

# MN316

# HTTP 用户手册

NB-IoT 系列

版本：V1.0.0

日期：2020 年 9 月

# 服务与支持

如果您有任何关于模组产品及产品手册的评论、疑问、想法，或者任何无法从本手册中找到答案的疑问，请通过以下方式联系我们。



## 中移物联网有限公司

OneMO 官网: [onemo10086.com](http://onemo10086.com)

邮箱: [SmartModule@cmiot.chinamobile.com](mailto:SmartModule@cmiot.chinamobile.com)

客户服务热线: 400-110-0866

微信公众号: CMOneMO



中国移动  
China Mobile

# 文档声明

## 注意

本手册描述的产品及其附件特性和功能，取决于当地网络设计或网络性能，同时也取决于用户预先安装的各种软件。由于当地网络运营商、ISP，或当地网络设置等原因，可能也会造成本手册中描述的全部或部分产品及其附件特性和功能未包含在您的购买或使用范围之内。

## 责任限制


除非合同另有约定，中移物联网有限公司对本文档内容不做任何明示或暗示的声明或保证，并且不对特定目的适销性及适用性或者任何间接的、特殊的或连带的损失承担任何责任。

在适用法律允许的范围内，在任何情况下，中移物联网有限公司均不对用户因使用本手册内容和本手册中描述的产品而引起的任何特殊的、间接的、附带的或后果性的损坏、利润损失、数据丢失、声誉和预期的节省而负责。

因使用本手册中所述的产品而引起的中移物联网有限公司对用户的最大赔偿（除在涉及人身伤害的情况中根据适用法律规定的损害赔偿外），不应超过用户为购买此产品而支付的金额。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。公司保留随时修改本手册中任何信息的权利，无需进行提前通知且不承担任何责任。

## 商标声明

 中国移动  
China Mobile 为中国移动有限公司注册商标。

本手册和本手册描述的产品中出现的其他商标、产品名称、服务名称和公司名称，均为其各自所有者的财产。

## 进出口法规

出口、转口或进口本手册中描述的产品（包括但不限于产品软件和技术数据），用户应遵守相关进出口法律和法规。

## 隐私保护

关于我们如何保护用户的个人信息等隐私情况，请查看相关隐私政策。

## 操作系统更新声明

操作系统仅支持官方升级；如用户自己刷非官方系统，导致安全风险和损失由用户负责。

## 固件包完整性风险声明

固件仅支持官方升级；如用户自己刷非官方固件，导致安全风险和损失由用户负责。

## 版权所有©中移物联网有限公司。保留一切权利。

本手册中描述的产品，可能包含中移物联网有限公司及其存在的许可人享有版权的软件，除非获得相关权利人的许可，否则，非经本公司书面同意，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并以任何形式传播。



# 关于文档

## 修订记录

版本	日期	作者	描述
V1.0.0	2020/9/23	张雄威	初版



# 目录

服务与支持.....	2
文档声明 .....	3
关于文档 .....	5
修订记录.....	5
目录.....	6
1 HTTP AT 命令集.....	7
1.1 AT+HTTPCREATE 创建 HTTP 实例 .....	7
1.2 AT+HTTPHEADER 设置/查询 HTTP Header.....	8
1.3 AT+HTTPCONTENT 设置/查询 HTTP Content .....	9
1.4 AT+HTTPSEND 发送 HTTP 请求 .....	10
1.5 AT+HTTPCFG 配置 HTTP 参数 .....	12
1.6 AT+HTTPCLOSE 关闭 HTTP 实例.....	13
1.7 +HTTPNMIH 输出接收到的 Header 信息.....	13
1.8 +HTTPNMIC 输出接收到的 Content 信息 .....	14
1.9 +HTTDPDROP 丢弃信息 .....	15
1.10 +HTTPPER 错误信息 .....	15
1.11 +HTTPDISCONN 断开连接信息 .....	15
2 HTTP 应用举例.....	16
2.1 HTTP 服务器 GET 数据.....	16
2.2 POST 数据.....	19
2.3 OneNET HTTP API 应用举例.....	20
2.3.1 API-创建 LWM2M 设备.....	20
2.3.2 API-设备即时命令-读设备资源.....	21
3 HTTP 响应码.....	23
3.1 提示信息 .....	23
3.2 响应成功 .....	23
3.3 重定向 .....	23
3.4 客户端错误.....	24
3.5 服务器错误类 .....	25
4 HTTP 错误码.....	26
4.1 错误代码 .....	26

# 1 HTTP AT 命令集

本章介绍 MN316 HTTP 相关命令。

## 1.1 AT+HTTPCREATE 创建 HTTP 实例

该命令用于创建 HTTP 实例。

AT+HTTPCREATE	
语法	
<p><b>配置命令</b></p> <p>AT+HTTPCREATE=&lt;URL&gt;[,&lt;user&gt;,&lt;password&gt;]</p>	<p><b>响应</b></p> <p><b>成功:</b></p> <p>+HTTPCREATE:&lt;id&gt;</p> <p>OK</p> <p><b>失败:</b></p> <p>+CME ERROR:&lt;err&gt;</p>
参数说明	
<URL> 字符串	
服务器域名或者 IP 地址	
<user> 字符串	
用户名	
<password> 字符串	
密码	
<id> 整型	
返回的 HTTP 实例索引	
示例	
<p>AT+HTTPCREATE="http://iot.10086.cn:80/joinus/index/name/joinus.html"</p> <p>+HTTPCREATE:0</p> <p>OK</p>	
//返回的 HTTP 实例 ID，后续操作以此为参考。	



- 如果不输入 URL 前缀 http://，默认为 HTTP，URL 中的 path 字段将被忽略；
- 目前仅支持 HTTP，user 和 password 用于需要鉴权的网站，仅支持基本鉴权。

## 1.2 AT+HTTPHEADER 设置/查询 HTTP Header

该命令设置并查询 HTTP 头部信息。

AT+HTTPHEADER	
语法	
<b>配置命令</b> AT+HTTPHEADER=<id>[,<header>[,<encode_method>]]	<b>响应</b> 如果仅输入 id 参数字段，如果当前头部信息为空： OK  否则将返回已输入的 HTTP 头部信息： +HTTPHEADER:<id>,<length>,<header>  如果设置 id 以及 header 的格式正确，设置头部信息成功： OK  <b>失败：</b> +CME ERROR:<err>
参数说明	
<b>&lt;id&gt; 整型</b>	
通过+HTTPCREATE 返回的实例索引。	
<b>&lt;header&gt; 字符串</b>	
将要配置的头部信息。	
<b>&lt;encode_method&gt; 字符或者十六进制解析，当为字符时，支持转义字符。</b>	
0	转义字符串（缺省值）
1	十六进制字符串
<b>&lt;length&gt; 整型</b>	
返回的已输入的 header 长度。	
示例	
AT+HTTPHEADER=0,"User-Agent: Unkown\r\nContent-Type: application/text\r\n"	
OK	//增加 Header 字段。
AT+HTTPHEADER=0,"Connection: close\r\n"	
OK	//显示当前已输入的 Header。
AT+HTTPHEADER=0	
+HTTPHEADER:0,71,User-Agent: Unkown	//已输入 Header 长度 71 字节，内容如下。
Content-Type: application/text	
Connection: close	
OK	
AT+HTTPHEADER=0,""	
OK	//清空 Header 输入。
AT+HTTPHEADER=0	
OK	//此时 Header 为空，直接返回 OK。



- HTTP 协议要求 Header 字段之间以\r\n 分隔；
- 当输入的 Header 字段为空时，将清空此前已输入 Header，否则新输入的 Header 将添加在上一次 Header 输入之后。



### 1.3 AT+HTTPCONTENT 设置/查询 HTTP Content

该命令设置并查询 HTTP content 信息。

AT+HTTPCONTENT	
语法	
<b>配置命令</b> AT+HTTPCONTENT=<id>[,<content>[,<encode_method>]]	<b>响应</b> 如果仅输入 id 参数字段，如果当前 content 为空，则返回： OK  否则将返回已输入的 content 信息： +HTTPCONTENT:<id>,<length>,<content>  如果设置 id 以及 content 的格式正确，设置信息成功返回： OK  失败： +CME ERROR:<err>
参数说明	
<b>&lt;id&gt; 整型</b>	
通过+HTTPCREATE 返回的实例索引。	
<b>&lt;content&gt; 字符串</b>	
将要配置的 content 信息。	
<b>&lt;encode_method&gt; 字符或者十六进制解析，当为字符时，支持转义字符。</b>	
0	转义字符串（缺省值）
1	十六进制字符串
<b>&lt;length&gt; 整型</b>	
返回的已输入的 content 长度。	



AT+HTTPCONTENT 的用法与 AT+HTTPHEADER 类似，不做赘述。

# 1.4 AT+HTTPSEND 发送 HTTP 请求

该命令发送 HTTP 请求。

AT+HTTPSEND	
语法	
配置命令 AT+HTTPSEND=<id>,<method>,<path> >	响应 成功: OK  失败: +CME ERROR:<err>
参数说明	
<id> 整型	
通过+HTTPCREATE 返回的实例索引。	
<method> HTTP 请求类型	
0	GET
1	POST
2	PUT
3	DELETE
4	HEAD
<path> 字符串	
请求的 URL path 绝对路径参数。	
示例	
AT+HTTPSEND=0,0,"/page/index/name/zhangcheng" OK CONNECT OK +HTTPNMIH:0,1,443 Server: Tengine Content-Type: image/svg+xml Transfer-Encoding: chunked Connection: keep-alive ..... +HTTPNMIH:0,0,435 age-class: Standard Content-MD5: yW+WEp1yfdf9J59//yKoVg== x-oss-server-time: 3 ..... +HTTPNMIC:0,1,906,18 ..... +HTTPNMIC:0,1,906,460 ..... +HTTPNMIC:0,1,906,381 ..... +HTTPNMIC:0,1,927,21 ..... +HTTPNMIC:0,1,937,10 ..... +HTTPNMIC:0,0,937,0 .....	



- Path 参数不可为空；
- 成功发送 HTTP 请求后，header 以及 content 字段将自动清空以释放空间；
- 根据 ITU 规范，AT 命令中逗号+双引号 (, "及" )，双引号+分号 (";) 为 AT 命令参数分割符，如需输入需要转义或加间隔处理。



中国移动  
China Mobile

# 1.5 AT+HTTPCFG 配置 HTTP 参数

该命令用于查询或配置客户端参数。

AT+HTTPCFG	
语法	
<b>配置命令</b> AT+HTTPCFG=<id>,<type>[,<value>[,<encode>]]	<b>响应</b> 如果输入格式正确返回： OK  如果仅设置<id>以及<type>，则返回 value 查询结果： +HTTPCFG:<id>,<type>,<value>  <b>失败：</b> +CME ERROR:<err>
参数说明	
<id> 整型	
通过+HTTPCREATE 返回的实例索引。	
<type> 配置的参数类型	
1	服务器 cert 证书（暂不支持）
2	客户端 cert 证书（暂不支持）
3	客户端 pk（暂不支持）
4	预留
5	预留
6	串口输出接收到的 header 及 content 字段时分包大小（缺省值 400）。
7	串口输出接收到的 header 及 content 字段时分包延时时间间隔(ms)，最大支持 3000ms。
8	串口输出接收到的 header 及 content 字段时是否转换为 16 进制输出。
<value> 字符串	
type=8 时，value 为字符串“0”或“1”。	
<encode> 整型，仅当配置 type=1/2/3 时有此项参数。	
0	字符串输入（缺省值）
1	Hex 字符串输入
示例	
AT+HTTPCFG=0,6,512	//配置数据输出包大小 512 字节。
OK	
AT+HTTPCFG=0,6	//读取数据输出包大小。
+HTTPCFG:0,6,512	//当前为 512 字节。
OK	



如果控制输出速率较低时，有可能引起 HTTPDROPP 现象。

## 1.6 AT+HTTPCLOSE 关闭 HTTP 实例

该命令用于关闭 HTTP 实例。

AT+HTTPCLOSE	
语法	
配置命令 AT+HTTPCLOSE=<id>	响应 成功： OK  失败： +CME ERROR:<err>
参数说明	
<id> 整型	
通过+HTTPCREATE 返回的实例索引。	

## 1.7 +HTTPNMIH 输出接收到的 Header 信息

主动上报接收到的 HTTP 头部信息。

+HTTPNMIH	
语法	
+HTTPNMIH	响应 +HTTPNMIH:<id>,<more>,<length> <header>
参数说明	
<id> 整型	
通过+HTTPCREATE 返回的实例索引。	
<more> 整型	
是否后续还有头部数据。	
<length> 整型	
本次输出的头部信息长度。	
<header> 字符串	
头部内容。	

# 1.8 +HTTPNMIC 输出接收到的 Content 信息

主动上报接收到 HTTP content 数据。

+HTTPNMIC	
语法	
+HTTPNMIC	响应 +HTTPNMIC:<id>,<more>,<total>,<summation>,<cur_len> <content>
参数说明	
<id> 整型	
通过+HTTPCREATE 返回的实例索引。	
<more> 字符串	
是否后续还有数据。	
<total> 整型	
本次输出的 Content-length 总长度或 chunked 传输分包总长度；如果为 chunked 模式，total 字段会随着解析到的 chunked 字段不断增加。	
<summation> 整型	
已接收到的 content 总长度。	
<cur_len> 整型	
本包输出长度。	
<content> 字符串	
http 数据内容。	



- <total>参数根据服务器下发策略模组自动识别；
- 当为 chunked 模式时，必须当 more 输出 0 时才能确定最终接收总长度。

## 1.9 +HTTDPDROP 丢弃信息

上报底层 TCP 数据缓冲区满，数据丢失。

+HTTDPDROP	
语法	
+HTTDPDROP	响应 +HTTDPDROP:<id>,<length>
参数说明	
<id> 整型	
通过+HTTPCREATE 返回的实例索引。	
<length> 整型	
drop 数据长度，如果出现数据丢失将引起后续数据解析异常，建议提高模组波特率并丢弃已接收数据，重新发送 HTTP 请求。	

## 1.10 +HTTPERR 错误信息

上报 HTTP 错误代码。

+HTTPERR	
语法	
+HTTPERR	响应 +HTTPERR:<id>,<code>
参数说明	
<id> 整型	
通过+HTTPCREATE 返回的实例索引。	
<code> 整型	
错误类型，包括 HTTP 错误响应码或者 HTTP 连接错误码，如果是解析到 HTTP 响应码异常，将额外输出服务器信息，如 404 NOT FOUND，参考响应码与错误码章节。	

## 1.11 +HTTPDISCONN 断开连接信息

上报承载 HTTP 的 TCP 连接断开，下次发送请求时会自动重连。

+HTTPDISCONN	
语法	
+HTTPDISCONN	响应 +HTTPDISCONN:<id>
参数说明	
<id> 整型	
通过+HTTPCREATE 返回的实例索引。	

## 2 HTTP 应用举例

本章详细描述了 HTTP 对接 OneNET 平台的流程和相应的 AT 命令流程。

### 2.1 HTTP 服务器 GET 数据

<pre> AT+HTTPCREATE="http://iot.10086.cn/" +HTTPCREATE:0 OK AT+HTTPHEADER=0,"User-Agent: Unkown\r\nConnection: close\r\n" OK AT+HTTPSEND=0,0,"page/index/name/culture"  OK CONNECT OK +HTTPNMIH:0,0,313 Server: nginx Date: Sun, 10 Jun 2018 02:54:38 GMT Content-Type: text/html; charset=utf-8 Transfer-Encoding: chunked  Connection: close Set-Cookie: PHPSESSID=7it2op5guravshrvf3j29kpr1; path=/; HttpOnly Expires: Thu, 19 Nov 1981 08:52:00 GMT Pragma: no-cache Cache-control: private X-Powered-By: ThinkPHP +HTTPNMIC:0,1,7960,1122,1122 &lt;!DOCTYPE html&gt; &lt;html lang="en"&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt;   &lt;link href="/Public/Img/favicon.ico" rel="shortcut icon" type="image/vnd.microsoft.icon"&gt; &lt;meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user- scalable=no"&gt;   &lt;meta      name="apple-mobile-web-app- capable" content="yes"&gt;   &lt;meta      name="apple-mobile-web-app- status-bar-style" content="black"&gt;   &lt;meta      name="format-detection" content="telephone=no"&gt;   &lt;meta name="robots" content="all" /&gt; ..... +HTTPNMIC:0,1,7960,2582,1460 &lt;meta name="keywords" content=" </pre>	<pre> //创建 http 实例，可指定 URL，除 HTTP 标识和域名外，其他 URL 部分将被忽略。 //模组返回 httpclient id:0。  //配置 HTTP 头部，注意 HTTP 协议规定每个头部字段之间有换行 分隔，部分 HTTP 服务器要求配置 User-Agent 字段。 //发送 GET (method 参数为 0)请求包，URL 资源绝对路径 "/page/index/name/culture"。 //服务器响应头部信息。  //本次传输为 chunked 模式，服务器数据将分片传输，不指定 Content-Length。  //服务器响应内容字段，首包数据。  //下一包数据。 </pre>
---	---



```

.....
+HTTPNMIC:0,1,7960,4042,1460
ass="navbar-header">
    <button type="button" class="navbar-
toggle" data-toggle="collapse" data-target="#b-
menu-1">
        <span class="sr-only">Toggle
navigation</span>
        <span class="icon-bar"></sp
+HTTPNMIC:0,1,7960,4042,1460
<div class="navbar-header">
    <button type="button" class="navbar-
toggle" data-toggle="collapse" data-target="#b-
menu-1">
        <span class="sr-only">Togg
.....
.....
+HTTPNMIC:0,1,36632,36632,502
<script type="text/javascript">
    document.onkeydown=function(event){
        var e = event || window.event ||
arguments.callee.caller.arguments[0];

        if(e      &&      e.keyCode==13
&&$("#fb_pass").focus()){ // enter  閤?
            go_group();
        }
    };
.....
    $(".btn-close").click(function(){

        window.history.back(-1);           //最后一包数据，chunked 传输结束。
        $(".fill").hide();

    });
</script>
+HTTPNMIC:0,0,37539,37539,0
AT+HTTPHEADER=0,"User-Agent:
Unkown\r\nConnection: close\r\n"
OK
AT+HTTPSEND=0,0,"/Public/Home/assets/imag
e/chiMod_icon1.png"
OK
+HTTPNMIH:0,0,209
Server: nginx
Date: Sun, 10 Jun 2018 03:16:55 GMT
Content-Type: image/png
Content-Length: 7862

Last-Modified: Thu, 29 Mar 2018 06:27:39 GMT
Connection: close
ETag: "5abc875b-1eb6"
Accept-Ranges: bytes
+HTTPNMIC:0,1,7862,1460,1460
.....
HTTPNMIC:0,0,7862,7862,562
.....

AT+HTTPSEND=0,0,"/Public/Home/assets/imag
e/chiMod_icon11.png"
OK

```

+HTTPErr:0,404

404 Not Found

+HTTPNMIH:0,0,139

Server: nginx

Date: Sun, 10 Jun 2018 03:36:54 GMT

Content-Type: text/html; charset=UTF-8

Transfer-Encoding: chunked

Server: nginx

Date: Sun, 10 Jun 2018 03:36:54 GMT

Content-Type: text/html; charset=UTF-8

Transfer-Encoding: chunked

Connection: close

+HTTPNMIC:0,0,0,0,0

AT+HTTPCLOSE=0

OK



如果模组接收数据时出现+HTTDPDROp:0,1460 上报，说明模组接收缓冲区满引起数据 drop，数据丢失后将出现后续接收过程解析错误，请检测当前是否波特率较低影响数据交互，如 M5310-A 使用 +NATSPEED 将波特率调高至 115200 或 230400（波特率高于 9600 会自动禁止模组进入 PSM），HTTP 功能使用完毕之后调回较低波特率以恢复使能 PSM 的状态。



## 2.2 POST 数据

```

AT+HTTPCREATE="http://iot.10086.cn/"           //创建 http 实例。
+HTTPCREATE:0                                   //模组返回 httpclient id:0。
OK
AT+HTTPHEADER=0,"Connection: keep-alive\r\nAccept: //配置发送 HTTP header 信息。
*/*\r\nOrigin: http://iot.10086.cn\r\nX-Requested-With:
XMLHttpRequest\r\nUser-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/67.0.3396.79 Safari/537.36\r\nContent-Type:
application/x-www-form-urlencoded; charset=UTF-
8\r\nReferer:
http://iot.10086.cn/cooperation/index/cid/174\r\nAccept-
Encoding: gzip, deflate\r\nAccept-Language: zh-
CN,zh;q=0.9\r\n"
OK
AT+HTTPCONTENT=0,"num=1&type=1&info=%E5%8F%B3
%E4%BE%A7-%E8%81%94%E7%B3%BB%E6%88%91%E4%
BB%AC"
OK                                               //配置发送 HTTP content 信息。
AT+HTTPSEND=0,1,"/Commom/buried_point"
"/Commom/buried_point"
OK
+HTTPNMIH:0,0,468                             //发送 POST (method 参数为 1)请求包，URL 资源绝对
Server: nginx                                  路径。
Date: Sun, 10 Jun 2018 04:02:38 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: think_language=zh-CN; expires=Sun, 10-Jun-
2018 05:02:38 GMT; Max-Age=3600; path=/
Set-Cookie: PHPSESSID=l077puu6ri95pnb35n61obk2b0;
path=/; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-
check=0, pre-check=0
Pragma: no-cache
Content-Encoding: gzip
+HTTPNMIC:0,1,63,63,63                       //成功收到服务器响应 header 信息。
?0000000000 ——生*.I.)-V?設侍 K 菱璉?5324?55 暇 S 襪 JI,I
——??0#??0000
+HTTPNMIC:0,0,63,63,0
+HTTPERR:0,-14
+HTTPDISCON:0                                //content 字段结束。
//服务器连接断开提示。

```

## 2.3 OneNET HTTP API 应用举例

API 使用说明请参考 OneNET 文档中心>>资源下载>>1.1 说明文档>>NB-IoT 协议设备开发文档。

<https://open.iot.10086.cn/doc/art398.html#97>

本段仅举例创建设备与即时读设备资源命令。

### 2.3.1 API-创建 LWM2M 设备

```

AT+HTTPCREATE="http://api.heclouds.com/"
+HTTPCREATE:0
OK

AT+HTTPHEADER=0,"api-                                     //APIKey 需替换为实际值。
key:ChangeForYourOwnKey\r\nContent-
Type:application/json\r\nConnection: Close\r\n"
OK
AT+HTTPCONTENT=0,"{"title": "865820030265172\r","desc":
"865820030265172\r","protocol": "LWM2M\r","auth_info":
{"865820030265172": "460043783403288"}}"
OK
AT+HTTPSEND=0,1,"/devices"
OK
CONNECT OK                                     //发送 post 请求，创建设备。
+HTTPNMIH:0,0,158
Date: Mon, 22 Oct 2018 09:17:06 GMT
Content-Type: application/json
Content-Length: 59
Connection: keep-alive
Server: Apache-Coyote/1.1
Pragma: no-cache
+HTTPNMIH:0,0,59,59,59
{"errno":0,"data":{"device_id":"500500178"},"error":"succ"}
+HTTPDISCON:0
AT+MIPLCREATEEX="nbiotbt.heclouds.com:5683",1,""
+MIPLCREATEEX:0
OK
AT+MIPLADDOBJ=0,3200,1,"1",0,1                                     //添加 object。
OK
AT+MIPLOPEN=0,3000,30                                     //登录 OneNET。
OK
+MIPLEVENT:0,2                                     //登录成功。
+MIPLEVENT:0,6
+MIPLOBSERVE:0,21341,1,3200,0,-1
+MIPLDISCOVER:0,21342,3200

```

## 2.3.2 API-设备即时命令-读设备资源

AT+HTTPHEADER=0,"api-key:	ChangeForYourOwnKey	//APIKey 需替换实际值。
\r\nConnection: Close\r\n"		
OK		
AT+HTTPSEND=0,0,"/nbiot?imei=865820030265172&obj_id=3		//控制平台下发 READ 命令。
200"		
OK		
CONNECT OK		
+MIPLREAD:0,21345,3200,-1,-1		
AT+MIPLREADRSP=0,21345,1,3200,0,5500,5,1,"0",2,1		//模组收到平台下发的 READ 提示。
OK		//模组回复 READ。
AT+MIPLREADRSP=0,21345,1,3200,0,5501,3,3,"-12",1,2		
OK		//模组回复 READ。
AT+MIPLREADRSP=0,21345,1,3200,0,5750,1,4,"test",0,0OK		//模组回复 READ 并开始传输回复数据。
+HTTPNMIH:0,0,148		
Date: Tue, 23 Oct 2018 02:18:11 GMT		//读设备资源 HTTP API 返回 HTTP 数据头。
Content-Type: application/json;charset=UTF-8		
Transfer-Encoding: chunked		
Connection: close		
Pragma: no-cache		
+HTTPNMIC:0,1,144,144,144		
{"errno":0,"error":"succ","data":[{"res":[{"val":false,"res_id":5500},{		//读设备资源 HTTP API 返回的 HTTP 内容，为设备
val":-		回复 READ 上报值。
12,"res_id":5501},{val:"test","res_id":5750}],obj_inst_id":0}}}		
+HTTPNMIC:0,0,144,144,0		
+HTTPDISCON:0		
AT+MIPLCLOSE=0		
OK		//HTTP 连接断开。
AT+MIPLDELETE=0		//关闭 OneNET 连接。
OK		

当 IMEI 已被注册时将返回注册错误；即时读设备资源命令当设备不在线将返回设备不在线。当设备长时间不与平台通信可能引起路由信息老化或处于休眠模式时而无法收到平台下发的数据。当一定时间后未回复响应，API 将返回读取超时(unsucccess: TIME\_OUT)。其他注意事项请参考 OneNET 平台 API 手册。

以上过程仅用于展示 HTTP 操作示例，实际运用中应在客户服务端使用 OneNET API 控制模组进行相应动作。创建设备 HTTP API 中需要在 content 字段输入 json 字符串，由于 json 字符串中存在 AT 命令解析无法识别的双引号+逗号(",)，需要在不影响服务器解析的情况下在逗号前插入换行符避免解析失败；或者全文使用十六进制转换后输入；+HTTSEND 命令暂不支持此操作，请尽量将此类数据放入 content 或者 header 字段上传。

举例如下。

```
AT+HTTPCONTENT=0,{"title":      "865820030265172\r","desc":
"865820030265172\r","protocol":    "LWM2M\r","auth_info":
{"865820030265172": "460043783403288"},}
OK
AT+HTTPCONTENT=0
+HTTPCONTENT:0,130
{"title": "865820030265172"
,"desc": "865820030265172"
,"protocol": "LWM2M"
,"auth_info": {"865820030265172": "460043783403288"},}
OK
AT+HTTPCONTENT=0,""
OK
AT+HTTPCONTENT=0,7B227469746C65223A202238363538323
0303330323635313732222C2264657363223A20223836353832
30303330323635313732222C2270726F746F636F6C223A20224
C574D324D22C22617574685F696E666F223A207B2238363538
3230303330323635313732223A20223436303034333738333430
33323838227D2C7D,1
OK
AT+HTTPCONTENT=0
+HTTPCONTENT:0,127
{"title": "865820030265172","desc": "865820030265172","protocol": "
LWM2M","auth_info": {"865820030265172": "460043783403288"},}
OK
```

//清空 content 输入。

//以 16 进制字符串输入 content。

## 3 HTTP 响应码

### 3.1 提示信息

状态码	描述
100	Continue
101	Switching Protocols
102	Processing

### 3.2 响应成功

状态码	描述
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
207	Multi-Status

### 3.3 重定向

状态码	描述
300	Multiple Choices
301	Moved Permanently
302	Move temporarily
303	See Other
304	Not Modified
305	Use Proxy
306	Switch Proxy
307	Temporary Redirect

## 3.4 客户端错误

状态码	描述
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
415	Unsupported Media Type
416	Requested Range Not Satisfiable
417	Expectation Failed
421	too many connections
422	Unprocessable Entity
423	Locked
424	Failed Dependency
449	Retry With
451	Unavailable For Legal Reasons



### 3.5 服务器错误类

状态码	描述
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version Not Supported
506	Variant Also Negotiates
507	Insufficient Storage
509	Bandwidth Limit Exceeded
510	Not Extended
600	Unparseable Response Headers



## 4 HTTP 错误码

### 4.1 错误代码

状态码	描述
9100	无效的 HTTP ID
9101	无效的证书类型
9102	创建失败
9103	不支持 SSL 方式
9104	正在连接
9105	发送错误
9106	关闭错误
9107	Header 设置错误
9108	Content 设置错误

