成大写;
4. 第三步得到的字符串即为token;
5. 客户端在登录成功后访问所有的HTTP 接口均要在头部加上token;
6. 同理服务端在接收登录信息后进行以上计算得到token, 用于验证客户端的访问.

示例：

密钥secertkey = ‘1111111111’（由TTDD提供）

登录接口参数有: userIP = ‘127.0.0.1’, userPort = ’4000’,

baseInfor=’20120408123}DPS-400AB-31 A}S03}2次站}第一段}焊接站}’

此时token 计算方式如下：

1. 连接字符串：secertkey + userIP + userPort+ baseInfor

即为 1111111111127.0.0.1400020120408123}DPS-400AB-31 A}S03}2次站}第一段}焊接站}

1. 将上面字符串进行MD5加密并转大写,即为

token = D5AF66DE0C5925BEB6794FCED99010DA

1. 此时头部headers = {"Content-Type": "application/json","token":token}