BG96 HTTP

AT Commands Manual

LTE Module Series

Rev. BG96_HTTP_AT_Commands_Manual_V1.0

Date: 2017-06-22

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

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

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

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

GENERAL NOTES

QUECTEL OFFERS THE 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	2017-06-22	Sherlock ZHAO	Initial

Contents

Table Index

1 Introduction

BG96 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 BG96 TCP/IP AT commands, customers can configure a PDP context, activate/deactivate the PDP context and query the PDP context status. And through BG96 HTTP AT command, customers can send HTTP GET/POST requests to HTTP(S) server, and read HTTP response from HTTP(S) server. The general process is as follows:

Step 1: Configure <APN>, <username>, <password> and other parameters of a PDP context by AT+QICSGP. Please refer to *Quectel_BG96_TCP(IP)_AT_Commands_Manual* for details. If QoS settings need to be updated, configure them by AT+CGQMIN, AT+CGEQMIN, AT+CGQREQ and AT+CGEQREQ commands. For more details, please refer to *Quectel_BG96_AT_Commands_Manual*.

Step 2: Activate the PDP context by AT+QIACT, then the assigned IP address can be queried by AT+QIACT?. Please refer to *Quectel_BG96_TCP(IP)_AT_Commands_Manual* for details.

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

Step 4: Configure SSL context parameters by AT+QSSLCFG command. For more details, please refer to *Quectel_BG96_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, and 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 the PDP context by AT+QIDEACT command. For more details, please refer to *Quectel_BG96_TCP(IP)_AT_Commands_Manual*.

1.2. Description of HTTP Header

1.2.1. Customize HTTP Request Header

HTTP request header is filled by the module automatically. HTTP request header can be customized by configuring <request_header> as 1 via AT+QHTTPCFG command, and then inputting HTTP request header according to the following requirements:

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: 220.180.239.212:8011<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

1.2.2. Output HTTP Response Header

HTTP response header will not be outputted automatically. HTTP response header information can be obtained by configuring <response header> to 1 via AT+QHTTPCFG command, and then HTTP response header will be outputted with HTTP response body after executing AT+QHTTPREAD or AT+QHTTPREADFILE command.

1.3. Description of Data Mode

The COM port of BG96 module has two working modes: AT command mode and data mode. In AT command mode, the inputted data via COM port will be treated as AT command. While in data mode, it will be treated as data.

Inputting “+++” or pulling up DTR (AT&D1 should be set first) can make the COM port exit from data mode. To prevent the “+++” from being misinterpreted as data, the following sequence should be followed:

1) Do not input any character within 1s or longer before inputting “+++”.
2) Input “+++” within 1s, and no other characters can be inputted during the time.
3) Do not input any character within 1s after “+++” has been inputted.

When AT+QHTTPURL, AT+QHTTPOST and AT+QHTTPREAD are executed, the COM port will enter data mode. If customers are using “+++” or DTR to make the port exit from data mode, the executing procedure of these commands will be interrupted before the response is returned. In such case, the COM port cannot reenter data mode by executing ATO command.

1.4. Error Handling

1.4.1. Executing HTTP AT Commands Fails

When executing HTTP AT commands, if “ERROR” response is received from the module, please check whether the (U)SIM card is inserted and whether it is “+CPIN: Ready” returned when executing AT+CPIN?.

1.4.2. PDP Activation Fails

If it is failed to active a PDP context by AT+QIACT command, please check the following configurations:

1. Query whether the PS domain is attached or not by AT+CGATT? command. If not, please execute AT+CGATT=1 command to attach the 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 According to 3GPP specifications, the module only supports three PDP contexts activated simultaneously, so customers must make sure the number of activated PDP contexts is less than 3.

If all above configurations are correct, but activating the PDP context by AT+QIACT command still fails, please reboot the module to resolve this issue. After rebooting the module, please check the configurations mentioned above at least three times and each time at an interval of 10 minutes to avoid frequently rebooting 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(S) server is valid.
2. Query the status of the PDP context by AT+QIACT? command to make sure the specified PDP context has been activated successfully.
3. Query the address of DNS server by AT+QIDNSCFG command to make sure the address of DNS server is not "0.0.0.0".

If the DNS server address is "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 Data Mode Fails

When executing AT+QHTTPURL, AT+QHTTPGET, AT+QHTTPPOST and AT+QHTTPREAD commands, if "+CME ERROR: 704" (704: HTTP UART busy) is returned, please check whether there are other ports in data mode, since the module only supports one port in data mode simultaneously. If any, please re-execute these commands after other ports have exited from data mode.

1.4.5. Sending GET/POST Requests Fails

When executing AT+QHTTPGET, AT+QHTTPPOST and QHTTPPOSTFILE commands, if a failed result is received, please check the following configurations:

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

If all above configurations are correct, but sending GET/POST requests by AT+QHTTPGET, AT+QHTTPPOST and AT+QHTTPPOSTFILE commands still fails, please deactivate the PDP context by AT+QIDEACT and re-activate the PDP context by AT+QIACT to resolve this issue. If activating the PDP context fails, please refer to ***Chapter 1.4.2*** to resolve it.

1.4.6. Reading Response Fails

Before reading response by AT+QHTTPREAD and AT+QHTTPREADFILE commands, customers should execute AT+QHTTPGET, AT+QHTTPPOST and AT+QHTTPPOSTFILE commands and the following URC information will be reported:

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

During executing AT+QHTTTPREAD and AT+QHTTPREADFILE commands, if customers encounter some errors, such as: “+CME ERROR: 717” (717: HTTP socket read error), please resend HTTP GET/POST requests to HTTP(S) server by AT+QHTTPGET, AT+QHTTPPOST and AT+QHTTPPOSTFILE commands. If sending GET/POST requests to HTTP(S) server fails, please refer to ***Chapter 1.4.5*** to resolve it.

2 Description of HTTP AT Commands

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

The command is used to configure the parameters for HTTP(S) server, including configuring a PDP context ID, customizing HTTP request header, 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>** If <contextID> is omitted, 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>** If <request_header> is omitted , 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>** If <response_header> is omitted , 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>** If <sslctxID> is omitted, 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. PDP context ID. The range is 1-16, and the default value is 1.
<request_header>	Numeric type. Disable or enable to customize HTTP request header. 0 Disable 1 Enable
<response_header>	Numeric type. Disable or enable to output HTTP response header. 0 Disable 1 Enable
<sslctxID>	Numeric type. SSL context ID used for HTTP. The range is 0-5, and the default value is 1. Customers should configure the SSL parameters by AT+QSSLCFG. For details, please refer to *Quectel_BG96_SSL_AT_Commands_Manual*.

<content_type>	Numeric type. Data type of HTTP body. 0 application/x-www-form-urlencoded 1 text/plain 2 application/octet-stream 3 multipart/form-data
<err>	Integer type. The error code of the operation. Please refer to ***Chapter 4***.

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

URL must begin with “http://” or “https://”, which indicates you will access to an HTTP server or an HTTPS server respectively.

AT+QHTTPURL Set URL of HTTP(S) Server	
Test Command **AT+QHTTPURL=?**	Response **+QHTTPURL: (1-700),(1-65535)** **OK**
Write Command **AT+QHTTPURL=<URL_length>[,<timeout>]**	Response a) If the parameter format is right, and it is not sending HTTP GET/POST requests at present: **CONNECT** TA switches to transparent access mode, and the URL can be inputted. When the total size of the inputted data reaches <URL_length>, TA will return to command mode and report the following code: **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 the 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. The length of URL. The range is 1-700. Unit: byte.
<timeout>	Numeric type. The maximum time for inputting URL. The range is 1-65535, and the default value is 60. Unit: second.
<err>	Integer type. The error code of the operation. Please refer to ***Chapter 4***.

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

According to the configured <request_header> parameter in AT+QHTTPCFG="requestheader"[, <request_header>] command, the AT+HTTPGET Write Command has two different formats. If <request_header> is set to 1, after AT+QHTTPGET command has been sent, "CONNECT" may be outputted in 125s to indicate that the connection is successful. If it is not outputted during the time, the "+CME ERROR: <err>" will be outputted.

After AT+HTTPGET write command has been sent, it is suggested to wait a specific period of time (refer to the maximum response time below) for "+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]" to be outputted after "OK" is reported.

In "+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]", the <httprspcode> parameter can only be reported when <err> equals 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 0 (disable to customize HTTP request header) Write Command **AT+QHTTPGET[=<rsptime>]**	Response a) If the parameter format is right and no other errors occur: **OK** When the module has received response from HTTP(S) server, it will report the following URC: **+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]** b) If the parameter format is not right or other errors occur: **+CME ERROR: <err>**
If <request_header> equals 1 (enable to customize HTTP request header) Write Command **AT+QHTTPGET=<rsptime>,<data_len**	Response a) If HTTP(S) server is connected successfully: **CONNECT**

gth>[,<input_time>]	TA switches to transparent access mode, and the HTTP GET request header can be inputted. When the total size of the inputted data reaches <data_length>, TA will return to command mode and report the following code: **OK** When the module has received response from HTTP(S) 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 the 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, and the default value is 60. Unit: second. It is used to configure the timeout for the HTTP GET response "+QHTTPGET: <err>[,<httprspcode>,<content_length>]" to be outputted after "OK" is returned.
<data_length>	Numeric type. The length of HTTP request information, including HTTP request header and HTTP request body, the range is 1-2048. Unit: byte.
<input_time>	Numeric type. The maximum time for inputting HTTP request information. The range is 1-65535, and the default value is 60. Unit: second.
<err>	Integer type. The error code of the operation. 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 via UART/USB

The command is used to send HTTP POST request. According to the configured <request_header> parameter in AT+QHTTPCFG="requestheader"[,<request_header>] command, the AT+HTTPPOST Write Command has two different formats, if <request_header> is set to 0, please input post body via UART/USB port, if <request_header> is set to 1, please input post header and body via UART/USB port.

After AT+QHTTPPOST command has been sent, “CONNECT” may be outputted in 125s to indicate the connection is successful. If it is not received during the time, “+CME ERROR: <err>” will be outputted.

It is suggested to wait a specific period of time (refer to the maximum response time below) for “+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]” to be outputted after “OK” is reported.

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

	When the module has received response from HTTP(S) 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 the 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. The range is 1-1024000. Unit: byte.
<input_time>	Numeric type. The maximum time for inputting post body or HTTP request information. The range is 1-65535, and the default value is 60. Unit: second.
<rsptime>	Numeric type. The range is 1-65535, and the default value is 60. Unit: second. It is used to configure the timeout for the HTTP POST response "+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]" to be outputted after "OK" is returned.
<err>	Integer type. The error code of the operation. Please refer to ***Chapter 4***.
<httprspcode>	Please refer to ***Chapter 5***.
<request_header>	Please refer to ***Chapter 2.1***.
<content_length>	Numeric type. The length of HTTP response body. Unit: byte.

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

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

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

It is suggested to wait a specific period of time (refer to the maximum response time below) for "+QHTTPPOSTFILE: <err>[,<httprspcode>[,<content_length>]]" to be outputted after "OK" is reported.

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 1, the specified file must contain HTTP request header information.	Response a) If parameter format is right and HTTP(S) server is connected successfully: **OK** When the module has received response from HTTP(S) 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, and the default value is 60. Unit: second. It is used to configure the timeout for the HTTP POST response "+QHTTPPOSTFILE: <err>[,<httprspcode>,<content_length>]" to be outputted after "OK" is returned.
<err>	Integer type. The error code of the operation. Please refer to ***Chapter 4***.
<httprspcode>	Please refer to ***Chapter 5***.
<request_header>	Please refer to ***Chapter 2.1***.
<content_length>	Numeric type. The length of HTTP response body.

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

After sending HTTP GET/POST requests, customers can retrieve HTTP response information from HTTP(S) server via UART/USB port by AT+QHTTPREAD command. "+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]", "+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]" or "+QHTTPPOSTFILE: <err>[,<httprspcode>,<content_length>]" information must be received before executing AT+QHTTPREAD command.

AT+QHTTPREAD Read Response from HTTP(S) Server via UART/USB	
Test Command **AT+QHTTPREAD=?**	Response **+QHTTPREAD: (1-65535)** **OK**
Write Command **AT+QHTTPREAD[=<wait_time>]**	Response a) If the 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 been outputted completely, it will report the following code: **+CME ERROR: <err>** b) If the parameter format is not right or other errors occur: **+CME ERROR: <err>**

Parameter

<wait_time>	Numeric type. The maximum interval time between receiving two packets of data. The range is 1-65535, and the default value is 60. Unit: second.
<err>	Integer type. The error code of the operation. Please refer to ***Chapter 4***.

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

After sending HTTP GET/POST requests, you can retrieve HTTP response information from HTTP(S) server via file by AT+QHTTPREADFILE. And "+QHTTPGET: <err>[,<httprspcode>[,<content_length>]]",

"+QHTTPPOST: <err>[,<httprspcode>[,<content_length>]]" or "+QHTTPPOSTFILE: <err>[,<httprspcode>,<content_length>]" information must be received before executing AT+QHTTPREADFILE command.

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

Parameter

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

3 Examples

3.1. Access to HTTP Server

3.1.1. Send HTTP GET Request

The following examples show how to send HTTP GET request with a customer HTTP request header and how to read HTTP GET response.

```
//Example of how to send HTTP GET response.

AT+QHTTPCFG="contextid",1              //Configure 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 PDP context 1, APN is "UNINET" for China Unicom.
OK
AT+QIACT=1                                 //Activate context 1.
OK                                         //Activated 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 whose length is 23 (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.

//Example of how to read HTTP response.
```

```
//Solution 1: Read HTTP response information and output via UART port.
AT+QHTTPREAD=80                    //Read HTTP response information and output via 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

The following examples show how to send HTTP POST request and retrieve post body via UART port and how to read HTTP POST response.

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

```
AT+QIACT=1                                  //Activate context 1.
OK                                          //Activated 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 whose length is 59 (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 via UART, maximum input body time is 80s and maximum response time is 80s.
CONNECT
Message=HelloQuectel                        //Input post body whose length is 20 (post body is an example, please input the correct post body in practical test).
OK

+QHTTPPOST: 0,200,177                       //If the HTTP response header contains “CONTENT-LENGTH” information, the <content_length> information will be reported.
AT+QHTTPREAD=80                             //Read HTTP response body and output via 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 outputted successfully.
```

3.1.2.2. Post Body Obtained from File System

The following examples show how to send HTTP POST request and retrieve post body via file system and how to store HTTP POST response to file system.

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

```
AT+QIACT=1                                  //Activate context 1.
OK                                          //Activated 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 whose length is 59 (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 successfully, you can execute AT+QHTTPREAD command.
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 HTTPS GET Request

The following examples show how to send HTTP GET request with a customer HTTP request header and how to read HTTP GET response.

```
//Example of how to send HTTP GET request.
AT+QHTTPCFG="contextid",1                   //Configure 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 PDP context 1, APN is "UNINET" for China Unicom.
OK
AT+QIACT=1                                  //Activate context 1.
```

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

OK
AT+QHTTPCFG="sslctxid",1                    //Set SSL context ID.
OK
AT+QSSLCFG="sslversion",1,1                 //Set SSL version as 1, it means TLSV1.0.
OK
AT+QSSLCFG="ciphersuite",1,0x0005           //Set SSL cipher suite as 0x0005, it means RC4-SHA.
OK
AT+QSSLCFG="seclevel",1,0                   //Set SSL verify level as 0, it means you do not need any CA
                                              certificate.
OK
AT+QHTTPURL=22,80                           //Set the URL which will be accessed.
CONNECT
https://www.alipay.com                      //Input URL whose length is 19 (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: 0,200,21472                      //If the HTTP response header contains "CONTENT-LENGTH"
                                              information, the <content_length> information will be
                                              reported.

//Example of how to read HTTP response.

//Solution 1: Read HTTP response information and output via UART.

AT+QHTTPREAD=80                             //Read HTTP response information and output via 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 HTTPS POST Request

3.2.2.1. Post Body Obtained from UART/USB

The following examples show how to send HTTP POST request and retrieve post body via UART port and how to read HTTP POST response.

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

OK
AT+QHTTPCFG="sslctxid",1           //Set SSL context ID.
OK
AT+QSSLCFG="sslversion",1,1        //Set SSL version as 1, it means TLsV1.0.
OK
AT+QSSLCFG="ciphersuite",1,0x0005 //Set SSL cipher suite as 0x0005, it means RC4-SHA.
OK
AT+QSSLCFG="seclevel",1,2          //Set SSL verify level as 2, it means you should upload CA certificate, client certificate 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://220.180.239.212:8011/processorder.php    //Input URL whose length is 45 (this 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 whose length is 48 (this post body is an example, please input the correct one in practical test).
OK

+QHTTPPOST: 0,200,285                //If the HTTP response header contains “CONTENT-LENGTH” information, the <content_length> information will be reported.
AT+QHTTPREAD=80                      //Read HTTP response body and output via 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 outputted successfully.
```

3.2.2.2. Post Body Obtained from File System

The following examples show how to send HTTP POST request and retrieve post body from file system and how to store HTTP POST response to file system.

```
AT+QHTTPCFG=“contextid”,1            //Configure the PDP context ID as 1.
OK
```

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

OK
AT+QHTTPCFG=“sslctxid”,1                    //Set SSL context ID.
OK
AT+QSSLCFG=“sslversion”,1,1                 //Set SSL version as 1, it means TLsV1.0.
OK
AT+QSSLCFG=“ciphersuite”,1,0x0005           //Set SSL cipher suite as 0x0005, it means RC4-SHA.
OK
AT+QSSLCFG=“seclevel”,1,2                   //Set SSL verify level as 2, it means you should upload CA
                                              certificate, client certificate 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:// 220.180.239.212:8011/processorder.php //Input URL whose length is 45 (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 is sent successfully, 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 Codes

The error code <err> indicates an error related to mobile equipment or network. The details about <err> are described in the following table.

Table 1: Summary of Error Codes

<err>	Meaning
0	Operation successful
701	HTTP unknown error
702	HTTP timeout
703	HTTP busy
704	HTTP UART busy
705	HTTP no GET/POST requests
706	HTTP network busy
707	HTTP network open failed
708	HTTP network no configuration
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 closed
720	HTTP data encode error
721	HTTP data decode error
722	HTTP read timeout
723	HTTP response failed
724	Incoming call busy
725	Voice call busy
726	Input timeout
727	Wait data timeout
728	Wait HTTP response timeout
729	Memory allocation failed
730	Invalid parameter

5 Summary of HTTP Response Codes

<httprspcode> indicates the response codes from HTTP(S) server. The details about <httprspcode> are described in the following table.

Table 2: Summary of HTTP Response Codes

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

6 Appendix A References

Table 3: Related Documents

SN	Document Name	Remark
[1]	RFC2616	Hyper Text Transport Protocol
[2]	Quectel_BG96_TCP(IP)_AT_Commands_Manual	Introduction about BG96 TCP/IP AT commands
[3]	Quectel_BG96_FILE_Application_Note	BG96 FILE application note
[4]	Quectel_BG96_AT_Commands_Manual	BG96 AT commands manual
[5]	Quectel_BG96_SSL_AT_Commands_Manual	Introduction about BG96 SSL AT commands

Table 4: Terms and Abbreviations

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