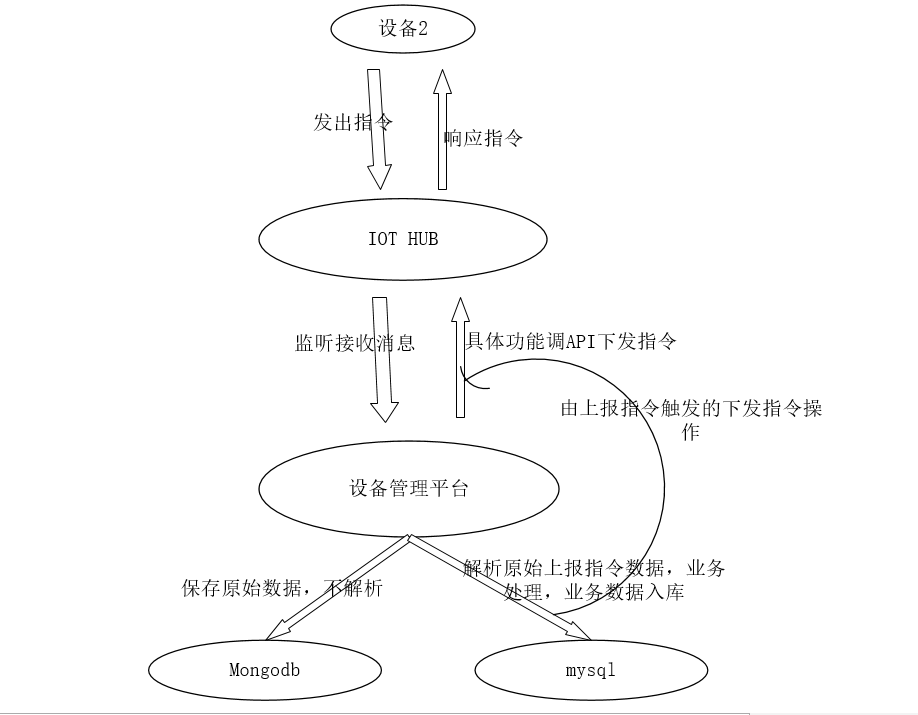
# 对接IOT Hub模块

1. **设备指令模块处理**
2. 流程图



1. 调用DeviceOperateService类的receiveDeviceOrder()方法保存数据到 mongodb , 将json指令字符串传给数据解析模块进行业务数据处理入库
2. 上下线消息

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 字段类型 | 必传 | 说明 |
| msgType | Integer | Y | 订阅消息类型 1-上下线 |
| clientID | String | Y | 设备唯一标识 |
| ip | String | Y | 设备IP地址 |
| cleanSession | boolean | Y | 是否清除mqtt会话 |
| msgCode | String | Y | 消息类型 100-上线 101-下线 |
| ts | Long | Y | 时间戳 |

格式:

{"msgType": 1,"clientID": "00001","ip": "192.168.3.36","cleanSession": "false","msgCode": "100","ts": 10100933393939}

将设备信息保存到mongodb 集合(deviceInfo\_tab) 根据clientID，ip检查是否存在设备记录，存在就更新，不存在新增,

|  |  |  |
| --- | --- | --- |
| 参数名称 | 类型 | 说明 |
| id | String | 主键 |
| clientID | String | 设备ID |
| ip | String | 设备IP地址 |
| createTime | Date | 指令创建时间 |
| status | String | 上下线状态 100-上线 101-下线 |
| orderTime | Long | 指令上报时间戳 |

1. 设备上报普通指令消息

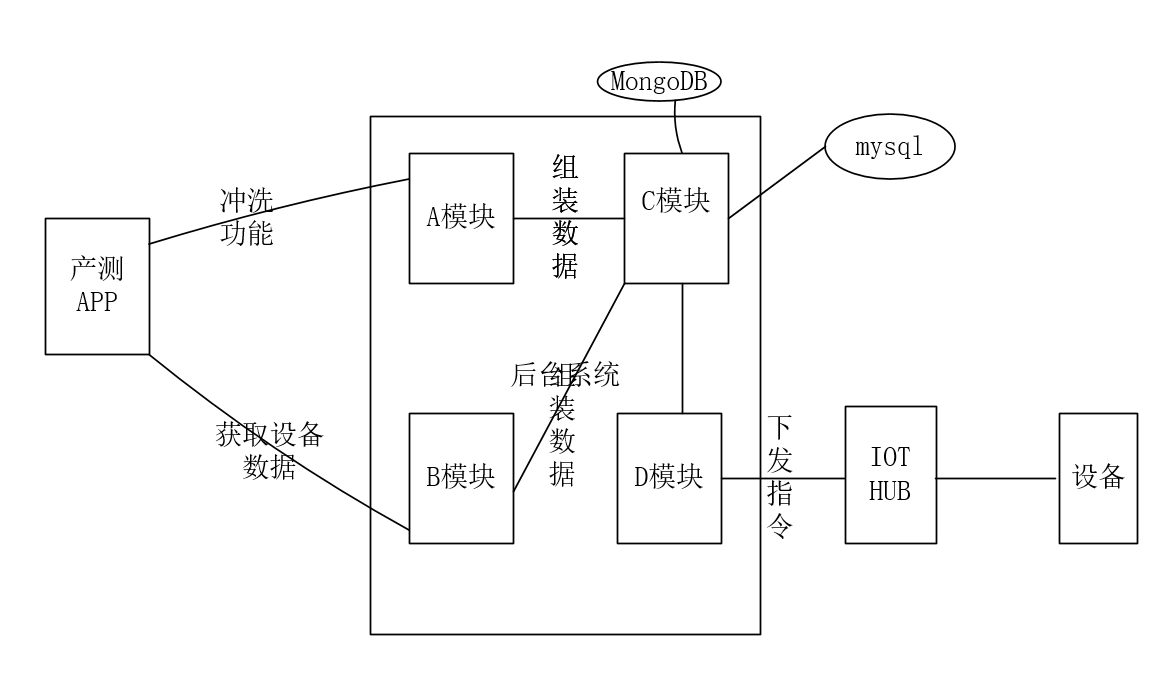
|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 字段类型 | 必传 | 说明 |
| msgType | Integer | Y | 订阅消息类型 0-普通消息 |
| clientID | String | Y | 设备唯一标识 |
| content | String | Y | 消息内容 |
| ts | Long | Y | 时间戳 |

{"msgType": 0,"clientID": "00001","content": "hdjdjdj1234","ts": 10100933393939}

消息直接保存到(集合: device\_order\_tab)，同一条消息根据clientID，data，orderTime检查是否存在，防止重复消费；mondodb保存完交给业务组件解析，并标记是否已处理

|  |  |  |
| --- | --- | --- |
| 参数名称 | 类型 | 说明 |
| id | String | 主键 |
| data | String | 指令JSON字符串 |
| clientID | String | 设备唯一标识 |
| status | String | 0-未处理 1-已处理 |
| createTime | Date | 指令创建时间 |
| orderTime | Long | 指令上报时间戳 |

1. 下发指令统一接口：



说明: A,B模块相当于具体处理APP请求的controller,调用C模块进行业务数据封装后将指令数据给到D模块，由D模块统一与IOT交互。

调用 DeviceOperateController类的 issuedDeviceOrder()接口处理下发设备指令：

请求地址: /device/issurOrder 类型：post

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必传（Y/N） | 说明 |
| deviceId | String | y | 设备ID |
| dataJson | String | y | JSON指令字符串 |

返回:

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必传(Y/N) | 说明 |
| code | String | Y | 0-成功 1-失败 |
| msg | String | Y | 请求结果 |
| data | object |  | 结果数据 |

Data

|  |  |  |  |
| --- | --- | --- | --- |
| deviceId | String | Y | 设备ID |
| result | String | Y | IOT返回的结果串 |

注意：跟IOT的交互接口参考第三部分

{

“code”:”0”,

“msg”:”请求成功”,

“data”:{

“deviceId”:”001”,’

“result”:”hdjdjddkkkfkfkk”

}

}

1. **设备调用平台接口**
2. 设备获取连接字符串

描述：设备配网成功后，需要从管理平台获取到连接IOT的ip地址信息；

接口地址：/device/getConnectionStr 类型：get

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 说明 | 是否必传 |
| clientID | String | 设备Id | Y |

出参：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| code | String | 0-成功 1-失败 |
| msg | String | 请求结果 |
| data | Map |  |
|  |  |  |

Map:

|  |  |  |
| --- | --- | --- |
| clientID | String | 设备ID |
| iotHost | String | IOT连接地址 |
| “” | String | productKey&deviceName |

{

“code”:”0”,

“msg”:”请求成功”,

“data”:{

“clientID”:”00001”,

“iotHost”:”7299229929”,

“sharedAccessKey”:”demo@0001”,

}

}

1. 查询上下线设备详情

描述：用于APP连接IOT平台成功后检查设备是否已经上线

请求地址：/device/findOrderOne/{clientID} 类型: get

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 说明 | 是否必传 |
| clientID | String | 设备Id | Y |

出参：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| code | String | 0-成功 1-失败 |
| Msg | String | 请求结果 |
| data | Object | 结果为DeviceEntity |

DeviceEntity:

|  |  |  |
| --- | --- | --- |
| clientID | String | 设备ID |
| status | String | 100-上线 101-下线 |
| id | String | Mongdb主键 |
| ip | String | 设备IP |
| cleanSession | String | 是否清除会话 |
| orderTime | Long | 指令时间戳 |

{

“code”:”0”,

“msg”:”请求成功”,

“data”:{

“clientID”:”00001”,

“iotHost”:”7299229929”,

“status”:”100”,

“ip”:”192.168.6.10”,

“id”:”0001”,

“cleanSession”:false,

“orderTime”:1020202039

}

}

1. 查询上下线设备集合

描述: 可以根据上下线状态查看多个设备情况

请求地址：/device/ findDeviceAll/{status} 类型: get

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 说明 | 是否必传 |
| status | String | 100-上线 101-下线 | N |

出参：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| code | String | 0-成功 1-失败 |
| Msg | String | 请求结果 |
| data | List<DeviceEntity> | 集合 |

DeviceEntity:

|  |  |  |
| --- | --- | --- |
| clientID | String | 设备ID |
| status | String | 100-上线 101-下线 |
| id | String | Mongdb主键 |
| ip | String | 设备IP |
| cleanSession | String | 是否清除会话 |
| orderTime | Long | 指令时间戳 |

{

“code”:”0”,

“msg”:”请求成功”,

“data”:[

{

“clientID”:”00001”,

“iotHost”:”7299229929”,

“status”:”100”,

“ip”:”192.168.6.10”,

“id”:”0001”,

“cleanSession”:false,

“orderTime”:1020202039

},

{

“clientID”:”00002”,

“iotHost”:”7299229929”,

“status”:”100”,

“ip”:”192.168.6.10”,

“id”:”0002”,

“cleanSession”:false,

“orderTime”:1020202039

}

]

}

1. 查询指令集合

请求地址：/device/findOrderAll

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 说明 | 必传 |
| clientID | String | 设备ID | N |
| status | String | 消息处理状态 0-未处理 1-已处理 | N |

出参：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| code | String | 0-成功 1-失败 |
| Msg | String | 请求结果 |
| data | List< DeviceReportOrderEntity> | 集合 |

DeviceReportOrderEntity

|  |  |  |
| --- | --- | --- |
| clientID | String | 设备ID |
| data | String | 消息内容 |
| id | String | Mongdb主键 |
| status | String | 0-未处理 1-已处理 |
| orderTime | Long | 指令时间戳 |

{

“code”:”0”,

“msg”:”请求成功”,

“data”:[

{

“clientID”:”00001”,

“data”:”hello shenzhen”,

“status”:”0”,

“orderTime”:1020202039,

“id”:”001”

},

{

“clientID”:”00001”,

“data”:”hello shenzhen”,

“status”:”0”,

“orderTime”:1020202039,

“id”:”002”

}

]

}

1. 查询指令详情

请求地址：/device/ /findOrderOne/{id}

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 说明 | 必传 |
| id | String | 设备ID | Y |

出参：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| code | String | 0-成功 1-失败 |
| Msg | String | 请求结果 |
| data | DeviceReportOrderEntity | 对象 |

DeviceReportOrderEntity

|  |  |  |
| --- | --- | --- |
| clientID | String | 设备ID |
| data | String | 消息内容 |
| id | String | Mongdb主键 |
| status | String | 0-未处理 1-已处理 |
| orderTime | Long | 指令时间戳 |

{

“code”:”0”,

“msg”:”请求成功”,

“data”: {

“clientID”:”00001”,

“data”:”hello shenzhen”,

“status”:”0”,

“orderTime”:1020202039,

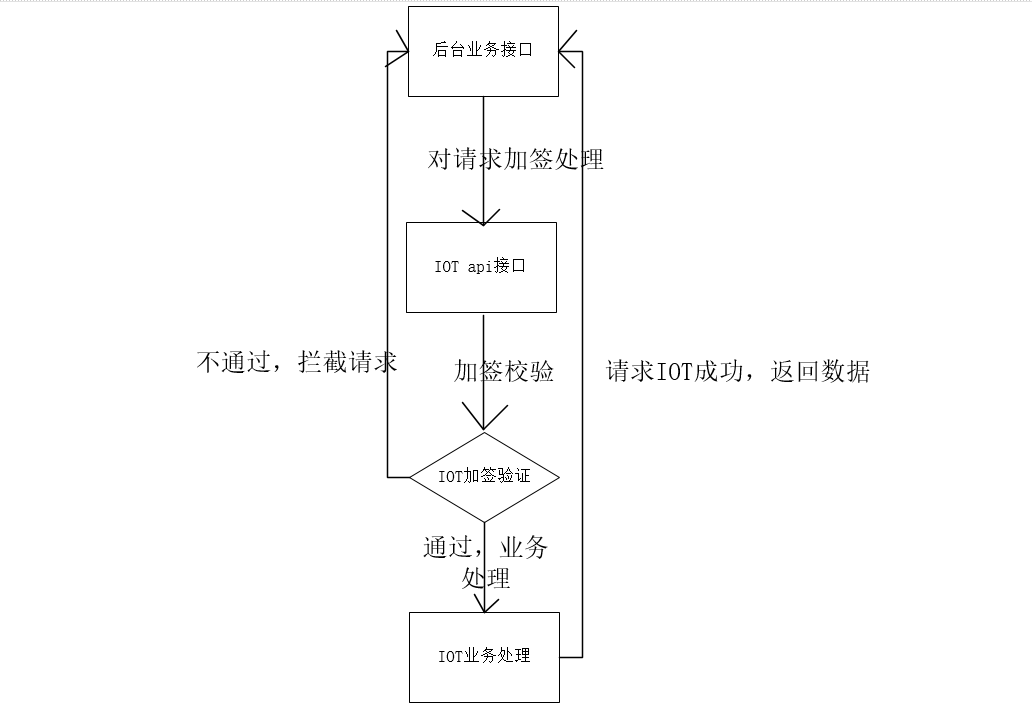
“id”:”001”

}

}

1. **设备平台请求IOT接口**

流程图



公共入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必须 | 描述 |
| accessKeyId | String | Y | Access key id |
| signatureMethod | String | Y | 签名方法 |
| signatureNonce | String | Y | 请求随机数 |
| signatureVersion | String | Y | 签名版本 |
| timestamp | String | Y | 请求时间戳 |
| version | String | Y | Api版本 |
| signature | String | Y | 请求签名 |

公共出参

|  |  |  |  |
| --- | --- | --- | --- |
| requestId | String | Y | Trace id，每次请求新生成 |
| code | String | Y | 请求结果状态 |
| message | String | Y | 请求结果国际化的消息 |
| resultCode | String | Y | 请求结果状态码 A09000-Z99999 |

1. 服务下发指令接口：调用DeviceOperateService类的issuedDeviceOrder()方法，调用请求工具类实时与IOT API接口交互(参考物联网 IOT接口文档 Publish接口)

1) 发送普通消息

请求地址:https://api.yun.pingan.com/v1?Action=Publish&<公共参数>

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| productKey | String | Y | 产品key |
| deviceName | String | Y | 设备名 |
| topicName | String | Y | 消息主题 |
| content | String | Y | 消息内容 |
| gos | String | Y | 发送频次 0-最多送达一次 1-至少送达一次 |

出参：

|  |  |  |  |
| --- | --- | --- | --- |
| status | Boolean | Y | 接口调用状态 true-成功 false:失败 |
| desc | String | Y | 描述信息 |
| content | String |  |  |

2) 发送广播消息

请求地址:https://api.yun.pingan.com/v1?Action=PublishBroadcast&<公共参数>

公共入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必须 | 描述 |
| accessKeyId | String | Y | Access key id |
| signatureMethod | String | Y | 签名方法 |
| signatureNonce | String | Y | 请求随机数 |
| signatureVersion | String | Y | 签名版本 |
| timestamp | String | Y | 请求时间戳 |
| version | String | Y | Api版本 |
| signature | String | Y | 请求签名 |

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| productKey | String | Y | 产品key |
| content | String | Y | 消息内容 |
| gos | String | Y | 发送频次 0-最多送达一次 1-至少送达一次 |

出参：

|  |  |  |  |
| --- | --- | --- | --- |
| status | Boolean | Y | 接口调用状态 true-成功 false:失败 |
| desc | String | Y | 描述信息 |
| content | String |  |  |

1. 设备注册接口