基建感知层通信协议规范

2021年5月10日

目录

[1 范围 1](#_Toc71580637)

[2 缩略语及名词解释 1](#_Toc71580638)

[3 专题感知设备及通信接口 1](#_Toc71580639)

[4 边缘物联代理装置接口协议 3](#_Toc71580640)

[5 感知层总体通信架构 3](#_Toc71580641)

[6 边缘物联代理与基建平台交互协议规范 5](#_Toc71580642)

[6.1 通信架构 5](#_Toc71580643)

[6.2 风险作业获取 5](#_Toc71580650)

[6.3 作业票信息获取 8](#_Toc71580651)

[6.4 人员信息获取 11](#_Toc71580652)

[7 边缘物联代理与物联管理平台交互协议规范 14](#_Toc71580653)

[7.1 通信架构 14](#_Toc71580654)

[7.2 规约结构 14](#_Toc71580655)

[7.3 交互机制 15](#_Toc71580656)

[7.4 报文格式 15](#_Toc71580657)

[7.5 交互主题 16](#_Toc71580658)

[8 边缘物联代理与业务APP及端设备交互协议规范 17](#_Toc71580659)

[8.1 通信架构 17](#_Toc71580660)

[8.2 边缘物联代理内部交互及Mqtt协议接入 18](#_Toc71580661)

[8.3 MODBUS协议接入 33](#_Toc71580662)

[8.4 二进制协议接入 36](#_Toc71580663)

[8.5 HTTP协议接入 40](#_Toc71580664)

[9 可扩展性 55](#_Toc71580671)

1. 范围

本规范规定了适用于基建现场感知层应用中边缘物联代理、基建业务平台、基建业务应用和现场各类感知设备之间的通信传输规约及数据通信协议。

本规范适用于基建现场感知层应用中边缘物联代理、基建业务平台、基建业务应用和现场各类感知设备通信接口的设计、开发、检验和验收。

1. 缩略语及名词解释

E基建APP：基建全过程数字化综合管理平台移动端应用软件

业务APP：各专题基于业务需求开发的应用程序，后台部署在边缘物联代理上，前端是E基建APP一部分。

LoRa：基于扩频技术的超远距离无线传输方案

LoRaWAN：LoRaWAN是为LoRa远距离通信网络设计的一套通讯协议和系统架构。在协议和网络架构的设计上，充分考虑了节点功耗，网络容量，QoS，安全性和网络应用多样性等几个因素。

蓝牙5.0：蓝牙5.0是由蓝牙技术联盟在2016年提出的蓝牙技术标准，蓝牙5.0针对低功耗设备有相应提升和优化，提高传输速度，增加有效工作距离，与4.2LE版本相比其功耗更低，且兼容老的版本。

Modbus：一种串行通信协议

MQTT：消息队列遥测传输（Message Queuing Telemetry Transport）

NFC：近距离无线通信技术（Near Field Communication）

RFID：射频识别（Radio Frequency Identification）

RS485：平衡传输标准

RS232：串行二进制数据交换接口技术标准

RJ45：是标准计算机网络的8位模块化接口

Wi-Fi：无线局域网(Wireless Fidelity)

TCP/IP：传输控制协议/网际协议

HTTP：超文本传输协议（Hypertext Transfer Protocol，HTTP）

HTTPS：安全版超文本传输协议（Hyper Text Transfer Protocol over SecureSocket Layer，HTTPS）

1. 专题感知设备及通信接口

根据2021年六大专题提供的感知层工作方案及成果交付物，梳理统计六大专题感知层设备类型及接口信息，统计信息如下表所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **专题** | **子专题** | **包含设备** | **接口** | **协议** |
| 人员状态自动感知 | 人工智能视频识别设备 | 布控球 | Wi-Fi、RJ45 | HTTP、HTTPS |
| 区域（UWB）定位设备 | UWB | RJ45 | HTTP、HTTPS |
| 心率监测手环 | 手环 | Wi-Fi、LoRa | HTTP、HTTPS、私有协议 |
| 智能班组管控箱 | 智能班组管控箱 | Wi-Fi | HTTP、HTTPS |
| 环境自动监测 | 室外环境监测站 | 室外环境监测设备 | LoRa | 私有协议 |
| 风险环境监测 | 风险环境监测设备 | LoRa | 私有协议 |
| 有限空间环境监测 | 有限空间环境监测设备 | LoRa | 私有协议 |
| 边坡位移监测 | 边坡监测 | RS232、Wi-Fi、LoRa | MQTT、HTTP、私有协议 |
| 沉降监测 | 沉降监测 | RS232、Wi-Fi、LoRa | Modbus、HTTP、私有协议 |
| 工程成品质量变形分析监测 | 测距仪、静力水准仪、多点位移计、雨量计、土压计、沉降计 | RS232、Wi-Fi、LoRa | Modbus、HTTP、私有协议 |
| 施工装备智能化改造 | 组塔抱杆施工状态监测 | 拉力传感器、倾角传感器 | LoRa、WI-FI | MQTT、私有协议 |
| 施工机械管理 | 施工机械管理设备 | LoRa | 私有协议 |
| 电缆敷设状态监测 | 拉力传感器 | Wi-Fi、LoRa | MQTT、私有协议 |
| 施工装备智能化改造 | 真空净油机组、真空抽气机组 | RS232、Wi-Fi、LoRa | MQTT、私有协议 |
| 施工装备物联状态监测 | 滤油机、牵张机、六氟化硫回收装置、压接机 | RS485、RJ45、WI-FI、LoRa | MQTT、私有协议 |
| 桩基作业智能监测 | 碎石桩、CFG桩、水泥搅拌桩 | Wi-Fi | MQTT、私有协议 |
| 压实作业智能监测 | 压实传感器 | Wi-Fi | MQTT、私有协议 |
| 施工现场智能识别 | 视频识别 | 布控球、枪/球机、激光点云、无人机、手持终端等 | RJ45、Wi-Fi、RS485 | HTTP、HTTPS、MQTT |
| 质量检验检测工具 | 交接试验仪器设备 | 直流高压发生器、SF6气体综合测试仪、SF6气体检漏仪、SF6密度继电器校验仪 | Wi-Fi | 私有协议 |
| 实测实量设备 | 蓝牙扭矩扳手、蓝牙激光测距仪、沉降观测传感器、埋件标高传感器 | Wi-Fi、蓝牙 | 私有协议 |
| 无人机线路交跨测量 | 无人机 | Wi-Fi | 私有协议 |
| 安全工器具智 | 安全工器具 | 定位安全帽、全功能型安全帽、安全带、穿戴式近电感知设备 | Wi-Fi、LoRa | MQTT、私有协议 |

1. 边缘物联代理装置接口协议

通过对现场感知层传感设备的调研及六大专题感知传感设备接入需求与接口的收集整理，现场感知层传感设备主要通过RJ45、Wi-Fi、LoRa、蓝牙及串口等接口接入边缘物联代理，通过MQTT、MODBUS、HTTP及私有协议进行传感数据的采集。

由于现场传感设备都是不同设备厂商独自接入，通信协议不统一，新设备接入需要重新开发应用程序，多个采集程序的部署造成边缘物联代理内存资源占用。为了更好的兼容现场感知层传感设备的接入要求，实现传感网络接口标准化，边缘代理设备目前配备了RJ45、Wi-Fi、LoRa、蓝牙、串口五类接口。为便于感知层传感设备的统一接入及通信通道畅通,边缘物联代理制定统一的设备接入通信规约，感知层各类传感设备基于统一通信规约通过MQTT、MODBUS、HTTP、二进制协议与边缘物联代理进行通信交互。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **接口类型** | **传输协议** | **接口协议** | **频段** |
| 1 | 蓝牙 | 二进制 | 5.0 | 2400~2483.5MHz |
| 2 | LoRa | 二进制 | LoRaWAN | 470~510MHz |
| 3 | Wi-Fi | HTTP/MODBUS/MQTT | TCP/IP | 2.4G/5.8G |
| 4 | RJ45 | HTTP/MODBUS/MQTT | TCP/IP | -- |
| 5 | 串口 | MODBUS | RS485/RS232 | -- |

1. 感知层总体通信架构



1. 感知层通信架构图

感知层的整体通信架构如上图所示，根据不同的业务应用场景，主要分为直接上传到省侧平台的视频类数据、通过边缘代理上传至省侧平台的现场侧数据及直接上传至省测平台的实验类数据三种。

* 视频类数据直接通过电网视频监控B接口协议上送数据到内外网统一视频平台；同时边缘物联代理通过ONVIF协议获取视频类设备的实时图像数据并进行本地智能分析运算，根据HTTP协议将分析结果上传至基建平台、或通过物管平台根据MQTT协议上传至基建平台。
* e基建APP通过蓝牙采集实验类设备及实测实量设备的数据，根据HTTP协议与基建平台进行数据交互。
* 部分现场感知设备根据HTTP协议接入到省公司基建平台或通过物管平台根据MQTT协议与基建平台进行数据交互。
* 现场侧智能传感设备通过边缘代理统一进行数据采集、整理、加密，并根据HTTP协议与基建平台进行数据交互或通过物管平台根据MQTT协议与基建平台进行数据交互。
* 边缘物联代理与基建平台进行通信时可根据安全要求采用HTTPS数据加密方式进行传输。
* 边缘计算框架采用MQTT协议上送物管平台，基建平台可根据HTTP、MQTT或Kafka协议与物管平台交互。

1. 边缘物联代理与基建平台交互协议规范
   1. 通信架构

本部分定义了边缘物联代理（简称边设备）与省公司基建平台通信中对http协议的使用。如图二所示。



1. 通信通信架构示意图
   1. 项目信息获取
      1. 接口描述

项目信息获取属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + 1. 接口流程

项目信息获取的接口流程见下图。



1. 风险作业获取流程

主要功能流程如下：

1. F1：边缘物联代理发送HTTP请求获取项目名称，消息中携带模糊匹配的项目名称条件；
2. F2：省公司基建平台返回符合项目信息列表。
   * 1. 接口参数
        1. HTTP响应码

HTTP响应码见表1。

表1 HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - 1. 参数定义

项目信息获取请求命令相关的参数定义见表2。

表2查询项目信息请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| projectName | 必选 | String | 项目名称 |

项目信息请求响应相关的参数定义见表3。

表3 项目信息获取的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| id | 必选 | INT32 | 项目ID |
| projectname | 必选 | INT32 | 项目名称 |
| projectno | 必选 | String | 项目编码 |
| buildergno | 必选 | String | 建管单位编码 |
| buildergname | 必选 | String | 建管单位 |
| vollevle | 必选 | String | 电压等级 |
| projecttype | 必选 | String | 项目类型 |

* + 1. 消息示例
       1. 项目信息获取请求

请求URL：http://ip:port/ getProjectByName

请求方式：POST

表头：Content-Type:application/json

{

" projectName ": "项目名称" 按照项目名称模糊查询，为空返回所有项目列表

}

* + - 1. 项目信息获取响应

{

"code":"",

"msg":"ok",

"data":{

"id": "", // 项目ID

"projectname": "", // 项目名称

"projectno": "", // 项目编码

"builderno": "", // 建管单位编码

"buildername": "", // 建管单位

"vollevle": "", // 电压等级

"projecttype": "" // 项目类型

}}

* 1. 单项信息获取
     1. 接口描述

单项信息获取属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + 1. 接口流程

风险作业获取的接口流程见下图。



1. 单项信息获取流程

主要功能流程如下：

1. F1：边缘物联代理发送HTTP请求获取单项信息，消息中携带项目编码条件；
2. F2：省公司基建平台返回符合单项信息列表。
   * 1. 接口参数
        1. HTTP响应码

HTTP响应码见表1。

表1 HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - 1. 参数定义

请求命令相关的参数定义见表2。

表2单项信息获取请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| prjNo | 必选 | String | 项目编码 |

响应相关的参数定义见表3。

表3 单项信息获取的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| id | 必选 | INT32 | 单项id |
| SINGLENAME | 必选 | INT32 | 项目单体名称 |
| singleno | 必选 | String | 项目单体编码 |
| wbscode | 必选 | String | WBS编码 |
| prjno | 必选 | String | 项目编码 |
| singlecon | 必选 | String | 单项工程描述 |
| vollevle | 必选 | INT32 | 电压等级 |
| linelength | 必选 | INT32 | 可研线路长度 |
| subscap | 必选 | INT32 | 可研变电容量 |
| substypecode | 必选 | INT32 | 变电站类型包括:01 土建,02 电气,03 变电 |
| estimatetypecode | 必选 | String | 估算类型 |
| buildorgno | 必选 | String | 唯一标识建设管理单位 |
| startdate | 必选 | String | 实际开工时间 |
| findate | 必选 | String | 实际投产时间 |
| compdate | 必选 | String | 实际竣工时间 |

* + 1. 消息示例
       1. 单项信息获取请求

请求URL：http://ip:port/ /project/singleProject/getPrjsinInfoByPrjNo

请求方式：POST

表头：Content-Type:application/json

{

"prjNo": "项目编码"

}

* + - 1. 单项信息获取响应

{

"code":0,

"msg":"ok",

data:{

id:"单项id",

SINGLENAME:"项目单体名称",

singleno:"项目单体编码",

wbscode:"WBS编码",

prjno:"项目编码",

singlecon:"单项工程描述",

vollevle:"电压等级",

linelength:"可研线路长度",

subscap:"可研变电容量",

substypecode:"变电站类型包括:01 土建,02 电气,03 变电",

estimatetypecode:"估算类型",

buildorgno:"唯一标识建设管理单位",

startdate:"实际开工时间",

findate:"实际投产时间",

compdate:"实际竣工时间",

}

}

* 1. 标段信息获取
     1. 接口描述

标段信息获取属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + 1. 接口流程

标段信息获取的接口流程见下图。



1. 标段信息获取流程

主要功能流程如下：

1. F1：边缘物联代理发送HTTP请求获取标段信息；
2. F2：省公司基建平台返回符合条件的标段信息列表。
   * 1. 接口参数
        1. HTTP响应码

HTTP响应码见表1。

表1 HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - 1. 参数定义

请求命令相关的参数定义见表2。

表2标段信息获取请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| sinCodeList | 必选 | String | 单项编码 |

响应相关的参数定义见表3。

表3 标段信息获取的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| id | 必选 | INT32 | 主键ID |
| BIDCODE | 必选 | String | 标段编码 |
| BIDNAME | 必选 | String | 标段名称 |
| BIDDESC | 必选 | String | 标段描述 |
| prjCode | 必选 | String | 项目编码 |
| STATUS | 必选 | String | 数据状态 |
| bidType | 必选 | INT32 | 0：设计，1：监理，2：施工，3：监理加施工 |
| chiefSuperVice | 必选 | String | 监理总工程师 |
| buildManager | 必选 | String | 施工项目经理 |
| substypecode | 必选 | INT32 | 变电站类型包括:01 土建,02 电气,03 变电 |
| startdate | 必选 | String | 实际开工时间 |
| findate | 必选 | String | 实际投产时间 |
| compdate | 必选 | String | 实际竣工时间 |

* + 1. 消息示例
       1. 标段信息获取请求

请求URL：http://ip:port /project/designBid/getSGTenderListBySinList

请求方式：POST

表头：Content-Type:application/json

{

"sinCodeList":"sinCode1, sinCode2 " //"标段编码列表"

}

* + - 1. 标段信息获取响应

{

"code":0,

"msg":"ok",

data:{

id:"主键ID",

BIDCODE:"标段编码",

BIDNAME:"标段名称",

BIDDESC:"标段描述",

prjCode:"项目编码",

STATUS:"数据状态",

bidType:"0：设计，1：监理，2：施工，3：监理加施工",

chiefSuperVice:"监理总工程师",

buildManager:"施工项目经理",

substypecode:"变电站类型包括:01 土建,02 电气,03 变电",

startdate:"实际开工时间",

findate:"实际投产时间",

compdate:"实际竣工时间",

}

}

1. 6. 风险作业获取
      1. 接口描述

风险作业获取属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + 1. 接口流程

风险作业获取的接口流程见下图。



1. 风险作业获取流程

主要功能流程如下：

1. F1：边缘物联代理发送HTTP请求获取风险作业，消息中携带查询条件；
2. F2：省公司基建平台返回符合查询的风险作业。
   * 1. 接口参数
        1. HTTP响应码

HTTP响应码见表1。

表1 HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - 1. 参数定义

风险作业获取请求命令相关的参数定义见表2。

表2查询风险作业请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| bidId | 必选 | String | 标段票编码 |

查询风险作业响应相关的参数定义见表3。

表3 查询风险作业的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| id | 必选 | INT32 | 主键 |
| projectSectionId | 必选 | INT32 | 工程标段 |
| projectSection | 必选 | String | 工程标段名称 |
| startDate | 必选 | String | 开始时间 |
| endDate | 必选 | String | 结束时间 |
| premeas | 必选 | String | 预防措施 |
| addPremeas | 必选 | String | 补充管控措施 |
| createUser | 必选 | String | 创建人 |
| createUserId | 必选 | INT32 | 创建人ID |
| createUnit | 必选 | String | 创建单位 |
| createUnitId | 必选 | INT32 | 创建单位ID |
| createDate | 必选 | String | 创建日期 |
| updateUser | 必选 | String | 更新人 |
| updateUserId | 必选 | INT32 | 更新人ID |
| updateUnit | 必选 | String | 更新单位 |
| updateUnitId | 必选 | INT32 | 更新单位ID |
| updateDate | 必选 | String | 更新日期 |
| worksn | 必选 | String | 工序 |
| wcont | 必选 | String | 作业部位 |
| ticketTypeId | 必选 | INT32 | 作业票类型ID |
| worktype | 必选 | String | 作业类型 |
| fixriskclass | 必选 | String | 风险等级 |
| workProgress | 必选 | String | 作业进度 |
| workStatus | 必选 | String | 作业状态 |
| riskBookId | 必选 | INT32 | 风险清册ID |
| projectCode | 必选 | String | 项目编码 |
| singlePrjCode | 必选 | String | 单项编码 |
| bidno | 必选 | String | 标段CODE |

* + 1. 消息示例
       1. 风险作业获取请求

请求URL：http://ip:port/getRiskWork

请求方式：POST

表头：Content-Type:application/json

{

" bidId ": "标号id"

}

* + - 1. 风险作业获取响应

{

"code":"",

"msg":"ok",

"data":{

"id":"" //主键

,"projectSectionId":"" //工程标段

,"projectSection":"" //工程标段名称

,"startDate":"" //开始时间

,"endDate":"" //结束时间

,"premeas":"" //预防措施

,"addPremeas":"" //补充管控措施

,"createUser":"" //创建人

,"createUserId":"" //创建人ID

,"createUnit":"" //创建单位

,"createUnitId":"" //创建单位ID

,"createDate":"" //创建日期

,"updateUser":"" //更新人

,"updateUserId":"" //更新人ID

,"updateUnit":"" //更新单位

,"updateUnitId":"" //更新单位ID

,"updateDate":"" //更新日期

,"worksn":"" //工序

,"wcont":"" //作业部位

,"ticketTypeId":"" //作业票类型ID

,"worktype":"" //作业类型

,"fixriskclass":"" //风险等级

,"workProgress":"" //作业进度

,"workStatus":"" //作业状态

,"riskBookId":"" //风险清册ID

,"projectCode":"" //项目编码

,"singlePrjCode":"" //单项编码

,"bidno":"" //标段CODE

}}

* 1. 作业票信息获取
     1. 接口描述

作业票信息获取属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + 1. 接口流程

作业票信息获取的接口流程见下图。



1. 作业票信息获取流程

主要功能流程如下：

1. F1：边缘物联代理发送HTTP请求作业票信息，消息中携带查询条件；
2. F2：省公司基建平台返回符合查询的作业票信息。
   * 1. 接口参数
        1. HTTP响应码

HTTP响应码见表1。

表1 HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - 1. 参数定义

作业票信息获取请求命令相关的参数定义见表2。

表2作业票信息获取请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| bidId | 必选 | String | 标段编码 |

作业票信息获取响应相关的参数定义见表3。

表3 作业票信息获取的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| id | 必选 | INT32 | *主键* |
| projectId | 必选 | INT32 | *工程ID* |
| workClassId | 必选 | INT32 | *施工班组ID* |
| riskLevelOrig | 必选 | String | *初勘风险等级* |
| riskLevelRepe | 必选 | String | *复测后风险等级* |
| worktype | 必选 | String | *作业类型* |
| worksn | 必选 | String | *工序* |
| wcont | 必选 | String | *作业部位* |
| startDate | 必选 | String | *开始时间* |
| endDate | 必选 | String | *结束时间* |
| workplanId | 必选 | INT32 | *施工方案交底ID* |
| vitalpoints | 必选 | String | *技术方案要点* |
| leaderId | 必选 | INT32 | *工作负责人ID* |
| leader | 必选 | String | *工作负责人* |
| supervisorId | 必选 | INT32 | *安全监护人ID* |
| supervisor | 必选 | String | *安全监护人* |
| engineerId | 必选 | INT32 | *施工技术员ID* |
| engineer | 必选 | String | *施工技术员* |
| mainRisk | 必选 | String | *主要风险* |
| createUser | 必选 | String | *创建人* |
| createUserId | 必选 | INT32 | *创建人ID* |
| createUnit | 必选 | String | *创建单位* |
| createUnitId | 必选 | INT32 | *创建单位ID* |
| createDate | 必选 | String | *创建日期* |
| updateUser | 必选 | String | *更新人* |
| updateUserId | 必选 | INT32 | *更新人ID* |
| updateUnit | 必选 | String | *更新单位* |
| updateUnitId | 必选 | INT32 | *更新单位ID* |
| updateDate | 必选 | String | *更新日期* |
| jobticketNo | 必选 | String | *票号* |
| ticketPreCon | 必选 | String | *作业票作业必备条件* |
| projectSectionId | 必选 | String | *工程标段* |
| projectSection | 必选 | String | *工程标段名称* |
| ticketTypeId | 必选 | INT32 | *作业票类型ID* |
| ticketTypeName | 必选 | String | *作业票类型名称* |
| riskBookId | 必选 | INT32 | *风险清册ID* |
| isExpand | 必选 | String | *是否延期* |
| siteReChange | 必选 | String | *复测现场变化* |
| siteRePremeas | 必选 | String | *复测现场控制措施* |
| alarmId | 必选 | INT32 | *风险预警ID* |
| projectCode | 必选 | String | *项目编码* |
| singlePrjCode | 必选 | String | *单项编码* |
| staffInfo | 必选 | Object | *参与人员* |

* + 1. 消息示例
       1. 作业票获取请求

请求URL：http://ip:port/ getJobTicket

请求方式：POST

表头：Content-Type:application/json

{

"bidId": "标号id "

}

* + - 1. 作业票获取响应

{"code":"",

"msg":"ok",

"data":{

"id":"" //主键

,"projectId":""//工程ID

,"workClassId":""//施工班组ID

,"riskLevelOrig":""//初勘风险等级

,"riskLevelRepe":""//复测后风险等级

,"worktype":""//作业类型

,"worksn":"" //工序

,"wcont":""//作业部位

,"startDate":""//开始时间

,"endDate":""//结束时间

,"workplanId":"" //施工方案交底ID

,"vitalpoints":""//技术方案要点

,"leaderId":""//工作负责人ID

,"leader":"" //工作负责人

,"supervisorId":""//安全监护人ID

,"supervisor":"" //安全监护人

,"engineerId":"" //施工技术员ID

,"engineer":""//施工技术员

,"mainRisk":""//主要风险

,"createUser":"" //创建人

,"createUserId":""//创建人ID

,"createUnit":"" //创建单位

,"createUnitId":""//创建单位ID

,"createDate":"" //创建日期

,"updateUser":"" //更新人

,"updateUserId":""//更新人ID

,"updateUnit":"" //更新单位

,"updateUnitId":""//更新单位ID

,"updateDate":"" //更新日期

,"jobticketNo":""//票号

,"ticketPreCon" :""//作业票作业必备条件

,"projectSectionId":""//工程标段

,"projectSection":"" //工程标段名称

,"ticketTypeId":""//作业票类型ID

,"ticketTypeName":"" //作业票类型名称

,"riskBookId":"" //风险清册ID

,"isExpand":""//是否延期

,"siteReChange":""//复测现场变化

,"siteRePremeas":""//复测现场控制措施

,"alarmId":""//风险预警ID

,"projectCode":""//项目编码

,"singlePrjCode":""//单项编码

,"staffInfo"://参与人员

[

{

"userId":"",

"userName":"",

"roleCode":""

}

]

}}

* 1. 人员信息获取
     1. 接口描述

人员信息获取属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + 1. 接口流程

人员信息获取的接口流程见下图。



1. 人员信息查询流程

主要功能流程如下：

1. F1：边缘物联代理发送HTTP请求人员信息，消息中携带查询条件；
2. F2：省公司基建平台返回符合查询的人员信息。
   * 1. 接口参数
        1. HTTP响应码

HTTP响应码见表1。

表1 HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - 1. 参数定义

人员信息获取请求命令相关的参数定义见表2。

表2查询人员信息请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| bidId | 必选 | String | 标段编码 |

查询人员信息响应相关的参数定义见表3。

表3 查询人员信息的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| id | 必选 | INT32 | *主键* |
| deptName | 必选 | String | *单位名称* |
| projectName | 必选 | String | *项目名称* |
| name | 必选 | String | *人员名称* |
| personType | 必选 | String | *人员类型* |
| idCard | 必选 | String | *身份证号码* |
| genderName | 必选 | String | *性别* |
| nation | 必选 | String | *民族* |
| mobile | 必选 | String | *手机号码* |
| age | 必选 | INT32 | *年龄* |
| talentLevel | 必选 | String | *人才层次* |
| politicAppera | 必选 | String | *政治面貌* |
| nativePlace | 必选 | String | *籍贯* |
| degree | 必选 | String | *文化程度* |
| inWork | 必选 | INT32 | *参加工作年限* |
| status | 必选 | String | *状态* |

* + 1. 消息示例
       1. 人员信息获取请求

请求URL：http://ip:port/ getPeopleInfo

请求方式：POST

表头：Content-Type:application/json

{

"bidId": "标号id"

}

* + - 1. 人员信息获取响应

{

"code":"",

"msg":"ok",

"data":{

"id":"" //主键

,"deptName":"" //单位名称

,"projectName":"" //项目名称

,"name":"" //人员名称

,"personType":"" //人员类型

,"idCard":"" //身份证号码

,"genderName":"" //性别

,"nation":"" //民族

,"mobile":"" //手机号码

,"age":"" //年龄

,"talentLevel":"" //人才层次

,"politicAppera":"" //政治面貌

,"nativePlace":"" //籍贯

,"degree":"" //文化程度（学历编码（0：中专；1：高中； 2：大专；3：本科；4：研究生；5：博士及以上；6：其他））

,"inWork":"" //参加工作年限

,"status":"" //状态

}}

1. 边缘物联代理与物联管理平台交互协议规范
   1. 通信架构

本部分定义了边缘物联代理（简称边设备）与物联管理平台通信中对MQTT协议的使用。如图二所示。

通信架构（宋体6号）

1. 通信通信架构示意图
   1. 规约结构

本部分规定了边设备与物联管理平台之间所采用的MQTT通信协议在ISO七层体系中所属位置以及协议栈数据结构，如表1所示。

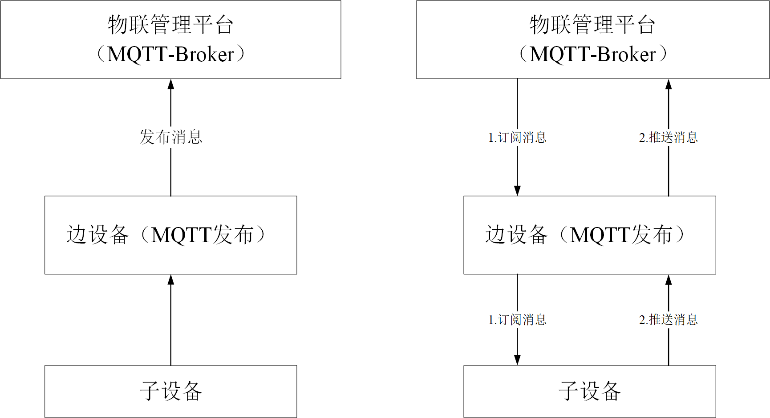
1. MQTT规约结构

|  |  |
| --- | --- |
| 应用功能 | 用户进程 |
| MQTT | 应用层（第7层） |
| TCP/IP协议 | 传输层（第4层） |
| 网络层（第3层） |
| 802.3、4G/5G、NB-IoT等协议 | 链路层（第2层） |
| 物理层（第1层） |
| **注：** 第5，第6层未用 | |



1. MQTT协议栈数据结构
   1. 交互机制

MQTT协议采用发布/订阅机制来完成消息交互，该机制能够提供一对多消息分发，其交互机制如图3所示：



1. 交互机制示意图
   1. 报文格式
      1. 请求报文

边设备与物联管理平台交互的请求报文均以JSON格式描述，通用报文字段如表2　所示：

1. 请求报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必选 | 描述 |
| mid | number | 是 | 在请求报文中该值为请求ID |
| deviceId | string | 否 | 设备唯一标识 |
| timestamp | number | 是 | 消息发送的时间戳，UTC时间,精度到秒 |
| expire | number | 否 | 此报文过期的相对时间（相对timestamp），单位秒，不包含该字段，或者该字段为-1时表示永不过期 |
| type | string | 是 | 消息类型，全大写，单词中间用“\_”隔开 |
| param | object | 否 | 报文内容，详见各报文定义 |

* + 1. 应答报文

边设备与物联管理平台交互的应答报文均以JSON格式描述，通用报文字段如表3　所示：

1. 应答报文格式

| 名称 | 类型 | 是否必选 | 描述 |
| --- | --- | --- | --- |
| mid | number | 是 | 在应答报文中，该值为应答所对应请求报文的请求ID |
| deviceId | string | 否 | 设备唯一标识 |
| timestamp | number | 是 | 消息发送的时间戳，UTC时间，精度到秒 |
| type | string | 是 | 消息类型,全大写，单词中间用“\_”隔开 |
| param | object | 否 | 报文内容，详见各报文定义 |
| code | number | 是 | 标识应答的返回码 |
| msg | string | 否 | 应答结果描述，字符串，最大256字符 |

应答的“code”字段可以选择的值如表4　所示：

1. 应答code编码格式

|  |  |
| --- | --- |
| code | 描述 |
| 200 | 请求成功 |
| 202 | 请求被接受，但是服务器未处理完 |
| 400 | 请求失败 |
| 401 | 请求未认证/认证错误 |
| 403 | 请求被拒绝 |
| 404 | 请求的资源不存在 |
| 600 | 其他错误 |

* 1. 交互主题

边设备与物联管理平台交互采用发布订阅模式进行通信，订阅与发布基于主题。主题格式为:/主题版本/边设备或直连设备ID/消息执行类别/消息动作，/{v1}/{edgeId}/{Topic Type}/{Topic Command}，同时平台和边设备可自行扩展以实现业务特有的功能，其中：

1. {v1}: 表示当前定义的主题版本为v1, 后期业务发展还会继续开发v2、v3等版本；
2. {edgeId}: 表示直接与物联管理平台建立MQTT长连接的边设备或直连设备ID，该ID为物联管理平台内部唯一，可采用多种途径获取，例如如MQTT、HTTP方式等；
3. {Topic Type }：Topic的类别，主要有device、container、app、service、topo五种。device表示该topic为设备类消息，service表示为业务类消息，app表示应用类消息，container表示容器消息，topo表示拓扑类消息；
4. {Topic Command }:表示消息的动作如控制、采集的数据、请求或响应。Topic Command的具体格式参考对应的业务模型规范。

Topic的主要内容和用途见表5，其中:方向取值为：up、down。当为up时表示边设备或直连终端通发布、物联管理平台订阅的topic；为down时表示物联管理平台发布、边设备或直连终端订阅的topic。

1. Topic汇总

| 分类 | topic内容 | 方向 | 用途 |
| --- | --- | --- | --- |
| 设备管理 | /v1/{edgeId}/device/command | down | 用于物联管理平台向边设备发送设备控制命令，如设备升级、控制设备等 |
| /v1/{edgeId}/device/reply | up | 用于对物联管理平台发送的设备控制命令的响应 |
| /v1/{edgeId}/device/request | up | 用于边设备向物联管理平台发送设备管理相关的请求命令，如请求连接等 |
| /v1/{edgeId}/device/response | down | 用于对边设备发送的设备管理相关的请求命令的响应 |
| /v1/{edgeId}/device/data | up | 用于边设备向物联管理平台主动上报设备相关的状态、事件等 |
| /v1/{edgeId}/topo/request | up | 用于边设备向物联管理平台发送拓扑信息的请求命令 |
| /v1/{edgeId}/topo/response | down | 用于对边设备发送的拓扑信息的请求命令的响应 |
| 容器管理 | /v1/{edgeId}/container/command | down | 用于物联管理平台向边设备发送的容器控制请求命令，如容器安装、启动、停止等 |
| /v1/{edgeId}/container/reply | up | 用于对物联管理平台的容器控制请求命令的响应 |
| /v1/{edgeId}/container/data | up | 用于边设备向物联管理平台主动上报容器相关的状态、事件等 |
| 应用管理 | /v1/{edgeId}/app/command | down | 用于物联管理平台向边设备发送应用控制请求命令，如应用安装、启动、停止等 |
| /v1/{edgeId}/app/reply | up | 用于对物联管理平台发送的应用控制请求命令的响应 |
| /v1/{edgeId}/app/data | up | 用于边设备向物联管理平台主动上报应用相关的状态、事件等 |
| 业务交互 | /v1/{edgeId}/service/command | down | 用于物联管理向边设备发送业务交互请求命令 |
| /v1/{edgeId}/service/reply | up | 用于对物联管理的业务交互请求命令的响应 |
| /v1/{edgeId}/service/data | up | 用于边设备向物联管理平台主动上报业务相关的数据、事件等 |

1. 边缘物联代理与业务APP及端设备交互协议规范
   1. 通信架构



1. 边缘物联代理通信架构图

由于之前现场传感设备接入都是由传感设备厂商自己开发接入，现场通信协议不统一，每次有新的厂家设备接入都需要重新开发部署新的应用程序，浪费人力、占用大量边缘物联代理的内存资源，又会出现边缘代理上部署了大量采集程序，管理困难。

为便于不同业务应用开发,边缘物联代理通过制定MQTT、HTTP、Modbus等通用接入协议，对符合通用协议基建现场感知层传感设备的统一接入，并采用边缘计算框架方式与物管平台统一对接，各业务应用只需与边缘计算框架交互，业务APP通过HTTP和MQTT协议与框架进行交互，业务功能包括端设备的接入、删除、状态更新，数据、事件主动上报，业务控制命令响应以及APP的相关配置。边缘计算框架与业务APP运行在同一设备中，通过连接本地MQTT和HTTP服务进行业务通信。

* 传感设备通过串口、Wi-Fi、LoRa、RJ45等接口连接到边缘物联代理装置，由边缘物联代理的接入层进行统一的传感设备接入管理、传感设备数据采集等。
* 边缘物联代理接入层将Modbus、二进制协议统一转换为指定MQTT格式，各业务应用可通过边缘计算框架获取各自业务相关数据；具有一定处理计算能力的感知设备或网关类设备可以采用MQTT协议直接上传到边缘计算框架。
* 移动终端可通过Wi-Fi连接边缘物联代理，基建APP离线模块通过物联代理中的业务应用实现设备信息、实时数据、历史数据等信息的展示，同时基建APP离线模块可以通过业务应用对传感设备进行配置管理。
* 边缘物联代理根据业务需求通过HTTP协议从平台获取数据，并按需将数据上送到基建平台。

* 1. 边缘物联代理内部交互及Mqtt协议接入
     1. 交互主题

内部交互主题如表所示：

表 1内部交互主题

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 业务交互 | 业务功能 | 交互主题 | 用途 | 备注 |
| 数据上报 | /v1/${appName}/service/data | 用于业务APP对边缘计算框架发送实时数据 |  |
| /dev/${appName}/service/data | 用于端设备对边缘计算框架发送实时数据 |  |
| 事件上报 | /v1/${appName}/service/event | 用于业务APP对边缘计算框架发送紧急事件 |  |
| /dev/${appName}/service/event | 用于端设备对边缘计算框架发送紧急事件 |  |
| 控制命令下发 | /v1/${appName}/service/command | 用于边缘计算框架转发物管平台对下的控制命令 |  |
| /dev/${appName}/service/command | 用于边缘计算框架对端设备对下发控制命令 |  |
| /v1/${appName}/service/reply | 用于业务APP对边缘计算框架发送服务响应 |  |
| /dev/${appName}/service/reply | 用于端设备对边缘计算框架发送服务响应 |  |
| 业务配置更新 | /v1/${appName}/app/command | 用于边缘计算框架通知业务APP更新业务配置文件 |  |
| /v1/${appName}/app/reply | 用于业务APP对边缘计算框架发送配置文件更新的响应 |  |
| 端设备管理 | /v1/${appName}/topo/request | 用于业务APP对边缘计算框架发送添加端设备、删除端设备、更新请求 |  |
| /dev/${appName}/topo/request | 用于终端发送添加端设备、删除端设备、更新请求到框架 |  |
| /v1/${appName}/topo/response | 用于边缘计算框架对业务APP发送端设备操作请求的响应 |  |
| /dev/${appName}/topo/response | 用于边缘计算框架对终端发送端设备操作请求的响应 |  |

* + 1. 协议格式

协议报文均以JSON格式描述，通用报文字段如下表所示：

表 2通用报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必选** | **描述** |
| type | string | 是 | 消息类型，全大写，单词中间用“\_”隔开 |
| mid | number | 是 | 消息ID，会话发起者自主生成，响应者与其一致 |
| param | object | 是 | 报文内容，详见各报文定义 |
| timestamp | number | 是 | 时间戳，毫秒精度，根据当前时间生成 |

* + 1. 端设备添加接口



* + - 1. 端设备添加信息上报

方向：APP -> 边缘框架

APP主题：/v1/${appName}/topo/request

设备主题：/dev/${appName}/topo/request

端设备的新增、状态更新、删除等操作均由APP发起，由边缘框架转发到平台，平台下发为每个端设备生成的deviceid，在后续上报数据阶段，APP上报数据中需带有deviceid标明数据的来源。

业务APP向边缘计算框架发起添加端设备请求，其字段说明如下：type字段取值为：CMD\_TOPO\_ADD；

表 3 端设备添加报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | String | 是 | CMD\_TOPO\_ADD |
| mid | number | 是 | 该消息的编号，自主生成 |
| timestamp | number | 是 | 毫秒精度，根据当前时间生成 |
| expire | number | 是 | 默认为-1 表示永不过期 |
| param | object | 是 | 详见下表 |

Param字段包含内容如下表所示：

表 4 param字段内容

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| nodeInfos | array | 是 | 端设备信息列表，列表大小1~100 |

表 5 Array<nodeInfos>字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必选** | **描述** |
| nodeId | string | 是 | 端设备唯一标识，设备直接MQTT接入边设备时由厂商自定义唯一标识（可采用序列号/MAC地址等） |
| name | string | 是 | 端设备名称 |
| description | string | 是 | 端设备描述 |
| mfgInfo | string | 是 | 端设备厂商信息 |
| nodeModel | string | 是 | 端设备型号 |
| modelId | string | 否 | 端设备模型编号 |

典型报文样例如下：

{

"type":"CMD\_TOPO\_ADD",

"mid":1000000000020028,

"timestamp":1581384683012,

"expire":-1,

"param":{

"nodeInfos":[

{

"nodeId":"testSN001",

"name":"test001",

"description":"test",

"mfgInfo":"NARI",

"nodeModel":"NARI"

},

{

"nodeId":"testSN002",

"name":"test002",

"description":"test",

"mfgInfo":"NARI",

"nodeModel":"nari"

}

]

}

}

* + - 1. 端设备添加信息响应

方向：边缘框架 -> APP

APP主题：/v1/${appName}/topo/response

设备主题：/dev/${appName}/topo/response

边缘计算框架在获取了平台的响应后，返回业务APP响应。

其报文字段说明如下：

* type字段取值为：CMD\_TOPO\_ADD；
* result字段如所示：

表6 添加端设备应答字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | string | 是 | CMD\_TOPO\_ADD |
| mid | number | 是 | 在报文中该值为对应控制报文的ID |
| deviceId | string | 是 | 边设备ID |
| code | number | 是 | 标识应答的返回码 |
| msg | string | 否 | 应答结果描述，字符串，最大256字符 |
| param | object | 是 | 端设备添加的响应结果 |

表7 param 字段说明

| **字段** | **类型** | **是否必选** | **参数描述** |
| --- | --- | --- | --- |
| result | array | 是 | 设备添加的结果信息 |

表 8 Array<result>字段说明

| **字段** | **类型** | **是否必选** | **参数描述** |
| --- | --- | --- | --- |
| statusCode | number | 是 | 请求处理的结果码，“200”表示成功，“400”表示失败 |
| statusDesc | string | 否 | 响应状态描述 |
| nodeId | string | 是 | 端设备唯一标识 |
| deviceId | string | 是 | 平台生成的设备唯一标识 |
| Profile | Object | 否 | 该端设备的过滤模型，由边缘框架下载、过滤 |

典型报文样例如下：

{

"mid":1000000000020028,

"deviceId":"1000000001154323",//边设备ID

"timestamp":15212345678900,

"type":"CMD\_TOPO\_ADD",

"code":200,

"msg":"SUCCESS!",

"param":{

"result":[

{

"statusCode":200,

"statusDesc":"SUCCESS!",

"nodeId":"testSN001",

"deviceId":"152223495866321",

//返回值添加物模型

"profile":{

"url":"http://xxxxx",

"name":"",

"size":23,

"md5":"xxx"

}

},

{

"statusCode":200,

"statusDesc":"SUCCESS!",

"nodeId":"testSN002",

"deviceId":"152223495866322",

"profile":{

"url":"http://",

"name":"",

"size":23,

"md5":"xxx"

}

}

]

}

}

* + 1. 端设备删除接口



* + - 1. 端设备删除信息上报

方向：APP ->边缘框架

APP主题：/v1/${appName}/topo/request

设备主题：/dev/${appName}/topo/request

业务APP向边缘计算框架发起删除设备请求。其字段说明如下：type字段取值为：CMD\_TOPO\_DEL，param字段如下表所示：

表9 删除端设备命令字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | String | 是 | CMD\_TOPO\_DEL |
| mid | number | 是 | 该消息的编号，自主生成 |
| param | object | 是 | 参数，如下表 |
| timestamp | number | 是 | 毫秒精度 |
| expire | number | 是 | 默认为-1 表示永不过期 |

表 10 删除端设备命令param字段说明

| **字段** | **类型** | **是否必选** | **描述** |
| --- | --- | --- | --- |
| nodeIds | array | 是 | 端设备ID列表，列表大小1~100，string数组 |

典型报文样例如下：

{

"mid":1000000000020029,

"timestamp":15212345678900,

"type":"CMD\_TOPO\_DEL",

"expire":-1,

"param":{

"nodeIds":["152223495866321","152223495866322"]

}

}

* + - 1. 端设备删除信息响应

方向：边缘框架 ->APP

APP主题：/v1/ ${appName}/topo/response

设备主题：/dev/ ${appName}/topo/response

边缘计算框架在获取了平台的响应后，返回业务APP响应。

其报文字段说明如下：

* type字段取值为：CMD\_TOPO\_DEL；
* param字段如下表所示：

表 11 端设备删除报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | string | 是 | CMD\_TOPO\_ DEL |
| mid | number | 是 | 在报文中该值为对应控制报文的ID |
| param | object | 是 | 设备删除的结果响应 |
| code | number | 是 | 标识应答的返回码 |
| msg | string | 否 | 应答结果描述，字符串，最大256字符 |
| deviceId | string | 是 | 边设备ID |

表 12 param字段说明

| **字段** | **类型** | **是否必选** | **参数描述** |
| --- | --- | --- | --- |
| result | array | 否 | 设备删除的结果信息 |

表 13 result段说明

| **字段** | **类型** | **是否必选** | **参数描述** |
| --- | --- | --- | --- |
| statusCode | number | 是 | 请求处理的结果码，“200”表示成功，“400”表示失败 |
| statusDesc | string | 否 | 结果描述 |
| deviceId | string | 是 | 平台生成的设备唯一标识 |

典型报文样例：

{

"mid":1000000000020029,

"deviceId":"1000000001154323",//边设备ID

"timestamp":15212345678900,

"type":"CMD\_TOPO\_DEL",

"code":200,

"msg":"SUCCESS!",

"param":{

"result":[

{

"statusCode":200,

"statusDesc":"SUCCESS!",

"deviceId":"152223495866321"

},

{

"statusCode":200,

"statusDesc":"SUCCESS!",

"deviceId":"152223495866322"

}

]

}

}

* + 1. 端设备状态更新接口



* + - 1. 端设备更新信息上报

方向：APP ->边缘框架

APP主题：/v1/${appName}/topo/request

设备主题：/dev/${appName}/topo/request

业务APP向边缘计算框架发起更新终端设备信息请求。其字段说明如下：type字段取值为：CMD\_TOPO\_UPDATE，param字段如下表所示：

表 14 报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | String | 是 | CMD\_TOPO\_ UPDATE |
| mid | number | 是 | 该消息的编号，自主生成 |
| param | object | 是 |  |
| timestamp | number | 是 | 毫秒精度 |
| expire | number | 是 | 默认为-1 表示永不过期 |

表15 更新端设备字段说明

| **字段** | **类型** | **是否必选** | **参数描述** |
| --- | --- | --- | --- |
| nodeStatuses | array | 是 | 端设备状态列表，列表大小1~100 |

表16 deviceStatus结构体字段说明

| **字段** | **类型** | **是否必选** | **参数描述** |
| --- | --- | --- | --- |
| deviceId | string | 是 | 平台生成的设备唯一标识 |
| status | string | 是 | 端设备状态：  OFFLINE：设备离线  ONLINE： 设备上线 |

{

"mid":1000000000020030,

"timestamp":15212345678900,

"type":"CMD\_TOPO\_UPDATE",

"expire":-1,

"param":{

"nodeStatuses":[

{

"deviceId":"152223495866321",

"status":"ONLINE"

},

{

"deviceId":"152223495866322",

"status":"OFFLINE"

}

]

}

}

* + - 1. 端设备更新信息响应

方向：边缘框架 ->APP

APP主题：/v1/ ${appName}/topo/response

设备主题：/dev/ ${appName}/topo/response

边缘计算框架收到物联管理平台的响应后，向APP发送响应报文。其字段说明如下：type字段取值为：CMD\_TOPO\_UPDATE，param字段如下表所示：

表17 报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | string | 是 | CMD\_TOPO\_UPDATE |
| mid | number | 是 | 在报文中该值为对应控制报文的ID |
| param | object | 是 | 设备更新的结果响应 |
| code | number | 是 | 标识应答的返回码 |
| msg | string | 否 | 应答结果描述，字符串，最大256字符 |

表18 param结构体字段说明

| **字段** | **类型** | **是否必选** | **参数描述** |
| --- | --- | --- | --- |
| result | array | 否 | 更新设备状态信息 |

表19 Array<result>结构体字段说明

| **字段** | **类型** | **是否必选** | **参数描述** |
| --- | --- | --- | --- |
| statusCode | number | 是 | 请求处理的结果码，“200”表示成功，“400”表示失败 |
| statusDesc | string | 否 | 结果状态描述 |
| deviceId | string | 是 | 端设备唯一标识 |

典型报文样例：

{

"mid":1000000000020031,

"deviceId":"1000000001154323",//边设备ID

"timestamp":15212345678900,

"type":"CMD\_TOPO\_UPDATE",

"code":200,

"msg":"SUCCESS!",

"param":{

"result":[

{

"statusCode":200,

"statusDesc":"SUCCESS!",

"deviceId":"152223495866321"

},

{

"statusCode":200,

"statusDesc":"SUCCESS!",

"deviceId":"152223495866322"

}

]

}

}

* + 1. 业务数据主动上报接口



* + - 1. 数据主动上报

方向：APP ->边缘框架

APP主题：/v1/${appName}/service/data

设备主题：/dev/${appName}/service/data

APP向边缘计算框架主动上报业务数据，边缘框架根据对应的物模型过滤后上报。报文格式如下表所示，其中type字段取值为：CMD\_REPORTDATA，param如下表所示：

表20报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | string | 是 | CMD\_REPORTDATA |
| mid | number | 是 | 在报文中该值为报文ID，自主生成 |
| param | object | 是 | 根据物模型规范定义 |
| deviceId | string | 是 | 报文中的数据属于哪个端设备 |
| timestamp | number | 是 | 毫秒精度 |
| expire | number | 是 | 默认为-1 表示永不过期 |

表 22 param结构体字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必选** | **描述** |
| cmd | string | 是 | 数据属于哪个类别，与业务相关，上送到业务主站需要 |
| deviceId | String | 是 | 主站为端设备分配的唯一ID |
| data | json | 是 | 数据以物模型规范的数据格式上报 |

2

典型数据样例如下：

{

"mid":1000000000020033,

"type":"CMD\_REPORTDATA",

"timestamp":1574930415425,

"expire":-1,

"deviceId":"1000000001154323", //端设备

"param":{

"cmd":"B\_phs\_alert",

"deviceId":"1000000001154323", //端设备

"data":{

"Ua":220.3

}

}

}

* + 1. 业务事件主动上报接口



* + - 1. 事件主动上报

方向：APP ->边缘框架

业务主题：/v1/${appName}/ service/event

设备主题：/dev/${appName}/ service/event

APP向边设备主动上报紧急事件。报文格式如下表所示，其中type字段取值为：CMD\_REPORTEVENT，param如下表所示：

表 23 报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| type | String | 是 | CMD\_REPORTEVENT |
| mid | number | 是 | 该消息的编号，自主生成 |
| deviceId | string | 是 | 报文中的数据属于哪个端设备 |
| param | object | 是 | 根据物模型规范定义 |
| timestamp | number | 是 | 毫秒精度 |
| expire | number | 是 | 默认为-1 表示永不过期 |

表 24 param格式

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| deviceId | String | 是 | 主站为端设备分配的唯一ID |
| event | String | 是 | 事件类型 |
| data | JSON对象 | 否 | 根据物模型规范定义紧急事件的描述 |

典型报文样例如下：

{

"mid":1000000000020032,

"type":"CMD\_REPORTEVENT",

"timestamp":1574930415425,

"deviceId":"1000000001154323",//端设备

"expire":-1,

"param":{

"deviceId ":"1000000001154323", //端设备

"event":"resource\_alarm",

"data":{

"cpuAlm":"85",

"memAlm":"90",

"diskAlm":"95"

}

}

}

* + 1. 业务控制命令下发接口



* + - 1. 业务控制命令下发

方向：边缘框架 ->APP

APP主题：/v1/${appName}/service/command

设备主题：/dev/${appName}/service/command

边缘计算框架转发物联管理平台对终端设备的控制命令，App主动订阅主题，其中param字段需与物模型定义相同，此处不做要求（对于业务自定义的控制报文，也需要在模型文件中给出描述）。报文格式如下：

表25业务控制报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | string | 是 | CMD\_SERVICE |
| mid | number | 是 | 在报文中该值为报文ID |
| param | object | 是 | 根据物模型规范定义 |
| deviceId | string | 是 | 报文中的数据属于哪个端设备 |
| expire | number | 否 | 超时时间 |
| timestamp | number | 是 | 毫秒精度 |

典型控制报文样例如下：

{

"mid":1000000000020034,

"type":"CMD\_SERVICE",

"deviceId": "1000000001154323", //端设备ID

"timestamp":1574930415425,

"expire":-1,

"param":{

"cmd":"analog\_Get",

"paras":{

"body":

{

"PhV\_phsA":"",

"PhV\_phsB":""

}

}

}

}

* + - 1. 业务控制命令响应

方向：APP ->边缘框架

APP主题：/v1/${appName}/service/reply

设备主题：/dev/${appName}/service/reply

边缘计算框架转发的控制命令在APP完成执行后，返回响应报文。报文格式如下表所示，其中mid为对应控制报文的ID

表26 报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | string | 是 | CMD\_SERVICE |
| mid | number | 是 | 与下发报文一致 |
| timestamp | number | 是 | 毫秒精度 |
| param | object | 否 | 根据物模型规范定义 |
| code | number | 是 | 标识应答的返回码 |
| msg | string | 否 | 应答结果描述，字符串，最大256字符 |

典型报文样例如下：

{

"mid":1000000000020034,

"type":"CMD\_SERVICE",

"timestamp":1574930415425,

"deviceId":"1000000001154323",//端设备ID

"code": 200,

"msg": "SUCCESS",

"param":{

"cmd":"analog\_Get",

"paras":{

"body":

{

"PhV\_phsA":"220.4",

"PhV\_phsB":"220.7"

}

}

}

}

* + 1. APP业务配置接口



* + - 1. 更新命令下发

方向：边缘框架 ->APP

主题：/v1/${ appName }/app/ command

边缘计算框架向业务APP发送更新配置文件的报文，App主动订阅，appName表示APP的名称。报文格式如下表所示。其中param中包含的内容见下表。

表 27 业务配置命令报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| type | String | 是 | CMD\_APP\_REMOTE\_CONFIG\_DOWN |
| mid | number | 是 | 该消息的编号 |
| deviceId | String | 是 | 边设备ID |
| param | object | 是 | 详见下表 |
| expire | number | 否 | 过期时间 |
| timestamp | number | 是 | 毫秒精度 |

表 28 Param格式

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必需 | 描述 |
| container | String | 是 | 容器名称 |
| appName | String | 是 | App名称 |
| name | String | 是 | 文件名称 |
| configFile | String | 是 | 配置命令内容Base64编码内容 |

典型报文样例如下所示：

{

"mid": 1000000000020019,

"deviceId": "1000000001154323",//边设备ID

"type": "CMD\_APP\_REMOTE\_CONFIG\_DOWN",

"timestamp": 1562746993127,

"expire": -1,

"param": {

"container": "容器名称",

"appName": "XXXApp名称",

"name": "XXXApp配置文件名称",

"configFile": "配置命令内容Base64编码内容"

},

}

* + - 1. 更新命令响应

方向：APP ->边缘框架

主题：/v1/${appName}/app/reply

边缘计算框架转发的更新配文件的命令在APP完成执行后，返回响应报文，其中topic中需包含app的名称。报文格式如下表所示，其中mid为对应控制报文的ID。

表 29 业务配置响应报文格式

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必需** | **描述** |
| type | string | 是 | CMD\_CONFIG\_REPLY |
| mid | number | 是 | 与下发报文一致 |
| code | number | 是 | 标识应答的返回码 |
| msg | string | 否 | 应答结果描述，字符串，最大256字符 |

* 1. MODBUS协议接入
     1. 协议格式

通用MODBUS帧格式如下：

地址

功能码

差错校验

ADU

PDU

数据

APU：应用数据单元。

PDU：协议数据单元。

当通讯命令由发送设备（主机）发送至接收设备（从机）时，符合相应地址码的从机接收通讯命令，并根据功能码及相关要求读取信息，如果CRC校验无误，则执行相应的任务，然后把执行结果（数据）返送给主机。返回的信息中包括地址码、功能码、执行后的数据以及CRC校验码。如果CRC校验出错就不返回任何信息。

RTU协议中的指令由地址码(一个字节），功能码（一个字节），数据（N个字节），校验码（两个字节）组成。

地址码是每次通讯信息帧的第一字节（8位），从0到255。这个字节表明由用户设置地址的从机将接收由主机发送来的信息。每个从机都必须有唯一的地址码，并且只有符合地址码的从机才能响应回送信息。当从机回送信息时，回送数据均以各自的地址码开始。主机发送的地址码表明将发送到的从机地址，而从机返回的地址码表明回送的从机地址。相应的地址码表明该信息来自于何处。

功能码是每次通讯信息帧传送的第二个字节。MODBUS通讯规约可定义的功能码为1到127。作为主机请求发送，通过功能码告诉从机应执行什么动作。作为从机响应，从机返回的功能码与从主机发送来的功能码一样，并表明从机已响应主机并且已进行相关的操作。

数据区包括需要由从机返送何种信息或执行什么动作。这些信息可以是数据（如：开关量输入/输出、模拟量输入/输出、寄存器等等）、参考地址等。例如，主机通过功能码03告诉从机返回寄存器的值（包含要读取寄存器的起始地址及读取寄存器的长度），则返回的数据包括寄存器的数据长度及数据内容。对于不同的从机，地址和数据信息都不相同（应给出通讯信息表）。

主机或从机可用校验码进行判别接收信息是否正确。由于电子噪声或一些其它干扰，信息在传输过程中有时会发生错误，错误校验码（CRC）可以检验主机或从机在通讯数据传送过程中的信息是否有误，错误的数据可以放弃（无论是发送还是接收）。 MODBUS通讯协议的CRC（冗余循环码）包含2个字节，即16位二进制数。CRC码由发送设备（主机）计算，放置于发送信息帧的尾部。接收信息的设备（从机）再重新计算接收到信息的CRC，比较计算得到的CRC是否与接收到的相符，如果两者不相符，则表明出错。

MODBUS协议簇主要有三种协议：Modbus-RTU、Modbus-ASCII、Modbus-TCP，格式定义如下，当通信数据量大而且时二进制数据时，RTU相比较ASCII具备更紧凑的报文流，传输效率更高，串口通信推荐采用Modbus RTU协议。

地址

PDU（功能码+数据）

CRC

MODBUS-RTU

Modbus TCP协议是在RTU协议前面添加MBAP报文头，共七个字节长度，其分别的意义是：1.传输标志，两个字节长度，标志MODBUS询问/应答的传输，一般默认是00 00。2.协议标志，两个字节长度，0表示是MODBUS，1表示UNI-TE协议，默认为00 00。3.后续字节计数，两个字节长度，其实际意义就是后面的字节长度。4.单元标志，一个字节长度，默认为00，单元标志对应于Modbus RTU协议中的地址码。MODBUS默认采用的TCP端口号是502。

MBAP

PDU

MODBUS-TCP

* + 1. 功能码定义

| **功能码** | **功能名称** | **描述** |
| --- | --- | --- |
| 01 | 读线圈状态 | 读取线圈状态，可读取多个连续线圈的状态 |
| 02 | 读输入离散量 | 读取输入状态，可读取多个连续输入的状态 |
| 03 | 读多个寄存器 | 读取保持寄存器的值，可读取多个连续输入保持寄存器 |
| 04 | 读输入寄存器 | 读取输入寄存器的值，可读取多个连续输入输入寄存器 |
| 05 | 写单个线圈 | 写入单个线圈的状态 |
| 06 | 写单个保存寄存器 | 写入单个寄存器 |
| 0F | 写多个线圈 | 写入多个连续线圈的状态 |
| 10 | 写多个保存寄存器 | 写入多个连续寄存器 |

* + 1. 客户端命令格式

| **字节序号** | **字节长度** | **功能名称** | **描述** |
| --- | --- | --- | --- |
| 0 | 1 | 设备ID | 1~255 |
| 1 | 1 | 功能码 | 见功能码定义表 |
| 2 | 2 | 起始地址 | 地址高位在前，低位在后 |
| 4 | 2 | 读取长度 | 长度高位在前，低位在后 |
| 6 | 2 | CRC校验 | CRC低位在前，高位在后 |

* + 1. 服务端响应格式

| **字节序号** | **字节长度** | **功能名称** | **描述** |
| --- | --- | --- | --- |
| 0 | 1 | 设备ID | 1~255 |
| 1 | 1 | 功能码 | 见功能码定义表 |
| 2 | 1 | 字节总数 | 根据命令返回：8个开关量1个字节，1个模拟量2个字节 |
| 3 | 字节总数 | 数据寄存器 | 开关量：低位到高位 模拟量：寄存器高位在前，低位在后 |
| 3+字节总数 | 2 | CRC校验 | CRC低位在前，高位在后 |

* 1. 二进制协议接入
     1. 协议格式

协议数据由数据头、数据体、校验数据部分组成，如下表所示。

表1 通讯数据结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字节位 | BYTE0 | BYTE1 | BYTE2:BYTE7 | BYTE8 | BYTE9 |
| 参数 | START | LEN | MAC | SERIAL\_NUM | PACKAGE\_COUNT |
| 数据类型 | BYTE | BYTE | BYTE | BYTE | BYTE |
| 字节位 | BYTE10 | | BYTE11 | BYTE12:BYTE N-2 | BYTE N-1: BYTE N |
| 参数 | PACKAGE\_IDX | | ENCRYPTION | PAYLOAD | CRC |
| 数据类型 | BYTE | | BYTE | BYTE | WORD |

通讯数据包说明：

1. START

START为启动符，固定为68H，定义了数据流中的起点。

1. LEN

LEN表示后面数据总长度。

1. MAC

MAC字段表示设备地址，如下表所示。

表2 MAC数据结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| MAC[0] | MAC[1] | MAC[2] | MAC[3] | MAC[4] | MAC[5] |

1. SERIAL\_NUM

字段表示消息流水号。

1. PACKAGE\_COUNT

字段表示当前报文传输数据总包数。

1. PAGKAGE\_IDX

字段表示当前报文包数据为总数据包中的第几包。

1. ENCRYPTION

字段表示数据包是否加密,并设定加密方式，如下表所示。

表3 ENCRYPTION字段值

| 数据 | 类型 |
| --- | --- |
| 0 | 不加密 |
| 1 | AES-128 |

加解密范围为PAYLOAD数据体，加密时对所有数据体进行加密，然后再进行分包处理，解密时需把所有分包中PAYLOAD部分拼接起来后解密。

1. PAYLOAD

PAYLOAD字段代表当前消息的负载，整个DATA数据体需把所有分包中的PAYLOAD字段拼接后生成。

整体数据体结构说明：

字段数据结构如下表所示。

表4 PAYLOAD字段数据结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节位 | BYTE0 | BYTE1 | BYTE2 | BYTE N |
| 参数 | DEV\_TYPE | DATA\_TYPE | INS | DATA |
| 数据类型 | BYTE | BYTE | BYTE | BYTE[N] |

1. DEV\_TYPE

字段代表设备类型。

1. DATA\_TYPE

字段代表数据类型，如下表所示。

表5 DATA\_TYPE字段值

|  |  |
| --- | --- |
| 数据 | 类型 |
| 0x01 | 上行数据 |
| 0x02 | 下行数据 |

1. INS

字段代表当前消息指令。

1. DATA

字段为当前指令的数据体。

* + 1. 终端注册

指令（INS字段）：0x01

DATA数据结构：

1. 上行数据

其DATA字段说明如下表所示。

表6 DATA字段说明

|  |  |
| --- | --- |
| 字节位 | BYTE15 |
| 参数 | REGISTE |
| 数据类型 | BYTE |

其DATA字段值如下表所示。

表7 DATA字段值

| 数据 | 类型 |
| --- | --- |
| 0x01 | 注册 |
| 0x02 | 注销 |

1. 下行数据

其DATA字段说明如下表所示。

表 8 DATA字段说明

|  |  |
| --- | --- |
| 字节位 | BYTE15 |
| 参数 | RESULT |
| 数据类型 | BYTE |

其DATA字段值如下表所示。

表9 DATA字段值

|  |  |
| --- | --- |
| 数据 | 类型 |
| 0x01 | 成功 |
| 0x02 | 失败 |

* + 1. 数据上报

指令（INS字段）：0x02

DATA数据结构：

1. 上行数据

其DATA数据结构如下表所示。

表 10 DATA数据结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节位 | BYTE0:BYTE1 | BYTE N | | |
| 参数 | VALUECOUNT | VALUEARRAY | … | VALUEARRAY |
| 数据类型 | WORD | BYTE | … | BYTE |

VALUECOUNT代表后面VALUEARRAY参数个数。

VALUEARRAY结构如下表所示。

表 11 VALUEARRAY数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位 | BYTE0:BYTE1 | BYTE2 | BYTE3:BYTE N |
| 参数 | VALUENAME | VALUELEN | VALUE |
| 数据类型 | WORD | BYTE | 根据参数类型 |

1. 下行数据

其DATA字段说明如下表所示。

表 12 DATA字段说明

|  |  |
| --- | --- |
| 字节位 | BYTE15 |
| 参数 | RESULT |
| 数据类型 | BYTE |

其DATA字段值如下表所示。

表 13 DATA字段值

|  |  |
| --- | --- |
| 数据 | 类型 |
| 0x01 | 成功 |
| 0x02 | 失败 |

* + 1. 参数读取

指令（INS字段）：0x03

DATA数据结构：

1. 下行数据

其DATA字段说明如下表所示。

表 14 DATA字段说明

|  |  |
| --- | --- |
| 字节位 | BYTE15 |
| 参数 | TYPE |
| 数据类型 | BYTE |

TYPE参数这里固定为0x02。

1. 上行数据

其DATA字段说明如下表所示。

表 15 DATA字段说明

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字节位 | BYTE0 | BYTE1:BYTE2 | BYTE N | | |
| 参数 | RESULT | PARAMCOUNT | PARAMARRAY | … | PARAMARRAY |
| 数据类型 | BYTE | WORD | BYTE[n] | … | BYTE[n] |

RESULT参数的DATA字段值如下表所示。

表16 DATA字段值

|  |  |
| --- | --- |
| 数据 | 类型 |
| 0x01 | 成功 |
| 0x02 | 失败 |

PARAMCOUNT代表后面PARAMARRAY参数个数。

PARAMARRAY结构如下表所示。

表 17 PARAMARRAY数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位 | BYTE0:BYTE1 | BYTE2 | BYTE3:BYTE N |
| 参数 | PARAMNAME | PARAMLEN | PARAM |
| 数据类型 | WORD | BYTE | 根据参数类型 |

* + 1. 设置参数

指令（INS字段）：0x03

1. 下行数据

其DATA数据结构如下表所示。

表 18 DATA数据结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字节位 | BYTE0 | BYTE1:BYTE2 | BYTE N | | |
| 参数 | TYPE | PARAMCOUNT | PARAMARRAY | … | PARAMARRAY |
| 数据类型 | BYTE | WORD | BYTE[n] | … | BYTE[n] |

TYPE参数这里固定为0x01。

PARAMCOUNT代表后面PARAMARRAY参数个数。

PARAMARRAY结构如下表所示。

表 19 PARAMARRAY数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位 | BYTE0:BYTE1 | BYTE2 | BYTE3:BYTE N |
| 参数 | PARAMNAME | PARAMLEN | PARAM |
| 数据类型 | WORD | BYTE | 根据参数类型 |

PARAM参数为需要设置的参数值。

1. 上行数据

其数据结构如下表所示。

表 20 上行数据结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字节位 | BYTE0 | BYTE1:BYTE2 | BYTE N | | |
| 参数 | RESULT | PARAMCOUNT | PARAMARRAY | … | PARAMARRAY |
| 数据类型 | BYTE | WORD | BYTE[n] | … | BYTE[n] |

RESULT参数的返回值如下表所示。

表 21 返回值

|  |  |
| --- | --- |
| 数据 | 类型 |
| 0x01 | 成功 |
| 0x02 | 失败 |

PARAMCOUNT代表后面PARAMARRAY参数个数。

PARAMARRAY结构如下表所示。

表 22 PARAMARRAY数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字节位 | BYTE0:BYTE1 | BYTE2 | BYTE3:BYTE N |
| 参数 | PARAMNAME | PARAMLEN | PARAM |
| 数据类型 | WORD | BYTE | 根据参数类型 |

返回结果为成功返回的参数为当前设置成功的参数值。

返回结果为失败则返回的参数为当前设备中的原有的参数值。

* 1. HTTP协议接入

1. 5. 1. 查询算法能力
         1. 接口描述

查询算法能力属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + - 1. 接口流程

查询算法能力的接口流程见图A.1。



* 1. 查询算法能力流程

主要功能流程如下：

1. F1：系统1的用户发送HTTP检索系统2的查询算法能力，消息中携带查询条件；
2. F2：系统2返回符合查询的算法能力。
   * + 1. 接口参数
          1. HTTP响应码

HTTP响应码见表A.1。

* 1. HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - * 1. 参数定义

查询算法能力请求命令相关的参数定义见表A.2。

* 1. 查询算法能力请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| EventType | 必选 | String | 类型 |
| FrontCode | 必选 | String | 编码 |

查询算法能力响应相关的参数定义见表A.3。

* 1. 查询算法能力的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| AbilityInfo | 必选 | Object | 算法能力列表 |
| Number | 必选 | INT32 | 算法个数 |
| Ability | 必选 | Array | 算法数组 |
| AlgCode | 必选 | INT32 | 算法能力编码 |
| CodeDesc | 必选 | String | 算法能力描述 |

* + - 1. 消息示例
         1. 查询算法能力请求

请求URL：http://ip:port/analysis/interface

请求方式：POST

表头：Content-Type:application/json

{

"EventType": "alg\_ability\_request"

}

* + - * 1. 查询算法能力响应

{

"EventType": "alg\_ability\_request",

"Result": {

"Code": 200,

"Desc": ""

},

"AbilityInfo": {

"Number": 2,

"Ability": [{

"AlgCode": 1001,

"CodeDesc": ""

},

{

"AlgCode": 1002,

"CodeDesc": ""

}

]

}

}

* + 1. 查询算法任务
       1. 接口描述

查询算法任务属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + - 1. 接口流程

查询算法任务的接口流程见图A.2。



* 1. 查询算法任务流程

主要功能流程如下：

1. F1：系统1的用户发送HTTP检索系统2的查询算法任务，消息中携带查询条件；
2. F2：系统2返回符合检索条件的查询算法任务。
   * + 1. 接口参数
          1. HTTP响应码

HTTP响应码见表A.4。

* 1. HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - * 1. 参数定义

查询算法任务请求命令相关的参数定义见表A.5。

* 1. 查询算法任务请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| EventType | 必选 | String | 类型 |
| FrontCode | 必选 | String | 编码 |

查询算法任务响应相关的参数定义见表A.6。

* 1. 查询已有计划任务的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| FrontTaskNumber | 必选 | INT32 | 前端系统任务数量 |
| FrontTask | 必选 | Array | 前端系统任务 |
| FrontCode | 必选 | String | 前端系统编码 |
| TaskNumber | 必选 | INT32 | 分析任务数量 |
| Task | 必选 | Array | 分析任务 |
| TaskId | 必选 | String | 分析任务ID |
| AlgCode | 必选 | String | 算法能力编码 |
| TaskStatus | 必选 | INT32 | 分析任务状态，任务异常停止（-2），任务创建失败（-1），任务空闲（0），任务正在运行（1），任务正在创建（2） |

* + - 1. 消息示例
         1. 查询算法任务请求

请求URL：http://ip:port/analysis/interface

请求方式：POST

表头：Content-Type:application/json

{

"EventType": "alg\_task\_request",

"FrontCode": "100010000201040000",

}

* + - * 1. 查询算法任务响应

{

"EventType": "alg\_task\_request",

"Result": {

"Code": 200,

"Desc": ""

},

"FrontTaskNumber": 1,

"FrontTask": [{

"FrontCode": "100010000201040000",

"TaskNumber": 1,

"Task": [{

"TaskId": "AlgTask123",

"AlgCode": 1001,

"TaskStatus": 1

}]

}]

}

* + 1. 算法任务配置
       1. 接口描述

算法任务配置属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + - 1. 接口流程

算法任务配置的接口流程见图A.3。



* 1. 算法任务配置流程

主要功能流程如下：

1. F1：系统1的用户发送HTTP配置系统2的算法任务，消息中携带配置参数；
2. F2：系统2返回算法任务配置结果。
   * + 1. 接口参数
          1. HTTP响应码

HTTP响应码见表A.7。

* 1. HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - * 1. 参数定义

算法任务配置请求命令相关的参数定义见表A.8。

* 1. 算法任务配置请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| FrontCode | 必选 | String | 字符串，前端系统编码 |
| TaskId | 必选 | String | 字符串，分析任务ID |
| Command | 必选 | String | 整型，配置命令，0-停止，1-开始，2-删除 |
| AlgCode | 可选 | INT32 | 整型，算法能力编码 |
| AlarmInterval | 可选 | INT32 | 整型，告警间隔，单位秒 |
| AlarmSubcribe | 可选 | INT32 | 整型，是否告警订阅，0-不订阅告警，1-订阅告警 |
| AlarmPushUrl | 可选 | String | 字符串，告警推送地址 |
| PointNum | 可选 | INT32 | 整型，规则框点个数 |
| Point | 可选 | Array | Array型，点数组 |
| X | 可选 | INT32 | 整型，点的X坐标 |
| Y | 可选 | INT32 | 整型，点的Y坐标 |

算法任务配置响应相关的参数定义见表A.9。

* 1. 算法任务配置的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| EventType | 必选 | String | 类型 |
| FrontCode | 必选 | String | 编码 |
| TaskId | 必选 | String | 字符串，分析任务ID |

* + - 1. 消息示例
         1. 算法任务配置请求

请求URL：http://ip:port/analysis/interface

请求方式：POST

表头：Content-Type:application/json

{

"EventType": "alg\_task\_config",

"FrontCode": "100010000201040000",

"TaskId": "AlgTask123",

"Command": 1,

"AlgCode": 1001,

"AlarmInterval": 5,

"AlarmSubcribe": 1,

"AlarmPushUrl": "http://192.168.1.200:3400/analysis/alarm\_push",

"AreaNum": 2,

"AreaField": [{

"PointNum": 2,

"Point": [{

"X": 100,

"Y": 100

},

{

"X": 200,

"Y": 200

}

]

},

{

"PointNum": 2,

"Point": [{

"X": 100,

"Y": 100

},

{

"X": 200,

"Y": 200

}

]

}

]

}

* + - * 1. 算法任务配置响应

{

"EventType": "alg\_task\_config",

"FrontCode": "100010000201040000",

"TaskId": "AlgTask123",

"Result": {

"Code": 200,

"Desc": ""

}

}

* + 1. 分析结果告警推送
       1. 接口描述

分析结果告警推送属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + - 1. 接口流程

分析结果告警推送的接口流程见图A.4。



* 1. 分析结果告警推送流程

主要功能流程如下：

1. F1：系统1的用户发送HTTP配置系统2的算法任务，消息中携带配置参数；
2. F2：系统2返回算法任务配置结果。
   * + 1. 接口参数
          1. HTTP响应码

HTTP响应码见表A.10。

* 1. HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - * 1. 参数定义

分析结果告警推送命令相关的参数定义见表A.11。

* 1. 分析结果告警推送请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| FrontCode | 必选 | String | 前端系统编码 |
| AlarmInfo | 必选 | Object | 告警信息 |
| Number | 必选 | INT32 | 告警数量 |
| Item | 必选 | Array | 告警 |
| AlgCode | 必选 | INT32 | 算法能力编码 |
| TaskId | 必选 | String | 分析任务ID |
| AlgCode | 必选 | String | 算法能力编码 |
| AlarmTime | 必选 | DATE | 告警发生时间 |
| Desc | 必选 | String | 告警描述 |
| PictureName | 必选 | String | 告警图片名称 |
| Type | 必选 | INT32 | 告警图片类型 |
| Width | 可选 | INT32 | 告警图片的宽度，像素 |
| Height | 可选 | INT32 | 告警图片的高度，像素 |
| Size | 可选 | INT32 | 告警图片的大小，字节 |

分析结果告警推送响应相关的参数定义见表A.12。

* 1. 分析结果告警推送响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| EventType | 必选 | String | 类型 |
| FrontCode | 必选 | String | 编码 |
| Desc | 必选 | String | 描述 |

* + - 1. 消息示例
         1. 分析结果告警推送请求

请求URL：http://ip:port/analysis/alarm\_push

请求方式：POST

表头：Content-Type:application/json

{

"EventType": "alg\_task\_alarm\_push",

"FrontCode": "100010000201040000",

"AlarmInfo": {

"Number": 1,

"Item": [{

"AlgCode": 1001,

"TaskId": "AlgTask123",

"AlarmTime": "2020-01-01 12:00:00",

"Desc": "xxxxxxxxx",

"PictureName": "xxx.jpg",

"Type": 0,

"Width": 1080,

"Height": 720,

"Size": 180309

}]

}

}

* + - * 1. 分析结果告警推送响应

{

"EventType": "alg\_task\_alarm\_push",

"FrontCode": "100010000201040000",

"Result": {

"Code": 200,

"Desc": ""

}

}

* + 1. 分析任务告警图片推送
       1. 接口描述

分析任务告警图片推送属于数据接口，采用HTTP的POST方法,将图片数据上传到指定的文件服务地址。

* + - 1. 接口流程

分析任务告警图片推送接口流程见图A.5。



* 1. 分析任务告警图片推送流程

主要功能流程如下：

1. F1：系统1收到系统2发送的图像数据
2. F2：系统1返回接收结果。
   * + 1. 接口参数
          1. HTTP响应码

HTTP响应码见表A.13。

* 1. HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - * 1. 参数定义

分析任务告警图片推送请求命令相关的参数定义见表A.14。

* 1. 分析任务告警图片推送请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| FrontCode | 必选 | String | 字符串，前端系统编码 |
| TaskId | 必选 | String | 字符串，分析任务ID |
| Command | 必选 | String | 整型，配置命令，0-停止，1-开始，2-删除 |
| AlgCode | 可选 | INT32 | 整型，算法能力编码 |
| AlarmInterval | 可选 | INT32 | 整型，告警间隔，单位秒 |
| AlarmSubcribe | 可选 | INT32 | 整型，是否告警订阅，0-不订阅告警，1-订阅告警 |
| AlarmPushUrl | 可选 | String | 字符串，告警推送地址 |
| PointNum | 可选 | INT32 | 整型，规则框点个数 |
| Point | 可选 | Array | Array型，点数组 |
| X | 可选 | INT32 | 整型，点的X坐标 |
| Y | 可选 | INT32 | 整型，点的Y坐标 |

分析任务告警图片推送响应相关的参数定义见表A.15。

* 1. 分析任务告警图片推送的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| EventType | 必选 | String | 类型 |
| FrontCode | 必选 | String | 编码 |
| TaskId | 必选 | String | 分析任务ID |

* + - 1. 消息示例
         1. 分析任务告警图片推送请求

请求URL：http://ip:port/analysis/alarm\_push

请求方式：POST

表头：multipart/form-data,boundary="\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" #boundary自动生成

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" #boundary自动生成

Content-Disposition: form-data; name="image"; filename="ImageName.jpg"

Content-Type: image/jpeg

{

#二进制图像数据

}

* + - * 1. 分析任务告警图片推送响应

{

"EventType": "alg\_task\_alarm\_push",

"Result": {

"Code": 200,

"Desc": ""

}

}

* + 1. 查询系统资源状态
       1. 接口描述

查询系统资源状态属于数据接口，使用HTTP+JSON，请求使用标准的HTTP Request，采用HTTP的POST方法，消息体应采用JSON封装。

* + - 1. 接口流程

查询系统资源状态的接口流程见图A.1。



* 1. 查询算法能力流程

主要功能流程如下：

1. F1：系统1的用户发送HTTP检索系统2的查询系统资源状态；
2. F2：系统2返回系统资源状态。
   * + 1. 接口参数
          1. HTTP响应码

HTTP响应码见表A.1。

* 1. HTTP响应码

|  |  |
| --- | --- |
| 状态码 | 描述 |
| 200 | 请求成功 |
| 400 | 错误的消息格式 |
| 403 | 请求被禁止，无权限 |
| 404 | 请求的节点不存在 |
| 500 | 当前内部错误，无法提供服务 |
| 503 | 当前负荷满，稍后再尝试请求 |

* + - * 1. 参数定义

查询系统资源状态请求命令相关的参数定义见表A.2。

* 1. 查询算法能力请求命令参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| EventType | 必选 | String | 类型 |
| FrontCode | 必选 | String | 编码 |

查询算法能力响应相关的参数定义见表A.3。

* 1. 查询算法能力的响应参数定义

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 选项 | 参数类型 | 参数描述 |
| CPU | 必选 | String | CPU占用百分比 |
| RAMTotal | 必选 | String | 内存总大小 |
| RAMFree | 必选 | String | 内存空闲大小 |
| RAM | 必选 | INT32 | 内存占用百分比 |
| Disk | 必选 | INT32 | 硬盘占用百分比 |
| DiskTotal | 必选 | String | 硬盘总空间 |
| DiskFree | 必选 | String | 硬盘剩余空间 |
| AITotal | 必选 | String | AI算力总大小 |
| AIUsed | 必选 | String | AI算力占用百分比 |
| APPNum | 必选 | INT32 | APP数目 |
| Run | 必选 | INT32 | APP运行状态 |

* + - 1. 消息示例
         1. 查询系统资源状态

请求URL：http://ip:port/analysis/interface

请求方式：POST

表头：Content-Type:application/json

{

"EventType": "edge\_status\_request"

}

* + - * 1. 查询算法能力响应

{

"EventType": "edge\_status\_request",

"Result": {

"Code": 200,

"Desc": ""

},

"Status": {

"CPU": 0.4,

"RAMTotal": "1000M",

"RAMFree": "500M",

"DiskTotal": "500G",

"DiskFree": "200G",

"AITotal": "32T",

"APPNum": 2,

"APP": [{

"APPName": "APP1",

"Run": 1,

"CPU": 0.3,

"RAM": 0.2,

"Disk":0.2,

"AIUsed": 0,

}

{

"APPName": "APP2",

"Run": 0,

"CPU": 0.1,

"RAM": 0.1,

"Disk":0.2,

" AIUsed ": 0.3,

}]

}

1. 可扩展性

边缘物联代理与省公司基建平台之间采用http协议进行数据交互，数据的传输规约可根据业务需求进行灵活的扩展。

边缘物联代理与物联管理平台之间采用mqtt协议进行数据交互，在遵循本边缘物联代理与物联管理平台交互协议规范的基础上，传输内容及物模型可根据业务需求灵活调整。

基建现场新接入的智能感知设备在遵循边缘物联代理与业务APP及端设备交互协议规范的基础上，通过配置即可接入边缘物联代理进行使用。同时本规范也具备一定程度的扩展能力，智能感知设备也可通过扩展相应业务APP的功能接入到边缘物联代理进行使用。