

递易物联网平台技术方案评审文档

作者：李允智 时间： 2020-08-13 版本： 1.1

前言

在IoT版本1.0中，主要是做了设备的基础通信。在版本1.1中，主要做两件事情

- 原先业务方控制格口的权利，转交给IoT
- 开箱等下方设备指令走IoT

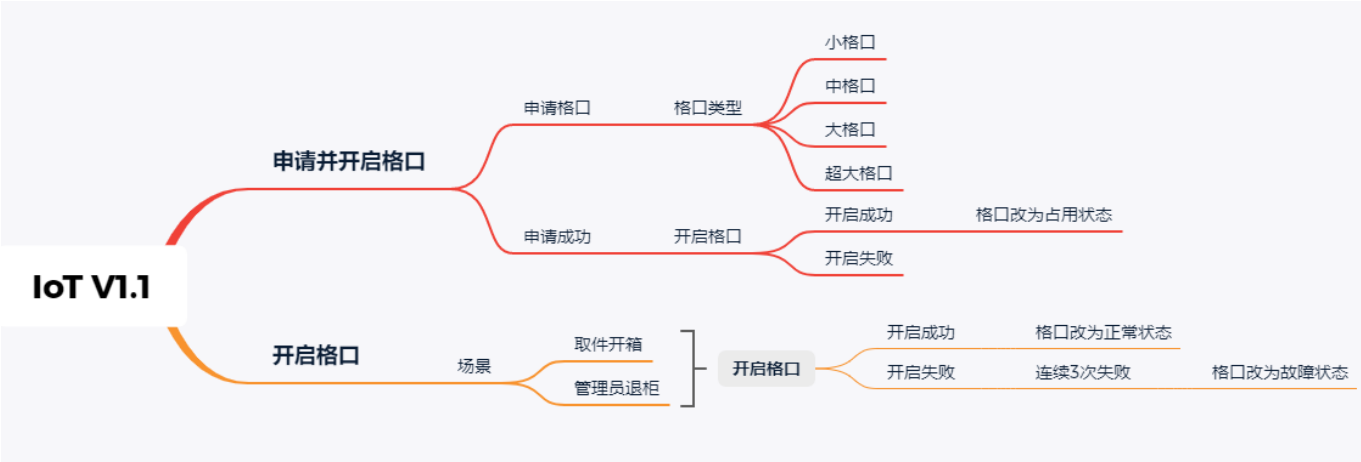
模块仍分为

- Device 设备模块
- Box 柜机模块

所有接口返回

- Code状态码 200为正常，0为错误
- Message错误提示 当为0时有值

思维导图



导图中状态为IoT格口状态

Device

V1.1无更新

Box 柜机模块

1、申请并开启格口接口

提供给业务方，申请并开启格口，原业务系统中【获取一个可用格口】，将格口的控制权转交给IoT，IoT来统一分配格口。如果有可用，开启格口

注意

- IoT在开箱成功之后会进行格口占用，该格口占用为IoT格口占用，非业务系统格口状态

接口请求入参

请求地址：Box/ApplyOpenCell

| 参数名 | 必选 | 类型 | 说明 |
|------------|----|--------|----------|
| DeviceSn | 是 | string | 设备编号 |
| CellType | 是 | int | 格口类型 |
| BizOrderId | 是 | string | 来源业务单号ID |

- 格口类型Code

- 小格口 1
- 中格口 2
- 大格口 3
- 超大格口 4

- 格口位置Code

- 左 0
- 右 1
- 主柜 2【中间的，收衣柜不显示】

接口请求出参

| 参数名 | 类型 | 说明 |
|---------------------|---------|-----------|
| Data - IsSuccess | bool | 申请并开启是否成功 |
| Data - CellSn | string | 格口编码 |
| Data - CellId | string | 格口Id |
| Data - CellPosition | int | 格口位置 |
| Code | integer | 返回状态码 |
| Message | string | 错误提示 |

IsSuccess为false错误提示信息有

- 暂无可用格口
- 设备无响应
- 格口开启失败

2、开启格口接口

提供给业务方，开启格口。该接口需要对应业务场景进行格口开启

业务场景分为

- 取件开箱 1
- 管理员退柜开箱 2
- 仅开箱 3

注意

- IoT在开箱成功之后会将格口改为正常可用状态【IoT格口状态】

接口请求入参

请求地址：Box/OpenCell

| 参数名 | 必选 | 类型 | 说明 |
|--------------|----|--------|----------|
| DeviceSn | 是 | string | 设备编号 |
| CellSn | 是 | string | 格口编码 |
| CellId | 是 | string | 格口Id |
| BizSceneType | 是 | int | 业务场景类型 |
| BizOrderId | 是 | string | 来源业务单号ID |

第三方业务单号ID，如果没有可传运单号，目的是为了保证，开启的时候格口号和业务单号Id一致才能开

接口请求出参

| 参数名 | 类型 | 说明 |
|------------------|---------|--------|
| Data - IsSuccess | bool | 开启是否成功 |
| Code | integer | 返回状态码 |
| Message | string | 错误提示 |

IsSuccess为true正确提示信息为

- 格口开启成功

IsSuccess为false错误提示信息有

- 设备无响应
- 格口开启失败

格口开启失败，设为故障逻辑

3次设备返回【格口开启失败】则将格口设为故障，该故障目前IoT会设置，也会将业务方的格口状态设为故障【临时方案，直到所有格口控制权转交IoT】该格口再3次之前开启成功，则清除标记次数；

3、获取格口数量

提供给业务方，获取各种类型格口的【可用】格口数量

- 格口类型Code
 - 小格口 1
 - 中格口 2
 - 大格口 3
 - 超大格口 4

接口请求入参

请求地址：Box/QueryCellCount

| 参数名 | 必选 | 类型 | 说明 |
|----------|----|--------|------|
| DeviceSn | 是 | string | 设备编号 |

接口请求出参

| 参数名 | 类型 | 说明 |
|--------------------|---------|----------|
| Data - SmallCount | int | 小格口可用数量 |
| Data - CenterCount | int | 中格口可用数量 |
| Data - BigCount | int | 大格口可用数量 |
| Data - LargeCount | int | 超大格口可用数量 |
| Code | integer | 返回状态码 |
| Message | string | 错误提示 |

4、格口状态(开启/关闭)

提供给业务方，获取设备格口的格口状态

接口请求入参

请求地址：Box/QueryCellStatus

| 参数名 | 必选 | 类型 | 说明 |
|------------|----|--------|----------|
| DeviceSn | 是 | string | 设备编号 |
| CellSn | 是 | string | 设备编号 |
| BizOrderId | 是 | string | 来源业务单号ID |

第三方业务单号ID，如果没有可传运单号，目的是为了保证，判断格口号和业务单号Id一致才会做判断

接口请求出参

| 参数名 | 类型 | 说明 |
|-----|----|----|
|-----|----|----|

| 参数名 | 类型 | 说明 |
|-------------------|---------|--------|
| Data - CellStatus | int | 开启还是关闭 |
| Code | integer | 返回状态码 |
| Message | string | 错误提示 |

- 格口状态CellStatus
 - 开启 1
 - 关闭 0

5、完成格口

提供给业务方，在用户场景为主动使用完格口时调用，如 拒收 场景

接口请求入参

请求地址：Box/FinishCell

| 参数名 | 必选 | 类型 | 说明 |
|------------|----|--------|----------|
| DeviceSn | 是 | string | 设备编号 |
| CellSn | 是 | string | 设备编号 |
| BizOrderId | 是 | string | 来源业务单号ID |

第三方业务单号ID，如果没有可传运单号，目的是为了保证，判断格口号和业务单号Id一致才会做处理

接口请求出参

| 参数名 | 类型 | 说明 |
|------------------|---------|--------------|
| Data - IsSuccess | bool | 完成格口操作成功还是失败 |
| Code | integer | 返回状态码 |
| Message | string | 错误提示 |

2020-08-24更新日志

- 在对设备 开箱操作 查询格口关闭 3个接口 增加了 并发限制，如果多次重复请求会提示 "设备繁忙，请稍后再试"
- 申请打开格口增加了CellId出参，打开格口增加了CellId入参
- 增加了完成格口接口 【对应收衣柜拒收场景】

2020-08-25更新日志

- 申请并打开格口接口，去除了商户ID参数，该参数通过设备号自动获取
- 去除了接口错误的需要JwtToken认证
- 现在申请并开启格口 会根据业务订单号 未完成的格口 仍打开上一次格口 对应禅道bug11087


- 去除了完成格口接收【拒收场景放在开箱】OpenCell接口增加场景 场景ID 3 3为拒收 仅开箱操作，其他场景仍会去做解锁格口操作【取件开箱和管理员退柜开箱】

2020-08-26更新日志

- 完成格口仍需要了
- 修复了德强测出的bug
- 去除了OpenCell接口的CellId入参 CellId在申请时候存入缓存，OpenCell可以通过缓存自动获取

2020-08-27更新日志

- 改进了Swagger UI 现在基于Swagger可以做接口调试了【相当于集成了个postman】
- 增加了日志记录【请求和返回都记录了】，并且可以通过唯一RequestId 查询到对应一来一回，很方便

图示: 企业微信截图_15985132428634 测试环境: <http://212.64.103.97:5601/app/kibana> 索引: devlog

2020-09-01

- ApplyOpenCell接口增加格口位置出参