

API接口文档

IOT 对外接口文档

Changelog

1. 20180503 create by jinwentao
2. 20180515 新增固件相关接口
3. 20180614 固件接口支持分包

接口调用说明:

本文中所有调用云信服务端接口的请求都需要进行此规则校验。

1. api checksum校验

本文档所有API请求时，需要带上四个Http Request Header

参数	参数说明
ProductKey	分配的ProductKey
Nonce	随机数（随机数，最大长度128个字符）
CurTime	当前UTC时间戳，从1970年1月1日0点0分0秒开始到现在的秒数(String) 必须是五分钟以内的时间。
Checksum	SHA1(ProductSecret+ Nonce + CurTime), 三个参数拼接的字符串，进行SHA1哈希计算，转化成16进制字符(String, 小写)
DupNonceEnable	如果此header值为true，表示用户此方法需要支持去重功能。非必需参数，服务器只有更新接口支持此功能。

Checksum有效期：出于安全性考虑，每个checksum的有效期为5分钟，建议每次请求都生成新的“checksum”，同时请确认发起请求的服务器是与标准时间同步的，比如有NTP服务。

Checksum值校验失败，会返回414. 其它状态码参考状态码表

Checksumjava

|

```

import java.security.MessageDigest;

public class CheckSumBuilder {
    // CheckSum
    public static String getCheckSum(String productSecret, String nonce, String curTime) {
        return encode("sha1", productSecret + nonce + curTime);
    }

    // md5
    public static String getMD5(String requestBody) {
        return encode("md5", requestBody);
    }

    private static String encode(String algorithm, String value) {
        if (value == null) {
            return null;
        }
        try {
            MessageDigest messageDigest
                = MessageDigest.getInstance(algorithm);
            messageDigest.update(value.getBytes());
            return getFormattedText(messageDigest.digest());
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    private static String getFormattedText(byte[] bytes) {
        int len = bytes.length;
        StringBuilder buf = new StringBuilder(len * 2);
        for (int j = 0; j < len; j++) {
            buf.append(HEX_DIGITS[(bytes[j] >> 4) & 0x0f]);
            buf.append(HEX_DIGITS[bytes[j] & 0x0f]);
        }
        return buf.toString();
    }

    private static final char[] HEX_DIGITS = { '0', '1', '2', '3', '4', '5',
        '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };
}

```

2. 接口说明

1. 所有接口请求Content-Type类型为: application/x-www-form-urlencoded;charset=utf-8
2. 所有接口返回类型为JSON, 同时进行UTF-8编码。

错误码:

code	描述
200	成功
500	服务端异常
414	传参错误
403	禁止调用
1001	产品不存在
1002	产品不支持动态注册
1003	产品已存在, 不能重复注册
1101	设备不存在
1102	设备名称重复
1201	rrpc调用超时
1301	设备影子不存在
1302	更新设备影子传参有误
1401	分页查询单页数据量过大

一. 设备相关接口

1 RegisterDevice

请求说明	<pre>POST https://api-iot-dev.netease.im/RegisterDevice HTTP/1.1 Content-Type: application/json;charset=UTF-8</pre>
接口描述	注册设备

参数	参数类型	是否必须	说明
DeviceName	String	否	设备名称, 不填的话会按默认规则生成, 最长128个字符

curl请求示例:

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/json;charset=UTF-8" -d 'DeviceName=123456' 'https://api-iot-dev.netease.im/RegisterDevice'
```

返回说明	http 响应: json
	<pre>"Content-Type": "application/json; charset=utf-8" { "data": { "DeviceName": "123456", "ProductKey": "fe416640c8e8a72734219e1847ad2547", "DeviceSecret": "22fd30ec291b4dc6b20180fbb1531090", "IotId": "094c818191424d2ba46d911c7549a55c" }, "requestId": "89445550571048e88fabc957ae78f9fb", "code": 200, "msg": "OK" }</pre>

返回data参数定义:

名称	类型	描述
ProductKey	String	产品PK
DeviceName	String	设备名称
DeviceSecret	String	设备秘钥
IotId	String	设备唯一id

2 QueryDeviceDetail

请求说明	<pre>GET https://api-iot-dev.netease.im/QueryDeviceDetail HTTP/1.1 Content-Type: application/x-www-form-urlencoded;charset=utf-8</pre>
接口描述	查询设备详情

参数	参数类型	是否必须	说明
DeviceName	String	否	设备名称
IotId	String	否	设备的唯一标识

注意:

- ProductKey与DeviceName组合作为设备唯一标识符，与IotId是一一对应的关系。因此入参输入，DeviceName、IotId两者选其一即可，如果两者都传，以IotId为为准。

curl请求示例:

```
curl -X GET -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "CheckSum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/x-www-form-urlencoded" 'https://api-iot-dev.netease.im/QueryDeviceDetail?IotId=094c818191424d2ba46d911c7549a55c'
```

返回说明 http 响应: json

```
"Content-Type": "application/json; charset=utf-8"
{
  "data": {
    "Status": "UNACTIVE",
    "GmtActive": null,
    "DeviceName": "123456",
    "Region": null,
    "ProductKey": "fe416640c8e8a72734219e1847ad2547",
    "FirmwareVersion": "",
    "NodeType": 0,
    "DeviceSecret": "22fd30ec291b4dc6b20180fbb1531090",
    "GmtOnline": null,
    "IpAddress": null,
    "ProductName": "test12345",
    "GmtCreate": "2018-05-03 15:55:04",
    "IotId": "094c818191424d2ba46d911c7549a55c"
  },
  "requestId": "e9bf9a3cfe0a428685f74d1132771463",
  "code": 200,
  "msg": "OK"
}
```

返回data参数定义:

名称	类型	描述
ProductKey	String	ProductKey, 全局唯一PK
ProductName	String	产品名称
DeviceName	String	设备名称
DeviceSecret	String	设备密钥
IotId	String	设备全局唯一Id, 与ProductKey&deviceName同时作为设备的唯一标识符
GmtCreate	String	创建时间
GmtActive	String	激活时间
GmtOnline	String	在线时间
Status	String	设备状态
FirmwareVersion	String	固件版本号
IpAddress	String	设备IP地址

NodeType	Integer	节点类型, 0 — 设备, 1 — 网关
Region	String	所在地区

Status设备状态字段说明

序号	status	说明
1	ONLINE	设备在线
2	OFFLINE	设备离线
3	UNACTIVE	设备未激活
4	DISABLE	设备已禁用

3. QueryDevice

请求说明	<pre>GET https://api-iot-dev.netease.im/QueryDevice HTTP/1.1 Content-Type: application/x-www-form-urlencoded;charset=utf-8</pre>
接口描述	分页查询设备列表

参数	参数类型	是否必须	说明
PageSize	Integer	是	设备列表显示页面大小
CurrentPage	Integer	是	设备列表显示页面当前页数

curl请求示例:

```
curl -X GET -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/x-www-form-urlencoded" 'https://api-iot-dev.netease.im/QueryDevice?PageSize=10&CurrentPage=1'
```

返回说明 http 响应: json

```
"Content-Type": "application/json; charset=utf-8"
{
  "data": {
    "list": [
      {
        "deviceId": "094c818191424d2ba46d911c7549a55c",
        "gmtModified": "2018-05-03 15:55:04",
        "deviceName": "123456",
        "deviceSecret": "22fd30ec291b4dc6b20180fbb1531090",
        "gmtCreate": "2018-05-03 15:55:04"
      },
      {
        "deviceId": "960969323b634ae7b8db98455e59523f",
        "gmtModified": "2018-05-03 16:38:45",
        "deviceName": "6ec6153142ab4c87b867252d8912ed48",
        "deviceSecret": "f657988bf8c04c7298612c6687c3b637",
        "gmtCreate": "2018-05-03 16:38:45"
      },
      {
        "deviceId": "78ea3a6a51e848aeab9f813f23a2e737",
        "gmtModified": "2018-05-03 16:38:46",
        "deviceName": "c79a7d3e3c9d437a89543f35767e94de",
        "deviceSecret": "01679621af5e4010bbde34184a7f24f4",
        "gmtCreate": "2018-05-03 16:38:46"
      }
    ],
    "Total": 3
  },
  "requestId": "23be7f1dc6704122abf007abea8b6f09",
  "code": 200,
  "msg": "OK"
}
```

返回data参数定义:

名称	类型	描述
DeviceId	String	设备ID
DeviceName	String	设备名称
DeviceSecret	String	设备密钥
GmtCreate	String	创建时间
GmtModified	String	修改时间

4 BatchQueryDeviceDetail

请求说明	POST https://api-iot-dev.netease.im/BatchQueryDeviceDetail HTTP/1.1 Content-Type: application/json;charset=UTF-8
接口描述	批量查询设备详情(最多查询200个设备)

参数	参数类型	是否必须	说明
IotIds	String	是	设备id列表, example:["id1","id2"]

curl请求示例:

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/json;charset=UTF-8" -d 'IotIds=["id1"]' 'https://api-iot-dev.netease.im/BatchQueryDeviceDetail'
```

返回说明 http 响应: json

```
"Content-Type": "application/json; charset=utf-8"
{
  "data": [
    {
      "Status": "UNACTIVE",
      "ProductName": "test111",
      "DeviceSecret": "71bba7a5dd52de63d3340f7dfd8b9608",
      "FirmwareVersion": "",
      "IotId": "bed3d1df79c04badb981e9b3794438a6",
      "GmtCreate": "2018-09-12 17:04:26",
      "GmtActive": null,
      "NodeType": 0,
      "Region": null,
      "IpAddress": null,
      "GmtOnline": null,
      "ProductKey": "c400b5654a73a5a1976c04910d79eebf",
      "DeviceName": "01d9d47e3bdf4411bce9806c90901206"
    },
    {
      "Status": "UNACTIVE",
      "ProductName": "test111",
      "DeviceSecret": "71bba7a5dd52de63d3340f7dfd8b9608",
      "FirmwareVersion": "",
      "IotId": "bed3d1df79c04badb981e9b3794438a6",
      "GmtCreate": "2018-09-12 17:04:26",
      "GmtActive": null,
      "NodeType": 0,
      "Region": null,
      "IpAddress": null,
      "GmtOnline": null,
      "ProductKey": "c400b5654a73a5a1976c04910d79eebf",
      "DeviceName": "01d9d47e3bdf4411bce9806c90901206"
    }
  ],
  "requestId": "f1ab51d59eb5495a9367c9377b806960",
  "code": 200,
  "msg": "OK"
}
```

返回data参数定义:

名称	类型	描述
ProductKey	String	ProductKey, 全局唯一PK
ProductName	String	产品名称
DeviceName	String	设备名称
DeviceSecret	String	设备密钥
IotId	String	设备全局唯一Id, 与ProductKey&deviceName同时作为设备的唯一标识符
GmtCreate	String	创建时间
GmtActive	String	激活时间
GmtOnline	String	在线时间
Status	String	设备状态
FirmwareVersion	String	固件版本号

IpAddress	String	设备IP地址
NodeType	Integer	节点类型, 0 — 设备, 1 — 网关
Region	String	所在地区

Status设备状态字段说明

序号	status	说明
1	ONLINE	设备在线
2	OFFLINE	设备离线
3	UNACTIVE	设备未激活
4	DISABLE	设备已禁用

5 BatchQueryDeviceStatus

请求说明	<pre>POST https://api-iot-dev.netease.im/BatchQueryDeviceStatus HTTP/1.1 Content-Type: application/json;charset=UTF-8</pre>
接口描述	批量查询设备状态(最多查询200个设备)

参数	参数类型	是否必须	说明
DeviceNames	String	是	设备名称列表, example:["name1", "name2"]

curl请求示例:

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/json;charset=UTF-8" -d 'DeviceNames=["name1"]' 'https://api-iot-dev.netease.im/BatchQueryDeviceStatus'
```

返回说明	<p>http 响应: json</p> <pre>"Content-Type": "application/json; charset=utf-8" { "data": [{ "Status": "UNACTIVE", "DeviceName": "01d9d47e3bdf4411bce9806c90901206" }, { "Status": "UNACTIVE", "DeviceName": "01d9d47e3bdf4411bce9806c90901206" }], "requestId": "f1ab51d59eb5495a9367c9377b806960", "code": 200, "msg": "OK" }</pre>
------	---

Status设备状态字段说明

序号	status	说明
1	ONLINE	设备在线
2	OFFLINE	设备离线
3	UNACTIVE	设备未激活
4	DISABLE	设备已禁用

二. 通信相关

1. RRPC

请求说明	<pre>POST https://api-iot-dev.netease.im/RRpc HTTP/1.1 Content-Type: application/x-www-form-urlencoded;charset=utf-8</pre>
接口描述	发消息给设备并同步返回响应
调用说明	<ol style="list-style-type: none"> 1. 设备端需订阅/rrpc/{ProductKey}/{DeviceName}/get/+ 2. 设备端在此topic收到消息后，需要回复到/rrpc/{ProductKey}/{DeviceName}/update/{messageId} 3. messageId为/rrpc/{ProductKey}/{DeviceName}/get/+ 中+的内容

参数	参数类型	是否必须	说明
DeviceName	String	是	设备名称
RequestBase64Byte	String	是	请求消息内容，需要是经过Base64编码得到的字符串数据
Timeout	Integer	是	表示等待设备回复消息的时间，单位是毫秒，允许范围是1000~5000

curl请求示例:

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/x-www-form-urlencoded" -d 'DeviceName=123456&RequestBase64Byte=aGVsbG8gd29ybGQ%3D&Timeout=1000' 'https://api-iot-dev.netease.im/RRpc'
```

返回说明	<p>http 响应: json</p> <pre>"Content-Type": "application/json; charset=utf-8" { "data": "hello world", "requestId": "06bef4229e914eec8752ab98e4210e99", "code": 200, "msg": "OK" }</pre>
------	--

返回data参数定义: 待约定

2. Pub

请求说明	<pre>POST https://api-iot-dev.netease.im/Pub HTTP/1.1 Content-Type: application/json;charset=UTF-8</pre>
接口描述	发布消息到Topic
调用说明	<ol style="list-style-type: none"> 1. 设备端需订阅/{ProductKey}/{DeviceName}/get 2. 此topic即为参数TopicFullName

参数	参数类型	是否必须	说明
TopicFullName	String	是	消息发送目的Topic名字
MessageContent	String	是	发送的消息，将消息内容二进制进行Base64编码后得到的字符串
Qos	Integer	否	指定消息发送的方式，默认为0 0：最多发送一次，1：最少发送一次

curl请求示例：

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "CheckSum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/json;charset=UTF-8" -d 'TopicFullName=123456&MessageContent=aGVsbG8gd29ybGQ%3D&Qos=0' 'https://api-iot-dev.netease.im/Pub'
```

返回说明	<p>http 响应: json</p> <pre>"Content-Type": "application/json; charset=utf-8" { "requestId": "06bef4229e914eec8752ab98e4210e99", "code": 200, "msg": "OK" }</pre>
------	---

3. PubBroadcast

请求说明	<pre>POST https://api-iot-dev.netease.im/PubBroadcast HTTP/1.1 Content-Type: application/json;charset=UTF-8</pre>
接口描述	发布消息到Topic
调用说明	<ol style="list-style-type: none"> 1. 设备端需订阅/broadcast/\${ProductKey}/XXXX/get，形如/broadcast/\${ProductKey}/1/get 2. 此topic即为参数TopicFullName

参数	参数类型	是否必须	说明
MessageContent	String	是	发送的消息，将消息内容二进制进行BASE64转码后得到的字符串
TopicFullName	String	是	消息发送目的Topic名字，格式为 /broadcast/\${ProductKey}/1/get，Topic第三段数字可以自定义

curl请求示例：

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/json;charset=UTF-8" -d 'TopicFullName=123456&MessageContent=aGVsbG8gd29ybGQ%3D' 'https://api-iot-dev.netease.im/PubBroadcast'
```

返回说明 http 响应: json

```
"Content-Type": "application/json; charset=utf-8"
{
  "requestId": "06bef4229e914eec8752ab98e4210e99",
  "code": 200,
  "msg": "OK"
}
```

三. 设备影子

1. GetDeviceShadow

请求说明

```
GET https://api-iot-dev.netease.im/GetDeviceShadow HTTP/1.1
Content-Type: application/x-www-form-urlencoded;charset=utf-8
```

接口描述 查看设备影子信息

参数	参数类型	是否必须	说明
ProductKey	String	是	分配的产品key
DeviceName	String	是	设备名称

curl请求示例:

```
curl -X GET -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/x-www-form-urlencoded" 'https://api-iot-dev.netease.im/GetDeviceShadow?DeviceName=094c818191424d2ba46d911c7549a55c'
```

返回说明

http 响应: json

```
"Content-Type": "application/json; charset=utf-8"
{
  "data": {
    "iotId": "6c3d6f369dbc4173alf6e650098a057f",
    "timestamp": 1524742425248,
    "version": 2,
    "state": {
      "reported": {
        "color": "red"
      },
      "desired": {
        "color": "pink"
      }
    },
    "metadata": {
      "reported": {
        "color": 123456789
      },
      "desired": {
        "color": 1524742425248
      }
    }
  },
  "requestId": "856101678f6648469190c7c06f06b60d",
  "code": 200,
  "msg": "OK"
}
```

返回data参数定义:

```
{
  "iotId": "6c3d6f369dbc4173alf6e650098a057f",
  "timestamp": 1524742425248,
  "version": 2,
  "state": {
    "reported": {
      "color": "red" //自定义参数
    },
    "desired": {
      "color": "pink" //自定义参数
    }
  },
  "metadata": {
    "reported": {
      "color": 123456789
    },
    "desired": {
      "color": 1524742425248
    }
  }
}
```

2. UpdateDeviceShadow

请求说明

```
POST https://api-iot-dev.netease.im/UpdateDeviceShadow HTTP/1.1
Content-Type: application/json; charset=UTF-8
```

接口描述

查看设备影子信息

参数	参数类型	是否必须	说明
DeviceName	String	是	设备名称
ShadowMessage	String	是	设备影子信息

curl请求示例:

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/json;charset=UTF-8" -d 'DeviceName=094c818191424d2ba46d911c7549a55c&ShadowMessage=' 'https://api-iot-dev.netease.im/UpdateDeviceShadow'
```

返回说明	http 响应: json
	<pre>"Content-Type": "application/json; charset=utf-8" { "requestId": "06bef4229e914eec8752ab98e4210e99", "code": 200, "msg": null }</pre>

ShadowMessage格式如下:

```
{
  "method": "update",
  "state": {
    "reported": "",
    "desired": {
      "color": "green"
    }
  },
  "version": 3
}
```

四. 固件升级

1. ota/firmware/upload

请求说明	POST https://api-iot-dev.netease.im/ota/firmware/upload HTTP/1.1 Content-Type: multipart/form-data
接口描述	上传固件

参数	参数类型	是否必须	说明
File	Multipart	是	文件输入流
FirmwareName	String	是	固件名称
FirmwareType	Integer	是	固件类型(用于区分多个固件, 只有一个的话传1)
FirmwareVersion	String	是	固件版本
SeparatePacket	Integer	否	是否分包, 0: 不分包, 1: 分包, 默认不分包

SeparateSize	Integer	否	分包大小
Description	String	否	固件描述
Extra	String	否	扩展信息

curl请求示例:

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: multipart/form-data" -d 'File=gwettwgsqsgsgs323f' 'https://api-iot-dev.netease.im/ota/firmware/upload'
```

返回说明	<p>http 响应: json</p> <pre>"Content-Type": "application/json; charset=utf-8" { "data": null "requestId": "a7a07794e46b4fe0a7575793f2bf49cb", "code": 200, "msg": "OK" }</pre>
------	--

2. ota/firmware/list

请求说明	<p>GET https://api-iot-dev.netease.im/ota/firmware/list HTTP/1.1 Content-Type: application/x-www-form-urlencoded; charset=utf-8</p>
接口描述	分页查询固件列表

参数	参数类型	是否必须	说明
FirmwareType	Integer	是	固件类型(用于区分多个固件, 只有一个的话传1)
PageSize	Integer	是	设备列表显示页面大小
CurrentPage	Integer	是	设备列表显示页面当前页数

curl请求示例:

```
curl -X GET -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/x-www-form-urlencoded" 'https://api-iot-dev.netease.im/ota/firmware/list?FirmwareType=1&PageSize=10&CurrentPage=1'
```

返回说明

http 响应: json

```
"Content-Type": "application/json; charset=utf-8"
{
  "data": {
    "total":5,
    "list": [
      {
        "firmwareName":"test1267",
        "firmwareVersion":"1.0",
        "firmwareType":1,
        "md5":"c4ca4238a0b923820dcc509a6f75849b",
        "description":null,
        "extra":null,
        "createTime":"2018-05-16 19:57:17"
      },
      {
        "firmwareName":"test126",
        "firmwareVersion":"1.0",
        "firmwareType":1,
        "md5":"c4ca4238a0b923820dcc509a6f75849b",
        "description":null,
        "extra":null,
        "createTime":"2018-05-16 19:56:34"
      },
      {
        "firmwareName":"test125",
        "firmwareVersion":"1.0",
        "firmwareType":1,
        "md5":"c4ca4238a0b923820dcc509a6f75849b",
        "description":null,
        "extra":null,
        "createTime":"2018-05-16 19:56:12"
      },
      {
        "firmwareName":"test124",
        "firmwareVersion":"1.0",
        "firmwareType":1,
        "md5":"c4ca4238a0b923820dcc509a6f75849b",
        "description":null,
        "extra":null,
        "createTime":"2018-05-16 19:55:30"
      },
      {
        "firmwareName":"test",
        "firmwareVersion":"1.0",
        "firmwareType":1,
        "md5":"c4ca4238a0b923820dcc509a6f75849b",
        "description":null,
        "extra":null,
        "createTime":"2018-05-15 19:50:56"
      }
    ]
  },
  "requestId":"7994e966eda74d59a1622ba9c3bb8d54",
  "code":200,
  "msg":"OK"
}
```

返回data参数定义:

名称	类型	描述
firmwareName	String	固件名称
firmwareType	Integer	固件类型
firmwareVersion	String	固件版本
md5	String	md5摘要
description	String	描述

extra	String	扩展字段
createTime	String	创建时间

3. ota/firmware/delete

请求说明	POST https://api-iot-dev.netease.im/ota/firmware/delete HTTP/1.1 Content-Type: application/json;charset=UTF-8
接口描述	删除固件

参数	参数类型	是否必须	说明
FirmwareType	Integer	是	固件类型(只有一个固件类型的时候传1)
FirmwareVersion	String	是	固件版本

curl请求示例:

```
curl -X POST -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/json;charset=UTF-8" -d 'FirmwareType=1&FirmwareVersion=1.0' 'https://api-iot-dev.netease.im/ota/firmware/delete'
```

返回说明	http 响应: json <pre>"Content-Type": "application/json; charset=utf-8" { "data": null, "requestId": "63c3ee3ce55d4e51a9199c29c83df35d", "code": 200, "msg": "OK" }</pre>
------	---

4. ota/firmware/get

请求说明	GET https://api-iot-dev.netease.im/ota/firmware/get HTTP/1.1 Content-Type: application/x-www-form-urlencoded;charset=utf-8
接口描述	查询固件

参数	参数类型	是否必须	说明
----	------	------	----

FirmwareType	Integer	是	固件类型(只有一个固件类型的时候传1)
FirmwareVersion	String	是	固件版本号

curl请求示例:

```
curl -X GET -H "ProductKey: fe416640c8e8a72734219e1847ad2547" -H "Nonce: 4tgggergigwow323t23t" -H "CurTime: 1443592222" -H "Checksum: 9e9db3b6c9abb2e1962cf3e6f7316fcc55583f86" -H "Content-Type: application/x-www-form-urlencoded" 'https://api-iot-dev.netease.im/ota/firmware/get?FirmwareType=1&FirmwareVersion=1.0'
```

返回说明 http 响应: json

```
"Content-Type": "application/json; charset=utf-8"
{
  "data": {
    "firmwareType":2,
    "firmwareName":"test",
    "firmwareVersion":"3",
    "separateSize":10240,
    "separateCount":5,
    "downloadUrlPrefix":"https://nim.nosdn.127.net/1234567/2/3/1529033875/",
    "md5":"7e2f156cac35fd789becb26b83f7a7c3",
    "size":45172,
    "description":null,
    "extra":null,
    "createTime":"2018-06-15 11:37:56",
    "packets": [
      {
        "firmwareIndex":1,
        "downloadUrl":"https://nim.nosdn.127.net/1234567/2/3/1529033875/1",
        "packetSize":10240,
        "packetMd5":"72def90e41c73778e49b296eada95b0"
      },
      {
        "firmwareIndex":2,
        "downloadUrl":"https://nim.nosdn.127.net/1234567/2/3/1529033875/2",
        "packetSize":10240,
        "packetMd5":"46549be4fa778d28198ec32eb85363a5"
      },
      {
        "firmwareIndex":3,
        "downloadUrl":"https://nim.nosdn.127.net/1234567/2/3/1529033875/3",
        "packetSize":10240,
        "packetMd5":"25fb7df0a9e6d7a8df9bd1a5a559f2ed"
      },
      {
        "firmwareIndex":4,
        "downloadUrl":"https://nim.nosdn.127.net/1234567/2/3/1529033875/4",
        "packetSize":10240,
        "packetMd5":"c4267ad742ca6a23d77133a24bef7b63"
      },
      {
        "firmwareIndex":5,
        "downloadUrl":"https://nim.nosdn.127.net/1234567/2/3/1529033875/5",
        "packetSize":4212,
        "packetMd5":"cd25171a1e8ac7cb320a886ff70f2c4a"
      }
    ],
    "separatePacket":true
  },
  "requestId":"48a3a681c5804308b384a4ce54e1798c",
  "code":200,
  "msg":"OK"
}
```

返回data参数定义:

名称	类型	描述
firmwareName	String	固件名称
firmwareType	Integer	固件类型
firmwareVersion	String	固件版本
size	Integer	固件大小
separatePacket	Boolean	是否分包(false:不分包, true:分包)
separateSize	Integer	分包大小

separateCount	Integer	分包个数
downloadUrlPrefix	String	固件下载地址前缀
md5	String	md5摘要
description	String	描述
extra	String	扩展字段
createTime	String	创建时间

packets

名称	类型	描述
firmwareIndex	Integer	包编号
downloadUrl	String	固件下载地址
firmwareSize	Integer	包大小
packetMd5	String	包md5摘要