端到端安全OTA概要设计

(版本号：V 1.2)

<公司Logo>

文 档 信 息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **项目名称:** |  | | | | |
| **项目经理:** |  | **文档编号:** |  | 文档版本号: | V0.1 |

文 档 变 更 记 录

| **版本编号** | **版本日期** | **作 者** | **说 明** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. OTA云端设计 4](#_Toc27964)

[1.1 设计目标 4](#_Toc1686)

[1.2 整体架构 4](#_Toc21784)

[1.3 交互流程 4](#_Toc29013)

[1.4 版本管理模式 5](#_Toc16133)

[1.5 OTA后台管理 5](#_Toc428)

[1.5.1 基础数据管理 5](#_Toc15809)

[1.5.2 固件清单管理 5](#_Toc10202)

[1.5.3 升级对象管理 6](#_Toc20277)

[1.5.4 升级包及版本管理 6](#_Toc9761)

[1.5.5 升级计划管理 6](#_Toc153)

[1.6 升级计划调度 7](#_Toc16618)

[1.7 OTA终端的服务接口 7](#_Toc2169)

[1.7.1 OTA版本更新检查 7](#_Toc28984)

[1.7.2 OTA升级包下载 9](#_Toc11753)

[1.7.3 OTA升级包下载进度汇报 10](#_Toc13181)

[1.7.4 OTA升级过程汇报 10](#_Toc20088)

[1.7.5 OTA升级结果汇报 12](#_Toc7761)

[1.7.6 下载确认请求 12](#_Toc8351)

[1.7.7 安装确认请求 13](#_Toc1941)

[1.7.8 升级错误日志上传（暂时不实现） 14](#_Toc32448)

[1.8 OTA用户交互接口 14](#_Toc29709)

[1.8.1 用户下载确认 14](#_Toc9961)

[1.8.2 用户安装确认 15](#_Toc11387)

[1.8.3 设备升级记录查询 （待定） 16](#_Toc32256)

[1.9 TSP接口依赖 16](#_Toc27827)

[1.9.1 OTA指令下发到终端接口 16](#_Toc6403)

[1.9.2 OTA用户通知接口 17](#_Toc1099)

[1.10 OTA终端指令消息 17](#_Toc9593)

[1.10.1 OTA升级通知 17](#_Toc15826)

[1.10.2 OTA下载确认 18](#_Toc21070)

[1.10.3 OTA安装确认 18](#_Toc18610)

[1.11 OTA用户通知消息 19](#_Toc9391)

[1.11.1 OTA下载确认请求通知 19](#_Toc808)

[1.11.2 OTA下载安装请求通知 19](#_Toc16545)

1. **OTA云端设计**
   1. 设计目标
   * 统一平台
     + 一个独立的OTA平台，支持多车场或客户的设备的接入
   * 支持多种设备类型和软件类型
     + 支持多车系，多型号规格，多种类型设备软件（比如TBOX、IPDA、OBD、IVI等）的升级
     + 支持其他非车辆相关软件的升级，为复杂的软件升级提供一套通用的解决方案。
     + 支持OS系统、固件、驱动、SDK、导航软件、其他APP程序等的OTA。
   * 丰富的升级策略
     + 支持全量升级，补丁升级(补丁增量包)，差分升级（二进制级别）。
     + 支持用户指定时间升级（延期或特定时间）
     + 支持静默升级,非静默升级(下载确认，更新确认，升级状态提示)。
     + 支持主动升级和被动通知升级
     + 支持版本的迭代，更新，版本的分支等。
     + 采用批量任务管理分批升级，避免升级拥堵
   * 安全性
     + 支持软件包的加密/解密和签名/验签
     + 双向认证支持
   * 高效性
     + CDN分发，终端通过最近的接入点接入
     + 断点续传，部分错误重传，避免整包重新下载
   * 可靠性
     + 数据包的签名验证，确保数据的完整性和可靠性
   1. 整体架构

<架构图>

OTA云端主要包括OP管理平台，OTA升级服务，任务调度服务，OTA差分服务，OTA文件服务几大模块。

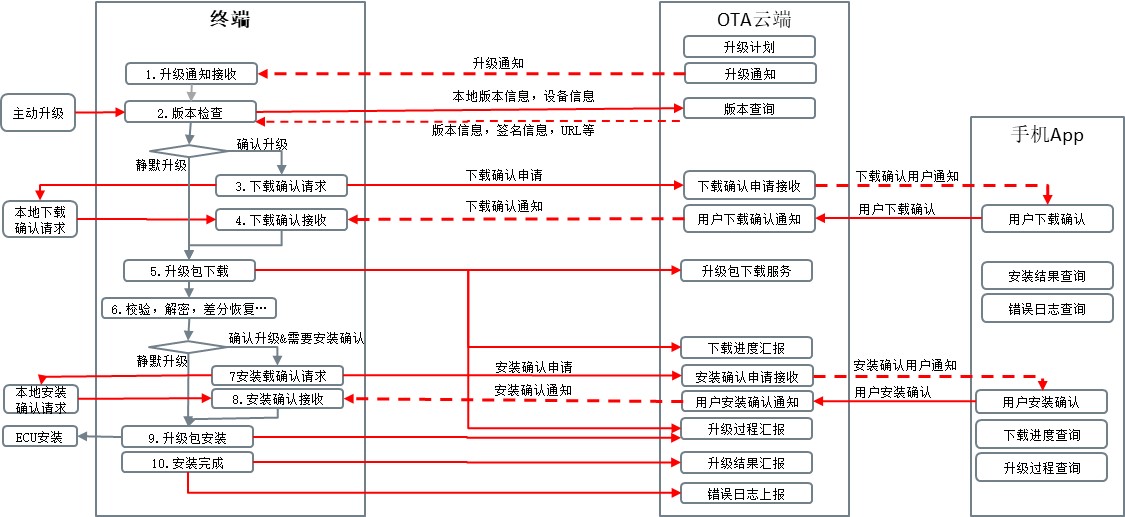
OP管理平台需要提供基础的用户，角色，权限，授权，鉴权等的管理和服务，需要提供基础的多项目能力支持。

OTA管理将以子模块的形式运行在OP平台。

任务调度系统包含任务调度服务和任务调度管理，两部分，任务调度管理作为OTA依赖的子模块运行于OP平台。

OTA通知指令的下行需要外部系统按接口要求实现和提供下行通知的接口。

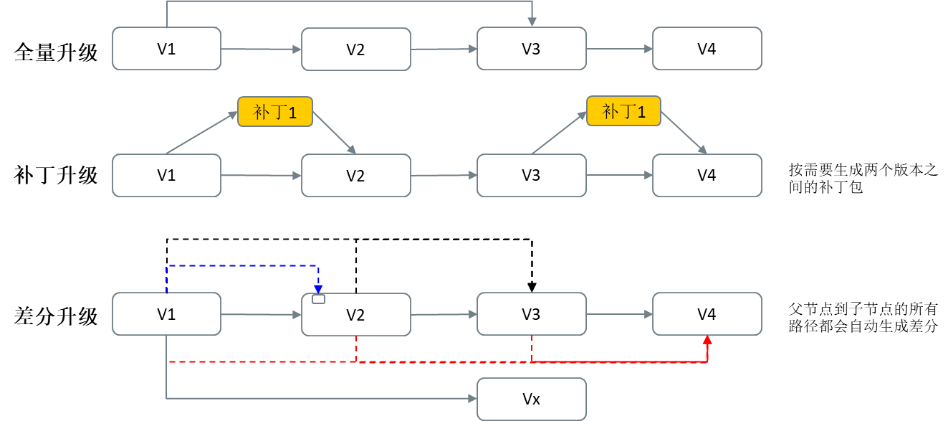
* 1. 交互流程



* 1. 版本管理模式

OTA版本采用单一路径升级模式，一个软件一个升级路径

路径各个节点之间允许存在差分包或者补丁包



* 1. OTA后台管理
     1. 基础数据管理

提供一个对车辆和零部件进行分类的树状结构的管理

* + 1. 固件清单管理

提供固件信息的导入和编辑能力，包含零部件信息以及对应的固件信息

导入固件信息时，需要选择其所属的设备节点



* + 1. 升级对象管理

提供管理需要升级的设备对象的功能，包括设备数据的导入和录入功能。各个项目需要各自管理各自项目的数据。

需要提供设备固件清单导入和管理的功能，设备固件清单，即需要通过该设备进行软件升级的固件的列表。

在设备数据创建或者导入的时候可以通过页面选择该一批设备对应的固件清单。

建立固件同固件软件之间的关系

* + 1. 升级包及版本管理

选择升级类型：全量，补丁，差分

如果是全量包

需要选择对应的软件，全量包对应的版本

上传全量包

创建版本信息

创建升级包信息

如果是补丁包

需要选择对应的软件，起始版本和目标版本

上传补丁包

创建版本信息

创建升级包信息

如果是差分包

需要选择对应的软件，全量包对应的版本，差分的起点

上传全量包

差分计算

升级路径生成

* + 1. 升级计划管理

调整要求：

1. 新增升级计划时，需要弹出新窗口，而不是打开一个新的tap
2. 界面太难看，采用图形化方式从树上选取版本
3. 选取设备的筛选条件很单一

这块需要结合设备的分类选择，比如按设备所在的区域，设备生产日期，销售日期等条件过滤，设备固件版本号等。

1. 支持升级通知
2. 按当前需升级的版本查询满足条件的设备
3. 支持计划任务的调度/无计划调度
4. OTA计划支持审批与发布
   1. 升级计划调度
   2. OTA终端的服务接口
      1. OTA版本更新检查
         1. 服务描述

OTA终端需要通过该接口向云端查询指定软件的最新版本信息或者指定软件版本的版本信息。

OTA终端可能是出于一个主动版本检查的需求向云端检查软件版本更新，或者是因为收到了来自云端的升级通知后向云端发起的一个版本更新检查。

考虑到某些情况下需要采用双系统交叉升级,在OTA版本更新检查时需要告诉云端当前设备的运行版本和待升级版本。

需要升级的设备的当前软件本部必须是目标版本的适应版本。

* + - 1. 服务地址

https post {root}/ota/versionCheck

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0,默认为0

objectId:升级对象的Id, 比如vin号

softwareCode:需要升级的部件中的某软件的软件代码

runningVersion: 当前运行的版本

upgradeVersion:升级的升级起始版本,在双系统时，需要升级备份版本

targetVersion: 可选的，在某些场景下，期望升级到特定版本

* + - 1. 处理过程

1. 检查当前升级对象是否在一个有效的升级计划或者白名单中（白名单未来规划）

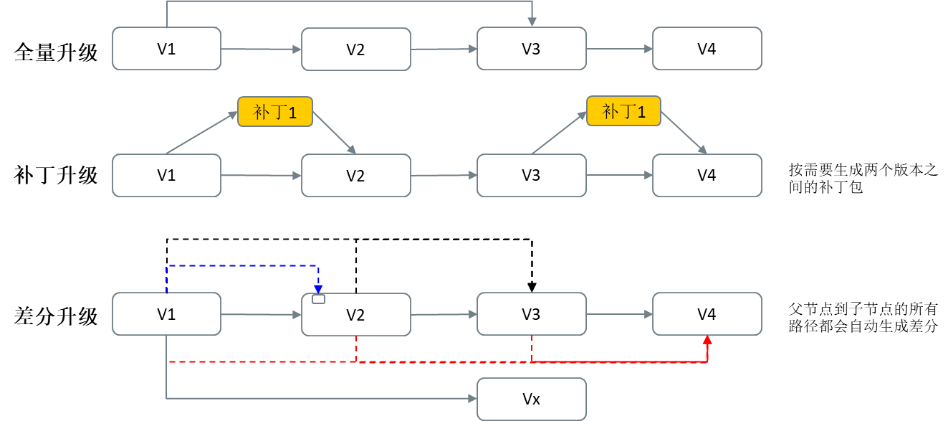
只要当前升级对象在一个有效的升级计划或者白名单即可升级。

1. 获取需要升级的软件信息和版本信息

联合“OTA升级对象“，“OTA升级对象部件清单”，“OTA升级部件软件清单”，“OTA软件”表根据objectType, objectId, softwareCode查询出OTA软件信息,需要升级的升级对象指定软件的软件版本信息(初始化版本，运行版本，待升级版本等)。

目的：

1. 检查被升级的对象是否存在
2. 检查被升级的对象是否存在参数中指定的软件
3. 获取被升级的软件的配置信息（软件表）
4. 查询需要升级的软件的版本信息



如上图所示：

针对“全量升级”，能升级到V4的只有V3；V1,V2是无法升级的，即只支持单级升级。

两个版本之间的大小根据版本号规则比较

在“软件版本升级路径“表中查询满足下述条件的版本：

路径的起点版本=upgradeVersion

路径的终点版本!=runningVersion

路径的终点版本=targetVersion (如果targetVersion不为空)

升级路径类型，根据软件中的“升级包模式”判断如下：

如果为“全量升级”，则路径类型=“全量升级”

如果为“补丁升级”，则路径类型为“全量路径” or “补丁路径”

如果为“差分升级”，则路径类型为“全量路径” or “差分路径”

在上述查询的升级路径中，如果出现多个版本,取最新的版本(。

1. 创建升级记录

如果需要升级的软件不存在升级记录，则创建升级记录

* + - 1. 返回结果

**升级模式:**全量包升级/补丁包升级/差分包升级

**加密方式:**非对称加密/对称加密(证书/约定的对称密钥)

**钻孔升级:**是/否采用钻孔升级法，会从升级包中挖出一些残缺碎片，使升级包不完整

**静默升级/非静默升级：**0-强制静默升级 1-非静默升级

**通知下载类型：**0 - 先下载后通知 1 - 先通知后下载

**是否通知到手机：**

**是否需要下载确认：**

**是否需要安装确认：**

**APN通道：**

**适用硬件版本:**

**起始版本号：**

**目标版本号：**

**是否还有后续版本:**

**升级记录sid：**升级记录的内部sid

**全量包信息:**如果存在全量包，则全量包信息总是返回

**包url：**升级包下载路径

**包解密算法：**

**包解密密钥：**该密钥是采用加密的方式存储的

**包签名算法**:

**包签名:**

**包描述：**

**包大小：**加密后的大小,单位byte

**升级包打包算法：**

**错误处理机制：**

**错误重试次数：**

**残缺包信息:**

采用钻孔升级法，需要返回此残缺信息，含多残缺块

升级包碎片数据结构：

[N][Index][L][data][L][data][L][data]……

N: 一个字节，代表有多少个碎片

Index: 表示数据在包中的开始位置

L:表示该数据的长度

Data:为具体数据，长度为L

上述数据采用Base64编码的字符格式。

上述信息采用已被采用客户公钥非对称加密。

上述信息加密后采用Base64编码为字符格式

**增量信息(补丁/差分升级):**

如果没有对应的内容，则该字段的内容为空，但字段名保留返回

*数据格式参考上述“全量包”*

**升级依赖:所依赖的升级软件版本信息**

多个软件和软件版本对应的集合列表[{}]

依赖顺序： 依赖的顺序，1,2,…, 最小的需要最先安装

软件代码：所依赖的软件代码

软件版本：所依赖的软件版本

* + - 1. 使用示例
    1. OTA升级包下载
       1. 服务描述

通过OTA云端或CDN提供OTA升级包的文件下载，要求支持断点续传。

* + - 1. 服务地址

https get {root}/ota/file/…/filename

* + - 1. 请求参数
      2. 处理过程
      3. 返回结果
      4. 使用示例
    1. OTA升级包下载进度汇报
       1. 服务描述

提供更新下载进度的接口，终端每隔20s(可配置化)向云端汇报一次当前已下载的字节数。

当终端下载完成时，必须向后台汇报100%下载完成的进度，即“下载进度”更新。

* + - 1. 服务地址

https post {root}/ota/download/progress/{objectType}/{objectId}/{otaRecordSid}/{size}

* + - 1. 请求参数

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

otaRecordSid:升级记录内部sid (获取版本时返回)

size:已下载字节数(byte)

* + - 1. 处理过程

验证objectId和otaRecordSid之间的匹配关系是否正确

根据otaRecordSid查找到升级记录

在“OTA升级下载进度“表中插入下载进度(需要考虑插入的数据量)

如果下载刚开始

在“OTA升级过程信息”插入“软件下载过程”-开始时间和状态）

如果下载100%完成

在“OTA升级过程信息”更新“软件下载过程”-完成时间和状态

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
    1. OTA升级过程汇报
       1. 服务描述

OTA升级由一系列的升级过程组成,终端需要通过本接口向云端即时汇报升级的各个过程进度信息。

终端需要确保过程信息汇报到后台，需要具有错误重新汇报的机制

OTA升级过程：

1. 升级通知

* Ota通知下发时需要创建升级通知过程，写入过程的开始时间
* 终端收到OTA通知时，需要通过此接口汇报更新“升级通知“过程，写入过程的结束时间

1. 检查版本

* 版本更新检查接口需要创建“版本检查”过程，写入过程的开始时间和结束时间

1. 软件下载确认

* 面向终端的软件“下载确认请求“接口，需要创建本过程，写入过程的开始时间
* 终端收到”OTA下载确认“的指令后，需要通过此接口汇报更新” 软件下载确认“过程，写入过程的结束时间

1. 软件下载

* 终端开始下载时，需要通过此接口汇报”软件下载“过程的开始时间
* 终端下载完成时，需要通过此接口汇报”软件下载“过程的结束时间

1. 请求安装确认

* 面向终端的软件“安装确认请求“接口，需要创建本过程，写入过程的开始时间
* 终端收到”OTA安装确认“的指令后，需要通过此接口汇报更新” 软件安装确认“过程，写入过程的结束时间

1. 软件安装

* 终端开始安装时，需要通过此接口汇报”软件安装“过程的开始时间
* 终端安装完成时，需要通过此接口汇报”软件安装“过程的结束时间

1. 软件回滚

* 终端开始回滚时，需要通过此接口汇报”软件回滚“过程的开始时间
* 终端回滚完成时，需要通过此接口汇报”软件回滚“过程的结束时间
  + - 1. 服务地址

https post {root}/ota/upgrade/progress

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

otaRecordSid:升级记录内部sid (获取版本时返回)

progressID:过程ID

startTime:开始时间，对于瞬间过程，开始时间=结束时间

endTime：结束时间

progressStatus：S-成功/ F-失败

progressDesc：文字描述，比如一些错误的原因等

* + - 1. 处理过程

根据otaRecordSid查找到升级记录

在“OTA升级过程信息“表中插入升级过程信息

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
    1. OTA升级结果汇报
       1. 服务描述

OTA升级完成后（升级重启证明了软件升级成功了后）向云端汇报升级结果，针对某个OTA软件版本升级成功还是失败，升级后的版本号情况。

* + - 1. 服务地址

https post {root}/ota/upgrade/result

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

softwareCode:需要升级的部件中的某软件的软件代码

runningVersion: 软件当前运行的版本

upgradeVersion: 需要升级的版本（双系统下，下次升级的版本）

status:

remarks;

……

* + - 1. 处理过程

根据otaRecordSid查找到升级记录

更新升级记录中的状态和版本信息

更新设备的版本信息(怎么和设备服务集成？设备版本信息在设备还是在OTA?)

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
    1. 下载确认请求
       1. 服务描述

终端向云端发送请求，要求用户确认是否需要下载此版本更。

* + - 1. 服务地址

https post {root}/ota/downloadConfirmReq

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

softwareCode:依赖的软件代码

otaRecordSid:升级记录内部sid (获取版本时返回)

* + - 1. 处理过程

终端在完成版本检查后，如果确定了需要升级此软件，则

终端向云端发送下载确认请求，该请求将包含需要下载的软件名称，大小，描述等信息，云端将该请求放入升级过程表(状态：待下载确认)。

云端可以将此升级确认通过推送的方式推送到用户或者让用户通过主动查询的方式查询到该确认列表。

用户可以通过App对该请求进行确认,用户可以选择下载更新，取消或者不予理会。在用户做出选择后，云端将把用户的选择确认消息推送到终端。

是否需要等待用户的下载确认，依赖被升级软件或版本采用的升级策略（是否需要用户确认，该信息在版本检查时返回）。

依赖数据库表：升级记录表

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
    1. 安装确认请求
       1. 服务描述

终端通过此接口向云端确认是否需要安装此版本更新的升级包。

* + - 1. 服务地址

https post {root}/ota/installConfirmReq

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

softwareCode:依赖的软件代码

otaRecordSid:升级记录内部sid (获取版本时返回)

* + - 1. 处理过程

终端完成OTA升级包的下载和升级包校验，等待安装前，需要向用户发起安装确认请求，该请求主要包含待安装的升级记录ID。云端将更新升级记录为“待安装确认”状态。

云端可以将此安装确认通过推送的方式推送到用户或者让用户通过主动查询的方式查询到该确认列表。

用户可以通过App对该请求进行安装确认,用户可以选择安装，删除或者不予理会此更新。在用户做出选择后，云端将把用户的选择确认消息推送到终端。

是否需要等待用户的安装确认，依赖被升级软件或版本采用的安装策略（是否需要用户安装确认，该信息在版本检查时返回）。

依赖数据库表：升级记录表

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
    1. 升级错误日志上传（暂时不实现）
       1. 服务描述

提供上报错误升级日志的接口，在升级失败后，终端通过该接口将升级的错误日志上传到后台。以供后续的错误原因调查。

* + - 1. 服务地址

https post {root}/ota/uploadOtaLog

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

softwareCode:需要升级的部件中的某软件的软件代码

runningVersion: 软件当前运行的版本

upgradeVersion: 需要升级的版本（双系统下，下次升级的版本）

versionCheckResult:版本检查结果, 来源于OTA版本检查结果

otaLogs：相关的错误日志信息，以压缩+base64编码的形式保存

* + - 1. 处理过程

将升级错误日志保存到升级错误日志信息表中(大数据实现？)

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
  1. OTA用户交互接口
     1. 用户下载确认
        1. 服务描述

该接口提供给用户对来自终端的软件更新进行安全确认

* + - 1. 服务地址

https post {server}/ota/downloadConfirm

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

softwareCode:依赖的软件代码

otaRecordSid:升级记录内部sid (获取版本时返回)

confirmType: 0 – 同意下载 1 – 拒绝下载

* + - 1. 处理过程

云端需要提供接口接收来自用户的下载确认

当云端收到用户的下载确认后，将更新数据库中升级记录的状态(状态：已确认下载)，并将该确认消息通过“OTA下载确认下发”指令推送到终端。

依赖数据库表：升级记录表

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
    1. 用户安装确认
       1. 服务描述

该接口提供给用户对来自终端的软件更新进行安全确认

* + - 1. 服务地址

https post {server}/ota/installConfirm

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

softwareCode:软件代码

otaRecordSid:升级记录内部sid (获取版本时返回)

confirmType: 0 – 同意安装 1 – 删除安装包

* + - 1. 处理过程

云端需要提供接口接收来自用户的安装确认

当云端收到用户的安装确认后，将更新数据库中升级记录的状态(状态：已确认安装)，并将该确认消息通过“OTA安装确认下发”指令推送到终端。

依赖数据库表：升级记录表

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
    1. 设备升级记录查询 （待定）
       1. 服务描述

该接口主要暴露给外部业务系统查询指定设备当前正在升级的升级对象的升级清单（待升级列表，待下载确认，待安装确认，已下载，已安装等）

* + - 1. 服务地址
      2. 请求参数
      3. 处理过程
      4. 返回结果
      5. 使用示例
  1. TSP接口依赖
     1. OTA指令下发到终端接口
        1. 服务描述

该指令接口需要外部接口实现，用于OTA平台向OTA终端设备推送OTA升级消息。

* + - 1. 服务地址

https post {server}/ota/upgrade/notification

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

otaChannel:升级渠道

notifyType: 具体参考“面向后台OTA指令下发”

notifyMsg: 通知内容,具体参考“面向后台OTA指令下发”

* + - 1. 处理过程

无

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
    1. OTA用户通知接口
       1. 服务描述

该指令接口需要外部接口实现，用于OTA平台向OTA App用户推送升级确认消息。

该接口需要外部业务系统实现

* + - 1. 服务地址

https post {server}/ota/upgrade/notify/user

* + - 1. 请求参数

以json对象post

objectType:需要升级的对象类型，车辆传0

objectId:升级对象的Id, 比如vin号

otaChannel:升级渠道

notifyType: 参考” 后台OTA用户通知下发“

notifyMsg: 参考” 后台OTA用户通知下发“

* + - 1. 处理过程

无

* + - 1. 返回结果

返回通用的Restful 成功返回结果

* + - 1. 使用示例
  1. OTA终端指令消息
     1. OTA升级通知
        1. 描述

该接口提供给OTA后台用于向终端下发OTA升级通知的能力，OTA通知的下行依赖终端同云端已有的双向通讯机制和通讯协议，比如MQTT协议，TC/IP协议等。OTA本身不提供该通讯协议。

在下发OTA升级通知时，如果对应的软件升级记录（不包含依赖的软件）不存在则创建升级记录信息。

需要创建升级过程信息

* + - 1. 下行指令格式

{

planId: <计划的sid>

swList：

[

{

软件代码

可升级版本

发布日期

版本描述

安装顺序

}

]

}

* + - 1. 触发方式

OTA升级计划开始后，由任务调度触发该指令的下发

* + - 1. 使用示例
    1. OTA下载确认
       1. 描述

云端将来自用户的下载确认发送给终端

需要创建“下载升级过程”

* + - 1. 下行指令格式

{

softwareCode:软件代码

otaRecordSid:升级记录内部sid (获取版本时返回)

confirmType: 0 – 同意安装 1 – 删除安装包

}

* + - 1. 触发方式

业务平台调用OTA提供的“用户下载确认”接口将本消息下发给指定的终端

* + - 1. 使用示例
    1. OTA安装确认
       1. 描述

云端将来自用户的安装确认发送给终端

需要创建“安装升级过程”

* + - 1. 下行指令格式

{

softwareCode: 软件代码

otaRecordSid: 升级记录内部sid (获取版本时返回)

confirmType: 0 – 同意安装 1 – 删除安装包

}

* + - 1. 触发方式
      2. 使用示例
  1. OTA用户通知消息
     1. OTA下载确认请求通知
        1. 描述

OTA平台需要采用本消息格式将来自终端的下载确认请求通过业务系统提供的“OTA用户通知接口”通知到用户。

* + - 1. 下行指令格式

{

softwareName:软件名称

softwareCode:软件代码

softwareVersion:软件版本

size: 软件大小

estimateTime:预计下载时间

}

* + - 1. 触发方式
      2. 使用示例
    1. OTA下载安装请求通知
       1. 描述

OTA平台需要采用本消息格式将来自终端的安装确认请求通过业务系统提供的“OTA用户通知接口”通知到用户。

* + - 1. 下行指令格式

{

softwareCode:软件代码

softwareVersion:软件版本

}

* + - 1. 触发方式
      2. 使用示例