**云网关设备-快速接入文档**

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **修订版本** | **修改描述** | **作者** |
| 2020-4-14 | V1.0 | 添加 | YinRui |

**缩略语清单**

|  |  |  |
| --- | --- | --- |
| **缩略语** | **英文全名** | **中文解释** |
|  |  |  |
|  |  |  |

# 说明

便于用户能够快速地接入设备到平台平台，故编写此说明文档。

# 名词解析

PID：产品ID，创建产品时平台生成的产品唯一性数字标识

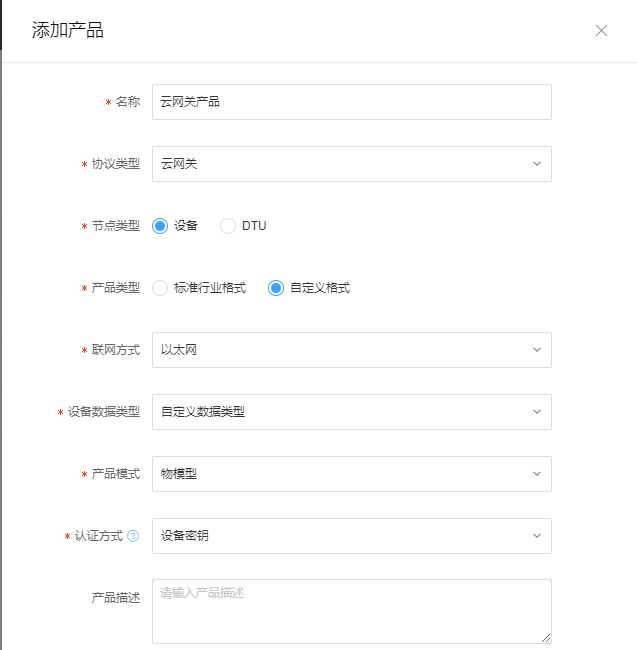
DID：产品ID，创建产品时平台生成的产品唯一性数字标识

AuthCode: 根据创建设备平台生成的设备密钥与设备PID和DID加密计算后得到

# 接入流程

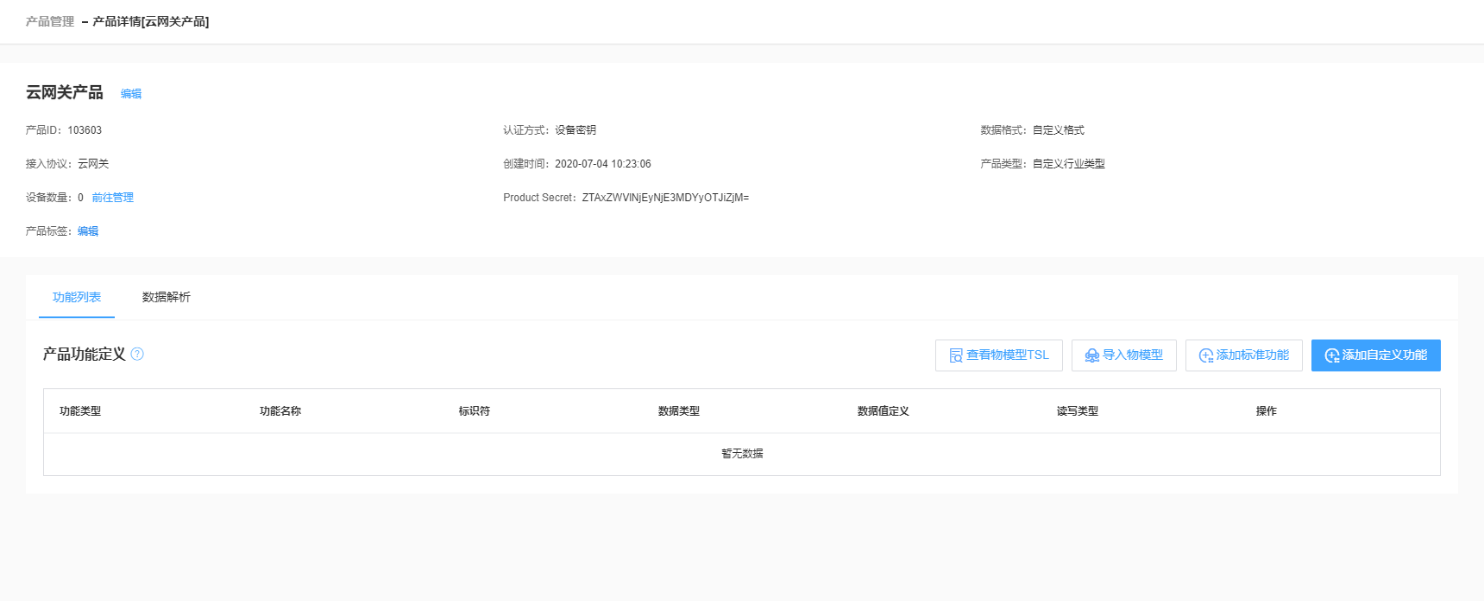
**第1步 创建产品**

登录平台，在左侧导航栏选择**物联网平台**，选择**设备管理->产品列表**根据实际情况创建自己的产品，如下图所示



**第2步 创建功能点**

创建好产品后在产品详情页**功能列表**点击**添加自定义功能**添加产品功能点，（此文档以 “temperature”举例说明）

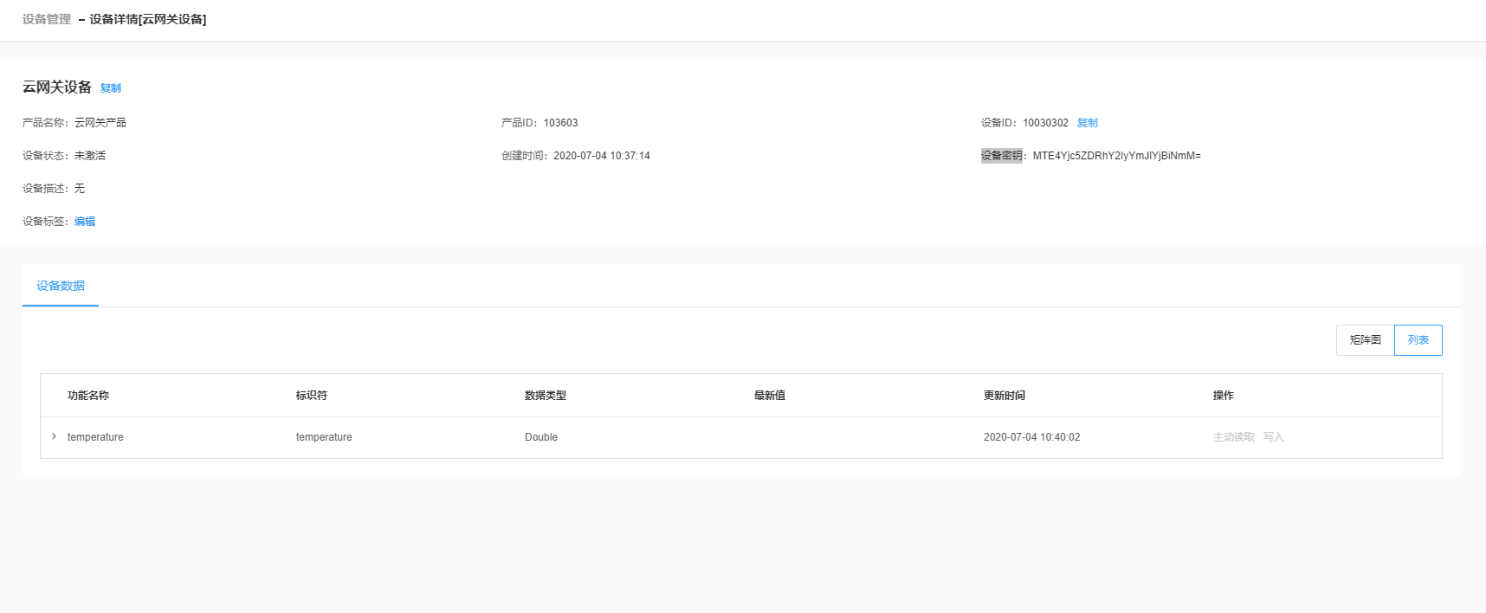




**第3步 创建设备**

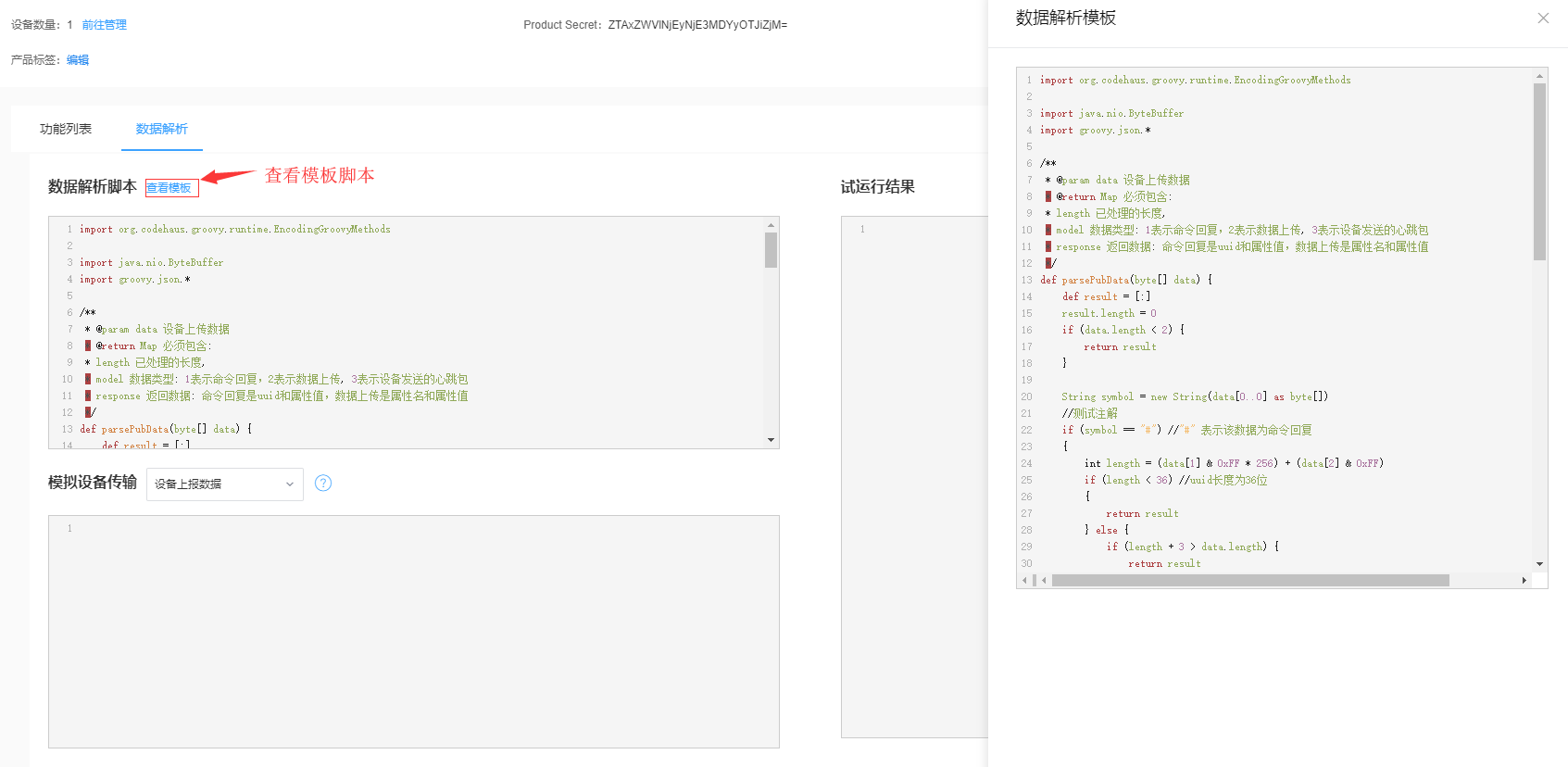
选择左侧导航栏的**设备管理->设备管理**，点击**添加设备**按钮进行添加设备，（此文档以 “云网关设备”举例说明）,如下图所示:

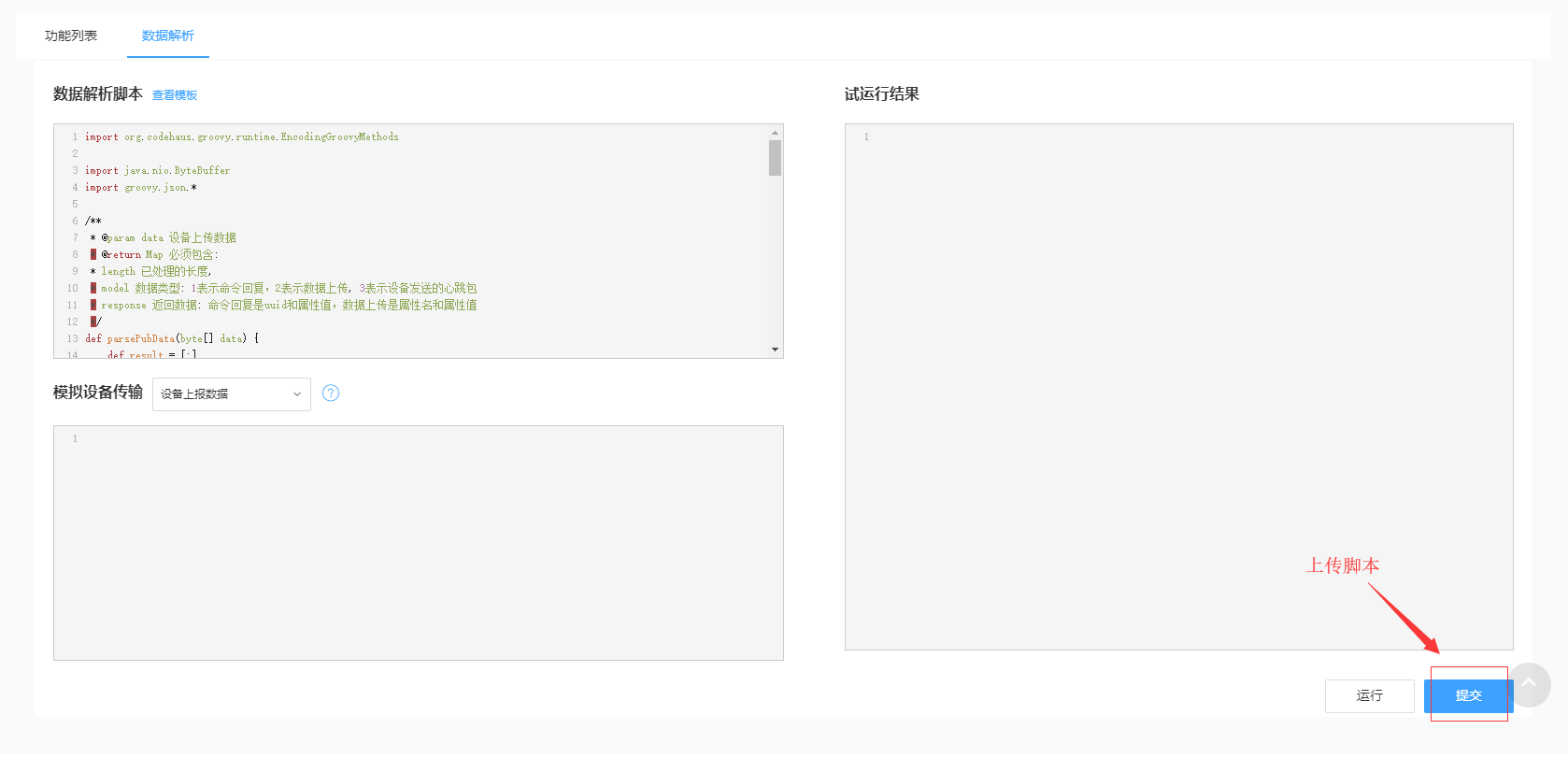




**第4步 上传解析脚本**

进入云网关产品的**详情**页面，点击**数据解析**标签卡，点击**查看模板**复制其中内容粘贴到“数据解析脚本“下方的脚本输入框，然后点击**提交**来上传脚本。如下图所示：





**第5步**

设备上电，发送登录报文到平台接入服务器进行鉴权。

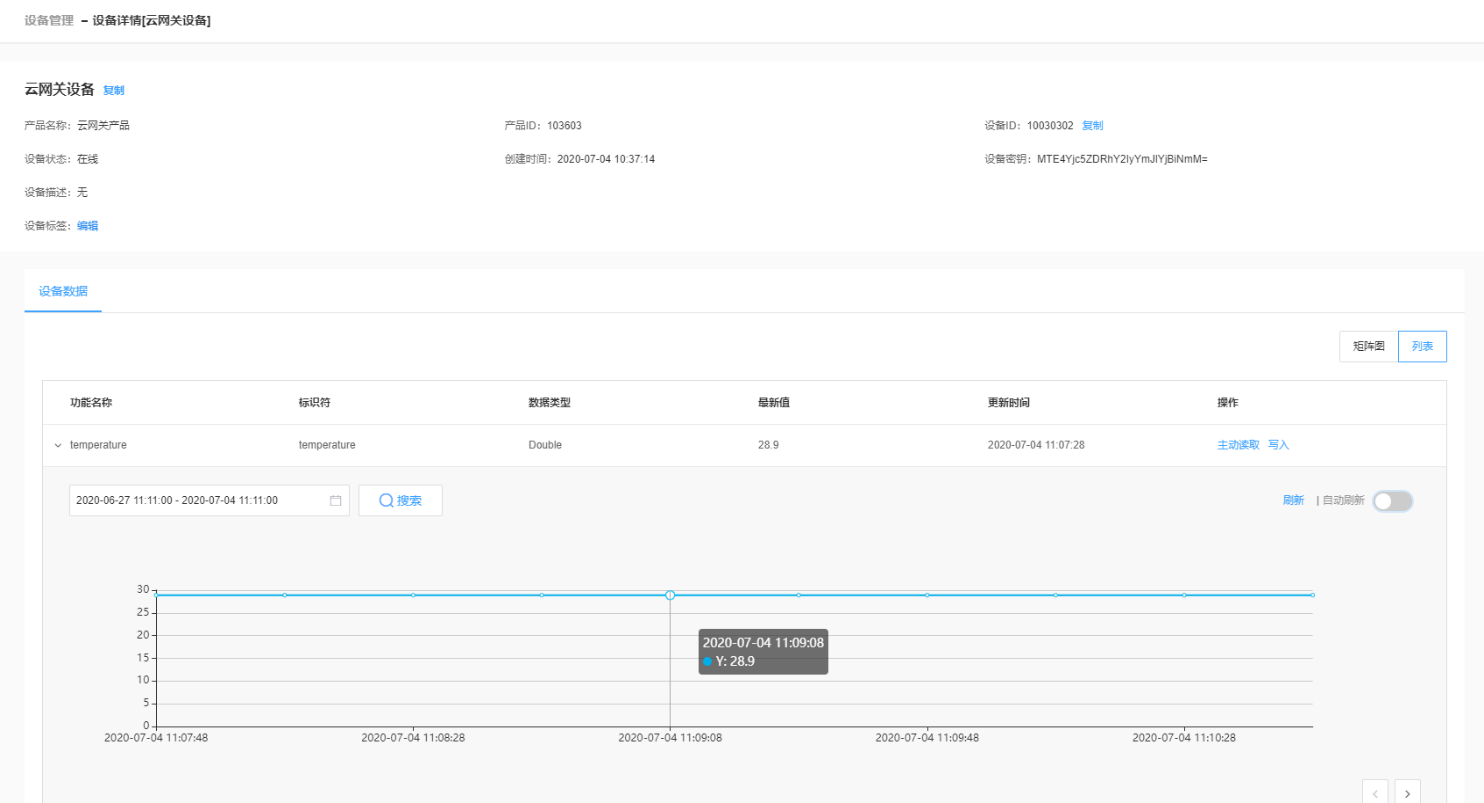
登录报文格式如下：

\*PID#DID#AuthCode\*

**第6步 查看设备数据点**

设备登录成功后，平台接入服务会加载登录报文中指定的脚本，然后根据脚本报文解析规则来解析设备上报的数据

进入云网关设备的**详情**页面，查看数据点情况，如下图所示：



# groovy脚本编写说明

* 1. **解析设备上传的数据**

脚本中需要定义名字为 “parsePubData”的方法，平台再收到设备上报数据时会调用该方法。方法传入参数为 byte[] 类型，代表设备上传的二进制数组报文。方法需返回为map，其中key为： “length”、 “model”、“response”。

length:18 //表示此次方法执行完成后方法入参byte[]数组所消耗字节数

model:2 //表示此次设备上传数据类型，1表示命令回复，2表示数据上传, 3表示设备发送的心跳（无实际意义，丢弃）

response: “temperature”:36 //表示设备上传报文的具体意义，如果设备上报数据为命令回复，则内容为是uuid和属性值（541168b5-8b8b-4753-92eb-61bde86dbc23value）如果设备上报数据内容为 设备属性名称和设备属性值（“temperature”:36）

* 1. **解析下发至设备数据**

脚本中需定义“parseCommandData”的方法用于将平台下发至设备的命令转换为设备所能识别的数据。方法入参为：“uuid”、 “params” 、“functionType”。

uuid:“541168b5-8b8b-4753-92eb-61bde86dbc23” //每条命令都带有一个uuid设备返回命令执行结果时需携带该uuid

params:“[“temperature”,…]”or{“temperature”:23,… } 如果为读命令，该参数为一个设备属性名称组成的字符串数组,如果为写命令，则该参数为多个属性名与属性值组成的map

functionType: “propertySet” //下发命令类型，propertySet表示设置值，propertyGet表示读取值

# 设备连接鉴权AuthCode计算方式

设备连接平台时AuthCode需要根据设备DID和PID以及device secret计算。首先将 did + “&“ + pid进行拼接，作为密文原始报文，再将device secret作为key先做sha1，再做哈希信息加密。最后做base64转化。

Go语言代码示例如下：

var token string

queryStr := "did" + "&" + "pid"

key := []byte("deviceSecret")

mac := hmac.New(sha1.New, key)

mac.Write([]byte(queryStr))

token = base64.StdEncoding.EncodeToString(mac.Sum(nil))

JAVA语言代码示例如下：

String queryStr = did + "&" + pid;

String key = deviceSecret;

SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(), "HmacSHA1");

Mac mac = Mac.getInstance("HmacSHA1");

mac.init(secretKeySpec);

byte[] bytes = mac.doFinal(queryStr.getBytes());

String result = Base64.encodeBase64String(bytes);