

设备升级管理

概述

- ,

接口详情

1. 创建自动升级任务

- 创建一个自动升级任务，需要提供起始版本和目标版本的固件信息。
- 了解固件信息可查看文档 产品与设备管理接口
- Api-Version请求头，默认不传为0，可选字段为2

Request

URL

```
POST /v2/upgrade/firmware/task
```

Header

```
Content-Type:"application/json"  
Access-Token:"/"  
Api-Version:"2"
```

Content

```
{  
  "product_id":"ID",  
  "name":"",  
  "description":"",  
  "from_version":"",  
  "from_version_url":"",  
  "from_version_md5":"MD5",  
  "from_version_size":"",  
  "target_version":"",  
  "target_version_url":"",  
  "target_version_md5":"MD5",  
  "target_version_size":"",  
  "type" : "",  
  "identify":"MCU",  
  "scope":{  
    "category":"", 0:, 1:",  
    "device_list":["ID1"]  
  },  
  "task_type":"",  
  "tag_type":"", 0:, 1:"  
}
```

• 概述

• 接口详情

1. 创建自动升级任务
2. 获得自动升级任务列表GET
3. 开始和停止自动升级任务
4. 删除升级任务
5. 查询最新升级任务
6. 上报升级结果
7. 获得自动升级任务列表POST
8. 下发ota升级日志
9. 查询ota升级日志
10. 升级任务由测试类型转为正式类型
11. 提交升级任务审核
12. 审核升级任务
13. 为测试任务变更指定设备
14. 获取升级任务
15. 查询操作日志接口
16. 查询自定义升级任务版本占比

• 升级附录

1. 设备固件类型
2. 设备升级任务类型

字段	是否必须	描述
----	------	----

product_id	是	产品ID
name	是	自动升级任务名称, 32个字符以内
description	是	描述, 250个字符以内
from_version	是	起始版本号, 32位整形
from_version_url	是	起始版本文件地址
from_version_md5	是	起始版本文件MD5值
from_version_size	是	起始版本文件大小
target_version	是	目标版本文件号
target_version_url	是	目标版本文件地址
target_version_md5	是	目标版本文件MD5
target_version_size	是	目标版本文件大小
type	否	升级任务类型, 新增, 见附件升级任务类型, 默认为wifi
identify	否	用来定位多MCU或多子设备的情况, 数字, type为mcu或多子设备时必须传, 默认为0, type/identify/from_version联合唯一
task_type	否	升级任务模式, 0自动模式, 1手动模式
scope	否	升级范围
scope.category	否	, 0:, 1:
scope.device_list	否	指定升级设备标识列表
tag_type	否	升级任务标签类型, 0:正式任务, 1:测试任务, 不传默认为正式任务

Response

Header

HTTP/1.1 200 OK

Content

```

{
  "id": "ID",
  "product_id": "ID",
  "name": "",
  "description": "",
  "from_version": "",
  "from_version_url": "",
  "from_version_md5": "MD5",
  "from_version_size": "",
  "target_version": "",
  "target_version_url": "",
  "target_version_md5": "MD5",
  "target_version_size": "",
  "upgrade_count": "",
  "status": "",
  "type": "",
  "identify": "MCU",
  "task_type": ", 0 , 1 ",
  "scope": {
    "category": ", 0:, 1:",
    "device_list": ["ID1"]
  },
  "tag_type": ", 0:, 1:",
  "create_time": ",
  yyyy-MM-dd'T'HH:mm:ss.SS'Z'"
}

```

字段	是否必须	描述
id	是	自动升级任务ID
product_id	是	产品ID
name	是	自动升级任务名称，32个字符以内
description	是	描述，250个字符以内
from_version	是	起始版本号，32位整形
from_version_url	是	起始版本文件地址
from_version_md5	是	起始版本文件MD5值
from_version_size	是	起始版本文件大小
target_version	是	目标版本文件号
target_version_url	是	目标版本文件地址
target_version_md5	是	目标版本文件MD5
target_version_size	是	目标版本文件大小
upgrade_count	是	已升级设备数量
status	是	任务状态：0：停止状态，1：启动状态，2：待执行，3：待审核 Api-Version为2时默认为待执行状态，其他默认为停止状态

type	是	升级任务类型, 新增, 见附件 升级任务类型, 默认为wifi
identify	是	用来定位多MCU或多子设备的情况, 数字, type为mcu或多子设备时存在, 默认为0, type/identify/from_version联合唯一
task_type	是	升级任务模式, 0自动模式, 1手动模式
scope	是	升级范围
scope.category	是	范围类别, 0:全网, 1:指定设备
scope.device_list	是	指定升级设备标识列表
tag_type	是	升级任务标签类型, 0:正式任务, 1:测试任务, 不传默认为正式任务
create_time	是	升级任务创建时间, 格式为yyyy-MM-dd'T'HH:mm:ss.SS'Z'

2. 获得自动升级任务列表GET

Request

URL

```
GET
/v2/upgrade/firmware/tasks?product_id={product_id}&offset=offset&limit=limit
```

Header

```
Content-Type: "application/json"
Access-Token: " /XLINK "
Api-Version: " 2 "
```

Content

Response

Header

```
HTTP/1.1 200 OK
```

Content

```
[{
  "id": "ID",
  "product_id": "ID",
  "name": "",
  "from_version": "",
  "from_version_url": "",
  "from_version_md5": "MD5",
  "from_version_size": "",
  "target_version": "",
  "target_version_url": "",
  "target_version_md5": "MD5",
  "target_version_size": "",
  "upgrade_count": "",
  "status": "",
  "type": "",
  "identify": "MCU",
  "description": "",
  "surplus_count": "",
  "task_type": ", 0, 1",
  "scope": {
    "category": ", 0:, 1:",
    "device_list": ["ID1"]
  },
  "tag_type": ", 0:, 1:",
  "audit_member_id": "",
  "create_time": "",
  "yyyy-MM-dd'T'HH:mm:ss.SS'Z'"
}]
```

字段	是否必须	描述
id	是	自动升级任务ID
product_id	是	产品ID
name	是	自动升级任务名称，32个字符以内
description	是	描述，250个字符以内
from_version	是	起始版本号，32位整形
from_version_url	是	起始版本文件地址
from_version_md5	是	起始版本文件MD5值
from_version_size	是	起始版本文件大小
target_version	是	目标版本文件号
target_version_url	是	目标版本文件地址
target_version_md5	是	目标版本文件MD5
target_version_size	是	目标版本文件大小
upgrade_count	是	已升级设备数量

status	是	任务状态：0：停止状态，1：启动状态，2：待执行，3：待审核 Api-Version为2时 有可能为0、1、2、3 Api-Version为其他时，只有0、1
type	是	升级任务类型, 新增, 见附件 升级任务类型, 默认为wifi
identify	是	用来定位多MCU或多子设备的情况, 数字, 默认为0, type/identify/from_version联合唯一
description	是	描述
surplus_count	否	当前剩余版本号为起始版本号的设备总数, 即是要升级的设备数, type为wifi时才有
task_type	是	升级任务模式, 0自动模式, 1手动模式
scope	是	升级范围
scope.category	是	范围类别, 0:全网, 1:指定设备
scope.device_list	是	指定升级设备标识列表
tag_type	是	升级任务标签类型, 0:正式任务, 1:测试任务, 不传默认为正式任务
audit_member_id	否	审批人标识
create_time	是	升级任务创建时间, 格式为yyyy-MM-dd'T'HH:mm:ss.SS'Z'

3. 开始和停止自动升级任务

Request

URL

POST /v2/upgrade/firmware/task/status

Header

Content-Type: "application/json"
Access-Token: "/"

Content

```
{
  "product_id": "ID",
  "upgrade_task_id": "ID",
  "status": ""
}
```

字段	是否必须	描述
product_id	是	产品ID
upgrade_task_id	是	升级任务ID
status	是	任务状态, 0: 停止状态, 1: 启动状态

Response

Header

```
HTTP/1.1 200 OK
```

Content

4. 删除升级任务

Request

URL

```
POST /v2/upgrade/firmware/task/delete
```

Header

```
Content-Type: "application/json"
Access-Token: "/"
```

Content

```
{
  "id": "ID",
  "product_id": "ID"
}
```

Response

Header

```
HTTP/1.1 200 OK
```

Content

5. 查询最新升级任务

```
,product_id, device_id, type, from_version,
```

Request

URL

```
POST /v2/upgrade/firmware/check/{device_id}
```

Header

```
Content-Type: "application/json"  
Access-Token: " / "
```

Content

```
{  
  "product_id": "ID",  
  "type": "",  
  "current_version": "",  
  "identify": "MCU"  
}
```

字段	是否必须	描述
product_id	是	产品ID
type	是	升级任务类型, 新增, 见附件升级任务类型, 默认为wifi
current_version	是	设备当前版本
identify	是	用来定位多MCU或多子设备的情况, type若为"wifi", identify设置为0。

Response

Header

HTTP/1.1 200 OK

Content

```
{
  "id": "ID",
  "product_id": "ID",
  "name": "",
  "description": "",
  "from_version": "",
  "from_version_url": "",
  "from_version_md5": "MD5",
  "from_version_size": "",
  "target_version": "",
  "target_version_url": "",
  "target_version_md5": "MD5",
  "target_version_size": "",
  "upgrade_count": "",
  "status": "",
  "type": "",
  "identify": "MCU",
  "scope": {
    "category": ", 0:, 1:",
    "device_list": ["ID1"]
  }
}
```

字段	是否必须	描述
id	是	自动升级任务ID
product_id	是	产品ID
name	是	自动升级任务名称, 32个字符以内
description	是	描述, 250个字符以内
from_version	是	起始版本号, 32位整形
from_version_url	是	起始版本文件地址
from_version_md5	是	起始版本文件MD5值
from_version_size	是	起始版本文件大小
target_version	是	目标版本文件号
target_version_url	是	目标版本文件地址
target_version_md5	是	目标版本文件MD5
target_version_size	是	目标版本文件大小
upgrade_count	是	已升级设备数量
status	是	任务状态: 0: 停止状态, 1: 启动状态, 默认为停止状态。
type	是	升级任务类型, 新增, 见附件升级任务类型, 默认为wifi

identify	是	用来定位多MCU或多子设备的情况, 数字, 默认为0
scope	否	升级范围
scope.category	否	范围类别, 0:全网, 1:指定设备
scope.device_list	否	指定升级设备标识列表

6. 上报升级结果

```
, type, identify, last_version, current_version, , id, mq
```

Request

URL

```
POST /v2/upgrade/firmware/report/{device_id}
```

Header

```
Content-Type: "application/json"
Access-Token: "//XLINK"
```

Content

```
{
  "type": "",
  "mod": "",
  "identify": "MCU",
  "last_version": "version",
  "current_version": "version",
  "result": "0",
  "desc": "",
  "start_date": "",
  "end_date": ""
}
```

字段	是否必须	描述
type	是	升级任务类型
mod	是	固件型号
identify	是	用来定位多MCU或多子设备的情况
last_version	是	升级前version
current_version	是	升级后version
result	是	升级结果, 0:成功, 1:失败
desc	否	升级状况描述

start_date	否	升级开始时间, 例: 2015-10-09T08:15:40.843Z
end_date	否	升级结束时间, 例: 2015-10-09T08:15:40.843Z

Response

Header

```
HTTP/1.1 200 OK
```

Content

7. 获得自动升级任务列表POST

Request

URL

```
POST /v2/upgrade/firmware/tasks
```

Header

```
Content-Type: "application/json"  
Access-Token: "//XLINK"  
Api-Version: "2"
```

Content

```

{
  "product_id": "id",
  "offset": "",
  "limit": "",
  "type": "",
  "tag_type": ", 0:, 1:",
  "query": {
    "_id": {
      "$eq": ""
    },
    "name": {
      "$like": ""
    }
  },
  "order": {
    "name": "desc/asc"
  }
}

```

字段	是否必须	描述
product_id	是	产品id
offset	否	请求偏移量, 默认值0
limit	否	每页请求量, 默认值10
type	否	不传时, 为全部类型
tag_type	否	任务标签类型, 0:正式任务, 1:测试任务
query	否	查询条件, 仅仅支持name和id 字段

Response

Header

```
HTTP/1.1 200 OK
```

Content

```

{
  "count": "",
  "list": [{
    "id": "ID",
    "product_id": "ID",
    "name": "",
    "from_version": "",
    "from_version_url": "",
    "from_version_md5": "MD5",
    "from_version_size": "",
    "target_version": "",
    "target_version_url": "",
    "target_version_md5": "MD5",
    "target_version_size": "",
    "upgrade_count": "",
    "status": "",
    "type": "",
    "identify": "MCU",
    "description": "",
    "surplus_count": "",
    "task_type": ", 0, 1",
    "scope": {
      "category": ", 0:, 1:",
      "device_list": ["ID1"]
    },
    "tag_type": ", 0:, 1:",
    "audit_member_id": "",
    "create_time": "",
    yyyy-MM-dd'T'HH:mm:ss.SS'Z'
  }]
}

```

字段	是否必须	描述
count	是	返回总数
id	是	自动升级任务ID
product_id	是	产品ID
name	是	自动升级任务名称，32个字符以内
description	是	描述，250个字符以内
from_version	是	起始版本号，32位整形
from_version_url	是	起始版本文件地址
from_version_md5	是	起始版本文件MD5值
from_version_size	是	起始版本文件大小
target_version	是	目标版本文件号
target_version_url	是	目标版本文件地址
target_version_md5	是	目标版本文件MD5
target_version_size	是	目标版本文件大小
upgrade_count	是	已升级设备数量

status	是	任务状态：0：停止状态，1：启动状态，2：待执行，3：待审核 Api-Version为2时 有可能为0、1、2、3 Api-Version为其他时，只有0、1
type	是	升级任务类型, 新增, 见附件 升级任务类型, 默认为wifi
identify	是	用来定位多MCU或多子设备的情况, 数字, 默认为0, type/identify/from_version联合唯一
description	是	描述
surplus_count	否	当前剩余版本号为起始版本号的设备总数, 即是要升级的设备数, type为wifi时才有
task_type	是	升级任务模式, 0自动模式, 1手动模式
scope	是	升级范围
scope.category	是	范围类别, 0:全网, 1:指定设备
scope.device_list	是	指定升级设备标识列表
tag_type	是	升级任务标签类型, 0:正式任务, 1:测试任务, 不传默认为正式任务
audit_member_id	否	审批人标识
create_time	是	升级任务创建时间, 格式为yyyy-MM-dd'T'HH:mm:ss.SS'Z'

8. 下发ota升级日志

- 升级网关向设备下发升级指令成功后, 由升级网关向API服务增加一条下发ota升级日志

Request

URL

```
POST /v2/upgrade/ota-log
```

Header

```
Content-Type: "application/json"
Access-Token: "XLINK"
Corp-ID: " "
```

Content

```
{
  "device_id": "",
  "task_id": ""
}
```

字段	是否必须	描述
device_id	是	设备标识
task_id	是	升级任务标识

Response

Header

```
HTTP/1.1 200 OK
```

Content

9. 查询ota升级日志

- 查询ota升级日志，用于管理平台获取使用

Request

URL

```
POST /v2/upgrade/ota-logs
```

Header

```
Content-Type:"application/json"
Access-Token:"/"
```

Content

```

{
  "offset": "",
  "limit": "",
  "filter": [
    "A",
    "B"
  ],
  "query": {
    "filed1": {
      "$like": ""
    },
    "filed3": {
      "$lt": ""
    }
  },
  "order": {
    "filed1": "desc",
    "filed2": "asc"
  }
}

```

字段	是否必须	描述
offset	否	从某个偏移量开始请求，默认为0
limit	否	请求的条目数量，默认为10
order	否	可以指定通过设备默认的某个字段排序，desc降序，asc升序
filter	否	字段过滤，集合类型，可以指定返回结果列表的字段，可以包含扩展属性字段
query	否	<p>查询条件，可以根据不同字段加上不同的比较指令来查询，查询条件字段包含设备所有默认字段，不支持扩展属性字段，支持比较指令包含如下：</p> <p>\$in: 包含于该列表任意一个值</p> <p>\$lt: 小于该字段值</p> <p>\$lte: 小于或等于字段值</p> <p>\$gt: 大于该字段值</p> <p>\$gte: 大于或等于该字段值</p> <p>\$like: 模糊匹配该字段值</p> <p>可提供字段有：</p> <ul style="list-style-type: none"> • task_id • device_id • device_mac • device_sn

Response

Header

HTTP/1.1 200 OK

Content

```
{
  "count": "",
  "list": [{
    "id": "ota",
    "product_id": "",
    "task_id": "",
    "device_id": "",
    "device_mac": "mac",
    "device_sn": "",
    "status": "",
    "execute_time": ""
  }]
}
```

字段	是否必须	描述
count	是	总数量
list	是	设备列表
id	是	ota升级日志标识
product_id	是	产品标识
task_id	是	升级任务标识
device_id	是	设备标识
device_mac	是	设备mac地址
device_sn	是	设备序列号
status	是	升级状态, 1: 已下发, 2: 已完成
execute_time	是	结束时间

10. 升级任务由测试类型转为正式类型

- 升级任务由测试类型转为正式类型
- 状态转为待执行状态
- 可以修改参数包括
 - name: 升级名称
 - description: 升级描述
 - scope

Request

URL

```
PUT
/v2/upgrade/firmware/tasks/{task_id}/product-
tag
```

Header

```
Content-Type:"application/json"
Access-Token:"/"
```

Content

```
{
  "name": "",
  "description": "",
  "scope": {
    "category": ", 0:", 1:",
    "device_list": ["ID1"]
  }
}
```

字段	是否必须	描述
name	否	升级任务名称
description	否	升级任务描述
scope	否	升级范围
scope.category	否	范围类型，当出现scope时，category必须
scope.device_list	否	当scope.category为指定设备时，必须

Response

Header

```
HTTP/1.1 200 OK
```

Content

11. 提交升级任务审核

- 任务类型必须为正式任务
- 任务状态必须为待执行状态
- 必须填写审核人(企业成员)
- 提交后状态重置为待审核状态

Request

URL

```
PUT
/v2/upgrade/firmware/tasks/{task_id}/submit-audit
```

Header

```
Content-Type: "application/json"
Access-Token: "/"
```

Content

```
{
  "audit_member_id": ""
}
```

字段	是否必须	描述
audit_member_id	是	审核人标识

Response

Header

```
HTTP/1.1 200 OK
```

Content

12. 审核升级任务

- 任务类型必须为正式任务
- 任务状态必须为待审核状态
- 接口调用者必须与任务审批人标识相等
- 如果审批人批准，则任务状态转换为执行中
- 如果审批人拒绝，则任务状态转换为待执行
- 将任务审批人置为空

Request

URL

```
PUT
/v2/upgrade/firmware/tasks/{task_id}/audit
```

Header

```
Content-Type: "application/json"
Access-Token: " / "
```

Content

```
{
  "audit_result": ""
}
```

字段	是否必须	描述
audit_result	是	审批结果, 1:批准, 2:拒绝

Response

Header

```
HTTP/1.1 200 OK
```

Content

13. 为测试任务变更指定设备

- 任务类型必须为测试任务
- 目前只能修改升级范围
- 全量变更

Request

URL

```
PUT
/v2/upgrade/firmware/tasks/{task_id}/devices
```

Header

```
Content-Type: "application/json"
Access-Token: " / "
```

Content

```
{
  "opt": "",
  "device_list": [
    "device_id1",
    "device_id2"
  ]
}
```

字段	是否必须	描述
opt	是	操作, 1:新增, 2:删除, 3:覆盖
device_list	是	设备标识列表

Response

Header

```
HTTP/1.1 200 OK
```

Content

14. 获取升级任务

- 根据标识获取升级任务

Request

URL

```
GET /v2/upgrade/firmware/tasks/{task_id}
```

Header

```
Content-Type:"application/json"
Access-Token:"/"
Api-Version:"2"
```

Content

Response

Header

HTTP/1.1 200 OK

Content

```
{
  "id": "ID",
  "product_id": "ID",
  "name": "",
  "from_version": "",
  "from_version_url": "",
  "from_version_md5": "MD5",
  "from_version_size": "",
  "target_version": "",
  "target_version_url": "",
  "target_version_md5": "MD5",
  "target_version_size": "",
  "upgrade_count": "",
  "status": "",
  "type": "",
  "identify": "MCU",
  "description": "",
  "surplus_count": "",
  "task_type": ", 0, 1",
  "scope": {
    "category": ", 0:, 1:",
    "device_list": ["ID1"]
  },
  "tag_type": ", 0:, 1:",
  "audit_member_id": "",
  "create_time": "",
  "yyyy-MM-dd'T'HH:mm:ss.SS'Z'"
}
```

字段	是否必须	描述
id	是	自动升级任务ID
product_id	是	产品ID
name	是	自动升级任务名称, 32个字符以内
description	是	描述, 250个字符以内
from_version	是	起始版本号, 32位整形
from_version_url	是	起始版本文件地址
from_version_md5	是	起始版本文件MD5值
from_version_size	是	起始版本文件大小
target_version	是	目标版本文件号
target_version_url	是	目标版本文件地址
target_version_md5	是	目标版本文件MD5
target_version_size	是	目标版本文件大小

upgrade_count	是	已升级设备数量
status	是	任务状态: 0: 停止状态, 1: 启动状态, 2: 待执行, 3: 待审核 Api-Version为2时 有可能为 0、1、2、3 Api-Version为其他时, 只有 0、1
type	是	升级任务类型, 新增, 见附件 升级任务类型, 默认为wifi
identify	是	用来定位多MCU或多子设备的情况, 数字, 默认为0, type/identify/from_version联合唯一
description	是	描述
surplus_count	否	当前剩余版本号为起始版本号的设备总数, 即是要升级的设备数, type为wifi时才有
task_type	是	升级任务模式, 0自动模式, 1手动模式
scope	是	升级范围
scope.category	是	范围类别, 0: 全网, 1: 指定设备
scope.device_list	是	指定升级设备标识列表
tag_type	是	升级任务标签类型, 0: 正式任务, 1: 测试任务, 不传默认为正式任务
audit_member_id	否	审批人标识
create_time	是	升级任务创建时间, 格式为yyyy-MM-dd'T'HH:mm:ss.SS'Z'

15. 查询操作日志接口

- 根据标识获取升级任务的操作日志

Request

URL

```
GET
/v2/upgrade/firmware/tasks/{task_id}/op-logs
```

Header

```
Content-Type: "application/json"
Access-Token: " / "
```

Content

Response

Header

```
HTTP/1.1 200 OK
```

Content

```
{
  "count": "",
  "list": [{
    "id": "",
    "content": "",
    "opt_id": "",
    "opt_name": "",
    "opt_account": "",
    "create_date": ""
  }]
}
```

字段	是否必须	描述
count	是	操作日志数量
id	是	操作日志标识
content	是	操作日志内容
opt_id	是	操作人标识
opt_name	是	操作人名称
opt_account	是	操作人账号
create_date	是	日志产生时间

16. 查询自定义升级任务版本占比

- 查询自定义升级任务的设备版本占比
- 以选定的设备总数为基数，各个版本设备数为占比

Request

URL

```
GET
/v2/statistics/ota/customize-task/{task_id}/device-firmware-version
```

Header

```
Content-Type: "application/json"
Access-Token: "/"
```

Content



Response

Header

```
HTTP/1.1 200 OK
```

Content

```
{
  "total": "",
  "ratios": [{
    "devices": "",
    "firmware_version": ""
  }, {
    "devices": "",
    "firmware_version": ""
  }]
}
```

字段	是否必须	描述
total	是	该升级任务包含设备总数
ratios	是	占比
devices	是	该版本设备总数
firmware_version	是	版本

升级附录

1. 设备固件类型

枚举值	说明
1	WIFI
2	MCU
3	子设备

2. 设备升级任务类型

枚举值	说明
1	WIFI
2	MCU
3	子设备