Wireless Module Expert

EC20 HTTP AT Commands Manual

LTE Module Series

Rev. EC20_HTTP_AT_Commands_Manual_V1.0

Date: 2015-07-02

www.quectel.com

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Office 501, Building 13, No.99, Tianzhou Road, Shanghai, China, 200233
Tel: +86 21 5108 6236
Mail: info@quectel.com

Or our local office, for more information, please visit:
http://www.quectel.com/support/salesupport.aspx

For technical support, to report documentation errors, please visit:
http://www.quectel.com/support/techsupport.aspx
Or Email: Support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2015-07-02	Chris PENG	Initial

Contents

Table Index

1 Introduction

EC20 provides HTTP application to HTTP(S) server. This document is a reference guide to all the AT commands defined for HTTP.

1.1. The Process of Using HTTP AT Commands

Through EC20 TCPIP AT command, you can configure PDP context, activate and deactivate PDP context, query PDP context status (Please refer to *Quectel_EC20_TCPIP_AT_Commands_Manual*), and through EC20 HTTP AT command, you can send HTTP GET/POST request to HTTP server, and read HTTP response from HTTP server. The general process is as follows:

Step 1: Configure the <apn>, <username>, <password> and other parameters of PDP context by AT+QICSGP (Please refer to *Quectel_EC20_TCPIP_AT_Commands_Manual*). If QoS settings need to be updated, configure them by the commands AT+CGQMIN, AT+CGEQMIN, AT+CGQREQ and AT+CGEQREQ (For more details, please refer to *Quectel_EC20_AT_Commands_Manual*).

Step 2: Activate PDP context by AT+QIACT, then the assigned IP address can be queried by AT+QIACT? (Please refer to *Quectel_EC20_TCPIP_AT_Commands_Manual*).

Step 3: Set PDP context ID and SSL context ID by AT+QHTTPCFG command.

Step 4: Configure SSL context parameters by AT+QSSLCFG command (Please refer to *Quectel_EC20_SSL_AT_Commands_Manual*).

Step 5: Set HTTP URL by AT+QHTTPURL command.

Step 6: Send HTTP request. AT+QHTTPGET command can be used for sending HTTP GET request, AT+QHTTPPOST or QHTTPOSTFILE command can be used for sending HTTP POST request.

Step 7: Read HTTP response information by AT+QHTTPREAD or QHTTPREADFILE command.

Step 8: Deactivate PDP context by AT+QIDEACT command. (For details, please refer to *Quectel_EC20_TCPIP_AT_Commands_Manual*).

1.2. Description of HTTP Header

1.2.1. Customize HTTP Request Header

HTTP request header is filled by the modem automatically. If you want to customize the HTTP request header, you can set <request_header> as 1 by AT+QHTTPCFG command, and then input HTTP request header as below:

1. Follow HTTP header syntax.
2. The value of URI in HTTP request line and the "Host:" header must be in line with the URL configured by AT+QHTTPURL command.
3. The HTTP request header must end with <CR><LF>.

The following example shows a valid HTTP POST request header:

```
POST/processorder.php HTTP/1.1<CR><LF>
Host: 220.180.239.201:8011<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: 42<CR><LF>
<CR><LF>
```

1.2.2. Output HTTP Response Header

HTTP response header will not output automatically. If you want to obtain HTTP response header information, you can set <response header> to 1 by AT+QHTTPCFG command, and the HTTP response header will be output with HTTP response body when you execute AT+QHTTPREAD or AT+QHTTPREADFILE command.

1.3. Description of Data Mode

The mode of the COM port includes AT command mode and data mode. The difference between them is as follows:

1) In AT command mode, the data input via COM port will be treated as AT command.
2) In data mode, the data input via COM port will be treated as data.

You can exit from data mode by using "+++" or pulling up DTR (AT&D1 should be set).To prevent the "+++" from being misinterpreted as data, it should comply with the following sequence:

1) Do not input any character within 1 second before inputting "+++".
2) Input "+++" within 1 second, and no other characters can be input during this time.
3) Do not input any character wit in 1 second after "+++" has been input.

When you execute QHTTPURL, QHTTPHEAD, QHTTPOST and QHTTPREAD, the modem will enter data mode. If you are using "+++" or DTR to make module exit from online data mode, the executing procedure of these command will be interrupted before the executed result is returned, and you can't reenter online data mode by executing ATO command.

1.4. Error Handling

1.4.1. Executing HTTP AT Command Fails

If you execute HTTP AT command and receive response "ERROR" from modem, please check whether the SIM card is inserted and the result of "AT+CPIN?" is "+CPIN: READY".

1.4.2. PDP Activation Fails

If you failed to active PDP context by AT+QIACT command, please check the following aspects:

1. Query whether the PS domain is attached by AT+CGATT? command. If not, execute AT+CGATT=1 command to attach PS domain.
2. Query the PS domain status by AT+CGREG? command and make sure the PS domain has been registered.
3. Query the PDP context parameters by AT+QICSGP command and make sure the APN of specified PDP context has been set.
4. Make sure the specified PDP context ID is neither used by PPP nor activated by AT+CGACT command.
5 The module only supports three PDP contexts activated simultaneously, so you must make sure the amount of activated PDP context is less than 3.

If the result of above checking is OK, but the executing of AT+QIACT command still fails, please reboot the modem to resolve this issue. After rebooting the modem, please follow the above checking at least three times and each time at an interval of 10 minutes to avoid frequent rebooting of the module.

1.4.3. DNS Parse Fails

When executing AT+QHTTPGET, AT+QHTTPPOST and QHTTPPOSTFILE commands, if "+CME ERROR: 714" (714: HTTP DNS error) is returned, please check the following aspects:

1. Make sure the domain name of HTTP server is valid.
2. Query the status of PDP context via AT+QIACT? command to make sure the specified PDP context has been activated successfully.
3. Query the address of DNS server via AT+QIDNSCFG command to make sure the address of DNS server is not equal to "0.0.0.0".

If the DNS server address is equal to "0.0.0.0", there are two solutions:

1. Reassign a valid DNS server address by AT+QIDNSCFG command.
2. Deactivate the PDP context by AT+QIDEACT command, and re-activate the PDP context via AT+QIACT command.

1.4.4. Entering Online Data Mode Fails

When executing QHTTPURL, QHTTPGET, QHTTPPOST and QHTTPREAD commands, if "+CME ERROR: 704" (704: HTTP UART busy) is returned, please check whether there are other ports in online data mode, since the modem only supports one port in online data mode simultaneously.

If you encounter this issue during executing QHTTPURL, QHTTPGET, QHTTPPOST and QHTTPREAD command, please re-execute these commands after other ports exit from online data mode.

1.4.5. Sending GET/POST Request Fails

If you receive a failed result during executing AT+QHTTPGET, AT+QHTTPPOST and QHTTPPOSTFILE command, please check the following aspects:

1. Make sure the URL input via HTTPURL command is valid and can be accessed.
2. Make sure the specified server supports GET/POST command.
3. Make sure the PDP context has been activated successfully.

If the result of above checking is OK, but the executing of AT+QHTTPGET, QHTTPPOST and QHTTPPOSTFILE command still fails, please deactivate the PDP context via AT+QIDEACT and reactivate the PDP context via AT+QIACT to resolve this issue. If you encounter failed result during activating the PDP context, please refer to Chapter 1.4.2 to resolve it.

1.4.6. Reading Response Fails

Before reading response by AT+QHTTPREAD and QHTTPREADFILE command, you should execute AT+QHTTPGET, AT+QHTTPPOST and QHTTPPOSTFILE command and receive the URC information:

"+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]"/
"+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]"/
"+QHTTPPOSTFILE: <err>[,<httprspcode>,<content_length>]"

During executing AT+QHTTTPREAD and QHTTPREADFILE command, if you encounter some errors, such as: "+CME ERROR: 717" (717: HTTP read error), please resend HTTP GET/POST request to HTTP server by AT+QHTTPGET, AT+QHTTPPOST and QHTTPPOSTFILE command. If you encounter failed result during sending GET/POST request to HTTP server, please refer to Chapter 1.4.5 to resolve it.

2 Description of AT Command

2.1. AT+QHTTPCFG Configure Parameters for HTTP(S) Server

AT+QHTTPCFG is used to configure the parameters for HTTP service, including PDP context ID, customizing HTTP request header and outputting HTTP response header and querying SSL settings. If the write command only executes one parameter, it will query the current settings.

AT+QHTTPCFG Configure Parameters for HTTP(s) Server	
Test Command **AT+QHTTPCFG=?**	Response **+QHTTPCFG: "contextid",(1-16)** **+QHTTPCFG: "requestheader",(0,1)** **+QHTTPCFG: "responseheader",(0,1)** **+QHTTPCFG: "sslctxid",(0-5)** **+QHTTPCFG: "contenttype",(0-3)** **OK**
Write Command **AT+QHTTPCFG="contextid"[,<contextID>]**	Response If <contextID> is not omitted: **OK** or **+CME ERROR: <err>** Else, query the current settings: **+QHTTPCFG: "contextid",<contextID>** **OK**
Write Command **AT+QHTTPCFG="requestheader"[,<request_header>]**	Response If <request_header> is not omitted: **OK** or **+CME ERROR: <err>** Else, query the current settings: **+QHTTPCFG: "requestheader",<request_header>** **OK**

Write Command **AT+QHTTPCFG="responseheader"[,<response_header>]**	Response If <response header> is not omitted: **OK** or **+CME ERROR: <err>** Else, query the current settings: **+QHTTPCFG: "responseheader",<response_header>** **OK**
Write Command **AT+QHTTPCFG="sslctxid"[,<sslctxID>]**	Response If <sslctxID> is not omitted: **OK** or **+CME ERROR: <err>** Else, query the current settings: **+QHTTPCFG: "sslctxid",<sslctxID>** **OK**
Read Command **AT+QHTTPCFG?**	Response **+QHTTPCFG: "contextid",<contextID>** **+QHTTPCFG: "requestheader",<request_header>** **+QHTTPCFG: "responseheader",<response_header>** **+QHTTPCFG: "sslctxid",<sslctxID>** **+QHTTPCFG: "contenttype",<content_type>** **OK**

Parameter

<contextID>	Numeric type, indicates PDP context ID, range is 1-16, default value is 1
<request_header>	Numeric type, indicates whether to customize HTTP request header 0 Disable 1 Enable
<response_header>	Numeric type, indicates whether to output HTTP response header or not 0 Disable 1 Enable
<sslctxID>	Numeric type, indicates this SSL context ID will be used for HTTP. The range is 0-5, default value is 1. You should configure the SSL parameter by AT+QSSLCFG. (For details, please refer to the command AT+QSSLCFG in *Quectel_EC20_SSL_AT_Commands_Manual*.)
<content_type>	Numeric type, indicates the data type of HTTP body 0 application/x-www-form-urlencoded

	1 text/plain 2 application/octet-stream 3 multipart/form-data
<err>	The type of error, please refer to Chapter 4

2.2. AT+QHTTPURL Set HTTP(S) Server URL

To input URL, you must begin with “http://” or “https://”. If the URL begins with “http://”, it indicates you will access to a HTTP server. If the URL begins with “https://”, it indicates you will access to a HTTPS server.

AT+QHTTPURL Set HTTP(S) Server URL	
Test Command **AT+QHTTPURL=?**	Response **+QHTTPURL: (1-700),(1-65535)** **OK**
Write Command **AT+QHTTPURL=<URL_length>[,<timeout>]**	Response a) If format is right, and it is not sending HTTP GET/POST request at present: **CONNECT** TA switches to the transparent access mode, and the URL can be input. When the total size of the input data reaches <URL_length>, TA will return to command mode and report the following codes: **OK** If the <timeout> has reached, but the received length of URL is less than <URL_length>, TA will return to command mode and report the following code: **+CME ERROR: <err>** b) If parameter format is not right or other errors occur: **+CME ERROR: <err>**
Read Command **AT+QHTTPURL?**	Response **[+QHTTPURL: <URL><CR><LF>]** **OK**

Parameter

<URL_length>	Numeric type, indicates the length of URL, range is 1-700, unit: byte
<timeout>	Numeric type, indicates the maximum time in seconds to input URL, range is

	1-65535, default value is 60, unit: second
<err>	The type of error, please refer to Chapter 4

2.3. AT+QHTTPGET Send GET Request to HTTP(S) Server

According to the configured parameter <request_header> in the command AT+QHTTPCFG="requestheader"[,<request_header>], the HTTPGET write command has two different formats, the details are shown as below. Please note that if <request_header> equals to 1, after the command AT+QHTTPGET has been sent, the CONNECT may be output in 125s to indicate that the connection is successful. If it is not received during this time, the +CME ERROR: <err> will be output.

After the HTTPGET write command has been sent, it is suggested to wait a specific time (refer to the Maximum Response Time below) after the result code OK is reported for the URC response "+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]".

In the URC "+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]", the parameter <httprspcode> can only be reported when <err> equals to 0. If HTTP response header contains "CONTENT-LENGTH" information, the <content_length> information will be reported.

AT+QHTTPGET Send GET Request to HTTP(S) Server	
Test Command **AT+QHTTPGET=?**	Response **+QHTTPGET: (1-65535),(1-2048),(1-65535)** **OK**
If <request_header> equals to 0 (disable customizing HTTP request header) **AT+QHTTPGET[=<rsptime>]**	Response a) If parameter format is right and no other errors occur: **OK** When modem has received response from HTTP server, it will report the following URC: **+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]** b) If parameter format is not right or other errors occur: **+CME ERROR: <err>**
If <request_header> equals to 1 (enable customizing HTTP request header) **AT+QHTTPGET=<rsptime>,<data_length>[,<input_time>]**	Response a) If connect HTTP server successfully: **CONNECT** TA switches to the transparent access mode, and the HTTP GET request header can be input. When the total size of the input data reaches <data length>, TA will return to command mode and report the following codes:

	OK When modem has received response from HTTP server, it will report the following URC: **+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]** If the <input_time> has reached, but the length of received data is less than <data length>, TA will return to command mode and report the following code: **+CME ERROR: <err>** b) If parameter format is not right or other errors occur: **+CME ERROR: <err>**
Maximum Response Time	Determined by <rsptime>

Parameter

<rsptime>	Numeric type, the range is 1-65535. The default value is 60, unit: second. It is used to configure the timeout for the HTTP GET response URC "+QHTTPGET: <err>[,<httprspcode>,<content_length>]" to report after the result code OK is returned
<data_length>	Numeric type, indicates the length of HTTP request information, including HTTP request header and HTTP request body, range is 1-2048, unit: byte
<input_time>	Numeric type, indicates the maximum time in seconds to input HTTP request information, range is 1-65535. The default value is 60, unit: second
<err>	Please refer to Chapter 4
<httprspcode>	Please refer to Chapter 5
<request_header>	Please refer to Chapter 2.1
<content_length>	Numeric type, indicates the length of HTTP response body, unit: byte.

2.4. AT+QHTTPPOST Send POST Request to HTTP(S) Server by UART/USB

You can send HTTP POST request via AT+QHTTPPOST command. According to the configured parameter <request_header> in the command AT+QHTTPCFG="requestheader"[,<request_header>], the HTTPPOST write command has two different formats, if you set <request_header> to 0, you should input post body by UART/USB, else if you set <request_header> to 1, you should input post header and body by UART/USB.

After the command AT+QHTTPPOST has been sent, the CONNECT may be output in 125s to indicate the connection is successful, if it isn't received during this time, the +CME ERROR: <err> will be output.

It is suggested to wait a specific time (refer to the Maximum Response Time below) after the result code OK is reported for the URC response "+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]".

AT+QHTTPPOST Send POST Request to HTTP(S) Server by UART/USB	
Test Command **AT+QHTTPPOST=?**	Response **+QHTTPPOST: (1-1024000),(1-65535),(1-65535)** **OK**
If <request_header> equals to 0 (disable customizing HTTP request header) **AT+QHTTPPOST=<data_length>[,<input_time>,<rsptime>]**	Response a) If parameter format is right and HTTP server is connected successfully and HTTP request header is sent completely, it will prompt you to input body: **CONNECT** TA switches to the transparent access mode, and the HTTP POST body can be input. When the total size of the input data reaches <data_length>, TA will return to command mode and report the following codes: **OK** When modem has received response from HTTP server, it will report the following URC: **+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]** If the <input_time> has reached, but the received length of data is less than <data_length>, TA will return to command mode and report the following code: **+CME ERROR: <err>** b) If parameter format is not right or other errors occur: **+CME ERROR: <err>**
If <request_header> equals to 1 (enable customizing HTTP request header) **AT+QHTTPPOST=<data_length>[,<input_time>,<rsptime>]**	Response a) If parameter format is right and HTTP server is connected successfully: **CONNECT** TA switches to the transparent access mode, and the HTTP POST header and body can be input. When the total size of the input data reaches <data_length>, TA will return to command mode and report the following codes: **OK**

	When modem has received response from HTTP server, it will report the following URC: **+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]** If the <input_time> has reached, but the length of received data is less than <data_length>, TA will return to command mode and report the following code: **+CME ERROR: <err>** b) If parameter format is not right or other errors occur: **+CME ERROR: <err>**
Maximum Response Time	Determined by network and <rsptime>

Parameter

<data_length>	Numeric type, if <request_header> is 0, it indicates the length of post body, and if <request_header> is 1, it indicates the length of HTTP request information, including HTTP request header and HTTP request body, range is 1-1024000, unit: byte
<input_time>	Numeric type, indicates the maximum time in seconds to input post body or HTTP request information, range is 1-65535. The default value is 60, unit: second
<rsptime>	Numeric type, range is 1-65535. The default value is 60, unit: second. It is used to configure the timeout for the HTTP POST response URC "+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]" to report after the result code OK is returned
<err>	Please refer to Chapter 4
<httprspcode>	Please refer to Chapter 5
<request_header>	Please refer to Chapter 2.1
<content_length>	Numeric type, indicates the length of HTTP response body, unit: byte

2.5. AT+QHTTPPOSTFILE Send POST Request to HTTP(S) Server by File

You also can send HTTP POST request in file via AT+QHTTPPOSTFILE command. According to the <request_header> in the command AT+QHTTPCFG="requestheader"[,<request_header>], the file operated through HTTPPOSTFILE command has two different formats, if set <request_header> to 0, the file in file system will be post body, else if set <request_header> to 1, the file in file system will be post header and body.

The modem will report information "+QHTTPPOSTFILE: <err>[,<httprspcode>[,<content_length>]]" to indicate the executing result of AT+QHTTPPOSFILE command. The parameter <httprspcode> can only be reported when <err> equals to 0.

It is suggested to wait a specific time (refer to the Maximum Response Time below) after the result code OK is reported for the URC response "+QHTTPPOSTFILE: <err>[,<httprspcode>[,<content_length>]]".

AT+QHTTPPOSTFILE	**Send POST Request to HTTP(S) Server by File**
Test Command **AT+QHTTPPOSTFILE=?**	Response **+QHTTPPOSTFILE: <file_name>,(1-65535)** **OK**
Write Command **AT+QHTTPPOSTFILE=<file_name>[,<rsptime>]** If <request_header> equals to 1, the specified file must contain HTTP request header information.	Response a) If parameter format is right and HTTP server is connected successfully: **OK** When modem has received response from HTTP server, it will report the following URC: **+QHTTPPOSTFILE: <err>[,<httprspcode>,<content_length>]** b) If parameter format is not right or other errors occur: **+CME ERROR: <err>**
Maximum Response Time	Determined by <rsptime>

Parameter

<file_name>	String type, file name, the max length of file name is 80, unit: byte
<rsptime>	Numeric type, the range is 1-65535. The default value is 60, unit: second. It is used to configure the timeout for the HTTP POST response URC "+QHTTPPOSTFILE: <err>[,<httprspcode>,<content_length>]" to report after the result code OK is returned.
<err>	Please refer to Chapter 4
<httprspcode>	Please refer to Chapter 5
<request_header>	Please refer to Chapter 2.1
<content_length>	Numeric type, indicates the length of HTTP response body

2.6. AT+QHTTPREAD Read Response from HTTP(S) Server by UART/USB

After sending HTTP GET/POST request, you can retrieve HTTP response information from HTTP server to UART/USB via AT+QHTTPREAD command. And before executing AT+QHTTPREAD, "+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]", "+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]" or "+QHTTPPOSTFILE: <err>[,<httprspcode>,<content_length>]" information must be received.

AT+QHTTPREAD Read Response from HTTP(S) Server by UART/USB	
Test Command **AT+QHTTPREAD=?**	Response **+QHTTPREAD: (1-65535)** **OK**
Write Command **AT+QHTTPREAD[=<wait_time>]**	Response a) If parameter format is right and read successfully: **CONNECT** <Output HTTP response information > **OK** **+QHTTPREAD: <err>** If <wait_time> reaches or other errors occur, but body has not output completely, it will report the following codes: **+CME ERROR: <err>** b) If parameter format is not right or other errors occur: **+CME ERROR: <err>**

Parameter

<wait_time>	Numeric type, indicates the maximum interval time between receiving two packets of data. The default value is 60, unit: second
<err>	Please refer to Chapter 4

2.7. AT+QHTTPREADFILE Read Response from HTTP(S) Server by File

After sending HTTP GET/POST request, you can retrieve HTTP response information from HTTP server to file via AT+QHTTPREADFILE command. And before executing AT+QHTTPREADFILE, "+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]", "+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]" or

"+QHTTPPOSTFILE: <err>[,<httprspcode>,<content_length>]" information must be received.

AT+QHTTPREADFILE Read Response from HTTP(S) Server by File	
Test Command **AT+QHTTPREADFILE=?**	Response **+QHTTPREADFILE: <file_name>,(1-65535)** **OK**
Write Command **AT+QHTTPREADFILE=<file_name>[,<wait_time>]**	Response a) If parameter format is right: **OK** When body is read over or <wait_time> reaches, it will report: **+QHTTPREADFILE: <err>** b) If parameter format is not right or other errors occur: **+CME ERROR: <err>**

Parameter

<wait_time>	Numeric type, indicates the maximum interval time between receiving two packets of data, range is 1-65535. The default value is 60, unit: second
<file_name>	String type, file name, the max length of file name is 80, unit: byte.
<err>	Please refer to Chapter 4

3 Example

3.1. Access to HTTP Server

3.1.1. Send HTTP GET Request

Following example shows how to send HTTP GET request with a customer HTTP request header, and how to read HTTP GET response.

```
//How to send HTTP GET response.

AT+QHTTPCFG="contextid",1            //Set the PDP context ID as 1.
OK
AT+QHTTPCFG="responseheader",1     //Allow to output HTTP response header.
OK
AT+QIACT?                           //Query the state of context.
OK
AT+QICSGP=1,1,"UNINET","","",1      //Configure context 1, APN is "UNINET" for China Unicom.
OK
AT+QIACT=1                          //Activate context 1.
OK                                  //Activate successfully.
AT+QIACT?                           //Query the state of context.
+QIACT: 1,1,1,"10.7.157.1"

OK
AT+QHTTPURL=23,80                   //Set the URL which will be accessed.
CONNECT
HTTP://www.sina.com.cn/             //Input URL which length is 23, note that this URL is an example,
                                      please input the correct URL in practical test.
OK
AT+QHTTPGET=80                      //Send HTTP GET request and maximum response time is 80s.
OK

+QHTTPGET: 0,200,547256             //If HTTP response header contains "CONTENT-LENGTH"
                                      information, the <content_length> information will be reported.
//How to read HTTP response.

//Solution 1: Read HTTP response information and output by UART
```

```
AT+QHTTPREAD=80                    //Read HTTP response information and output by UART, the
                                     maximum time to wait for HTTP session to close is 80s.
CONNECT
HTTP/1.1 200 OK <CR><LF>           //HTTP response header and body.
Content-Type: text/html<CR><LF>
Vary: Accept-Encoding<CR><LF>
X-Powered-By: shci_v1.03<CR><LF>
Server: nginx<CR><LF>
Date: Fri, 27 Dec 2013 02:21:43 GMT<CR><LF>
Last-Modified: Fri, 27 Dec 2013 02:20:01 GMT<CR><LF>
Expires: Fri, 27 Dec 2013 02:22:43 GMT<CR><LF>
Cache-Control: max-age=60<CR><LF>
Age: 1<CR><LF>
Content-Length: 547256<CR><LF>
X-Cache: HIT from xd33-85.sina.com.cn<CR><LF>
<CR><LF>
<body>
OK

+QHTTPREAD: 0                        //Read HTTP response header and body successfully.
//Solution 2: Read HTTP response information and store it to RAM file.
AT+QHTTPREADFILE="RAM:1.txt",80   //Read HTTP response header and body and store them to
                                     "RAM:1.txt", the maximum time to wait for HTTP session to
                                     close is 80s.
OK

+QHTTPREADFILE: 0                  //HTTP response header and body are stored successfully.
```

3.1.2. Send HTTP POST Request

3.1.2.1. Post Body Obtained from UART/USB

Following example shows how to send HTTP POST request and retrieve post body from UART/USB. In addition, it also shows how to read HTTP POST response.

```
AT+QHTTPCFG="contextid",1          //Set the PDP context ID as 1.
OK
AT+QIACT?                          //Query the state of context.
OK
AT+QICSGP=1,1,"UNINET","","",1     //Configure context 1, APN is "UNINET" for China Unicom.
OK
AT+QIACT=1                         //Activate context 1.
```

```
OK                                           //Activate successfully.
AT+QIACT?                                    //Query the state of context.
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPURL=59,80                            //Set the URL which will be accessed.
CONNECT
HTTP://api.efxnow.com/DEMOWebServices2.8/Service.asmx/Echo?   //Input URL which length is 59,
                                                                note that this URL is an
                                                                example, please input the
                                                                correct URL in practical test.
OK
AT+QHTTPPOST=20,80,80                        //Send HTTP POST request, POST body is obtained from UART,
                                               maximum input body time is 80s and maximum response time is
                                               80s.
CONNECT
Message=HelloQuectel                         //Input post body which length is 20, note that post body is an
                                               example, please input the correct post body in practical test.
OK

+QHTTPPOST: 0,200,177                        //If HTTP response header contains "CONTENT-LENGTH"
                                               information, the <content_length> information will be reported.
AT+QHTTPREAD=80                              //Read HTTP response body and output by UART, the maximum
                                               time to wait for HTTP session to close is 80s.
CONNECT
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="httpHTTPs://api.efxnow.com/webservices2.3">Message='HelloQuectel' ASCII:72
101 108 108 111 81 117 101 99 116 101 108 </string>     //Output HTTP response body.
OK

+QHTTPREAD: 0                                //HTTP response body is output successfully.
```

3.1.2.2. Post Body Obtained from File System

Following example shows how to send HTTP POST request and retrieve post body from file system. In addition, it also shows how to store HTTP POST response to file system.

```
AT+QHTTPCFG="contextid",1                    //Set the PDP context ID as 1.
OK
AT+QIACT?                                    //Query the state of context.
OK
AT+QICSGP=1,1,"UNINET","","",1               //Configure context 1, APN is "UNINET" for China Unicom.
OK
AT+QIACT=1                                   //Activate context 1.
```

```
OK                                              //Activate successfully.
AT+QIACT?                                       //Query the state of context.
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPURL=59,80                               //Set the URL which will be accessed.
CONNECT
HTTP://api.efxnow.com/DEMOWebServices2.8/Service.asmx/Echo?   //Input URL which length is 59,
                                                                note that this URL is an
                                                                example, please input the
                                                                correct URL in practical test.
OK
AT+QHTTPPOSTFILE=“RAM:2.txt”,80                 //Send HTTP POST request, POST body is obtained from
                                                  “RAM:2.txt”, and maximum response time is 80s.
OK

+QHTTPPOSTFILE: 0,200,177                       //After HTTP POST request is sent success, you can execute
                                                  command AT+QHTTPREAD.
AT+QHTTPREADFILE=“RAM:3.txt”,80                 //Read HTTP response body and store it to “RAM:3.txt”, the
                                                  maximum time to wait for HTTP session to close is 80s.
OK

+QHTTPREADFILE: 0                               //HTTP response body is stored successfully.
```

3.2. Access to HTTPS Server

3.2.1. Send HTTP GET Request

Following example shows how to send HTTP GET request with a customer HTTP request header, and how to read HTTP GET response.

```
//How to send HTTP GET request.

AT+QHTTPCFG=“contextid”,1                       //Set the PDP context ID as 1.
OK
AT+QHTTPCFG=“responseheader”,1                  //Allow to output HTTP response header.
OK
AT+QIACT?                                       //Query the state of context.
OK
AT+QICSGP=1,1,“UNINET”,“”,“”,1                  //Configure context 1, APN is “UNINET” for China Unicom.
OK
AT+QIACT=1                                      //Activate context 1.
OK                                              //Activate successfully.
```

```
AT+QIACT?                            //Query the state of context.
+QIACT: 1,1,1,"10.7.157.1"

OK
AT+QHTTPCFG=“sslctxid”,1             //Set the SSL context ID.
OK
AT+QSSLCFG=“sslversion”,1,1          //Set the SSL version as 1, it means TLsV1.0.
OK
AT+QSSLCFG=“ciphersuite”,1,0x0005    //Set the SSL ciphersuite as 0x0005, it means RC4-SHA.
OK
AT+QSSLCFG=“seclevel”,1,0            //Set the SSL verify level as 0, it means you don’t need any CA
                                       Cert.
OK
AT+QHTTPURL=19,80                    //Set the URL which will be accessed.
CONNECT
HTTPs://mail.qq.com/                 //Input URL which length is 19, note that this URL is an example,
                                       please input the correct URL in practical test.
OK
AT+QHTTPGET=80                       //Send HTTP GET request and maximum response time is 80s.
OK

+QHTTPGET: 0,200,10750               //If HTTP response header contains “CONTENT-LENGTH”
                                       information, the <content_length> information will be
                                       reported.

//How to read HTTP response.

//Solution 1: Read HTTP response information and output by UART.
AT+QHTTPREAD=80                      //Read HTTP response information and output by UART,
                                       the maximum time to wait for HTTP session to close is 80s.
CONNECT                              //HTTP response header and body.
HTTP/1.1 200 OK<CR><LF>
Server: nginx/1.2.7<CR><LF>
Date: Fri, 27 Dec 2013 02:38:27 GMT<CR><LF>
Content-Type: text/html; charset=GB18030<CR><LF>
Content-Length: 10750<CR><LF>
Connection: keep-alive<CR><LF>
<CR><LF>
<body>
OK

+QHTTPREAD: 0                        //Read HTTP response header and body successfully.

//Solution 2: Read HTTP response information and store it to RAM file.
```

```
AT+QHTTPREADFILE="RAM:4.txt",80  //Read HTTP response header and body and store it to
                                   "RAM:4.txt", the maximum time to wait for HTTP session to
                                   close is 80s.
OK

+QHTTPREADFILE: 0                //HTTP response header and body are stored successfully.
```

3.2.2. Send HTTP POST Request

3.2.2.1. Post Body Obtained from UART/USB

Following example shows how to send HTTP POST request and retrieve post body from UART/USB. In addition, it also shows how to read HTTP POST response.

```
AT+QHTTPCFG="contextid",1              //Set the PDP context ID as 1.
OK
AT+QIACT?                              //Query the state of context.
OK
AT+QICSGP=1,1,"UNINET","","",1         //Configure context 1, APN is "UNINET" for China Unicom.
OK
AT+QIACT=1                             //Activate context profile 1.
OK                                     //Activate successfully.
AT+QIACT?                              //Query the state of context.
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPCFG="sslctxid",1               //Set the SSL context ID.
OK
AT+QSSLCFG="sslversion",1,1            //Set the SSL Version as 1, it means TLsV1.0.
OK
AT+QSSLCFG="ciphersuite",1,0x0005 //Set the SSL ciphersuite as 0x0005, it means RC4-SHA.
OK
AT+QSSLCFG="seclevel",1,2             //Set the SSL verify level as 2, it means you should upload CA
                                       Cert, client Cert and client private key by AT+QFUPL command.
OK
AT+QSSLCFG="cacert",1, "RAM:cacert.pem"
OK
AT+QSSLCFG="clientcert",1,"RAM:clientcert.pem"
OK
AT+QSSLCFG="clientkey",1,"RAM:clientkey.pem"
OK
AT+QHTTPURL=45,80                     //Set the URL which will be accessed.
CONNECT
```

```
HTTPs://220.180.239.201:8011/processorder.php      //Input URL which length is 45, note that URL is an
                                                     example, please input the correct URL in
                                                     practical test.
OK
AT+QHTTPPOST=48,80,80                    //Send HTTP POST request , POST body is obtained from UART,
                                           maximum input body time is 80s and maximum response time is
                                           80s.
CONNECT
Message=1111&Appleqty=2222&Orangeqty=3333&find=1   //Input post body which length is 48, note
                                                     that this post body is an example, please
                                                     input the correct one in practical test.
OK

+QHTTPPOST: 0,200,285                     //If HTTP response header contains "CONTENT-LENGTH"
                                            information, the <content_length> information will be reported.
AT+QHTTPREAD=80                          //Read HTTP response body and output by UART, the maximum
                                           time to wait for HTTP session to close is 80s.
CONNECT                                  //Read HTTP response body successfully.
<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                           //HTTP response body is output successfully.
```

3.2.2.2. Post Body Obtained from File System

Following example shows how to send HTTP POST request and retrieve post body from file system. In addition, it also shows how to store HTTP POST response to file system.

```
AT+QHTTPCFG="contextid",1               //Set the PDP context ID as 1.
OK
AT+QIACT?                               //Query the state of context.
OK
```

```
AT+QICSGP=1,1,“UNINET”,“”,“”,1        //Configure context 1, APN is “UNINET” for China Unicom.
OK
AT+QIACT=1                            //Activate context 1.
OK                                    //Activate successfully.
AT+QIACT?                             //Query the state of context.
+QIACT: 1,1,1,"172.22.86.226"

OK
AT+QHTTPCFG=“sslctxid”,1              //Set the SSL context ID.
OK
AT+QSSLCFG=“sslversion”,1,1           //Set the SSL version as 1, it means TLsV1.0.
OK
AT+QSSLCFG=“ciphersuite”,1,0x0005 //Set the SSL ciphersuite as 0x0005, it means RC4-SHA.
OK
AT+QSSLCFG=“seclevel”,1,2            //Set the SSL verify level as 2, it means you should upload CA
                                      Cert, client Cert and client private key by AT+QFUPL command.
OK
AT+QSSLCFG=“cacert”,1,“RAM:cacert.pem”
OK
AT+QSSLCFG=“clientcert”,1,“RAM:clientcert.pem”
OK
AT+QSSLCFG=“clientkey”,1,“RAM:clientkey.pem”
OK
AT+QHTTPURL=45,80                     //Set the URL which will be accessed.
CONNECT
HTTPs:// 220.180.239.201:8011/processorder.php //Input URL which length is 45, note that this URL is
                                               an example, please input the correct URL in
                                               practical test.
OK
AT+QHTTPPOSTFILE=“RAM:5.txt”,80  //Send HTTP POST request, POST body is obtained from
                                   “RAM:5.txt”, and maximum response time is 80s.
OK

+QHTTPPOSTFILE: 0,200,285            //After HTTP POST request success, you can execute command
                                       AT+QHTTPREAD.
AT+QHTTPREADFILE=“RAM:6.txt”,80 //Read HTTP response body and store it to “RAM:6.txt”, the
                                  maximum time to wait for HTTP session to close is 80s.
OK

+QHTTPREADFILE: 0                    //HTTP response body is stored successfully.
```

4 Summary of ERROR Code

Table 1: Different Coding Schemes of +CME ERROR : <err>

<err>	Meaning
0	Operation successful
701	HTTP unknown error
702	HTTP timeout
703	HTTP busy
704	HTTP UART busy
705	HTTP no get/post request
706	HTTP network busy
707	HTTP network open fail
708	HTTP network no config
709	HTTP network deactivated
710	HTTP network error
711	HTTP URL error
712	HTTP empty URL
713	HTTP IP address error
714	HTTP DNS error
715	HTTP socket create error
716	HTTP socket connect error
717	HTTP socket read error

718	HTTP socket write error
719	HTTP socket close
720	HTTP data encode error
721	HTTP data decode error
722	HTTP read timeout
723	HTTP response fail
724	Incoming call busy
725	Voice call busy
726	Input timeout
727	Wait data timeout
728	Wait HTTP response timeout
729	Alloc memory fail
730	Invalid parameter

5 Summary of HTTP Response Code

Table 2: Different Coding Schemes of HTTP Response Code

<httprspcode>	Meaning
200	OK
403	Forbidden
404	Not found
409	Conflict
411	Length required
500	Internal Server error

6 Appendix A Reference

Table 3: Related Documents

SN	Document Name	Remark
[1]	RFC2616	Hyper Text Transport Protocol
[2]	Quectel_EC20_TCPIP_AT_Commands_Manual	Introduction about EC20 TCPIP AT commands
[3]	Quectel_EC20_FILE_AT_Commands_Manual	Introduction about EC20 FILE AT commands
[4]	Quectel_EC20_AT_Commands_Manual	EC20 AT commands manual
[5]	Quectel_EC20_SSL_AT_Commands_Manual	Introduction about EC20 SSL AT commands

Table 4: Terms and Abbreviations

Abbreviation	Description
HTTP	Hyper Text Transport Protocol
SSL	Security Socket Layer
DTR	Data Terminal Ready
PPP	Point-to-Point Protocol
DNS	Domain Name Server
URL	Uniform Resource Locator
URI	Uniform Resource Identifier