



A7600 Series_HTTP_AT Command Manual_V1.00

LTE Module

Shanghai SIMCom Wireless Solutions Ltd.
Building A, SIM Technology Building, No.633, Jinzhong Road
Changning District 200335
Tel:86-21-31575100/31575200
support@simcom.com
www.simcom.com

Document Title:	A7600 Series_HTTP_AT Command Manual
Version:	1.00
Date:	2019-05-28
Status:	Release
Document ID:	A7600 Series_HTTP_AT Command Manual_V1.00

General Notes

SIMCom offers this information as a service to its customers, to support application and engineering efforts that use the products designed by SIMCom. The information provided is based upon requirements specifically provided to SIMCom by the customers. SIMCom has not undertaken any independent search for additional relevant information, including any information that may be in the customer's possession. Furthermore, system validation of this product designed by SIMCom within a larger electronic system remains the responsibility of the customer or the customer's system integrator. All specifications supplied herein are subject to change.

Copyright

This document contains proprietary technical information which is the property of SIMCom Limited., copying of this document and giving it to others and the using or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights reserved in the event of grant of a patent or the registration of a utility model or design. All specification supplied herein are subject to change without notice at any time.

Copyright © Shanghai SIMCom Wireless Solutions Ltd. 2019

Version History

Version	Date	Chapter	What is new
V1.00	2019-05-28		New version

SIMCom Confidential

Contents

Version History	2
Contents	3
1 Introduction	4
1.1 The process of Using HTTP(S) AT Commands	4
2 Description of AT Command	4
2.1 AT+HTTPINIT Start HTTP service	4
2.2 AT+HTTPTERM Stop HTTP Service	5
2.3 AT+HTTPPARA Set HTTP Parameters value	5
2.4 AT+HTTPACTION HTTP Method Action.....	7
2.5 AT+HTTPHEAD Read the HTTP Header Information of Server Response	7
2.6 AT+HTTPREAD Read the response information of HTTP Server	8
2.7 AT+HTTPDATA Input HTTP Data	9
2.8 AT+HTTPPOSTFILE Send HTTP Request to HTTP(S) server by File.....	9
2.9 AT+HTTPREADFILE Receive HTTP Response Content to a file	10
3 Example	11
3.1 Access to HTTP server.....	11
3.1.1 Send HTTP GET Request	11
3.1.2 Send HTTP POST Request	13
3.1.3 Send HTTP HEAD Request	14
3.1.4 POSTFILE to HTTP server and read HTTP response content to a file	15
3.2 Access to HTTPS server	16
3.2.1 Send HTTPS GET Request	16
3.2.2 Send HTTPS POST Request	18
3.2.3 Send HTTPS HEAD Request	20
3.2.4 POSTFILE to HTTPS server and read HTTPS response content to a file	21
4 Summary of HTTP Response Code	22
5 Summary of HTTP error Code	23

This document is a reference guide to all the AT commands defined for HTTP(S). Through these HTTP AT commands, you can send HTTP GET/POST request to HTTP server, and read HTTP response from HTTP server.

1 Introduction

1.1 The process of Using HTTP(S) AT Commands

Step 1: Ensure GPRS network is available before performing HTTP(S) related operations.

Step 2: Enable PDP context.

Step 3: Activate the PDP context to start HTTP(S) service by AT+HTTPIPINIT.

Step 4: Set HTTP(S) URL by AT+HTTTPARA.

Step 5: Send HTTP(S) request by setting the parameter of AT+HTTTPACTION to different values, and when sending a HTTP(S) POST request, AT+HTTTPDATA must be executed to input data to post before AT+HTTTPACTION=2.

Step 6: Read HTTP(S) response header by AT+HTTTPHEAD, and get HTTP(S) response content by AT+HTTTPREAD or AT+HTTTPREADFILE.

Step 7: Deactivate the PDP context to stop HTTP(S) service by AT+HTTTPTERM.

2 Description of AT Command

2.1 AT+HTTPIPINIT Start HTTP service

AT+HTTPIPINIT is used to start HTTP service by activating PDP context. You must execute AT+HTTPIPINIT before any other HTTP related operations.

AT+HTTPIPINIT Start HTTP service	
Execute Command AT+HTTPIPINIT	Response a) If start HTTP service successfully: OK b) If failed: ERROR
Maximum Response Time	120000ms

Defined Values

<err>	The type of error, please refer to chapter 5
-------	--

2.2 AT+HTTPTERM Stop HTTP Service

AT+HTTPTERM is used to stop HTTP service.

AT+HTTPTERM STOP HTTP service	
Execute Command AT+HTTPTERM	Response a) If stop HTTP service successfully: OK b) If failed: ERROR

2.3 AT+HTTTPARA Set HTTP Parameters value

AT+HTTTPARA is used to set HTTP parameters value. When you want to access to a HTTP server, you should input <value> like <http://server/path:tcpPort>. In addition, <https://server/path:tcpPort> is used to access to a HTTPS server.

AT+HTTTPARA Set HTTP Parameters value	
Write Command AT+HTTTPARA="URL", "<url>"	Response a) If parameter format is right: OK b) If parameter format is not right or other errors occur: ERROR
Write Command AT+HTTTPARA="CONNECTTO", <con n_timeout>	Response a) If parameter format is right: OK b) If parameter format is not right or other errors occur: ERROR
Write Command AT+HTTTPARA="RCVTO", <recv_tim eout>	Response a) If parameter format is right: OK b) If parameter format is not right or other errors occur: ERROR
Write Command AT+HTTTPARA="CONTENT", "<conte nt_type>"	Response a) If parameter format is right: OK b) If parameter format is not right or other errors occur: ERROR
Write Command AT+HTTTPARA="ACCEPT", "<accept-t ype>"	Response a) If parameter format is right: OK

	<p>b) If parameter format is not right or other errors occur: ERROR</p>
<p>Write Command AT+HTTTPARA="SSLCFG", "<sslcfg_id>"</p>	<p>Response a) If parameter format is right: OK b) If parameter format is not right or other errors occur: ERROR</p>
<p>Write Command AT+HTTTPARA="AUTH", "<auth_token>"</p>	<p>Response a) If parameter format is right: OK b) If parameter format is not right or other errors occur: ERROR</p>
<p>Write Command AT+HTTTPARA="USERDATA", "<user_data>"</p>	<p>Response a) If parameter format is right: OK b) If parameter format is not right or other errors occur: ERROR</p>

Defined Values

<url>	<p>URL of network resource. String, start with "http://" or "https://" a) http://server'/path':tcpPort'. b) https://server'/path':tcpPort' "server": DNS domain name or IP address "path": path to a file or directory of a server "tcpPort": http default value is 80, https default value is 443. (can be omitted)</p>
<conn_timeout>	<p>Timeout for accessing server, Numeric type, range is 20-120s, default is 120s.</p>
<recv_timeout>	<p>Timeout for receiving data from server, Numeric type range is 2s-120s, default is 20s.</p>
<content_type>	<p>This is for HTTP "Content-Type" tag, String type, max length is 256, default is "text/plain".</p>
<accept-type>	<p>This is for HTTP "Accept-type" tag, String type, max length is 256, default is "*/*".</p>
<sslcfg_id>	<p>This is setting SSL context id, Numeric type, range is 0-9. Default is 0. Please refer to SIM7500_SIM7600_SIM7800 Series_SSL_AT Command Manual.</p>
<user_data>	<p>The customized HTTP header information. String type, max length is 256.</p>

2.4 AT+HTTPACTION HTTP Method Action

AT+HTTPACTION is used to perform a HTTP Method. You can use HTTPACTION to send a get/post request to a HTTP/HTTPS server.

AT+HTTPACTION HTTP Method Action	
Test Command AT+HTTPACTION=?	Response +HTTPACTION: (0-3) OK
Write Command AT+HTTPACTION=<method>	Response a) If parameter format is right : OK +HTTPACTION: <method>,<statuscode>,<datalen> b) If parameter format is not right or other errors occur: ERROR

Defined Values

<method>	HTTP method specification: 0: GET 1: POST 2: HEAD 3: DELETE
<statuscode>	Please refer to chapter 4 and 5
<datalen>	The length of data received

2.5 AT+HTTPHEAD Read the HTTP Header Information of Server Response

AT+HTTPHEAD is used to read the HTTP header information of server response when module receives the response data from server.

AT+HTTPHEAD Read the HTTP Header Information of Server Response	
Execute Command AT+HTTPHEAD	Response a)If read the header information successfully: +HTTPHEAD: <data_len> <data> OK b)If read failed: ERROR

Defined Values

<data_len>	The length of HTTP header
<data>	The header information of HTTP response

2.6 AT+HTTPREAD Read the response information of HTTP Server

After sending HTTP(S) GET/POST requests, you can retrieve HTTP(S) response information from HTTP(S) server via UART/USB port by AT+HTTPREAD. When the <datalen> of “+HTTPACTION: <method>,<statuscode>,<datalen>” is not equal to 0, You can execute AT+HTTPREAD=<start_offset>,<byte_size> to read out data to port. If parameter <byte_size> is set greater than the size of data saved in buffer, all data in cache will output to port.

Note: The response content received from server will be saved in cache, and would not be cleaned up by AT+HTTPREAD.

AT+HTTPREAD Read the Response information of HTTP Server	
Read Command AT+HTTPREAD?	Response a) If check successfully: +HTTPREAD: LEN,<len> OK b) If failed (no more data other error): ERROR
Write Command AT+HTTPREAD=<start_offset>,<byte_size>	Response a) If read the response info successfully: OK +HTTPREAD: <data_len> <data> +HTTPREAD: 0 If <byte_size> is bigger than the data size received, module will only return actual data size. b) If read failed: ERROR

Defined Values

<start_offset>	The start position of reading
<byte_size>	The length of data to read
<datalen>	The actual length of read data
<data>	Response content from HTTP server
<len>	Total size of data saved in buffer.

2.7 AT+HTTPDATA Input HTTP Data

You can use AT+HTTPDATA to input data to post when you send a HTTP/HTTPS POST request.

AT+HTTPDATA Input HTTP Data	
Write Command AT+HTTPDATA=<size>,<time>	Response a)if parameter format is right: DOWNLOAD <input data here> When the total size of the inputted data reaches <size>, TA will report the following code. Otherwise, the serial port will be blocked. OK b)If parameter format is wrong or other errors occur: ERROR

Defined Values

<size>	Size in bytes of the data to post. range is 1- 153600 (bytes)
<time>	Maximum time in milliseconds to input data.

2.8 AT+HTTPPOSTFILE Send HTTP Request to HTTP(S) server by File

You also can send HTTP request in a file via AT+HTTPPOSTFILE command. The URL must be set by AT+HTTPPARA before executing AT+HTTPPOSTFILE command. The parameter <path> can be used to set the file directory. When modem has received response from HTTP server, it will report the following URC:

+HTTPPOSTFILE: <httpstatuscode>,<content_length>

AT+HTTPPOSTFILE Send HTTP Request to HTTP(S) server by File	
Test Command AT+HTTPPOSTFILE=?	Response +HTTPPOSTFILE: <filename>[, (1-2)]
Write Command AT+HTTPPOSTFILE=<filename>[,<path>]	Response a) if parameter format is right and server connected successfully: OK +HTTPPOSTFILE: <httpstatuscode>,<content_len> b) if parameter format is right but server connected unsuccessfully:

	<p>OK</p> <p>+HTTPPOSTFILE: <errcode>,0</p> <p>c) if parameter format is not right or any other error occurs: ERROR</p>
--	--

Defined Values

<filename>	String type, filename, the max length is 112.unit:byte. The directory where the sent file saved. Numeric type, range is
<path>	1-2 1 – C:/ (local storage) 2 – D:/(sd card)

2.9 AT+HTTPREADFILE Receive HTTP Response Content to a file

After execute AT+HTTPACTION/AT+HTTPPOSTFILE command. You can receive the HTTP server response content to a file via AT+HTTPREADFILE.

Before AT+HTTPREADFILE executed, "+HTTPACTION:<method>,<httpstatuscode>,<content_len>" or "+HTTPPOSTFILE: <httpstatuscode>,<content_len>" must be received. The parameter <path> can be used to set the directory where to save the file. If omit parameter <path>, the file will be save to local storage.

AT+HTTPREADFILE Receive HTTP Response Content to a File

<p>Test Command</p> <p>AT+HTTPREADFILE=?</p>	<p>Response</p> <p>+HTTPREADFILE: <filename>[, (1-2)]</p> <p>OK</p>
<p>Write Command</p> <p>AT+HTTPREADFILE=<filename>[,<path>]</p>	<p>Response</p> <p>a)if parameter format is right : OK</p> <p>+HTTPREADFILE: <errcode></p> <p>b)if failed: +HTTPREADFILE: <errcode></p> <p>ERROR</p> <p>c)if parameter format is not right or any other error occurs: ERROR</p>

Defined Values

<filename>	String type, filename, the max length is 112.unit:byte.
-------------------------	---

<path>	1 – C:/ (local storage)
	2 – D:/(sd card)

3 Example

3.1 Access to HTTP server

3.1.1 Send HTTP GET Request

Following commands shows how to send a HTTP GET request to server, and how to read HTTP response.

```

AT+HTTPINIT //start HTTP service, activate PDP context
OK
//set the URL which will be accessed, for HTTP, the request URL begins with "HTTP://"
AT+HTTPPARA="URL","http://opinion.people.com.cn/GB/n1/2018/0815/c1003-30228758.html"
OK
AT+HTTPACTION=0 //send HTTP GET request
OK

+HTTPACTION: 0,200,22505 //22505 is the length of HTTP response information
AT+HTTPHEAD //read the HTTP response header
+HTTPHEAD: 387 //387 is the length of response header

HTTP/1.1 200 OK
Server: nginx
Content-Type: text/html
Connection: close
Date: Thu, 16 Aug 2018 05:13:36 GMT
Powered-By-ChinaCache: MISS from 06053423gG.15
Etag: W/"5b7379f5-57e9"

```

Last-Modified: Wed, 15 Aug 2018 00:55:17 GMT

Expires: Thu, 16 Aug 2018 05:18:36 GMT

Vary: Accept-Encoding

X-Cache-Hits: 14

Content-Length: 22505 // Content-Length indicates the length of HTTP response information is 22505 bytes

CC_CACHE: TCP_REFRESH_HIT

Accept-Ranges: bytes

OK

//read the response information of HTTP server, the length to read is 500 bytes

AT+HTTPREAD=0,500

OK

+HTTPREAD: 500

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=GB2312"/>
```

```
<meta http-equiv="Content-Language" content="utf-8" />
```

```
<meta content="all" name="robots" />
```

```
<title>人民日报钟声：牢记历史是为了更好开创未来--观点--人民网 </title>
```

```
<meta name="keywords" content="" />
```

```
<meta name="description" content=" 日方应在正确对待历史?"
```

+HTTPREAD: 0

AT+HTTPTERM

//stop HTTP Service

OK

3.1.2 Send HTTP POST Request

Following commands shows how to send HTTP POST request to server, and how to read HTTP response.

```
AT+HTTPINIT //start HTTP service, activate PDP context
OK
//set the URL which will be accessed, for HTTP, the request URL begins with "HTTP://"
AT+HTTPPARA="URL","http://api.efxnow.com/DEMOWebServices2.8/Service.asmx/Echo?"
OK
AT+HTTPDATA=18,1000 //send data to post, the length is 18 bytes
DOWNLOAD
Message=helloworld
OK
AT+HTTPACTION=1 //send HTTP POST request
OK
+HTTPACTION: 1,500,30 //30 is the length of HTTP response information
AT+HTTPHEAD //read the HTTP response header
+HTTPHEAD: 258
HTTP/1.1 500 Internal Server Error
Cache-Control: private
Content-Type: text/plain; charset=utf-8
Server: Microsoft-IIS/7.0
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Mon, 20 Aug 2018 04:18:58 GMT
Connection: close
Content-Length: 30
```

OK

//read the response information of HTTP server, the length to read is 30 bytes

AT+HTTPREAD=0,30

OK

+HTTPREAD: 30

Request format is invalid: .

+HTTPREAD: 0

AT+HTTPTERM

//stop HTTP Service

OK

3.1.3 Send HTTP HEAD Request

Following commands shows how to send HTTP HEAD request to server, and how to read HTTP response. HEAD request is used to only get HTTP response header from server, we use this method to test if we can connect to the server successfully.

AT+HTTPINIT

//start HTTP service, activate PDP context

OK

AT+HTTPPARA="URL","http://opinion.people.com.cn/GB/n1/2018/0815/c1003-30228758.html"

OK

AT+HTTPACTION=2

//send a HEAD request to server to only get header of HTTP response

OK

+HTTPACTION: 2,200,387

//387 is the length of HTTP response header

+HTTP_PEER_CLOSED

//server disconnect

AT+HTTPHEAD

//read HTTP response header

+HTTPHEAD: 387

```
HTTP/1.1 200 OK
Server: nginx
Content-Type: text/html
Connection: close
Vary: Accept-Encoding
Powered-By-ChinaCache: MISS from 06053423gG.15
ETag: W/"5b7379f5-57e9"
Last-Modified: Wed, 15 Aug 2018 00:55:17 GMT
Content-Length: 22505
X-Cache-Hits: 14
Date: Thu, 16 Aug 2018 10:58:00 GMT
Expires: Thu, 16 Aug 2018 11:03:00 GMT
CC_CACHE: TCP_REFRESH_HIT
Accept-Ranges: bytes
```

OK

```
AT+HTTPTERM //stop HTTP Service
```

OK

3.1.4 POSTFILE to HTTP server and read HTTP response content to a file

Following commands show us how to send a file to HTTP server, and how to read HTTP content as a file. We have our HTTP GET request in a file getbaidu.txt, and save the file in local storage before we execute AT+HTTPPOSTFILE.

```
AT+HTTPINIT //activate PDP,start HTTP service
OK
AT+HTTPPARA="URL","http://www.baidu.com" //set server URL
OK
AT+HTTPPOSTFILE="getbaidu.txt",1 //access server and send file getbaidu.txt to server
OK
```

```
+HTTPPOSTFILE: 200,14615
AT+HTTPHEAD //read the HTTP server response header information.

+HTTPHEAD: 773
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 14615
Content-Type: text/html
Date: Thu, 13 Sep 2018 05:14:30 GMT
Etag: "5b8641dc-3917"
Last-Modified: Wed, 29 Aug 2018 06:49:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BAIDUID=A374BCFD28DFEEAF0BA0C4EEAC77B0B0:FG=1; expires=Thu, 31-Dec-37
23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: BIDUPSID=A374BCFD28DFEEAF0BA0C4EEAC77B0B0; expires=Thu, 31-Dec-37 23:55:55
GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536815670; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/;
domain=.baidu.com
Vary: Accept-Encoding
X-Ua-Compatible: IE=Edge,chrome=1

OK
//read the HTTP server response content to a file named readbaidu.dat, saved to local storage
AT+HTTPREADFILE="readbaidu.dat"
OK

+HTTPREADFILE: 0
AT+HTTPTERM //stop HTTP Service

OK
```

3.2 Access to HTTPS server

3.2.1 Send HTTPS GET Request

Following commands shows how to send HTTPS GET request to server, and how to read HTTPS response.

```
AT+HTTPINIT //start HTTP service, activate PDP context
OK
//set the URL which will be accessed, for HTTPS, the request URL begins with "HTTPS://"
AT+HTTPPARA="URL","https://ss0.bdstatic.com/5aV1bjqh_Q23odCf/static/mancard/css/card_min_dee3
8e45.css"
OK
AT+HTTPACTION=0 // send HTTPS GET request
OK
+HTTPACTION: 0,200,52060 // 52060 is the length of HTTPS response information
AT+HTTPHEAD //read HTTPS response header
+HTTPHEAD: 390 //390 is the length of HTTPS response hreader

HTTP/1.1 200 OK
Server: bfe/1.0.8.13-sslpool-patch
Date: Thu, 16 Aug 2018 11:38:08 GMT
Content-Type: text/css
Content-Length: 52060
Connection: close
ETag: "5a323f72-cb5c"
Last-Modified: Thu, 14 Dec 2017 09:08:02 GMT
Expires: Sat, 18 Aug 2018 09:50:53 GMT
Age: 2425635
Accept-Ranges: bytes
Cache-Control: max-age=2592000
Vary: Accept-Encoding
Ohc-Response-Time: 1 0 0 0 0
```

OK

//read the response information of HTTPS server, the length to read is 500 bytes

AT+HTTPREAD=0,500

OK

+HTTPREAD: 500

```
.s-cardsetting{position:relative;text-align:left;padding:22px 25px 0 25px;border:1px solid #e3e3e3;width:843px}.main .sui-dialog-cardsetting{opacity:.98;filter:alpha(opacity=98);position:absolute;border:none;display:none;_height:186px}.sui-dialog-cardsetting{opacity:.98!important;filter:alpha(opacity=98)!important;border:none!important}.sui-dialog-cardsetting .sui-dialog-title{height:42px;line-height:42px;text-indent:21px}.s-cardsetting-content .s-mod-item b,.sui-dialog-cardsetting .sui-dialog-c
```

+HTTPREAD: 0

AT+HTTPTERM

//stop HTTP Service

OK

3.2.2 Send HTTPS POST Request

Following commands shows how to send HTTPS POST request to server, and how to read HTTPS response.

AT+HTTPINIT

//start HTTP service, activate PDP context

OK

//set the URL which will be accessed, for HTTPS, the request URL begins with "HTTPS://"

AT+HTTTPARA="URL","https://pv.csdn.net/csdnbi"

OK

AT+HTTPDATA=465,1000

//send data to post, the length is 465 bytes

DOWNLOAD

//prompt string which indicates you can input data here

```
[{"headers":{"component":"enterprise","datatype":"track","version":"v1"},"body":{"re":{"uid=merry1996&ref=https%3A%2F%2Fpassport.csdn.net%2Faccount%2Fverify%3Bjsessionid%3D7895A57BC64CE8616517F558940FD913.tomcat2&pid=www&mod=&con=&ck=-&curl=https%3A%2F%2Fwww.csdn.net%2F&session_id=10_1534696351647.160829&tos=12&referrer=https%3A%2F%2Fpassport.csdn.net%2Faccount%2Fverify%3Bjsessionid%3D7895A57BC64CE8616517F558940FD913.tomcat2&user_name=merry1996&type=pv\"}}}]
```

OK

```
AT+HTTPACTION=1 //send HTTPS post request
OK

+HTTPACTION: 1,200,2 // 2 is the length of HTTPS response information

+HTTP_PEER_CLOSED

AT+HTTPHEAD //read the HTTPS response header
+HTTPHEAD: 377
HTTP/1.1 200 OK
Server: openresty
Date: Mon, 20 Aug 2018 03:20:30 GMT
Content-Type: application/octet-stream
Connection: close
Set-Cookie: uuid_tt_dd=10_37481894210-1534735230305-445993; Expires=Thu, 01 Jan 2025 00:00:00 GMT; Path=/; Domain=.csdn.net;
Set-Cookie: dc_session_id=10_1534735230305.501284; Expires=Thu, 01 Jan 2025 00:00:00 GMT; Path=/; Domain=.csdn.net;

OK
//read the response information of HTTPS server, the length to read is 10 bytes
AT+HTTPREAD=0,10
OK

+HTTPREAD: 2
ok //ok is the content of HTTPS response information, 2 bytes
+HTTPREAD: 0
AT+HTTPTERM //stop HTTP Service
OK
```

3.2.3 Send HTTPS HEAD Request

Following commands shows how to send HTTPS HEAD request to server, and how to read HTTPS response.

```
AT+HTTPIPINIT //start HTTP service, activate PDP context
OK
//set the URL which will be accessed, for HTTPS, the request URL begins with "HTTPS://"
AT+HTTTPARA="URL","https://ss0.bdstatic.com/5aV1bjqh\_Q23odCf/static/mancard/css/card\_min\_dee38e45.css"
OK
AT+HTTTPACTION=2 // send HTTPS HEAD request
OK
+HTTTPACTION: 2,200,390 // 390 is the length of HTTPS response header
+HTTP_PEER_CLOSED
AT+HTTPHEAD //read HTTPS response header
+HTTPHEAD: 390
HTTP/1.1 200 OK
Server: bfe/1.0.8.13-sslpool-patch
Date: Thu, 16 Aug 2018 11:46:22 GMT
Content-Type: text/css
Content-Length: 52060
Connection: close
ETag: "5a323f72-cb5c"
Last-Modified: Thu, 14 Dec 2017 09:08:02 GMT
Expires: Sat, 18 Aug 2018 09:50:53 GMT
Age: 2426129
```

Accept-Ranges: bytes
Cache-Control: max-age=2592000
Vary: Accept-Encoding
Otc-Response-Time: 1 0 0 0 0 0

OK

AT+HTTPTERM //stop HTTP Service

OK

3.2.4 POSTFILE to HTTPS server and read HTTPS response content to a file

AT+HTTPINIT //activate PDP,start HTTP service

OK

AT+HTTPPARA="URL","https://www.baidu.com" //set server URL

OK

AT+HTTPPOSTFILE="getbaidu.txt",1 //access server and send file getbaidu.txt to server

OK

+HTTPPOSTFILE: 200,14615

AT+HTTPHEAD //read the HTTP server response header information.

+HTTPHEAD: 773

HTTP/1.1 200 OK

Accept-Ranges: bytes

Cache-Control: no-cache

Connection: Keep-Alive

Content-Length: 14615

Content-Type: text/html

Date: Thu, 13 Sep 2018 05:14:30 GMT

Etag: "5b8641dc-3917"

Last-Modified: Wed, 29 Aug 2018 06:49:00 GMT

P3p: CP=" OTI DSP COR IVA OUR IND COM "

Pragma: no-cache

Server: BWS/1.1

Set-Cookie: BAIDUID=A374BCFD28DFEEAF0BA0C4EEAC77B0B0:FG=1; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

Set-Cookie: BIDUPSID=A374BCFD28DFEEAF0BA0C4EEAC77B0B0; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

```
Set-Cookie: PSTM=1536815670; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/;
domain=.baidu.com
Vary: Accept-Encoding
X-Ua-Compatible: IE=Edge,chrome=1

OK
//read the HTTP server response content to a file readbaidu.dat, saved to local storage
AT+HTTPREADFILE="readbaidu.dat"
OK

+HTTPREADFILE: 0
AT+HTTPTERM //stop HTTPS Service
OK
```

4 Summary of HTTP Response Code

HTTP status code responded by remote server, refer to HTTP 1.1(RFC 2616).

<statuscode>	Meaning
100	Continue
101	Switching Protocols
200	OK
201	Created
201	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy
307	Temporary Redirect
400	Bad Request

<statuscode>	Meaning
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Large
415	Unsupported Media Type
416	Requested range not satisfiable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway timeout
505	HTTP Version not supported
600	Not HTTP PDU
601	Network Error
602	No memory
603	DNS Error
604	Stack Busy

5 Summary of HTTP error Code

HTTP code

HTTP code	Meaning
+HTTP_PEER_CLOSED	It's a notification message. While received, it means the connection has been closed by server.
+HTTP_NONET_EVENT	It's a notification message. While received, it means now the network is unavailable.

HTTP error code:

<errcode>	Meaning
0	Success
701	Alert state
702	Unknown error
703	Busy
704	Connection closed error
705	Timeout
706	Receive/send socket data failed
707	File not exists or other memory error
708	Invalid parameter
709	Network error
710	start a new ssl session failed
711	Wrong state
712	Failed to create socket
713	Get DNS failed
714	Connect socket failed
715	Handshake failed
716	Close socket failed
717	No network error
718	Send data timeout
719	CA missed