

# LTE Standard(E)系列 HTTP(S)应用指导

**LTE Standard 模块系列**

版本：1.3

日期：2024-11-13

状态：受控文件



上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233

电话：+86 21 5108 6236 邮箱：[info@quectel.com](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：<http://www.quectel.com/cn/support/sales.htm>。

如需技术支持或反馈我司技术文档中的问题，请随时登录网址：

<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：[support@quectel.com](mailto:support@quectel.com)。

## 前言

移远通信提供该文档内容以支持客户的产品设计。客户须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档。

## 使用和披露限制

### 许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

### 版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，除了正常的非独家、免版税的产品使用许可，任何形式的购买都不可被视为授予许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，移远通信有权追究法律责任。

### 商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

### 第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

## 隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

## 免责声明

- 1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
- 2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
- 3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
- 4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

版权所有 ©上海移远通信技术股份有限公司 2024，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2024.**

# 文档历史

## 修订记录

版本	日期	作者	变更表述
-	2022-06-07	Lawrence LIU	文档创建
1.0	2023-02-27	Lawrence LIU	受控版本
1.1	2023-10-30	Greyson DONG/ Fawei ZHOU	<ol style="list-style-type: none"> <li>1. 新增适用模块 EC600Z-CN 和 EC800Z-CN。</li> <li>2. 更新 AT+QHTTPCFG 中参数&lt;content_type&gt;的取值（第 2.3.1 章）。</li> <li>3. 新增 AT+QHTTPGETEX 中参数&lt;start_position&gt;相关备注（第 2.3.4 章）。</li> </ol>
1.2	2024-06-24	Elmo HUANG	<ol style="list-style-type: none"> <li>1. 新增适用模块 EC801E-CN。</li> <li>2. 新增有关 AT+QHTTPCFG="http2.0"支持情况的备注（第 2.3.1 章）。</li> </ol>
1.3	2024-11-13	Elmo HUANG	<ol style="list-style-type: none"> <li>1. 完善 AT 示例声明（第 2.2 章）。</li> <li>2. 更新 HTTP(S) POST 请求头信息示例（第 1.3.1 章）。</li> <li>3. 更新命令举例中的 URL（第 3 章）。</li> </ol>

## 目录

文档历史 .....	3
目录 .....	4
表格索引 .....	6
<b>1 引言 .....</b>	<b>7</b>
1.1. 适用模块 .....	7
1.2. HTTP(S)命令使用流程 .....	8
1.3. HTTP(S)请求头信息说明 .....	8
1.3.1. 自定义 HTTP(S)请求头信息 .....	8
1.3.2. 输出 HTTP(S)响应头信息 .....	9
1.4. 数据模式说明 .....	9
<b>2 HTTP(S) AT 命令详解 .....</b>	<b>10</b>
2.1. AT 命令说明 .....	10
2.1.1. 定义 .....	10
2.1.2. AT 命令语句 .....	10
2.2. AT 示例声明 .....	11
2.3. AT 命令详解 .....	11
2.3.1. AT+QHTTPCFG 配置 HTTP(S)服务器参数 .....	11
2.3.2. AT+QHTTPURL 设置 HTTP(S)服务器 URL .....	16
2.3.3. AT+QHTTPGET 发送 GET 请求到 HTTP(S)服务器 .....	17
2.3.4. AT+QHTTPGETEX 发送范围 GET 请求到 HTTP(S)服务器 .....	18
2.3.5. AT+QHTTPPOST 通过 UART/USB 发送 POST 请求到 HTTP(S)服务器 .....	19
2.3.6. AT+QHTTPPOSTFILE 通过文件发送 POST 请求到 HTTP(S)服务器 .....	21
2.3.7. AT+QHTTPPUT 通过 UART/USB 发送 PUT 请求到 HTTP(S)服务器 .....	23
2.3.8. AT+QHTTPPUTFILE 通过文件发送 PUT 请求到 HTTP(S)服务器 .....	24
2.3.9. AT+QHTTPREAD 通过 UART/USB 读取 HTTP(S)服务器响应信息 .....	26
2.3.10. AT+QHTTPREADFILE 通过文件读取 HTTP(S)服务器响应信息 .....	27
2.3.11. AT+QHTTPSTOP 取消 HTTP(S)请求 .....	27
<b>3 举例 .....</b>	<b>29</b>
3.1. 访问 HTTP 服务器 .....	29
3.1.1. 发送 HTTP GET 请求并读取响应信息 .....	29
3.1.2. 发送 HTTP POST 请求并读取响应信息 .....	30
3.1.2.1. 通过 UART/USB 获取 POST 请求体 .....	30
3.1.2.2. 从文件系统获取 POST 请求体 .....	31
3.1.3. 发送 HTTP PUT 请求并读取响应信息 .....	32
3.1.3.1. 通过 UART/USB 获取 PUT 请求体 .....	32
3.1.3.2. 从文件系统获取 PUT 请求体 .....	33
3.2. 访问 HTTPS 服务器 .....	34
3.2.1. 发送 HTTPS GET 请求并读取响应信息 .....	34
3.2.2. 发送 HTTPS POST 请求并读取响应信息 .....	36
3.2.2.1. 从 UART/USB 获取 POST 请求体 .....	36

3.2.2.2.	从文件系统获取 POST 请求体 .....	37
3.2.3.	发送 HTTPS PUT 请求并读取响应信息 .....	38
3.2.3.1.	从 UART/USB 获取 PUT 请求体 .....	38
3.2.3.2.	从文件系统获取 PUT 请求体 .....	40
4	常见问题处理.....	42
4.1.	HTTP(S) AT 命令执行失败.....	42
4.2.	PDP 激活失败.....	42
4.3.	DNS 解析失败 .....	42
4.4.	GET/POST/PUT 请求发送失败 .....	43
4.5.	响应信息读取失败 .....	43
5	错误代码.....	44
6	HTTP(S)响应错误代码 .....	46
7	附录 参考文档及术语缩写 .....	47

表格索引

表 1: 适用模块 ..... 7

表 2: AT 命令及响应类型 ..... 10

表 3: 错误代码列表..... 44

表 4: HTTP(S)响应代码列表 ..... 46

表 5: 参考文档 ..... 47

表 6: 术语缩写 ..... 47

# 1 引言

移远通信 LTE Standard(E)系列模块为 HTTP(S)服务器提供 HTTP(S)应用程序。

HTTP（HyperText Transfer Protocol，超文本传输协议）是一种用于分布式、协作式和超媒体信息系统的应⽤层协议。

HTTPS（HyperText Transfer Protocol Secure，超文本传输安全协议）是一种通过计算机网络进行安全通信的传输协议。HTTPS 经由 HTTP 进行通信，但利用 SSL/TLS 来加密数据包。HTTPS 开发的主要目的是提供对网站服务器的身份认证，保护交换资料的隐私与完整性。

本文档主要介绍了与 HTTP(S)相关的 AT 命令。

## 1.1. 适用模块

表 1：适用模块

模块系列	模块
LTE Standard(E)	EC600E-CN
	EC800E-CN
	EC600Z-CN
	EC800Z-CN
	EC801E-CN



## 1.2. HTTP(S)命令使用流程

通过模块 TCP/IP AT 命令，可配置 PDP 上下文，激活/去激活上下文以及查询 PDP 上下文状态。通过模块的 HTTP(S) AT 命令，可以发送 HTTP(S) GET/POST/PUT 请求到 HTTP(S)服务器，并读取来自 HTTP(S)服务器的响应结果。大致流程如下：

第一步，使用 **AT+QICSGP** 配置 PDP 上下文的<APN>、<username>、<password>和其他参数，详情请参考文档 [1]。

第二步，通过 **AT+QIACT** 激活 PDP 上下文后，可使用 **AT+QIACT?**查询已分配的 IP 地址，详情请参考文档 [1]。

第三步，通过 **AT+QHTTPCFG** 配置 PDP 和 SSL 上下文 ID。

第四步，通过 **AT+QSSLCFG** 配置 SSL 上下文参数，详情请参考文档 [2]。

第五步，通过 **AT+QHTTPURL** 设置 HTTP(S) URL。

第六步，发送 HTTP(S)请求。**AT+QHTTPGET** 用于发送 HTTP(S) GET 请求，**AT+QHTTPGETEX** 用于发送 HTTP(S)范围 GET 请求。**AT+QHTTPPOST** 或 **AT+QHTTPPOSTFILE** 用于发送 HTTP(S) POST 请求。**AT+QHTTPPUT** 或 **AT+QHTTPPUTFILE** 用于发送 HTTP(S) PUT 请求。

第七步，通过 **AT+QHTTPREAD** 或 **AT+QHTTPREADFILE** 读取 HTTP(S)响应信息。

第八步，通过 **AT+QIDEACT** 去激活 PDP 上下文，详情请参考文档 [1]。

## 1.3. HTTP(S)请求头信息说明

### 1.3.1. 自定义 HTTP(S)请求头信息

模块自动填补 HTTP(S)请求头信息，用户也可通过 **AT+QHTTPCFG** 将<request\_header>配置为 1 自定义 HTTP(S)请求头信息，但需遵循以下标准：

- 遵循 HTTP(S)请求头信息语句规范。
- HTTP(S)请求行中的 URI 值和 **Host:**请求头信息必须与 **AT+QHTTPURL** 配置的 URL 一致。
- HTTP(S)请求头信息必须以<CR><LF>结尾。

以下为标准 HTTP(S) POST 请求头信息示例：

```
POST /processorder.php HTTP/1.1<CR><LF>
Host:192.0.2.2:8011<CR><LF>
Accept: */*<CR><LF>
```

```
User-Agent: QUECTEL_MODULE<CR><LF>
Connection: Keep-Alive<CR><LF>
Content-Type: application/x-www-form-urlencoded<CR><LF>
Content-Length: 48<CR><LF>
<CR><LF>
Message=1111&Appleqty=2222&Orangeqty=3333&find=1
```

### 1.3.2. 输出 HTTP(S)响应头信息

模块不自动输出 HTTP(S)响应头信息，可通过 **AT+QHTTPCFG** 将 **<response\_header>** 配置为 1 获取 HTTP(S)响应头信息，然后执行 **AT+QHTTPREAD** 或 **AT+QHTTPREADFILE**，HTTP(S)响应头信息将以 HTTP(S)响应体形式输出。

## 1.4. 数据模式说明

模块的 COM 口有两种工作模式，一是 AT 命令模式，另一种是数据模式。AT 命令模式下，通过 COM 口输入的数据被认为是 AT 命令；数据模式下，则被认为是数据。

用户可以通过 **+++** 或者 **DTR**（需先设置 **AT+D1**）两种方式退出数据模式，为了防止 **+++** 被当成数据发送，实际操作时必须遵循以下标准：

- **+++** 输入前 1 秒内不能输入其它任何数据。
- 必须在 1 秒内输入 **+++**，并且不能输入其它任何数据。
- **+++** 输入后 1 秒内不能输入其它任何数据。

通过执行 **AT+QHTTPURL**、**AT+QHTTPPOST**、**AT+QHTTPPUT** 和 **AT+QHTTPREAD** 可使 COM 口进入数据模式，但在这些命令响应之前，若通过 **+++** 或拉高 **DTR** 退出数据模式，会中断这些命令的执行。在这种情况下，无法通过执行 **ATO** 命令使 COM 口重新进入数据模式。

## 2 HTTP(S) AT 命令详解

### 2.1. AT 命令说明

#### 2.1.1. 定义

- **<CR>** 回车符。
- **<LF>** 换行符。
- **<...>** 参数名称。实际命令行中不包含尖括号。
- **[...]** 可选参数或 TA 信息响应的可选部分。实际命令行中不包含方括号。若无特别说明，配置命令中的可选参数被省略时，将默认使用其之前已设置的值或其默认值。
- **下划线** 参数的默认设置。

#### 2.1.2. AT 命令语句

前缀 **AT** 或 **at** 必须加在每个命令行的开头。输入 **<CR>** 将终止命令行。通常，命令后面跟随形式为 **<CR><LF><response><CR><LF>** 的响应。在本文档中变现命令和响应的表格中，省略了 **<CR><LF>**，仅显示命令和响应。

表 2：AT 命令及响应类型

AT 命令类型	语句	描述
测试命令	<b>AT+&lt;cmd&gt;=?</b>	测试是否存在相应的命令，并返回有关其参数的类型、值或范围的信息。
查询命令	<b>AT+&lt;cmd&gt;?</b>	查询相应命令的当前参数值。
设置命令	<b>AT+&lt;cmd&gt;=&lt;p1&gt;[,&lt;p2&gt;[,&lt;p3&gt;[...]]]</b>	设置用户可定义的参数值。
执行命令	<b>AT+&lt;cmd&gt;</b>	返回特定的参数信息或执行特定的操作。

## 2.2. AT 示例声明

本文中的示例仅为方便用户了解 AT 命令的使用方法，不构成移远通信对终端流程设计的建议或意见，也不代表模块应被设置成相应示例中的状态。某些 AT 命令存在多个示例，这些示例之间不存在承接关系或连续性。AT 命令示例中存在的 URL、域名、IP 地址、用户名/账号以及密码等（若有），仅为示意以供说明之用；实际使用时须根据实际情况进行修改。

## 2.3. AT 命令详解

### 2.3.1. AT+QHTTPCFG 配置 HTTP(S)服务器参数

该命令用于配置 HTTP(S)服务器参数，包括配置 PDP 上下文 ID，自定义 HTTP(S)请求头信息，输出 HTTP(S)响应头信息和配置 SSL 上下文 ID 等。若执行设置命令时只保留一个参数，表示查询当前配置。

AT+QHTTPCFG 配置 HTTP(S)服务器参数	
测试命令 <b>AT+QHTTPCFG=?</b>	响应 <b>+QHTTPCFG: "contextid",</b> (支持的<contextID>列表) <b>+QHTTPCFG: "requestheader",</b> (支持的<request_header>列表) <b>+QHTTPCFG: "responseheader",</b> (支持的<response_header>列表) <b>+QHTTPCFG: "sslctxid",</b> (支持的<sslctxID>列表) <b>+QHTTPCFG: "contenttype",</b> (支持的<content_type>列表) <b>+QHTTPCFG: "rspout/auto",</b> (支持的<auto_outrsp>列表) <b>+QHTTPCFG: "closed/ind",</b> (支持的<closedind>列表) <b>+QHTTPCFG: "url",</b> <url_string> <b>+QHTTPCFG: "header",</b> <header_value> <b>+QHTTPCFG: "auth",</b> <username:password> <b>+QHTTPCFG: "http2.0",</b> (支持的<http2_enable>列表) <b>+QHTTPCFG: "form/option",</b> <name>,<file_name>,<content_type> <b>+QHTTPCFG: "reset"</b>  <b>OK</b>
查询命令 <b>AT+QHTTPCFG?</b>	响应 <b>+QHTTPCFG: "contextid",</b> <contextID> <b>+QHTTPCFG: "requestheader",</b> <request_header> <b>+QHTTPCFG: "responseheader",</b> <response_header> <b>+QHTTPCFG: "sslctxid",</b> <sslctxID> <b>+QHTTPCFG: "contenttype",</b> <content_type> <b>+QHTTPCFG: "rspout/auto",</b> <auto_outrsp>

	<b>+QHTTPCFG: "closed/ind",&lt;closedind&gt;</b> <b>+QHTTPCFG: "url",&lt;url_string&gt;</b> <b>+QHTTPCFG: "header",&lt;header_value&gt;</b> <b>+QHTTPCFG: "auth",&lt;username:password&gt;</b> <b>+QHTTPCFG: "http2.0",&lt;http2_enable&gt;</b> <b>+QHTTPCFG:"form/option",&lt;name&gt;,&lt;file_name&gt;,&lt;content_type&gt;</b> <b>+QHTTPCFG: "reset"</b>  <b>OK</b>
设置命令 <b>AT+QHTTPCFG="contextid",&lt;contextID&gt;]</b>	响应 若省略可选参数，则查询当前配置： <b>+QHTTPCFG: "contextid",&lt;contextID&gt;</b>  <b>OK</b>  若指定可选参数，则配置 PDP 上下文 ID： <b>OK</b> 或者 <b>+CME ERROR: &lt;err&gt;</b>
设置命令 <b>AT+QHTTPCFG="requestheader",&lt;request_header&gt;]</b>	响应 若省略可选参数，则查询当前配置： <b>+QHTTPCFG: "requestheader",&lt;request_header&gt;</b>  <b>OK</b>  若指定可选参数，则禁用或启用自定义 HTTP(S)请求头信息： <b>OK</b> 或者 <b>+CME ERROR: &lt;err&gt;</b>
设置命令 <b>AT+QHTTPCFG="responseheader",&lt;response_header&gt;]</b>	响应 若省略可选参数，则查询当前配置： <b>+QHTTPCFG: "responseheader",&lt;response_header&gt;</b>  <b>OK</b>  若指定可选参数，则禁用或启用输出 HTTP(S)响应头信息： <b>OK</b> 或者 <b>+CME ERROR: &lt;err&gt;</b>
设置命令 <b>AT+QHTTPCFG="sslctxid",&lt;sslctxID&gt;]</b>	响应 若省略可选参数，则查询当前配置： <b>+QHTTPCFG: "sslctxid",&lt;sslctxID&gt;</b>

	<p>OK</p> <p>若指定可选参数，则配置用于 HTTP(S)的 SSL 上下文 ID:</p> <p>OK</p> <p>或者</p> <p>+CME ERROR: &lt;err&gt;</p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="contenttype",&lt;content_type&gt;]</b></p>	<p>响应</p> <p>若省略可选参数，则查询当前配置:</p> <p><b>+QHTTPCFG: "contenttype",&lt;content_type&gt;</b></p> <p>OK</p> <p>若指定可选参数，则配置 HTTP(S)体的数据类型:</p> <p>OK</p> <p>或者</p> <p>+CME ERROR: &lt;err&gt;</p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="rspout/auto",&lt;auto_outrsp&gt;]</b></p>	<p>响应</p> <p>若省略可选参数，则查询当前配置:</p> <p><b>+QHTTPCFG: "rspout/auto",&lt;auto_outrsp&gt;</b></p> <p>OK</p> <p>若指定可选参数，则禁用或启用自动输出 HTTP(S)响应头信息:</p> <p>OK</p> <p>或者</p> <p>+CME ERROR: &lt;err&gt;</p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="closed/ind",&lt;closedind&gt;]</b></p>	<p>响应</p> <p>若省略可选参数，则查询当前配置:</p> <p><b>+QHTTPCFG: "closed/ind",&lt;closedind&gt;</b></p> <p>OK</p> <p>若指定可选参数，则禁用或启用上报 HTTP(S)会话关闭 URC</p> <p><b>+QHTTPURC: "closed":</b></p> <p>OK</p> <p>或者</p> <p>+CME ERROR: &lt;err&gt;</p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="url",&lt;url_string&gt;]</b></p>	<p>响应</p> <p>若省略可选参数，则查询当前配置:</p> <p><b>+QHTTPCFG: "url",&lt;url_string&gt;</b></p> <p>OK</p>

	<p>若指定可选参数，则配置 HTTP(S)的 URL:</p> <p><b>OK</b></p> <p>或者</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="header"[,&lt;header_value&gt;]</b></p>	<p>响应</p> <p>若省略可选参数，则查询当前配置:</p> <p><b>+QHTTPCFG: "header",&lt;header_value&gt; [...]</b></p> <p><b>OK</b></p> <p>若指定可选参数，则配置 HTTP(S)请求头部行/头部字段:</p> <p><b>OK</b></p> <p>或者</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="auth"[,&lt;username:password&gt;]</b></p>	<p>响应</p> <p>若省略可选参数，则查询当前配置:</p> <p><b>+QHTTPCFG: "auth",&lt;username:password&gt;</b></p> <p><b>OK</b></p> <p>若指定可选参数，则配置用户名及密码:</p> <p><b>OK</b></p> <p>或者</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="http2.0"[,&lt;http2_enable&gt;]</b></p>	<p>响应</p> <p>若省略可选参数，则查询当前配置:</p> <p><b>+QHTTPCFG: "http2.0",&lt;http2_enable&gt;</b></p> <p><b>OK</b></p> <p>若指定可选参数，则禁用或启用 HTTP 2.0:</p> <p><b>OK</b></p> <p>或者</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="form/option"[,&lt;name&gt;[,&lt;file_name&gt;[,&lt;content_type&gt;]]]</b></p>	<p>响应</p> <p>若省略可选参数，则查询当前配置:</p> <p><b>+QHTTPCFG: "form/option",&lt;name&gt;,&lt;file_name&gt;,&lt;content_type&gt; [...]</b></p> <p><b>OK</b></p>

	<p>若指定任意可选参数，则配置 form/option 的参数值：</p> <p><b>OK</b></p> <p>或者</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>设置命令</p> <p><b>AT+QHTTPCFG="reset"</b></p>	<p>响应</p> <p><b>OK</b></p> <p>或者</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
最大响应时间	-
特性说明	<p>该命令立即生效；</p> <p>参数配置不保存。</p>

## 参数

<b>&lt;contextID&gt;</b>	整型。PDP 上下文 ID。范围：1~15；默认值：1。
<b>&lt;request_header&gt;</b>	<p>整型。禁用或启用自定义 HTTP(S)请求头信息。</p> <p><u>0</u>     禁用</p> <p>1     启用</p>
<b>&lt;response_header&gt;</b>	<p>整型。禁用或启用输出 HTTP(S)响应头信息。</p> <p><u>0</u>     禁用</p> <p>1     启用</p>
<b>&lt;sslctxID&gt;</b>	<p>整型。HTTP(S)的 SSL 上下文 ID。范围：0~5；默认值：1。通过 <b>AT+QSSLCFG</b> 配置 SSL 参数，详情请参考文档 [2]。</p>
<b>&lt;content_type&gt;</b>	<p>整型。HTTP(S)体的数据类型。</p> <p><u>0</u>     application/x-www-form-urlencoded</p> <p>1     text/plain</p> <p>2     application/octet-stream</p> <p>3     multipart/form-data</p> <p>4     application/json</p> <p>5     image/jpeg</p>
<b>&lt;auto_outrsp&gt;</b>	<p>整型。禁用或启用自动输出 HTTP(S)响应头信息。若启用自动输出 HTTP(S)响应头信息，<b>AT+QHTTPREAD</b> 和 <b>AT+QHTTPREADFILE</b> 会执行失败。</p> <p><u>0</u>     禁用</p> <p>1     启用</p>
<b>&lt;closedind&gt;</b>	<p>整型。禁用或启用上报 HTTP(S)会话关闭 URC <b>+QHTTPURC: "closed"</b>。</p> <p><u>0</u>     禁用</p> <p>1     启用</p>
<b>&lt;url_string&gt;</b>	字符串类型。HTTP(S)的 URL。
<b>&lt;header_value&gt;</b>	字符串类型。HTTP(S)请求头部行/头部字段名，如："Content-type: text/plain" 或 "Content-type"。
<b>&lt;username:password&gt;</b>	字符串类型。用户名及密码。格式为："username:password"。



<name>	字符串类型。form/option 的 name 值。
<file_name>	字符串类型。form/option 的 filename 值。
<content_type>	字符串类型。form/option 的 content-type 值。
<http2_enable>	整型。禁用或启用 HTTP 2.0。 0 禁用 1 启用
<err>	错误代码。详情请参考第5章。

## 备注

EC600Z-CN、EC800Z-CN 和 EC801E-CN 模块不支持 AT+QHTTPCFG="http2.0"。

## 2.3.2. AT+QHTTPURL 设置 HTTP(S)服务器 URL

HTTP(S)服务器的 URL 必须以 http://或 https://开头，表示访问 HTTP 或 HTTPS 服务器。

AT+QHTTPURL 设置 HTTP(S)服务器 URL	
测试命令 AT+QHTTPURL=?	响应 +QHTTPURL: (支持的<URL_length>列表),(支持的<timeout>列表)  OK
查询命令 AT+QHTTPURL?	响应 [+QHTTPURL: <URL>]  OK
设置命令 AT+QHTTPURL=<URL_length>[,<timeout>]	响应 若参数格式正确，且不发送 HTTP(S) GET/POST/PUT 请求： CONNECT  TA 切换到透传模式（即数据模式），即可输入 URL。当输入数据的总大小达到<URL_length>，TA 将切换回命令模式并上报以下结果： OK  若输入时间达到 <timeout>，但接收的 URL 长度小于 <URL_length>，TA 将切换回命令模式并上报以下结果： +CME ERROR: <err>  若参数格式不正确或发生其他错误： +CME ERROR: <err>

最大响应时间	取决于<timeout>
特性说明	该命令立即生效； 参数配置不保存。

## 参数

<URL_length>	整型。URL 长度。范围：1~2048；单位：字节。
<timeout>	整型。URL 的最大输入时间。范围：1~65535；默认值：60；单位：秒。
<err>	错误代码。详情请参考第 5 章。

### 2.3.3. AT+QHTTPGET 发送 GET 请求到 HTTP(S)服务器

根据在 **AT+QHTTPCFG="requestheader",<request\_header>** 命令中配置的参数 <request\_header>，**AT+QHTTPGET** 设置命令有两种形式。若 <request\_header> 为 1，发送 **AT+QHTTPGET** 后，若 125 秒内输出 **CONNECT**，则表示 HTTP(S)服务器连接成功；若 125 秒内未上报 **CONNECT**，将输出 **+CME ERROR: <err>**。

发送 **AT+QHTTPGET** 设置命令上报 **OK** 后，需要等待一段时间（参考最大响应时间）才会输出 **URC +QHTTPGET: <err>[,<httprspcode>[,<content\_length>]]**。

**+QHTTPGET: <err>[,<httprspcode>[,<content\_length>]]** 中，仅当 <err> 为 0 时，才上报 <httprspcode>。若 HTTP(S)响应头信息包括 Content-Length，将上报<content\_length>信息。

#### AT+QHTTPGET 发送 GET 请求到 HTTP(S)服务器

测试命令 <b>AT+QHTTPGET=?</b>	响应 <b>+QHTTPGET: (支持的&lt;rsptime&gt;列表),(支持的&lt;data_length&gt;列表),(支持的&lt;input_time&gt;列表)</b>  <b>OK</b>
设置命令 若<request_header>为 0（禁用自定义 HTTP(S)请求头信息） <b>AT+QHTTPGET[=&lt;rsptime&gt;]</b>	响应 若参数格式正确且无其他错误发生： <b>OK</b>  模块收到来自 HTTP(S)服务器的响应后，将上报以下 URC： <b>+QHTTPGET: &lt;err&gt;[,&lt;httprspcode&gt;[,&lt;content_length&gt;]]</b>  若参数格式不正确或有其他错误发生： <b>+CME ERROR: &lt;err&gt;</b>
设置命令 若<request_header>为 1（启用自定义 HTTP(S)请求头信息）	响应 若 HTTP(S)服务器连接成功： <b>CONNECT</b>

AT+QHTTPGET=<rsptime>,<data_length>[,<input_time>]	<p>TA 切换到透传模式（即数据模式），即可输入 HTTP(S) GET 请求头信息。当输入数据的总大小达到&lt;data_length&gt;，TA 将切换回命令模式并上报以下结果：</p> <p><b>OK</b></p> <p>模块收到来自 HTTP(S)服务器的响应后，将上报以下 URC：</p> <p><b>+QHTTPGET: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b></p> <p>若输入时间达到&lt;input_time&gt;，但收到的数据长度小于&lt;data_length&gt;，TA 将切换回命令模式并上报以下结果：</p> <p><b>+QHTTPGET: &lt;err&gt;</b></p> <p>若参数格式不正确或有其他错误发生：</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
最大响应时间	取决于<rsptime>
特性说明	该命令立即生效； 参数配置不保存。

## 参数

<rsptime>	整型。上报 <b>OK</b> 后，此参数为 HTTP(S) GET 响应 <b>+QHTTPGET: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b> 的最大响应时间。范围：1~65535；默认值：60；单位：秒。
<data_length>	整型。HTTP(S)请求信息的长度，请求信息包括 HTTP(S)请求头信息和 HTTP(S)请求体。范围：1~65535；单位：字节。
<input_time>	整型。HTTP(S)请求信息的最大输入时间。范围：1~65535；默认值：60；单位：秒。
<httprcode>	HTTP(S)服务器响应代码。详情请参考第6章。
<request_header>	整型。禁用或启用自定义 HTTP(S)请求头信息。 0 禁用 1 启用
<content_length>	整型。HTTP(S)响应体长度。单位：字节。
<err>	错误代码。参考第5章。

### 2.3.4. AT+QHTTPGETEX 发送范围 GET 请求到 HTTP(S)服务器

与读取文件类似，MCU 可以通过 **AT+QHTTPGETEX** 从 HTTP(S)服务器获取指定位置和指定长度的数据，仅在 **AT+QHTTPCFG="requestheader",0** 时，方可执行该命令。之后，HTTP(S)服务器将始终以 **206** 代码响应范围 GET 请求。

## AT+QHTTPGETEX 发送范围 GET 请求到 HTTP(S)服务器

测试命令 <b>AT+QHTTPGETEX=?</b>	响应 <b>+QHTTPGETEX: (支持的&lt;rsptime&gt;列表),&lt;start_position&gt;,&lt;read_len&gt;</b>  <b>OK</b>
设置命令 <b>AT+QHTTPGETEX=&lt;rsptime&gt;,&lt;start_position&gt;,&lt;read_len&gt;</b>	响应 若参数格式正确且无其他错误发生: <b>OK</b>  模块收到来自 HTTP(S)服务器响应后, 将上报以下 URC: <b>+QHTTPGET: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b>  若参数格式不正确或有其他错误发生: <b>+CME ERROR: &lt;err&gt;</b>
最大响应时间	取决于<rsptime>
特性说明	该命令立即生效; 参数配置不保存。

### 参数

<rsptime>	整型。上报 <b>OK</b> 后, 此参数为 HTTP(S) GET 响应 <b>+QHTTPGET: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b> 的最大响应时间。范围: 1~65535; 默认值: 60; 单位: 秒。
<start_postion>	整型。HTTP(S)客户端需要 GET 数据的初始位置。
<read_len>	整型。HTTP(S)客户端需要 GET 数据的长度。
<httprcode>	HTTP(S)服务器响应代码。详情请参考第 6 章。
<content_length>	整型。HTTP(S)响应体长度。单位: 字节。
<err>	错误代码。详情请参考第 5 章。

### 备注

当第二次执行 **AT+QHTTPGETEX** 时, 若<start\_position>设置为-1, 分段下载从上次下载结束的位置开始下载。

## 2.3.5. AT+QHTTPPOST 通过 UART/USB 发送 POST 请求到 HTTP(S)服务器

该命令用于通过 UART/USB 发送 POST 请求到 HTTP(S) 服务器。根据在 **AT+QHTTPCFG="requestheader" [,<request\_header>]** 命令中配置的 <request\_header>, **AT+QHTTPPOST** 设置命令有两种形式。若<request\_header>为 0, 则通过 UART/USB 口输入 POST 请

求体；若<request\_header>为 1，则通过 UART/USB 口输入 POST 请求头信息和 POST 请求体。

发送 **AT+QHTTPPOST** 后，若 125 秒内返回 **CONNECT**，则表示 HTTP(S)服务器连接成功；若 125 秒内未返回 **CONNECT**，将输出**+CME ERROR: <err>**。

上报 **OK** 后，需要等待一段时间（参考最大响应时间）才会输出 URC **+QHTTPPOST: <err>[,<httprcode>[,<content\_length>]]**。

## AT+QHTTPPOST 通过 UART/USB 发送 POST 请求到 HTTP(S)服务器

测试命令 <b>AT+QHTTPPOST=?</b>	响应 <b>+QHTTPPOST: (支持的&lt;data_length&gt;列表),(支持的&lt;input_time&gt;列表),(支持的&lt;rsptime&gt;列表)</b>  <b>OK</b>
设置命令 若<request_header>为 0（禁用自定义 HTTP(S)请求头信息） <b>AT+QHTTPPOST=&lt;data_length&gt;[,&lt;input_time&gt;,&lt;rsptime&gt;]</b>	响应 若参数格式正确，HTTP(S)服务器连接成功且 HTTP(S)请求头信息发送完成： <b>CONNECT</b>  TA 切换到透传模式（即数据模式），即可输入 HTTP(S) POST 请求体。当输入数据总大小达到<data_length>，TA 将切换回命令模式并上报以下结果： <b>OK</b>  模块收到来自 HTTP(S)服务器响应后，将上报以下 URC： <b>+QHTTPPOST: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b>  若输入时间达到<input_time>，但收到的数据长度小于<data_length>，TA 将切换回命令模式并上报以下结果： <b>+QHTTPPOST: &lt;err&gt;</b>  若参数格式不正确或其他错误发生： <b>+CME ERROR: &lt;err&gt;</b>
设置命令 若<request_header>为 1（启用自定义 HTTP(S)请求头信息） <b>AT+QHTTPPOST=&lt;data_length&gt;[,&lt;input_time&gt;,&lt;rsptime&gt;]</b>	响应 若参数格式正确且 HTTP(S)服务器连接成功： <b>CONNECT</b>  TA 切换到透传模式（即数据模式），即可输入 HTTP(S) POST 请求体和 HTTP(S) POST 请求头信息。当输入数据总大小达到<data_length>，TA 将切换回命令模式并上报以下结果： <b>OK</b>  模块收到来自 HTTP(S)服务器响应后，将上报以下 URC：

	<p><b>+QHTTPPOST: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b></p> <p>若输入时间达到 <b>&lt;input_time&gt;</b>，但收到的数据长度小于 <b>&lt;data_length&gt;</b>，TA 将切换回命令模式并上报以下结果：</p> <p><b>+QHTTPPOST: &lt;err&gt;</b></p> <p>若参数格式不正确或有其他错误发生：</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
最大响应时间	取决于网络和 <b>&lt;rsptime&gt;</b>
特性说明	该命令立即生效； 参数配置不保存。

## 参数

<b>&lt;data_length&gt;</b>	整型。若 <b>&lt;request_header&gt;</b> 为0,表示 POST 请求体长度;若 <b>&lt;request_header&gt;</b> 为 1,表示 HTTP(S)请求信息的长度,包括 HTTP(S) POST 请求头信息和 HTTP(S) POST 请求体。范围:1~1024000;单位:字节。
<b>&lt;input_time&gt;</b>	整型。POST 请求体或 HTTP(S)请求信息的最大输入时间。范围:1~65535;默认值:60;单位:秒。
<b>&lt;rsptime&gt;</b>	整型。上报 OK 后,此参数为 HTTP(S) POST 响应 <b>+QHTTPPOST: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b> 的最大输出时间。范围:1~65535;默认值:60。单位:秒。
<b>&lt;httprcode&gt;</b>	HTTP(S)服务器响应代码。详情请参考第6章。
<b>&lt;request_header&gt;</b>	整型。禁用或启用自定义 HTTP(S)请求头信息。 0 禁用 1 启用
<b>&lt;content_length&gt;</b>	整型。HTTP(S)响应体长度。单位:字节。
<b>&lt;err&gt;</b>	错误代码。详情请参考第5章。

### 2.3.6. AT+QHTTPPOSTFILE 通过文件发送 POST 请求到 HTTP(S)服务器

该命令用于通过文件发送 POST 请求到 HTTP(S)服务器。根据在 **AT+QHTTPCFG="requestheader" [,<request\_header>]**命令中配置的**<request\_header>**，**AT+QHTTPPOSTFILE** 操作的文件有两种形式。若**<request\_header>**为 0，文件系统的文件将为 POST 请求体；若**<request\_header>**为 1，文件系统的文件将为 POST 请求头信息和 POST 请求体。

执行 **AT+QHTTPPOSTFILE** 后，模块会上报 URC **+QHTTPPOSTFILE: <err>[,<httprcode>[,<content\_length>]]**表示命令执行结果。仅当**<err>**为 0 时，才上报**<httprcode>**。

上报 OK 后，需要等待一段时间（参考最大响应时间）才会输出 URC **+QHTTPPOSTFILE: <err>[,<httprcode>[,<content\_length>]]**信息。

## AT+QHTTPPOSTFILE 通过文件发送 POST 请求到 HTTP(S)服务器

测试命令 <b>AT+QHTTPPOSTFILE=?</b>	响应 <b>+QHTTPPOSTFILE: &lt;file_name&gt;,(支持的&lt;rsptime&gt;列表), (支持的&lt;post_mode&gt;列表) OK</b>
设置命令 <b>AT+QHTTPPOSTFILE=&lt;file_name&gt;[, &lt;rsptime&gt;,&lt;post_mode&gt;]</b>	响应 若参数格式正确且 HTTP(S)服务器连接成功: <b>OK</b>  当模块收到来自 HTTP(S)服务器响应后: <b>+QHTTPPOSTFILE: &lt;err&gt;[,&lt;httprspcode&gt;[,&lt;content_length&gt;]]</b>  若参数格式不正确或有其他错误发生: <b>+CME ERROR: &lt;err&gt;</b>
最大响应时间	取决于<rsptime>
特性说明	该命令立即生效; 参数配置不保存。

### 参数

<b>&lt;file_name&gt;</b>	字符串类型。文件名称。最大长度为 64 字节。
<b>&lt;rsptime&gt;</b>	整型。上报 <b>OK</b> 后，此参数为 HTTP(S) POST 响应 <b>+QHTTPPOSTFILE: &lt;err&gt;[,&lt;httprspcode&gt;[,&lt;content_length&gt;]]</b> 的最大输出时间。范围：1~65535；默认值：60；单位：秒。
<b>&lt;httprspcode&gt;</b>	HTTP(S)服务器响应代码。详情请参考第 6 章。
<b>&lt;request_header&gt;</b>	整型。禁用或启用自定义 HTTP(S)请求头信息。若<request_header>为 1，指定的文件必须包含 HTTP(S)请求头信息。 0 禁用 1 启用
<b>&lt;content_length&gt;</b>	整型。HTTP(S)响应体长度。单位：字节。
<b>&lt;post_mode&gt;</b>	整型。HTTP(S)发送文件的模式。 0 直接发送文件内容 1 记录保存文件，暂不发送，等待和<post_mode>=2 时配置的文件一起发送 2 发送文件，连同<post_mode>=1 时保存的文件一起发送（只支持两个文件一起发送）
<b>&lt;err&gt;</b>	错误代码。详情请参考第 5 章。

### 2.3.7. AT+QHTTTPUT 通过 UART/USB 发送 PUT 请求到 HTTP(S)服务器

该命令用于通过 UART/USB 发送 PUT 请求到 HTTP(S) 服务器。根据在 **AT+QHTTPCFG="requestheader",<request\_header>** 命令中配置的 <request\_header>，**AT+QHTTTPUT** 设置命令有两种形式。若<request\_header>为 0，则通过 UART/USB 口输入 PUT 请求体；若<request\_header>为 1，则通过 UART/USB 口输入 PUT 请求头信息和 PUT 请求体。

发送 **AT+QHTTTPUT** 后，若 125 秒内返回 **CONNECT**，则表示 HTTP(S)服务器连接成功；若 125 秒内未返回 **CONNECT**，将输出**+CME ERROR: <err>**。

上报 **OK** 后，需要等待一段时间（参考最大响应时间）才会输出 **URC +QHTTTPUT: <err>[,<httprcode>[,<content\_length>]]**。

AT+QHTTTPUT 通过 UART/USB 发送 PUT 请求到 HTTP(S)服务器	
测试命令 <b>AT+QHTTTPUT=?</b>	<p>响应</p> <p><b>+QHTTTPUT: (支持的&lt;data_length&gt;列表),(支持的&lt;input_time&gt;列表),(支持的&lt;rsptime&gt;列表)</b></p> <p><b>OK</b></p>
设置命令 若<request_header>为 0（禁用自定义 HTTP(S)请求头信息） <b>AT+QHTTTPUT=&lt;data_length&gt;[,&lt;input_time&gt;,&lt;rsptime&gt;]</b>	<p>响应</p> <p>若参数格式正确，HTTP(S)服务器连接成功且 HTTP(S)请求头信息发送完成： <b>CONNECT</b></p> <p>TA 切换到透传模式（即数据模式），即可输入 HTTP(S) PUT 请求体。当输入数据总大小达到&lt;data_length&gt;，TA 将返回命令模式并上报以下结果： <b>OK</b></p> <p>模块收到来自 HTTP(S)服务器响应后，将上报以下 URC： <b>+QHTTTPUT: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b></p> <p>若输入时间达到&lt;input_time&gt;，但收到的数据长度小于&lt;data_length&gt;，TA 将返回命令模式并上报以下结果： <b>+QHTTTPUT: &lt;err&gt;</b></p> <p>若参数格式不正确或其他错误发生： <b>+CME ERROR: &lt;err&gt;</b></p>
设置命令 若<request_header>为 1（启用自定义 HTTP(S)请求头信息） <b>AT+QHTTTPUT=&lt;data_length&gt;[,&lt;input_time&gt;,&lt;rsptime&gt;]</b>	<p>响应</p> <p>若参数格式正确且 HTTP(S)服务器连接成功： <b>CONNECT</b></p> <p>TA 切换到透传模式（即数据模式），即可输入 HTTP(S) PUT</p>



	<p>请求体和 HTTP(S) PUT 请求头信息。当输入数据总大小达到 <b>&lt;data_length&gt;</b>，TA 将切换回命令模式并上报以下结果： <b>OK</b></p> <p>模块收到来自 HTTP(S)服务器响应后，将上报以下 URC： <b>+QHHTTPPUT: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b></p> <p>若输入时间达到 <b>&lt;input_time&gt;</b>，但收到的数据长度小于 <b>&lt;data_length&gt;</b>，TA 将切换回命令模式并上报以下结果： <b>+QHHTTPPUT: &lt;err&gt;</b></p> <p>若参数格式不正确或有其他错误发生： <b>+CME ERROR: &lt;err&gt;</b></p>
最大响应时间	取决于网络和 <b>&lt;rsptime&gt;</b>
特性说明	<p>该命令立即生效；</p> <p>参数配置不保存。</p>

## 参数

<b>&lt;data_length&gt;</b>	整型。若 <b>&lt;request_header&gt;</b> 为 0，表示 PUT 请求体长度；若 <b>&lt;request_header&gt;</b> 为 1，表示 HTTP(S)请求信息的长度，包括 HTTP(S) PUT 请求头信息和 HTTP(S) PUT 请求体。范围：1~1024000；单位：字节。
<b>&lt;input_time&gt;</b>	整型。PUT 请求体或 HTTP(S)请求信息的最大输入时间。范围：1~65535；默认值：60；单位：秒。
<b>&lt;rsptime&gt;</b>	整型。上报 <b>OK</b> 后，此参数为 HTTP(S) PUT 响应 <b>+QHHTTPPUT: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b> 的最大输出时间。范围：1~65535；默认值：60；单位：秒。
<b>&lt;httprcode&gt;</b>	HTTP(S)服务器响应代码。详情请参考第6章。
<b>&lt;request_header&gt;</b>	<p>整型。禁用或启用自定义 HTTP(S)请求头信息。</p> <p>0 禁用</p> <p>1 启用</p>
<b>&lt;content_length&gt;</b>	整型。HTTP(S)响应体长度。单位：字节。
<b>&lt;err&gt;</b>	错误代码。详情请参考第5章。

### 2.3.8. AT+QHHTTPPUTFILE 通过文件发送 PUT 请求到 HTTP(S)服务器

该命令用于通过文件发送 PUT 请求到 HTTP(S) 服务器。根据在 **AT+QHHTTPCFG="requestheader"[,<request\_header>]** 命令中配置的 **<request\_header>**，**AT+QHHTTPPUTFILE** 操作的文件有两种形式。若**<request\_header>**为 0，文件系统的文件将为 PUT 请求体；若**<request\_header>**为 1，文件系统的文件将为 PUT 请求头信息和 PUT 请求体。

执行 **AT+QHTTPPUTFILE** 后，模块会上报 URC **+QHTTPPUTFILE: <err>[,<httprcode>[,<content\_length>]]** 表示命令执行结果。仅当 **<err>** 为 0 时，才上报 **<httprcode>**。

上报 **OK** 后，需要等待一段时间（参考最大响应时间）才会输出 URC **+QHTTPPUTFILE: <err>[,<httprcode>[,<content\_length>]]** 信息。

## AT+QHTTPPUTFILE 通过文件发送 PUT 请求到 HTTP(S)服务器

测试命令 <b>AT+QHTTPPUTFILE=?</b>	响应 <b>+QHTTPPUTFILE: &lt;file_name&gt;,(支持的&lt;rsptime&gt;列表),(支持的&lt;put_mode&gt;列表)</b>  <b>OK</b>
设置命令 <b>AT+QHTTPPUTFILE=&lt;file_name&gt;[,&lt;rsptime&gt;,&lt;put_mode&gt;]</b>	响应 若参数格式正确且 HTTP(S)服务器连接成功： <b>OK</b>  当模块收到来自 HTTP(S)服务器响应后： <b>+QHTTPPUTFILE: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b>  若参数格式不正确或其他错误发生： <b>+CME ERROR: &lt;err&gt;</b>
最大响应时间	取决于 <b>&lt;rsptime&gt;</b>
特性说明	该命令立即生效； 参数配置不保存。

## 参数

<b>&lt;file_name&gt;</b>	字符串类型。文件名称。最大长度为 64 字节。
<b>&lt;rsptime&gt;</b>	整型。上报 <b>OK</b> 后，此参数为 HTTP(S) PUT 响应 <b>+QHTTPPUTFILE: &lt;err&gt;[,&lt;httprcode&gt;[,&lt;content_length&gt;]]</b> 的最大输出时间。范围：1~65535；默认值：60；单位：秒。
<b>&lt;httprcode&gt;</b>	HTTP(S)服务器响应代码。详情请参考第6章。
<b>&lt;request_header&gt;</b>	整型。禁用或启用自定义 HTTP(S)请求头信息。若 <b>&lt;request_header&gt;</b> 为 1，指定的文件必须包含 HTTP(S)请求头信息。 0 禁用 1 启用
<b>&lt;content_length&gt;</b>	整型。HTTP(S)响应体长度。单位：字节。
<b>&lt;put_mode&gt;</b>	整型。HTTP(S)发送文件的模式。 0 直接发送文件内容

- 1 记录保存文件，暂不发送，等待和<put\_mode>=2 时配置的文件一起发送
- 2 发送文件，连同<put\_mode>=1 时保存的文件一起发送（只支持两个文件一起发送）

<err> 错误代码。详情请参考第5章。

### 2.3.9. AT+QHTTPREAD 通过 UART/USB 读取 HTTP(S)服务器响应信息

发送 HTTP(S) GET/POST/PUT 请求后，可使用 **AT+QHTTPREAD** 通过 UART/USB 端口从 HTTP(S) 服务器读取 HTTP(S)响应信息。必须接收**+QHTTPGET: <err>[,<httprspcode>[,<content\_length>]]**、**+QHTTPPOST: <err>[,<httprspcode>[,<content\_length>]]**、**+QHTTPPOSTFILE: <err>[,<httprspcode>[,<content\_length>]]**和**+QHTTPPUTFILE: <err>[,<httprspcode>[,<content\_length>]]**信息后才能执行 **AT+QHTTPREAD**。

#### AT+QHTTPREAD 通过 UART/USB 读取 HTTP(S)服务器响应信息

测试命令 <b>AT+QHTTPREAD=?</b>	响应 <b>+QHTTPREAD: (支持的&lt;wait_time&gt;列表)</b>  <b>OK</b>
设置命令 <b>AT+QHTTPREAD[=&lt;wait_time&gt;]</b>	响应 若参数格式正确： <b>CONNECT</b> <输出 HTTP(S)响应信息> <b>OK</b>  当响应信息读取完成或两个数据包接收间隔时间达到 <b>&lt;wait_time&gt;</b> : <b>+QHTTPREAD: &lt;err&gt;</b>  若参数格式不正确或有其他错误发生： <b>+CME ERROR: &lt;err&gt;</b>
最大响应时间	取决于 <b>&lt;wait_time&gt;</b>
特性说明	该命令立即生效； 参数配置不保存。

#### 参数

<b>&lt;wait_time&gt;</b>	整型。接收两个数据包之间的最大间隔时间。范围：1~65535；默认值：60；单位：秒。
<b>&lt;err&gt;</b>	错误代码。详情请参考第5章。

### 2.3.10. AT+QHTTPREADFILE 通过文件读取 HTTP(S)服务器响应信息

发送 HTTP(S) GET/POST/PUT 请求后，可以使用 **AT+QHTTPREADFILE** 通过文件从 HTTP(S)服务器读取 HTTP(S)响应信息，必须接收**+QHTTPGET: <err>[,<httprspcode>[,<content\_length>]]**、**+QHTTPPOST: <err>[,<httprspcode>[,<content\_length>]]**、**+QHTTPPOSTFILE: <err>[,<httprspcode>[,<content\_length>]]**、**+QHTTPPUT: <err>[,<httprspcode>[,<content\_length>]]**或**+QHTTPPUTFILE: <err>[,<httprspcode>[,<content\_length>]]**信息后才能执行 **AT+QHTTPREADFILE**。

AT+QHTTPREADFILE 通过文件读取 HTTP(S)服务器响应信息	
测试命令 <b>AT+QHTTPREADFILE=?</b>	响应 <b>+QHTTPREADFILE: &lt;file_name&gt;,(支持的&lt;wait_time&gt;列表)</b>  <b>OK</b>
设置命令 <b>AT+QHTTPREADFILE=&lt;file_name&gt;[,&lt;wait_time&gt;]</b>	响应 若参数格式正确: <b>OK</b>  当响应信息读取完成或两个数据包接收间隔时间达到<wait_time>: <b>+QHTTPREADFILE: &lt;err&gt;</b>  若参数格式不正确或有其他错误发生: <b>+CME ERROR: &lt;err&gt;</b>
最大响应时间	取决于<wait_time>
特性说明	该命令立即生效; 参数配置不保存。

#### 参数

<wait_time>	整型。接收两个数据包之间的最大间隔时间。范围：1~65535；默认值：60；单位：秒。
<file_name>	字符串类型。文件名称。最大长度为 64 字节。
<err>	错误代码。详情请参考第5章。

### 2.3.11. AT+QHTTPSTOP 取消 HTTP(S)请求

MCU 可通过该命令取消 HTTP(S) GET/POST/PUT 请求，断开与 HTTP(S)的会话连接。

AT+QHTTPSTOP 取消 HTTP(S)请求	
测试命令 <b>AT+QHTTPSTOP=?</b>	响应 <b>OK</b>

执行命令 <b>AT+QHTTPSTOP</b>	响应 若参数格式正确且无其他错误发生： <b>OK</b>  若参数格式不正确或有其他错误发生： <b>+CME ERROR: &lt;err&gt;</b>
最大响应时间	10 秒
特性说明	该命令立即生效； 参数配置不保存。

## 参数

<err>	错误代码。详情请参考第5章。
-------	----------------

## 3 举例

### 3.1. 访问 HTTP 服务器

#### 3.1.1. 发送 HTTP GET 请求并读取响应信息

以下举例说明如何发送 HTTP GET 请求，启用 HTTP 响应头信息输出以及读取 HTTP GET 响应。

```
//发送 HTTP GET 响应示例
AT+QHTTPCFG="contextid",1           //配置PDP上下文ID为1。
OK
AT+QHTTPCFG="responseheader",1      //启用输出 HTTP 响应头信息。
OK
AT+QIACT?                            //查询 PDP 上下文状态。
OK                                   //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QICSGP=1,1,"UNINET","",1        //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置
AT+CFUN=1,1 使配置生效。

OK
AT+QIACT=1                           //激活 PDP 上下文 1。
OK                                   //激活成功。
AT+QIACT?                            //查询 PDP 上下文状态。
+QIACT: 1,1,1,"10.7.157.1"

OK
AT+QHTTPURL=23,80                    //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
HTTP://www.example.com/              //输入长度为 23 个字节的 URL。（此 URL 仅为示例。请根据实际情况输入正确的 URL。）

OK
AT+QHTTPGET=80                       //发送 HTTP GET 请求，最大响应时间为 80 秒。
OK

+QHTTPGET: 0,200,601710              //若 HTTP 响应头信息中包括 Content-Length，则将上报
<content_length>信息。

//读取 HTTP GET 响应信息示例。

//方法 1：读取 HTTP 响应信息并通过 UART 口将其输出。
```

```

AT+QHTTPREAD=80                                //读取 HTTP 响应信息并通过 UART 口将其输出。HTTP 会话关
                                                    闭的最长等待时间为 80 秒。

CONNECT
HTTP/1.1 200 OK <CR><LF>                            //HTTP 响应头信息和响应体。
Server: nginx<CR><LF>
Date: Tue, 12 Sep 2017 05:57:29 GMT<CR><LF>
Content-Type: text/html<CR><LF>
Content-Length: 601710<CR><LF>
Connection: close<CR><LF>
Last-Modified: Tue, 12 Sep 2017 05:54:48 GMT<CR><LF>
Vary: Accept-Encoding<CR><LF>
Expires: Tue, 12 Sep 2017 05:58:28 GMT<CR><LF>
Cache-Control: max-age=60<CR><LF>
X-Powered-By: shci_v1.03<CR><LF>
Age: 1<CR><LF>
.....<CR><LF>                                     //此处响应信息省略。
<CR><LF>
<body>
OK

+QHTTPREAD: 0                                    //成功读取 HTTP 响应头信息和响应体。

//方法 2：读取 HTTP 响应信息并将其储存到 UFS 文件中。

AT+QHTTPREADFILE="UFS:1.txt",80                //读取 HTTP 响应头信息和响应体并将其储存到 UFS:1.txt。
                                                    HTTP 会话关闭的最长等待时间为 80 秒。

OK

+QHTTPREADFILE: 0                                //成功储存 HTTP 响应头信息和响应体。

```

### 3.1.2. 发送 HTTP POST 请求并读取响应信息

#### 3.1.2.1. 通过 UART/USB 获取 POST 请求体

以下举例说明如何发送 HTTP POST 请求,通过 UART 口读取 POST 请求体以及如何读取 HTTP POST 响应信息。

```

AT+QHTTPCFG="contextid",1                        //配置PDP上下文ID为1。
OK
AT+QIACT?                                         //查询 PDP 上下文状态。
OK                                                  //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QICSGP=1,1,"UNINET","",1                    //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置
                                                    AT+CFUN=1,1 使配置生效。
OK

```

```

AT+QIACT=1 //激活 PDP 上下文 1。
OK //激活成功。
AT+QIACT? //查询 PDP 上下文状态。
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPURL=59,80 //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
http://example.com/DEMOWebServices2.8/Service.asmx/example? //输入长度为 59 个字节的 URL。
//（此 URL 仅为示例，请根据实际情况输入正确的 URL。）

OK
AT+QHTTPPOST=20,80,80 //发送 HTTP POST 请求，通过 UART 获取 POST 请求体。POST
//请求体的最大输入时间和响应时间皆为 80 秒。

CONNECT
Message=HelloQuectel //输入长度为 20 个字节的 POST 请求体。（此 POST 请求体仅
//为示例，请根据实际情况输入正确的 POST 请求体。）

OK

+QHTTPPOST: 0,200,177 //若 HTTP 响应头信息包含 Content-Length，则将上报
//<content_length>信息。

AT+QHTTPREAD=80 //读取 HTTP 响应信息并通过 UART 口将其输出。HTTP 会话关
//闭的最长等待时间为 80 秒。

CONNECT
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="HTTps://example.com/webservices2.3">Message='HelloQuectel' ASCII:72 101 108
108 111 81 117 101 99 116 101 108 </string> //输出 HTTP 响应信息。
OK

+QHTTPREAD: 0 //成功输出 HTTP 响应信息。

```

### 3.1.2.2. 从文件系统获取 POST 请求体

以下举例说明如何发送 HTTP POST 请求，通过文件系统读取 POST 请求体，以及将 HTTP POST 响应储存在文件系统。

```

AT+QHTTPCFG="contextid",1 //配置PDP上下文ID为1。
OK
AT+QIACT? //查询 PDP 上下文状态。
OK //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QICSGP=1,1,"UNINET","",1 //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置
//AT+CFUN=1,1 使配置生效。

OK
AT+QIACT=1 //激活 PDP 上下文 1。

```



```

OK //激活成功。
AT+QIACT? //查询 PDP 上下文状态。
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPURL=59,80 //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
http://example.com/DEMOWebServices2.8/Service.asmx/example? //输入长度为 59 个字节的 URL。
//（此 URL 仅为示例，请根据实际情况输入正确的 URL。）

OK
//来自 UFS 文件的 POST 请求信息，读取 HTTP 响应信息并将其储存到 UFS 文件中。
AT+QHTTPPOSTFILE="UFS:2.txt",80 //发送 HTTP POST 请求，从 UFS:2.txt 中获取 POST 请求体，
//最大响应时间为 80 秒。

OK

+QHTTPPOSTFILE: 0,200,177 //HTTP POST 请求发送成功后，方可执行
//AT+QHTTPREADFILE。
AT+QHTTPREADFILE="UFS:3.txt",80 //读取 HTTP 响应信息并将其储存到 UFS:3.txt。HTTP 会话关
//闭的最长等待时间为 80 秒。

OK

+QHTTPREADFILE: 0 //HTTP 响应信息储存成功。

```

### 3.1.3. 发送 HTTP PUT 请求并读取响应信息

#### 3.1.3.1. 通过 UART/USB 获取 PUT 请求体

以下举例说明如何发送 HTTP PUT 请求，通过 UART 口读取 PUT 请求体以及如何读取 HTTP PUT 响应信息。

```

AT+QHTTPCFG="contextid",1 //配置PDP上下文ID为1。
OK
AT+QIACT? //查询 PDP 上下文状态。
OK //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QICSGP=1,1,"UNINET","",1 //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置
//AT+CFUN=1,1 使配置生效。

OK
AT+QIACT=1 //激活 PDP 上下文 1。
OK //激活成功。
AT+QIACT? //查询 PDP 上下文状态。
+QIACT: 1,1,1,"172.22.86.226"

```

```

OK
AT+QHTTTPURL=59,80 //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
http://example.com/DEMOWebServices2.8/Service.asmx/example? //输入长度为 59 个字节的 URL。
//（此 URL 仅为示例，请根据实际情况输入正确的 URL。）

OK
AT+QHTTTPUT=20,80,80 //发送 HTTP PUT 请求，通过 UART 获取 PUT 请求体。PUT 请
//求体的最大输入时间和响应时间皆为 80 秒。

CONNECT
Message=HelloQuectel //输入长度为 20 个字节的 PUT 请求体。（此 PUT 请求体仅为示
//例，请根据实际情况输入正确的 PUT 请求体。）

OK

+QHTTTPUT: 0,200,177 //若 HTTP 响应头信息包含 Content-Length，则将上报
//<content_length>信息。

AT+QHTTTPREAD=80 //读取 HTTP 响应信息并通过 UART 口将其输出。HTTP 会话关
//闭的最长等待时间为 80 秒。

CONNECT
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="HTTPs://example.com/webservices2.3">Message='HelloQuectel' ASCII:72 101 108
108 111 81 117 101 99 116 101 108 </string> //输出 HTTP 响应信息。

OK

+QHTTTPREAD: 0 //成功输出 HTTP 响应信息。

```

### 3.1.3.2. 从文件系统获取 PUT 请求体

以下举例说明如何发送 HTTP PUT 请求，通过文件系统读取 PUT 请求体，以及将 HTTP PUT 响应储存在文件系统。

```

AT+QHTTPCFG="contextid",1 //配置PDP上下文ID为1。
OK
AT+QIACT? //查询 PDP 上下文状态。
OK //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QICSGP=1,1,"UNINET","",1 //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置
//AT+CFUN=1,1 使配置生效。

OK
AT+QIACT=1 //激活 PDP 上下文 1。
OK //激活成功。
AT+QIACT? //查询 PDP 上下文状态。
+QIACT: 1,1,1,"172.22.86.226"

OK

```

```

AT+QHTTPURL=59,80 //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
http://example.com/DEMOWebServices2.8/Service.asmx/example? //输入长度为 59 个字节的 URL。
                                     (此 URL 仅为示例，请根据实际情况输入正确的 URL。)

OK
//来自 UFS 文件的 PUT 请求信息，读取 HTTP 响应信息并将其储存到 UFS 文件中。
AT+QHTTPPUTFILE="UFS:2.txt",80 //发送 HTTP PUT 请求，从 UFS:2.txt 中获取 PUT 请求体，最大响应时间为 80 秒。

OK

+QHTTPPUTFILE: 0,200,177 //HTTP PUT 请求发送成功后，方可执行 AT+QHTTPREADFILE。

AT+QHTTPREADFILE="UFS:3.txt",80 //读取 HTTP 响应信息并将其储存到 UFS:3.txt。HTTP 会话关闭的最长等待时间为 80 秒。

OK

+QHTTPREADFILE: 0 //HTTP 响应信息储存成功。

```

## 3.2. 访问 HTTPS 服务器

### 3.2.1. 发送 HTTPS GET 请求并读取响应信息

以下举例说明如何发送 HTTPS GET 请求，启用 HTTPS 响应头信息输出以及如何读取 HTTPS GET 响应信息。

```

//发送HTTPS GET请求示例
AT+QHTTPCFG="contextid",1 //配置PDP上下文ID为1。
OK
AT+QHTTPCFG="responseheader",1 //启用输出 HTTPS 响应头信息。
OK
AT+QIACT? //查询 PDP 上下文状态。
OK //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QICSGP=1,1,"UNINET","",1 //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置 AT+CFUN=1,1 使配置生效。
OK
AT+QIACT=1 //激活 PDP 上下文 1。
OK //激活成功。
AT+QIACT? //查询 PDP 上下文状态。
+QIACT: 1,1,1,"10.7.157.1"
OK
AT+QHTTPCFG="sslctxid",1 //设置 SSL 上下文 ID 为 1。

```

```

OK
AT+QSSLCFG="sslversion",1,1 //设置 SSL 版本为 1，表示 TLSV1.0。
OK
AT+QSSLCFG="ciphersuite",1,0x0005 //设置 SSL 加密套件为 0x0005，表示 RC4-SHA。
OK
AT+QSSLCFG="secclevel",1,0 //设置 SSL 验证级别为 0，表示无身份验证模式。
OK
AT+QHTTPURL=22,80 //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
HTTP://www.example.cn/ //输入长度为 22 个字节的 URL。（此 URL 仅为示例，请根据实际情况输入正确的 URL。）

OK
AT+QHTTPGET=80 //发送 HTTPS GET 请求，最大响应时间为 80 秒。
OK

+QHTTPGET: 0,200,21472 //若 HTTPS 响应头信息包含 Content-Length，则将上报
<content_length>信息。

//读取 HTTPS 响应信息示例

//方法 1：读取 HTTPS 响应信息并将其通过 UART 输出。

AT+QHTTPREAD=80 //读取 HTTPS 响应信息并将其通过 UART 输出。HTTP 会话关
闭的最长等待时间为 80 秒。
CONNECT //HTTPS 响应头信息和响应体。
HTTP/1.1 200 OK<CR><LF>
Server: Tengine/2.1.0<CR><LF>
Date: Tue, 12 Sep 2017 05:54:34 GMT <CR><LF>
Content-Type: text/html; charset=utf-8<CR><LF>
Content-Length: 21451<CR><LF>
Connection: keep-alive <CR><LF>
..... <CR><LF> //此处响应信息省略。
<CR><LF>
<body>
OK

+QHTTPREAD: 0 //HTTPS 响应头信息和响应体读取成功。

//方法 2：读取 HTTPS 响应信息并将其储存在 UFS 文件中。

AT+QHTTPREADFILE="UFS:4.txt",80 //读取 HTTPS 响应头信息和响应体并将其储存在 UFS:4.txt。
HTTPS 会话关闭的最长等待时间为 80 秒。

OK

+QHTTPREADFILE: 0 //HTTPS 响应头信息和响应体储存成功。

```

### 3.2.2. 发送 HTTPS POST 请求并读取响应信息

#### 3.2.2.1. 从 UART/USB 获取 POST 请求体

以下举例说明如何发送 HTTPS POST 请求,通过 UART 读取 POST 请求体以及如何读取 HTTPS POST 响应信息。

```

AT+QHTTPCFG="contextid",1 //配置PDP上下文ID为1。
OK
AT+QIACT? //查询 PDP 上下文状态。
OK
AT+QICSGP=1,1,"UNINET","",1 //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置 AT+CFUN=1,1 使配置生效。
OK //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QIACT=1 //激活 PDP 上下文 1。
OK //激活成功。
AT+QIACT? //查询 PDP 上下文状态。
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPCFG="sslctxid",1 //设置 SSL 上下文 ID 为 1。
OK
AT+QSSLCFG="sslversion",1,1 //设置 SSL 版本为 1，表示 TLSV1.0。
OK
AT+QSSLCFG="ciphersuite",1,0x0005 //设置 SSL 加密套件为 0x0005，表示 RC4-SHA。
OK
AT+QSSLCFG="secclevel",1,2 //设置 SSL 验证级别为 2，表示需通过 AT+QFUPL 上传 CA 证书，客户端证书和客户端私钥。
OK
AT+QSSLCFG="cacert",1,"UFS:cacert.pem"
OK
AT+QSSLCFG="clientcert",1,"UFS:clientcert.pem"
OK
AT+QSSLCFG="clientkey",1,"UFS:clientkey.pem"
OK
AT+QHTTPURL=45,80 //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
http://www.example.cn/example/example/example //输入长度为 45 个字节的 URL。（此 URL 仅为示例，请根据实际情况输入正确的 URL。）
OK
AT+QHTTPPOST=48,80,80 //发送 HTTPS POST 请求，通过 UART 获取 POST 请求体。POST 请求体的最大输入时间和响应时间为 80 秒。
CONNECT
Message=1111&Appleqty=2222&Orangeqty=3333&find=1 //输入长度为 48 个字节的 POST 请求体。

```

（此 POST 请求体仅为示例，请根据实际情况输入正确的 POST 请求体。）

OK

+QHTTPPOST: 0,200,285

//若 HTTPS 响应头信息包含 Content-Length，则将上报 <content\_length> 信息。

AT+QHTTPREAD=80

//读取 HTTPS 响应信息并将其通过 UART 输出。HTTPS 会话关闭的最长等待时间为 80 秒。

CONNECT

//HTTPS 响应信息读取成功。

<html>

<head>

<title>Quectel's Auto Parts - Order Results</title>

</head>

<body>

<h1>Quectel's Auto Parts</h1>

<h2>Order Results</h2>

<p>Order processed at 02:49, 27th December</p><p>Your order is as follows: </p>1111 message<br />2222 apple<br />3333 orange<br /></body>

</html>

OK

+QHTTPREAD: 0

//HTTPS 响应信息输出成功。

### 3.2.2.2. 从文件系统获取 POST 请求体

以下举例说明如何发送 HTTPS POST 请求，从文件系统读取 POST 请求体以及如何将 HTTPS POST 响应信息储存到文件系统。

AT+QHTTPCFG="contextid",1

//配置PDP上下文ID为1。

OK

AT+QIACT?

//查询 PDP 上下文状态。

OK

//仅返回 OK 表示无被激活的 PDP 上下文。

AT+QICSGP=1,1,"UNINET","",1

//配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置 AT+CFUN=1,1 使配置生效。

OK

AT+QIACT=1

//激活 PDP 上下文 1。

OK

//激活成功。

AT+QIACT?

//查询 PDP 上下文状态。

+QIACT: 1,1,1,"172.22.86.226"

OK

AT+QHTTPCFG="sslctxid",1

//设置 SSL 上下文 ID 为 1。

OK

```

AT+QSSLCFG="sslversion",1,1           //设置 SSL 版本为 1，表示 TLSV1.0。
OK
AT+QSSLCFG="ciphersuite",1,0x0005     //设置 SSL 加密套件为 0x0005，表示 RC4-SHA。
OK
AT+QSSLCFG="secllevel",1,2            //设置 SSL 验证级别为 2，表示需通过 AT+QFUPL 上传 CA 证
书，客户端证书和客户端私钥。
OK
AT+QSSLCFG="cacert",1,"UFS:cacert.pem"
OK
AT+QSSLCFG="clientcert",1,"UFS:clientcert.pem"
OK
AT+QSSLCFG="clientkey",1,"UFS:clientkey.pem"
OK
AT+QHTTPURL=45,80                     //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
http://www.example.cn/example/example/example //输入长度为 45 个字节的 URL。（此 URL 仅为
示例，请根据实际情况输入正确的 URL。）
OK
//POST 请求信息来自 UFS 文件，读取 HTTPS 响应信息并将其储存在 UFS 文件中。
AT+QHTTPPOSTFILE="UFS:5.txt",80       //发送 HTTPS POST 请求，从 UFS:5.txt 获取 POST 请求体，
最大响应时间为 80 秒。
OK
+QHTTPPOSTFILE: 0,200,177             //HTTPS POST 请求发送成功后，方可执行
AT+QHTTPREADFILE。
AT+QHTTPREADFILE="UFS:6.txt",80       //读取 HTTPS 响应信息并将其储存在 UFS:6.txt。HTTPS 会
话关闭的最长等待时间为 80 秒。
OK
+QHTTPREADFILE: 0                     //HTTPS 响应信息储存成功。

```

### 3.2.3. 发送 HTTPS PUT 请求并读取响应信息

#### 3.2.3.1. 从 UART/USB 获取 PUT 请求体

以下举例说明如何发送 HTTPS PUT 请求，通过 UART 读取 PUT 请求体以及如何读取 HTTPS PUT 响应信息。

```

AT+QHTTPCFG="contextid",1           //配置PDP上下文ID为1。
OK
AT+QIACT?                            //查询 PDP 上下文状态。
OK                                   //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QICSGP=1,1,"UNINET","",1        //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设

```

```

置 AT+CFUN=1,1 使配置生效。

OK
AT+QIACT=1 //激活 PDP 上下文 1。
OK //激活成功。
AT+QIACT? //查询 PDP 上下文状态。
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPCFG="sslctxid",1 //设置 SSL 上下文 ID 为 1。
OK
AT+QSSLCFG="sslversion",1,1 //设置 SSL 版本为 1，表示 TLSV1.0。
OK
AT+QSSLCFG="ciphersuite",1,0x0005 //设置 SSL 加密套件为 0x0005，表示 RC4-SHA。
OK
AT+QSSLCFG="secllevel",1,2 //设置 SSL 验证级别为 2，表示需通过 AT+QFUPL 上传
CA 证书，客户端证书和客户端私钥。

OK
AT+QSSLCFG="cacert",1,"UFS:cacert.pem"
OK
AT+QSSLCFG="clientcert",1,"UFS:clientcert.pem"
OK
AT+QSSLCFG="clientkey",1,"UFS:clientkey.pem"
OK
AT+QHTTPURL=45,80 //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
http://www.example.cn/example/example/example //输入长度为 45 个字节的 URL。（此 URL 仅为
示例，请根据实际情况输入正确的 URL。）
OK
AT+QHTTPPUT=48,80,80 //发送 HTTPS PUT 请求，通过 UART 获取 PUT 请求体。PUT 请求
体的最大输入时间和响应时间为 80 秒。

CONNECT
Message=1111&Appleqty=2222&Orangeqty=3333&find=1 //输入长度为 48 个字节的 PUT 请求体。（此
PUT 请求体仅为示例，请根据实际情况输入
正确的 PUT 请求体。）

OK

+QHTTPPUT: 0,200,285 //若 HTTPS 响应头信息包含 Content-Length，则将上报
<content_length>信息。
AT+QHTTPREAD=80 //读取 HTTPS 响应信息并将其通过 UART 输出。HTTP 会话关闭
的最长等待时间为 80 秒。

CONNECT //HTTPS 响应信息读取成功。
<html>
<head>
<title>Quectel's Auto Parts - Order Results</title>

```



```

</head>
<body>
<h1>Quectel's Auto Parts</h1>
<h2>Order Results</h2>

<p>Order processed at 02:49, 27th December</p><p>Your order is as follows: </p>1111
message<br />2222  apple<br />3333 orange<br /></body>
</html>
OK

+QHTTPREAD: 0                                //HTTPS 响应信息输出成功。

```

### 3.2.3.2. 从文件系统获取 PUT 请求体

以下举例说明如何发送 HTTPS PUT 请求，从文件系统读取 PUT 请求体以及如何将 HTTPS PUT 响应信息储存在文件系统。

```

AT+QHTTPCFG="contextid",1                    //配置PDP上下文ID为1。
OK
AT+QIACT?                                     //查询 PDP 上下文状态。
OK                                             //仅返回 OK 表示无被激活的 PDP 上下文。
AT+QICSGP=1,1,"UNINET","",1                 //配置 PDP 上下文为 1，APN 为中国联通的"UNINET"。需设置
AT+CFUN=1,1 使配置生效。
OK
AT+QIACT=1                                    //激活 PDP 上下文 1。
OK                                             //激活成功。
AT+QIACT?                                     //查询 PDP 上下文状态。
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPCFG="sslctxid",1                     //设置 SSL 上下文 ID 为 1。
OK
AT+QSSLCFG="sslversion",1,1                 //设置 SSL 版本为 1，表示 TLSV1.0。
OK
AT+QSSLCFG="ciphersuite",1,0x0005           //设置 SSL 加密套件为 0x0005，表示 RC4-SHA。
OK
AT+QSSLCFG="secllevel",1,2                  //设置 SSL 验证级别为 2，表示需通过 AT+QFUPPL 上传 CA 证
书，客户端证书和客户端私钥。
OK
AT+QSSLCFG="cacert",1,"UFS:cacert.pem"
OK
AT+QSSLCFG="clientcert",1,"UFS:clientcert.pem"
OK
AT+QSSLCFG="clientkey",1,"UFS:clientkey.pem"

```

```

OK
AT+QHTTPURL=45,80 //设置要访问的 URL，设置超时时间为 80 秒。
CONNECT
http://www.example.cn/example/example/example //输入长度为 45 个字节的 URL。（此 URL 仅为示例，
请根据实际情况输入正确的 URL。）

OK
//PUT 请求信息来自 UFS 文件，读取 HTTPS 响应信息并将其储存在 UFS 文件中。
AT+QHTTPPUTFILE="UFS:5.txt",80 //发送 HTTPS PUT 请求，从 UFS:5.txt 获取 PUT 请求体，最
大响应时间为 80 秒。

OK

+QHTTPPUTFILE: 0,200,177 //HTTPS PUT 请求发送成功后，方可执行
AT+QHTTPREADFILE。
AT+QHTTPREADFILE="UFS:6.txt",80 //读取 HTTPS 响应信息并将其储存在 UFS:6.txt。HTTP 会话
关闭的最长等待时间为 80 秒。

OK

+QHTTPREADFILE: 0 //HTTPS 响应信息储存成功。

```

# 4 常见问题处理

## 4.1. HTTP(S) AT 命令执行失败

执行 HTTP(S) AT 命令后，若返回**+CME ERROR: <err>**，请检查是否插入了 USIM 卡，并注意执行**AT+CPIN?**后是否返回了**+CPIN: READY**。

## 4.2. PDP 激活失败

若使用 **AT+QIACT** 激活 PDP 失败，请检查以下配置：

1. 查询是否通过 **AT+CGATT?**附着了 PS 域。若未附着 PS 域，请先执行 **AT+CGATT=1** 附着 PS 域。
2. 通过 **AT+CGREG?**查询 PS 域状态并确保已注册了 PS 域。
3. 通过 **AT+QICSGP** 查询 PDP 上下文参数，并确保已设置了指定 PDP 上下文的 APN。
4. 确保指定 PDP 上下文 ID 既不被 PPP 使用，也不通过 **AT+CGACT** 激活。

若上述配置无误，使用 **AT+QIACT** 激活 PDP 上下文仍失败，请重启模块。重启模块后，请至少检查三次上述提及的配置，每次间隔 10 分钟，以避免频繁重启模块。

## 4.3. DNS 解析失败

执行 **AT+QHTTPGET**、**AT+QHTTPGETEX**、**AT+QHTTPPOST**、**AT+QHTTPPOSTFILE**、**AT+QHTTPPUT** 和 **AT+QHTTPPUTFILE** 后，若返回**+CME ERROR: 714**（714: HTTP(S) DNS 错误），请检查以下配置：

1. 确认 HTTP(S)服务器域名有效。
2. 通过 **AT+QIACT?**查询 PDP 上下文状态，确保成功激活了指定的 PDP 上下文。
3. 通过 **AT+QIDNSCFG** 查询 DNS 服务器地址，确保 DNS 服务器的地址不是**"0.0.0.0"**。

若 DNS 服务器的地址为"0.0.0.0"，有两种解决方法：

1. 通过 **AT+QIDNSCFG** 重新分配一个有效 DNS 服务器地址。
2. 通过 **AT+QIDEACT** 去激活 PDP 上下文，通过 **AT+QIACT** 重新激活 PDP 上下文。

## 4.4. GET/POST/PUT 请求发送失败

若执行 **AT+QHTTPGET**、**AT+QHTTPGETEX**、**AT+QHTTPPOST**、**AT+QHTTPPOSTFILE**、**AT+QHTTPPUT** 和 **AT+QHTTPPUTFILE** 失败，请检查以下配置：

1. 确保通过 **AT+QHTTPURL** 输入的 URL 有效且可以访问。
2. 确保指定服务器支持 GET/POST/PUT 相关命令。
3. 确保 PDP 上下文已激活。

若上述配置无误，通过 **AT+QHTTPGET**、**AT+QHTTPGETEX**、**AT+QHTTPPOST**、**AT+QHTTPPOSTFILE**、**AT+QHTTPPUT** 和 **AT+QHTTPPUTFILE** 发送 GET/POST/PUT 请求仍失败，请通过 **AT+QIDEACT** 去激活 PDP 上下文，并通过 **AT+QIACT** 重新激活 PDP 上下文。若激活 PDP 上下文失败，详细信息请参考第 4.2 章解决此问题。

## 4.5. 响应信息读取失败

通过 **AT+QHTTPREAD** 和 **AT+QHTTPREADFILE** 读取响应信息之前，首先要执行 **AT+QHTTPGET**、**AT+QHTTPGETEX**、**AT+QHTTPPOST**、**AT+QHTTPPOSTFILE**、**AT+QHTTPPUT** 或 **AT+QHTTPPUTFILE** 并接收以下 URC：

```
+QHTTPGET: <err>[,<httprcode>[,<content_length>]]
+QHTTPPOST: <err>[,<httprcode>[,<content_length>]]
+QHTTPPOSTFILE: <err>[,<httprcode>[,<content_length>]]
+QHTTPPUT: <err>[,<httprcode>[,<content_length>]]
+QHTTPPUTFILE: <err>[,<httprcode>[,<content_length>]]
```

执行 **AT+QHTTPREAD** 和 **AT+QHTTPREADFILE** 时，若发生错误，如上报**+CME ERROR: 717**(717: HTTP(S) socket 读取错误)，请通过 **AT+QHTTPGET**、**AT+QHTTPGETEX**、**AT+QHTTPPOST**、**AT+QHTTPPOSTFILE**、**AT+QHTTPPUT** 和 **AT+QHTTPPUTFILE** 向 HTTP(S)服务器重新发送 HTTP(S) GET/POST/PUT 请求；若请求仍发送失败，请参考第 4.4 章解决此问题。

# 5 错误代码

<err>表示与移动设备或网络有关的错误代码。相关详细信息参考下表。

表 3：错误代码列表

<err>错误代码	英文描述	中文描述
0	Operation successful	操作成功
701	HTTP(S) unknown error	HTTP(S)未知错误
702	HTTP(S) timeout	HTTP(S)超时
703	HTTP(S) busy	HTTP(S)繁忙
705	HTTP(S) no GET/POST requests	HTTP(S)无 GET/POST/PUT 请求
706	HTTP(S) network busy	HTTP(S)网络繁忙
707	HTTP(S) network open failed	HTTP(S)网络连接失败
708	HTTP(S) network no configuration	HTTP(S)网络未配置
709	HTTP(S) network deactivated	去激活 HTTP(S)网络
710	HTTP(S) network error	HTTP(S)网络错误
711	HTTP(S) URL error	HTTP(S) URL 错误
712	HTTP(S) empty URL	HTTP(S)空 URL
713	HTTP(S) IP address error	HTTP(S) IP 地址错误
714	HTTP(S) DNS error	HTTP(S) DNS 错误
715	HTTP(S) socket create error	HTTP(S) socket 创建错误
716	HTTP(S) socket connect error	HTTP(S) socket 连接错误
717	HTTP(S) socket read error	HTTP(S) socket 读取错误
718	HTTP(S) socket write error	HTTP(S) socket 写入错误

719	HTTP(S) socket closed	HTTP(S) socket 关闭
720	HTTP(S) data encode error	HTTP(S)数据编码错误
721	HTTP(S) data decode error	HTTP(S)数据解码错误
722	HTTP(S) read timeout	HTTP(S)读取超时
723	HTTP(S) response failed	HTTP(S)响应失败
724	Incoming call busy	来电忙
725	Voice call busy	语音通话忙
726	Input timeout	输入超时
727	Wait data timeout	数据等待超时
728	Wait HTTP(S) response timeout	HTTP(S)等待响应超时
729	Memory allocation failed	内存分配不足
730	Invalid parameter	无效参数
731	Wondblock	被阻塞
732	SSL Handshake Failed	SSL 握手失败

# 6 HTTP(S)响应错误代码

<httprspcode>表示 HTTP(S)服务器响应代码。相关详细信息参见下表。

表 4: HTTP(S)响应代码列表

<httprspcode>响应代码	英文描述	中文描述
200	OK	成功
206	Partial Content	部分内容
403	Forbidden	禁止
404	Not found	未发现
409	Conflict	冲突
411	Length required	需输入长度
500	Internal server error	内部服务器错误

# 7 附录 参考文档及术语缩写

表 5: 参考文档

文档名称
[1] Quectel_LTE_Standard(E)系列_TCP(IP)_应用指导
[2] Quectel_LTE_Standard(E)系列_SSL_应用指导

表 6: 术语缩写

缩写	英文全称	中文全称
APN	Access Point Name	接入点名称
CA	Certification Authority	数字证书认证系统
DNS	Domain Name Server	域名服务器
DTR	Data Terminal Ready	数据终端就绪
HTTP	Hypertext Transfer Protocol	超文本传输协议
HTTPS	Hypertext Transfer Protocol Secure	超文本传输安全协议
ID	Identification	身份识别
IP	Internet Protocol	网际互连协议
LTE	Long-Term Evolution	长期演进
MCU	Microprogrammed Control Unit	微程序控制器
PDP	Packet Data Protocol	分组数据协议
PPP	Point-to-Point Protocol	点对点协议
PS	Packet Switch	分组交换
SSL	Security Socket Layer	安全 Socket 层



TA	Terminal Adapter	终端适配器
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器
URI	Uniform Resource Identifier	统一资源标示符
URL	Uniform Resource Locator	统一资源定位符
URC	Unsolicited Result Code	未经请求的结果代码
USB	Universal Serial Bus	通用串行总线
UFS	UNIX File System	UNIX 文件系统
USIM	Universal Subscriber Identity Module	通用用户识别模块