OTA SDK解释文档

目录

[一、OTA升级流程 3](#_Toc3795963)

[1. 概述 3](#_Toc3795964)

[2. API 4](#_Toc3795965)

[2.1 鉴权参数 4](#_Toc3795966)

[2.2 检测任务 4](#_Toc3795967)

[2.3 校验TOKEN 6](#_Toc3795968)

[2.4 获取文件 7](#_Toc3795969)

[2.5 上报状态 8](#_Toc3795970)

[3. OTA SDK 9](#_Toc3795971)

[3.1 用户例程 9](#_Toc3795972)

[3.2 逻辑控制 10](#_Toc3795973)

[3.3 适配接口 11](#_Toc3795974)

[3.4 事件说明 12](#_Toc3795975)

[二、其他说明 15](#_Toc3795976)

# 一、OTA升级流程

## 1. 概述

OTA：公有协议适用的通用远程升级功能，包括模组升级和MCU应用升级两类，待升级设备通过HTTP协议RESTful标准格式与OneNET平台进行数据交互以完成升级。

南向设备目前所需的RESTful API一共有三类：检测任务、拉取文件、上报状态，其中拉取文件包括校验TOKEN和获取文件两个API。一次完整的升级流程如下图所示：

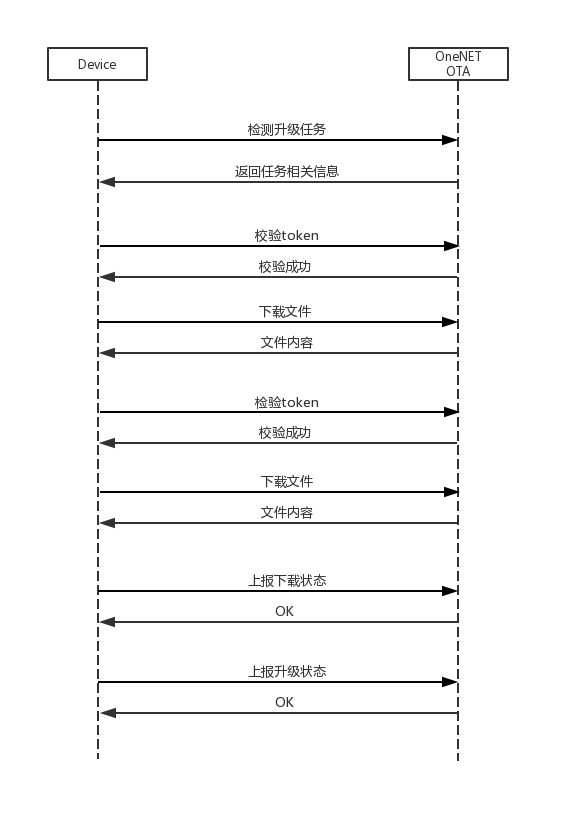


图1 升级流程

## 2. API

### 2.1 鉴权参数

在使用API进行OTA升级时，在HTTP头部需要填写Authorization，其value为：”version={version}&res=products/{product\_id}&et={et}&method={method}&sign={sign}”，既有5个参数构成，其中：

* {version}为当前鉴权版本号，固定取值为2018-10-31
* {product\_id}为用户产品ID
* {et}为本条API命令过期时间。注：Unix时间，例如2019/1/1 00:00:00为1546272000
* {method}为signature的计算方式，当前支持的签名计算方式有sha1、sha256和md5
* {sign}为对”{et}+\n+{method}+\n+products/{product\_id}+\n+{version}”这一字符串的签名结果

其中{sign}的生成步骤如下：

1. 在OneNET界面创建产品，得到access key和product\_id；
2. 按照et、method、product\_id、version的顺序拼接成字符串，例如et选择2025/1/1 00:00:00，method选择sha1加密，product\_id为12345，version为2018-10-31，则字符串为：”1735660800\nsha1\nproducts/12345\n2018-10-31”，其中”\n”为换行符既十六进制0x0A；
3. 对access key进行base64解码得到hmac\_key，用hmac\_key对拼接好的字符串进行hmac签名，签名方式为{method}，在本例中即为sha1，结果为”1WtPhCP34nOJVqg2crMXPfLplZk=“。

### 2.2 检测任务

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://ota.heclouds.com/ota/south/check |
| HTTP头部 | Content-Type:"application/json"  Authorization:"version=2018-10-31&res=products/{product\_id}&et={et}&method={method}&sign={sign}" |
| HTTP请求参数 | dev\_id=,//必填，设备id  manuf=,//必填，厂商编号  model=,//必填，模组编号  type=1|2,//必填，1：fota任务，2：sota任务  version=,//必填，当前版本  cdn=true|false//选填，是否返回拉取升级包ip |
| HTTP响应示例 | {  "errno": 0,  "error":"succ",  "data":{  "target":"1.2",// 升级任务的目标版本  "token":"abcdefg",  "size":123,//文件大小（字节）  "signal":12,//任务需在大于该信号强度的情况下进行  "power":90,//任务需在大于该电量的情况下进行  "retry":10,//重试次数  "interval":90,//重试间隔  "md5":"abcdefg",//升级文件的md5  "type":1|2 // 1:完整包，2：差分包  }  } |
| 备注 | * errno:   0:检测成功，设备存在升级任务  7:OneNET内部错误  8:检测失败，设备存在升级任务，但manuf或者model不一致  9:检测失败，设备存在升级任务，但version不一致  10:检查失败，设备存在升级任务，但signal或者power不一致  11:检查失败，设备不存在指定的type（fota/sota）所对应的升级任务  13:检查失败，设备存在升级任务，但是任务状态无效  19:检测失败，设备任务以暂停   * Token的有效期为2天 |

### 2.3 校验TOKEN

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://ota.heclouds.com/ota/south/download/{token}/check |
| HTTP头部 | Content-Type:"application/json"  Authorization:"version=2018-10-31&res=products/{product\_id}&et={et}&method={method}&sign={sign}" |
| HTTP请求参数 | dev\_id=，//必填，设备id |
| HTTP响应示例 | {  "errno": 0,  "error":"succ"  } |
| 备注 | * errno:   0:有效  7:OneNET内部错误  12:token过期，设备升级流程结束  15:不是最新获取的token  18:错误的token，设备升级流程应当结束  19:升级任务暂停，根据重试时间重新检查，不消耗重试次数  20:升级任务过期，设备升级流程结束  21:无效操作  22:升级任务取消，无需上报升级失败  23:token和dev\_id不一致   * URL中的{token}即为获取任务API中返回的token |

### 2.4 获取文件

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://ota.heclouds.com/ota/south/download/{token} |
| HTTP头部 | Range: bytes={start}-{end} |
| HTTP响应示例 | 二进制文件 |
| 备注 | * 如果header中没有Range字段，默认返回所有数据 * 响应报文中存在如下三个重要字段   Content-Disposition:该字段中包括filename，既文件名  Content-Range:当前请求获取的分片范围和总长度  Content-Length:当前请求获取的数据大小   * 头部Range字段解释：Range: bytes={start}-{end}，目前只支持如下几种模式   1、Range: bytes={start}-，获取第{start}+1个字节到最后的数据  例如：Range: bytes=0-，获取所有数据  Range: bytes=2-，获取第3个数据到最后1个数据  注意：如果{start}>=文件总长度，则默认{start}=0  2、Range: bytes={start}-{end}，获取第{start}+1个字节到第{end}+1个字节  例如：Range: bytes=0-99，获取前100个字节  注意：若{end}>=文件总长度len，则默认end=len-1  若{start}>{end}，start被设置为0  　　3、Range: bytes=-{end}，获取最后end个字节数据  　　　 例如：Range: bytes=-100，获取最后100个字节数据  　　　 注意：如果{end}>文件总长度len，则默认end=len（获取所有文件）   * 分片返回的状态码：206   不分片返回的状态码：200   * 每一次获取文件前都应该进行token的校验 * 如果响应的Header中没有返回Content-Length，说明拉取文件异常，这时响应内容只有一个字节数据（数字），该数字及代表的含义如下：   1：未找到资源，应上报下载失败（107）  2：文件大小不一致，应上报下载失败（107）  3：无效token，应上报下载失败（107） |

### 2.5 上报状态

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://ota.heclouds.com/ota/south/report |
| HTTP头部 | Authorization:"version=2018-10-31&res=products/{product\_id}&et={et}&method={method}&sign={sign}" |
| HTTP请求内容 | {  "result":101  } |
| HTTP响应示例 | {  "errno": 0,  "error":"succ"  } |
| 备注 | * result:   101:升级包下载成功  102:下载失败,空间不足  103:下载失败,内存溢出  104:下载失败,下载请求超时  105:下载失败,电量不足  106:下载失败,信号不良  107:下载失败,未知异常  201:升级成功  202:升级失败,电量不足  203:升级失败,内存溢出  204:升级失败,版本不一致  205:升级失败,md5校验失败  206:升级失败,未知异常   * error:   0:上报状态成功  1:上报状态失败，token鉴权失败  2:上报状态失败，状态码有误  20:上报状态失败，升级任务已经完成  21:无效的操作  22:上报状态失败，升级任务已被取消  24:请求内容result错误 |

## 3. OTA SDK

### 3.1 用户例程

OTA的用户例程为sample文件夹，在VS工程中对应为OtaSample项目。所涉及的函数有：

|  |  |  |
| --- | --- | --- |
| 函数名 | 作用说明 | 参数与备注 |
| ota\_init | ota上下文初始化函数 | check\_time：检查升级任务的时间间隔，单位：秒 |
| ota\_loop | ota逻辑控制函数 | check\_now：是否立即检查升级任务  备注：需要周期性调用 |
| ota\_deinit | ota上下文销毁函数 | 备注：在结束ota流程后应当调用该函数清除相关内存空间 |

### 3.2 逻辑控制

OTA的逻辑控制函数及相关功能文件为core文件夹，在VS工程中对应为CMIOT-OTA项目。core文件夹中的文件及二级文件夹有：

|  |  |
| --- | --- |
| 文件或文件夹名 | 说明 |
| ota\_core.c | 包含ota逻辑控制函数、功能函数 |
| ota\_parse.c | 包含解析报文控制函数 |
| ota\_api.h | ota外部使用头文件 |
| ota\_internal.h | ota内部使用头文件 |
| cjson/ | json解析组件 |
| http/ | http报文解析组件 |
| mbedtls/ | HMAC签名加密组件 |

在ota\_core.c文件中，包含了ota的逻辑控制函数及相关的解析功能函数，所涉及的函数有：

|  |  |  |
| --- | --- | --- |
| 函数名 | 作用说明 | 参数与备注 |
| OTA\_Pack\_Check\_Task | 查询任务API组包 | 无 |
| OTA\_Pack\_Check\_Token | Token检验API组包 | 无 |
| OTA\_Pack\_Get\_Package | 获取文件API组包 | 无 |
| OTA\_Pack\_Report | 上报状态API组包 | 无 |
| OTA\_Pack\_Authorization\_Info | Authorization参数计算及组包 | 无 |
| OTA\_CHECK\_DEVICE | 检查设备是否具备升级条件 | 备注：每次下载升级包前都应该检查电量及信号强度是否符合预设条件 |

### 3.3 适配接口

OTA的适配接口位于adapter文件夹，在VS工程中对应为OtaAdapter项目。接口共分为系统接口和网络接口两部分，其中系统接口涉及的函数有：

|  |  |  |
| --- | --- | --- |
| 函数名 | 作用说明 | 参数与备注 |
| OTA\_Event\_Handle | ota事件触发函数 | 无 |
| ota\_log\_printf | 日志打印 | 无 |
| ota\_atoi | 字符串转整形 | 无 |
| ota\_itoa\_decimal | 十进制整数转字符串 | 无 |
| ota\_hex2str | 十六进制转字符串 | 备注：将md5计算结果转成字符串用于比较 |
| ota\_malloc | 内存申请 | 无 |
| ota\_free | 内存释放 | 无 |
| ota\_get\_tick | 获取系统运行时间 | 备注：用于计时 |
| ota\_get\_deviceID | 获取设备ID | 无 |
| ota\_get\_manuf | 获取厂商编号 | 无 |
| ota\_get\_model | 获取模组编号 | 无 |
| ota\_get\_type | 获取升级type | 备注：1为fota任务，2为sota任务 |
| ota\_get\_version | 获取当前版本 | 无 |
| ota\_get\_signals | 获取信号强度 | 无 |
| ota\_get\_powers | 获取电池电量 | 无 |
| ota\_get\_auth\_version | 获取鉴权版本 | 备注：当前鉴权版本固定为” 2018-10-31” |
| ota\_get\_auth\_res | 获取res | 备注：在当前SDK中，既为获取产品ID |
| ota\_get\_auth\_et | 获取鉴权过期时间 | 备注：UNIX时间 |
| ota\_get\_auth\_method | 获取签名计算方式 | 备注：当前支持sha1/sha256/md5（全为小写） |
| ota\_get\_auth\_accesskey | 获取access key | 无 |
| ota\_read\_flash | 从flash中恢复ota状态 | 无 |
| ota\_save\_flash | 将ota状态保存至 flash | 无 |
| ota\_calculate\_range | 计算下次获取升级包的范围（range） | 无 |

其中网络接口涉及的函数有：

|  |  |  |
| --- | --- | --- |
| 函数名 | 作用说明 | 参数与备注 |
| ota\_net\_init | 网络连接初始化 | 无 |
| ota\_net\_send | 发送函数 | 无 |
| ota\_get\_host\_by\_name | 域名解析 | 备注：若不支持域名解析可以直接返回固定IP |

### 3.4 事件说明

SDK的逻辑控制和用户数据交互是通过事件触发实现的，包含在函数OTA\_Event\_Handle()中。其中所涉及的事件有：

|  |  |  |
| --- | --- | --- |
| 事件名 | 作用说明 | 参数与备注 |
| OTA\_EVENT\_TASK\_ManufOrModel\_ERROR | 检测任务失败，厂商编号或模组编号不一致 | 无 |
| OTA\_EVENT\_TASK\_Ver\_ERROR | 检测任务失败，差分包的初始版本号不一致 | 无 |
| OTA\_EVENT\_TASK\_STATE\_ERROR | 检测任务失败，任务状态失效 | 无 |
| OTA\_EVENT\_TASK\_NO\_TASK | 检测任务失败，没有对应的任务存在 | 无 |
| OTA\_EVENT\_TASK\_OTHER\_ERROR | 检测任务失败，由于其他原因失败 | 备注：其他原因包括OneNET内部错误、任务暂停等原因 |
| OTA\_EVENT\_TASK\_SUCCESS | 检测任务成功 | 备注：检测任务成功后应当将上下文的state参数更新为下载状态 |
| OTA\_EVENT\_TOKEN\_OK | 校验token成功 | 备注：token校验完毕后可以进行文件的获取 |
| OTA\_EVENT\_TOKEN\_INTERNAL\_ERROR | 校验token失败，OneNET内部错误 | 无 |
| OTA\_EVENT\_TOKEN\_EXPIRED\_ERROR | 校验token失败，token过期 | 备注：应当结束OTA本次流程，并销毁上下文 |
| OTA\_EVENT\_TOKEN\_RETRIEVE\_ERROR | 校验token失败，token过期 | 备注：可以重新检测一次任务，更新到最新的token |
| OTA\_EVENT\_TOKEN\_CHECK\_ERROR | 校验token失败，token错误 | 备注：应当结束OTA本次流程，并销毁上下文 |
| OTA\_EVENT\_TOKEN\_PAUSED\_ERROR | 校验token失败，任务已暂停 | 备注：根据重试时间进行重试，但是不消耗重试次数 |
| OTA\_EVENT\_TOKEN\_TASK\_EXPIRED\_ERROR | 校验token失败，任务已结束 | 备注：应当结束OTA本次流程，并销毁上下文 |
| OTA\_EVENT\_TOKEN\_TERMINATE\_ERROR | 校验token失败，任务已取消 | 备注：应当结束OTA本次流程，并销毁上下文 |
| OTA\_EVENT\_TOKEN\_OTHER\_ERROR | 校验token失败，由于其他原因失败 | 无 |
| OTA\_EVENT\_REPORT\_DOWNLOAD\_SUCCESS | 开始上报下载成功状态 | 无 |
| OTA\_EVENT\_REPORT\_LOW\_POWER | 开始上报下载失败状态，原因为电量低 | 无 |
| OTA\_EVENT\_REPORT\_BAD\_SIGNAL | 开始上报下载失败状态，原因为信号不良 | 无 |
| OTA\_EVENT\_REPORT\_UNKNOWN\_ERROR | 开始上报下载失败状态，其他原因 | 备注：其他原因包括重试次数消耗完毕、服务器未找到资源、文件大小不一致、token无效等原因 |
| OTA\_EVENT\_REPORT\_UPDATE\_SUCCESS | 开始上报升级成功状态 | 无 |
| OTA\_EVENT\_REPORT\_MD5\_ERROR | 开始上报升级失败状态，原因为md5校验失败 | 无 |
| OTA\_EVENT\_REPORT\_SUCCESS | 上报状态成功 | 备注：上报状态成功后，若是上报的下载成功，则可以开始进行升级，若为其他上报状态，则应当结束OTA本次流程，并销毁上下文 |
| OTA\_EVENT\_REPORT\_FAILED | 上报状态失败，由其他原因造成 | 备注：其他原因包括token失效、上传错误的状态码、升级任务已取消、升级任务已完成、无效操作等原因，其中若升级任务已取消或者已完成，建议结束OTA本次流程，并销毁上下文 |
| OTA\_EVENT\_custom\_retry\_download | 开始下载重试 | 备注：由于收到的字节数与content length不相等，故开始重试计时 |
| OTA\_EVENT\_custom\_save\_packet | 开始保存下载的文件 | 备注：将接收到的文件保存成文件 |
| OTA\_EVENT\_custom\_ready\_update | 开始升级 | 无 |
| OTA\_EVENT\_custom\_save\_task\_info | 保存任务信息至flash | 备注：将接收到的文件保存成文件 |
| OTA\_EVENT\_custom\_report\_error\_clear | 由于上报失败结束OTA流程 | 备注：当由于升级任务已取消、升级任务已完成导致上报状态失败时，建议结束OTA本次流程，并销毁上下文 |
| OTA\_EVENT\_custom\_delete\_package | 销毁上下文时清空已经保存的升级包数据 | 无 |

# 二、其他说明

1. 在win平台运行该工程时，只需要将default\_device\_id、default\_manuf、default\_model、default\_type、default\_version、default\_res和default\_access\_key修改即可运行。
2. 在事件OTA\_EVENT\_custom\_ready\_update中，应当增加其他方式判断升级成功与否（例如通过比对升级前后的version等），再上报升级状态；
3. 在函数OTA\_Pack\_Authorization\_Info()中会将签名加密的计算结果保存至上下文中的encrypt\_result变量中。在OTA\_EVENT\_custom\_save\_task\_info事件中可以将其保存至flash中或者下次恢复状态时重新计算；
4. 若超过一定时间未收到数据应当标记为重试（ctx\_net\_state重置为NET\_STATE\_NULL , isRetry置为DownLoadRetry\_True）；
5. 检测任务API中，如果设置了cdn为true可以返回升级包的下载IP；
6. 在函数ota\_save\_flash中，若需要将某些数据存入flash用作恢复时，应当保存检测任务成功时返回的token、信号要求、电量要求、重试次数、重试时间间隔、升级包大小、已下载升级包大小、升级包md5信息等，且当每次获取分片包成功并据存储完毕后，应当将已下载升级包大小更新到flash中；
7. 每次拉取文件的分片范围在函数ota\_calculate\_range()中计算，可以根据实际需要自由设计计算方式；
8. 下载升级包时若因为电量低或者信号差，可以选择在回调事件中将isRetry置为重试状态；
9. 如果需要升级包的文件名，可以在Content-Disposition中获取；
10. 每次获取分片包的最小范围限制为2个字节；
11. 上报状态时，若OneNET收到了状态码但是因某些原因（例如网络原因）导致sdk没收到回应，可以在OTA\_EVENT\_REPORT\_FAILED事件中根据实际需求选择直接销毁上下文。