# SIM7500_SIM7600_SIM7800 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:** | SIM7500_SIM7600_SIM7800 Series_HTTP_AT Command Manual |
| **Version:** | 1.00 |
| **Date:** | 2018-10-12 |
| **Status:** | Release |
| **Document ID:** | SIM7500_SIM7600_SIM7800 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.

# Version History

| Version | Date | Chapter | What is new |
|---|---|---|---|
| V1.00 | 2018-09-28 |  | New version |

# Contents

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+HTTPINIT.

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

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

**Step 6**: Read HTTP(S) response header by AT+HTTPHEAD, and get HTTP(S) response content by AT+HTTPREAD or AT+HTTPREADFILE.

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

# 2 Description of AT Command

## 2.1 AT+HTTPINIT Start HTTP service

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

| AT+HTTPINIT Start HTTP service | |
|---|---|
| Execute Command<br>**AT+HTTPINIT** | Response<br>a) If start HTTP service successfully:<br>**OK**<br>b) If failed:<br>**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<br>**AT+HTTPTERM** | Response<br>a)If stop HTTP service successfully:<br>**OK**<br>b)If failed:<br>**ERROR** |

## 2.3 AT+HTTPPARA Set HTTP Parameters value

AT+HTTPPARA 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+HTTPINIT Start HTTP service | |
|---|---|
| Write Command<br>**AT+HTTPPARA="URL","<url >"** | Response<br>a) If parameter format is right:<br>**OK**<br>b) If parameter format is not right or other errors occur:<br>**ERROR** |
| Write Command<br>**AT+HTTPPARA="CONNECTTO",<conn_timeout>** | Response<br>a) If parameter format is right:<br>**OK**<br>b) If parameter format is not right or other errors occur:<br>**ERROR** |
| Write Command<br>**AT+HTTPPARA="RECVTO",<recv_timeout>** | Response<br>a) If parameter format is right:<br>**OK**<br>b) If parameter format is not right or other errors occur:<br>**ERROR** |
| Write Command<br>**AT+HTTPPARA="CONTENT","<content_type>"** | Response<br>a) If parameter format is right:<br>**OK**<br>b) If parameter format is not right or other errors occur:<br>**ERROR** |
| Write Command<br>**AT+HTTPPARA="ACCEPT","**<accept-type>" | Response<br>a) If parameter format is right:<br>**OK** |

| | |
|---|---|
| | b) If parameter format is not right or other errors occur:<br>**ERROR** |
| Write Command<br>**AT+HTTPPARA="SSLCFG","<sslcfg_id >** | Response<br>a) If parameter format is right:<br>**OK**<br>b) If parameter format is not right or other errors occur:<br>**ERROR** |

**Defined Values**

| | |
|---|---|
| **<url>** | URL of network resource.String,start with "http://" or"https://"<br>a)http://'server'/'path':'tcpPort'.<br>b)https://'server'/'path':'tcpPort'<br>"server": DNS domain name or IP address<br>"path": path to a file or directory of a server<br>"tcpPort": http default value is 80,https default value is 443.(can be omitted) |
| **<conn_timeout>** | Timeout for accessing server, Numeric type, range is 20-120s, default is 120s. |
| **<recv_timeout>** | Timeout for receiving data from server, Numeric type range is 2-20s, default is 10s. |
| **< content_type>** | This is for HTTP "Content-Type" tag, String type, max length is 256,default is "text/plain". |
| **<accept-type >** | This is for HTTP "Accept-type" tag, String type, max length is 256,default is "*/*". |
| **<sslcfg_id >** | 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. |

## 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<br>**AT+HTTPACTION=?** | Response<br>**+HTTPACTION: (0-3)**<br><br>**OK** |
| Execute Command<br>**AT+HTTPACTION=<method>** | Response<br>a) If parameter format is right :<br>**OK** |

| | |
|---|---|
| | **+HTTPACTION: <method>,<statuscode>,<datalen>**<br>b) If parameter format is not right or other errors occur:<br>**ERROR** |

**Defined Values**

| | |
|---|---|
| **<method>** | HTTP method specification:<br>0: GET<br>1: POST<br>2: HEAD<br>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<br>**AT+HTTPHEAD** | Response<br>a)If read the header information successfully：<br>**+HTTPHEAD: <data_len>**<br>**<data>**<br>**OK**<br>b)If read failed:<br>**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 read the response information from HTTP(S) server by AT+HTTPREAD. You can execute AT+HTTPREAD? to check the total data saved in buffer, then AT+HTTPREAD=<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 buffer will output to port.

| **AT+HTTPREAD Read the Response of HTTP Server** |
|---|

| | |
|---|---|
| Read Command<br>**AT+HTTPREAD?** | Response<br>a) If check successfully:<br>**+HTTPREAD: LEN,<len>**<br><br>**OK**<br>b) If failed (no more data other error):<br>**ERROR** |
| Execute Command<br>**AT+HTTPREAD=[<start_addr>,]<byte_size>** | Response<br>a)If read the response info successfully：<br>**OK**<br>**+HTTPREAD: <data_len>**<br>**<data>**<br>**+HTTPREAD:0**<br>If <byte_size> is bigger than the data size received, module will only return actual data size.<br>b)If read failed:<br>**ERROR** |

**Defined Values**

| | |
|---|---|
| **<start_addr>** | The starting read position of the response information, can be omitted. |
| **<byte_size>** | The length of data to read, if there's only one parameter, that would be set as byte_size. |
| **<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** | |
|---|---|
| Execute Command<br>**AT+HTTPDATA=<size>,<time>** | Response<br>a)if parameter format is right:<br>**DOWNLOAD**<br>**<input data here>**<br><br>When the total size of the inputted data reaches **<size>**, TA will report the following code. Otherwise, the serial port will be blocked. |

| | |
|---|---|
| | **OK**<br>b)If parameter format is wrong or other errors occur:<br>**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. if set **<send_header>** to 0, you can customize any HTTP request in the file, module will send the file as HTTP header and body, else if set **<send_header>** to 1, modem will package a HTTP request itself, the file will be sent as HTTP body. 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: <method>[,<httpstatuscode>[,<content_length>]]**

Note: The parameter<method>,<send_header>,<path>can be omitted, and the default value of <send_header> is 0, the default<path>is 1 (/cache),default method is 1 (POST)

| **AT+HTTPPOSTFILE Send HTTP Request to HTTP(S) server by File** | |
|---|---|
| Test Command<br>**AT+HTTPPOSTFILE=?** | Response<br>**+HTTPPOSTFILE :<filename>[,(1-3)[,(0-3)[,(0-1)]]]** |
| Execute Command<br>**AT+HTTPPOSTFILE=<filename>[,<path>] [,<method>][,<send_header>]** | Response<br>a) if parameter format is right and server connected successfully:<br>**OK**<br><br>**+HTTPPOSTFILE: <method>,<httpstatuscode>,<content_len>**<br>b) if parameter format is right but server connected unsuccessfully:<br>**OK**<br><br>**+HTTPPOSTFILE: <method>,<errcode>,0**<br><br>c) if parameter format is not right or any other error occurs:<br>**ERROR** |

**Defined Values**

| | |
|---|---|
| **<filename>** | String type, filename, the max length is 256.unit:byte. |
| **<path>** | The directory where the sent file saved. Numeric type, range is |

| | |
|---|---|
| | 1-3<br>1 – F:/ (/cache)<br>2 – D:/(sd card)<br>3 – E:/ (/data/media/) |
| **<method>** | HTTP method specification:<br>0 – GET<br>1 – POST<br>2 – HEAD<br>3 – DELETE |
| **<send_header>** | Send file as HTTP header and Body Or Only as Body.Numeric type, the range is 0-1,the default is 0.<br>0 –Send file as HTTP header and body<br>1 – Send file as Body |

## 2.9 AT+HTTPREADFILE Receive HTTP Response Content to a file

After execute AT+HTTPACTION/AT+HTTPOSTFILE 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: <httpsatuscode>,<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 /cache.
**Note:** by setting <path> to 4, you can download CA files to directory /mssl_cert/, which is used for SSL verification, details please refer SIM7500_SIM7600_SIM7800 Series_SSL_AT_Commands_Manual.

| **AT+HTTPREADFILE Receive HTTP Response Content to a File** | |
|---|---|
| Test Command<br>**AT+HTTPREADFILE=?** | Response<br>**+HTTPREADTFILE :<filename>[,(1-4)]** |
| Execute Command<br>**AT+HTTPREADFILE=<filename>[,<path>]** | Response<br>a)if parameter format is right :<br>**OK**<br><br>**+HTTPREADFILE: <result>**<br>b)if parameter format is not right or any other error occurs:<br>**ERROR** |

**Defined Values**

| | |
|---|---|
| **<filename>** | String type, filename, the max length is 256.unit:byte. |
| **<path>** | 1 – F:/ (/cache/)<br>2 – D:/(sd card)<br>3 – E:/ (/data/media/)<br>4 – /mssl_cert/(this is for CA file downloading) |

# 3 Example

Before all HTTP related operations, we should ensure the following:

a) ensure GPRS network is available:

```
AT+CSQ

+CSQ: 23,0


OK

AT+CREG?

+CREG: 0,1


OK

AT+CGREG?

+CGREG: 0,1


OK
```

b) PDP context Enable:

```
// Specify the parameter value of the PDP context corresponding to cid
AT+CGSOCKCONT=1,"IP","CMNET"

OK

AT+CGPADDR

+CGPADDR: 1,10.49.14.68                    //ensure the first PDP context get a IP address

+CGPADDR: 4,0.0.0.0


OK

Note: usually CSOCKAUTH and CSOCKSETPN parameter are kept default if not care about.
```

## 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
 +HTTPREAD: DATA,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?
OK
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



+HTTP_PEER_CLOSED
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

+HTTPREAD: DATA,30

Request format is invalid: .


OK

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 directory "/data/media" 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",0,0     //access server and send file getbaidu.txt to server
OK

+HTTPPOSTFILE: 0,200,14615
AT+HTTPHEAD                                  //read the HTTP server response header information.

+HTTPHEAD: DATA,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, saved to "/data/media/readbaidu.dat"
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_dee38
```

```
e45.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 0


OK
//read the response information of HTTPS server, the length to read is 500 bytes
AT+HTTPREAD=0,500
+HTTPREAD: DATA,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;b
order: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
OK
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+HTTPPARA="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=mer
ry1996&ref=https%3A%2F%2Fpassport.csdn.net%2Faccount%2Fverify%3Bjsessionid%3D7895A57BC6
4CE8616517F558940FD913.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=m
erry1996&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
+HTTPREAD: DATA,2
ok                                     //ok is the content of HTTPS response information, 2 bytes
OK
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+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=2                      // send HTTPS HEAD request
OK


+HTTPACTION: 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
Ohc-Response-Time: 1 0 0 0 0 0
```

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

### 3.1.4 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",0,0      //access server and send file getbaidu.txt to server
OK

+HTTPPOSTFILE: 0,200,14615
AT+HTTPHEAD                                   //read the HTTP server response header information.

+HTTPHEAD: DATA,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, saved to "/data/media/readbaidus.dat"
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 |
| 401 | Unauthorized |
| 402 | Payment Required |
| 403 | Forbidden |
| 404 | Not Found |
| 405 | Method Not Allowed |
| 406 | Not Acceptable |
| 407 | Proxy Authentication Required |

| <statuscode> | Meaning |
|---|---|
| 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:

| <statuscode> | Meaning |
|---|---|
| 0 | Success |
| 701 | Alert state |
| 702 | Unknown error |
| 703 | Busy |
| 704 | Connection closed error |

| <statuscode> | Meaning |
|---|---|
| 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 |