SIM7672X & SIM7652X Series_HTTP(S)_ Application Note

LTE Module

SIMCom Wireless Solutions Limited
SIMCom Headquarters Building, Building 3, No. 289 Linhong Road, Changning District, Shanghai P.R. China
Tel: 86-21-31575100
support@simcom.com
www.simcom.com

Document Title:	SIM7672X & SIM7652X Series_HTTP(S)_Application Note
Version:	1.00
Date:	2023.05.22
Status:	Released

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

SIMCom Wireless Solutions Limited
SIMCom Headquarters Building, Building 3, No. 289 Linhong Road, Changning District, Shanghai P.R. China
Tel: +86 21 31575100
Email: simcom@simcom.com

For more information, please visit:
https://www.simcom.com/technical_files.html

For technical support, or to report documentation errors, please visit:
https://www.simcom.com/online_questions.html or email to: support@simcom.com

About Document

Version History

Version	Date	Owner	Description
V1.00	2023.5.22		New version

Scope

Based on module AT command manual, this document will introduce HTTP(S) application process. Developers could understand and develop application quickly and efficiently based on this document. This document applies to SIM7672X Series, SIM7652X Series.

Contents

1 Introduction

1.1 Purpose of the document

Based on module AT command manual, this document will introduce HTTP(S) application process. Developers could understand and develop application quickly and efficiently based on this document.

1.2 Related documents

[1] SIM7672X & SIM7652X Series_AT Command Manual

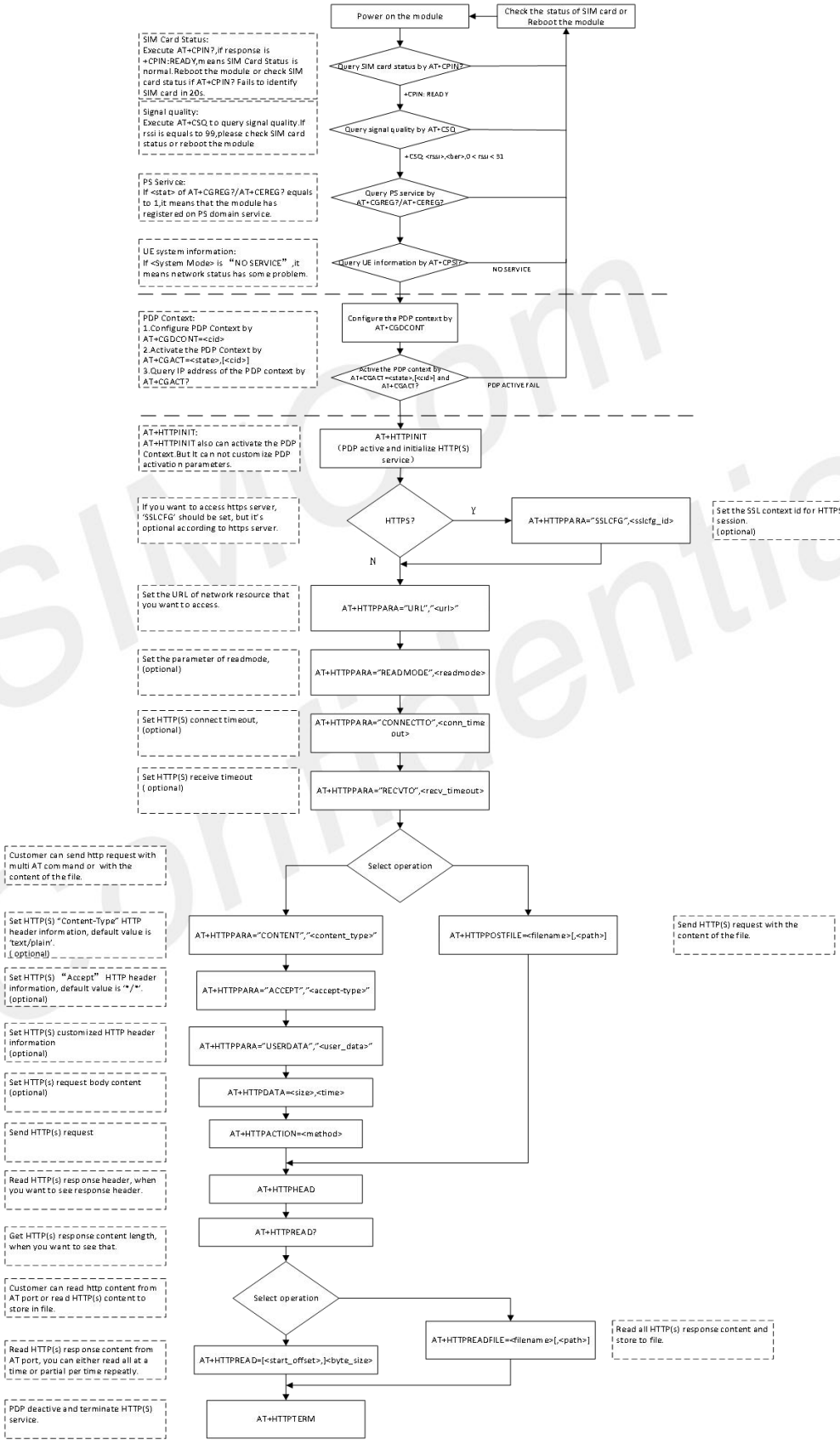
1.3 Conventions and abbreviations

In this document, the engines are referred as following term:
ME (Mobile Equipment);
MS (Mobile Station);
TA (Terminal Adapter);
DCE (Data Communication Equipment);

In application, controlling device controls the engine by sending AT Command via its serial interface. The controlling device at the other end of the serial line is referred as following term:
TE (Terminal Equipment);
DTE (Data Terminal Equipment) or plainly "the application" which is running on an embedded system;

Other Conventions:
PDP(Packet Data Protocol);
FTP(File Transfer Protocol);
SSL(Secure Sockets Layer);
TLS(Transport Layer Security);

1.4 The process of Using HTTP(S) AT Commands



1.5 Error Handling

1.5.1 Executing HTTP(S) AT Commands Fails

When executing HTTP(S) 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?***.

1.5.2 PDP Activation Fails

If it is failed to activate a PDP context with ***AT+CGACT*** command, please check the following configurations:
1. Query the PS domain status by ***AT+CGREG? / AT+CEREG?*** and make sure the PS domain has been registered.
2. Query the PDP context parameters by ***AT+CGDCONT?*** and make sure the APN of the specified PDP context has been set.
3. Make sure the specified PDP context ID is neither used by PPP nor activated by ***AT+CGACT*** command.
If all the above configurations are correct, but activating the PDP context by ***AT+CGACT*** command still fails, please reboot the module to resolve this issue. After rebooting the module, please check the configurations mentioned above for at least.

1.5.3 Error Response of HTTP(S) Server

When the < errcode > of ***+HTTPACTION: <method>,<errcode>,<datalen>*** or ***+HTTPPOSTFILE: <errcode>,<datalen>*** is not 200, it indicates an error code replied from HTTP(S) server.
For example, if < errcode > is 404, the URL can't be found. If < errcode > is 301, the URL is redirected. For more details, please refer to SIM7672X & SIM7652X Series_AT Command Manual.

2 AT Commands for HTTP(S)

2.1 Overview of AT Commands for HTTP(S)

Command	Description
AT+HTTPINIT	Start HTTP service
AT+HTTPTERM	Stop HTTP Service
AT+HTTPPARA	Set HTTP Parameters value
AT+HTTPACTION	HTTP Method Action
AT+HTTPHEAD	Read the HTTP Header Information of Server Response
AT+HTTPREAD	Read the response information of HTTP Server
AT+HTTPDATA	Input HTTP Data
AT+HTTPPOSTFILE	Send HTTP Request to HTTP(S) server by File
AT+HTTPREADFILE	Receive HTTP Response Content to a file

2.2 Detailed Description of AT Commands for HTTP(S)

2.2.1 AT+HTTPINIT Start HTTP Service

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

AT+HTTPINIT Start HTTP Service	
Test Command AT+HTTPINIT=?	Response **OK**
Execute Command AT+HTTPINIT	Response 1)If start HTTP service successfully: **OK** 2)If failed: **ERROR**

Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

<errcode>	Please refer to <errcode> list.

Examples

```
AT+HTTPINIT
OK
```

2.2.2 AT+HTTPTERM Stop HTTP Service

AT+HTTPTERM is used to stop HTTP service.

AT+HTTPTERM Stop HTTP Service	
Test Command **AT+HTTPTERM=?**	Response **OK**
Execute Command **AT+HTTPTERM**	Response 1)If stop HTTP service successfully: **OK** 2)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Examples

```
AT+HTTPTERM
OK
```

2.2.3 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+HTTPPARA Set HTTP Parameters value	
Test Command **AT+HTTPPARA=?**	Response **OK**
Write Command **AT+HTTPPARA="URL",<url>**	Response 1)If parameter format is right: **OK** 2)If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="CONNECTTO",<conn_timeout>**	Response 1)If parameter format is right: **OK** 2)If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="RECVTO",<recv_timeout>**	Response 1)If parameter format is right: **OK** 2)If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="CONTENT",<content_type>**	Response 1)If parameter format is right: **OK** 2)If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="ACCEPT",<accept-type>**	Response 1)If parameter format is right: **OK** 2)If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="SSLCFG",<sslcfg_id>**	Response 1)If parameter format is right: **OK** 2)If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="USERDATA",<**	Response 1)If parameter format is right:

user_data>	**OK** 2)If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="READMODE", <readmode>**	Response 1)If parameter format is right: **OK** 2)If parameter format is not right or other errors occur: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

<url>	URL of network resource. String type, 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)
<conn_timeout>	Timeout for accessing server. Numeric type, range is 20-120s, default is 120s.
<recv_timeout>	Timeout for receiving data from the server. Numeric type, range is 2s-120s, default is 20s.
<content_type>	This is for HTTP "Content-Type" tag. String type, max length is 256, and default is "text/plain".
<accept-type>	This is for HTTP "Accept-type" tag. String type, max length is 256, and default is "*/*".
<sslcfg_id>	This is setting SSL context id. Numeric type, range is 0-9. Default is 0.
<user_data>	The customized HTTP header information. String type, max length is 256.
<readmode>	For HTTPREAD. Numeric type, it can be set to 0 or 1. If set to 1, you can read the response content data from the same position repeatedly. The limit is that the size of HTTP server response content should be shorter than 1M.Default is 0.

Examples

```
AT+HTTPPARA="URL","http://www.baidu.com"
OK
AT+HTTPPARA="CONTENT","mutipart/form-da    //"mutipart/form-data" of <content-type> can be
```

ta" OK	used to transfer data. It will construct boundary header.

2.2.4 AT+HTTPACTION HTTP Method Action

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

AT+HTTPACTION HTTP Method Action	
Test Command AT+HTTPACTION=?	Response **+HTTPACTION: (0-4)** **OK**
Write Command AT+HTTPACTION=<method>	Response 1)If parameter format is right: **OK** **+HTTPACTION: <method>,<statuscode>,<datalen>** 2)If parameter format is right but server connected unsuccessfully: **OK** **+HTTPACTION: <method>,<errcode>,<datalen>** 3)If parameter format is not right or other errors occur: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

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

Examples

```
AT+HTTPACTION=?
+HTTPACTION: (0-4)

OK
AT+HTTPACTION=0
OK

+HTTPACTION: 0,200,104220
```

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

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

AT+HTTPHEAD Read the HTTP Header Information of Server Response	
Test Command **AT+HTTPHEAD=?**	Response **OK**
Execute Command **AT+HTTPHEAD**	Response 1)If read the header information successfully: **+HTTPHEAD: <data_len>** **<data>** **OK** 2)If read failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

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

Examples

```
AT+HTTPHEAD
+HTTPHEAD: 653
HTTP/1.1 200 OK
Content-Type: text/html
Connection: keep-alive
X-Cache: MISS from PDcache-04:opinion.people.com.cn
Date: Tue, 24 Mar 2020 03:12:09 GMT
Powered-By-ChinaCache: HIT from CNC-WB-b-D24
Powered-By-ChinaCache: HIT from CNC-WV-b-D1C
ETag: W/"5b7379f5-57e9"
x-cc-via: CNC-WB-b-D24[H,1], CNC-WV-b-D1C[H,62]
d-cc-upstream: CNC-WV-b-D1C
CACHE: TCP_HIT
Vary: Accept-Encoding
Last-Modified: Wed, 15 Aug 2018 00:55:17 GMT
Expires: Tue, 24 Mar 2020 03:17:09 GMT
x-cc-req-id: f4b9e1793697d1ef2950f530aeec4519
Content-Length: 22505
Age: 0
Accept-Ranges: bytes
Server: nginx
X-Frame-Options: ALLOW-FROM .*
CC_CACHE: TCP_REFRESH_HIT
OK
```

2.2.6 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 the parameter <byte_size> is set bigger than the size of the data stored in the buffer, then all data in the cache will be output to the port.

AT+HTTPREAD Read the response information of HTTP Server	
Test Command **AT+HTTPREAD=?**	Response **OK**
Read Command **AT+HTTPREAD?**	Response 1)If check successfully: **+HTTPREAD: LEN,<len>** **OK**

	2)If failed (no more data other error): **ERROR**
	Response 1)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. 2)If read failed: **ERROR**
Write Command **AT+HTTPREAD=[<start_offset>,]<byte_size>**	
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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.

Examples

```
AT+HTTPREAD?
+HTTPREAD: LEN,22505

OK
AT+HTTPREAD=0,500
OK

+HTTPREAD: 500
\0\0\0\0\0\0\0\0\0\0\0\0\0<!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>People's Daily Bell: Remembering History is for Better Creating the Future--Viewpoint--
```

```
People's Daily</title>
<meta name="keywords" content="" />
<meta name="description" content="      The Japanese side should be in the correct treatment of history?
+HTTPREAD: 0
```

NOTE

The response content received from the server will be stored in the cache, and will not be cleaned up by ***AT+HTTPREAD***.
When the total size of the data from the server is bigger than that and 'READMODE' is 0, you should read the data quickly, or you will fail to read it.

2.2.7 AT+HTTPDATA Input HTTP Data

AT+HTTPDATA can be used to input data to post when sending a HTTP/HTTPS POST request.

AT+HTTPDATA Input HTTP Data	
Test Command **AT+HTTPDATA=?**	Response **OK**
Write Command **AT+HTTPDATA=<size>,<time>**	Response 1)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** 2)If parameter format is wrong or other errors occur: **ERROR**
Parameter Saving Mode	
Max Response Time	
Reference	

Defined Values

<size>	Size in bytes of the data to post, range is 1- 153600 (bytes)

<time>	Maximum time in seconds to input data, range is 10-65535

Examples

```
AT+HTTPDATA=18,1000
DOWNLOAD
Message=helloworld
OK
```

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

You can also 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)[,(0-4)[,(0-1)]]]** **OK**
Write Command **AT+HTTPPOSTFILE=<filename>[,<path>[,<method>[,<send_header>]]]**	Response 1)if parameter format is right and server connected successfully: a)if parameter <method> is valid: **OK** **+HTTPPOSTFILE: <method>,<statuscode>,<content_len>** b)if parameter <method> is ignored: **OK** **+HTTPPOSTFILE: <statuscode>,<content_len>** 2)if parameter format is right but server connected unsuccessfully: a)if parameter <method> is valid: **OK** **+HTTPPOSTFILE: <method>,<errcode>,0** b)if parameter <method> is ignored: **OK** **+HTTPPOSTFILE: <errcode>,0**

	3)if parameter format is not right or any other error occurs: **ERROR**
Parameter Saving Mode	
Max Response Time	
Reference	

Defined Values

<filename>	String type, filename. Unit: byte. The max length is 55.
<path>	The directory where the sent file saved. Numeric type, range is 1-2 1 C:/ (local storage) 2 Reserved.
<method>	HTTP method specification: 0 GET 1 POST 2 HEAD 3 DELETE 4 PUT If this value is not provided, it is same to the value described in the post file.
<send_header>	Send file as HTTP header and Body or Only as Body. Numeric type, the range is 0-1, the default is 0. 0 Send file as HTTP header and Body 1 Send file as Body

Examples

```
AT+HTTPPOSTFILE=?
+HTTPPOSTFILE: <filename>[,(1-2)[,(0-4)[,(0-1)]]]

OK
AT+HTTPPOSTFILE="getbaidu.txt",1
OK

+HTTPPOSTFILE: 200,14615
AT+HTTPPOSTFILE="getbaidu.txt",1,1,1
OK

+HTTPPOSTFILE: 1,200,14615
```

2.2.9 AT+HTTPREADFILE Receive HTTP Response Content to a file

After executing ***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 local storage.

AT+HTTPREADFILE Receive HTTP Response Content to a File	
Test Command **AT+HTTPREADFILE=?**	Response **+HTTPREADFILE: <filename>[,(1-2)]** **OK**
Write Command **AT+HTTPREADFILE=<filename>[,<path>]**	Response 1)if parameter format is right: **OK** **+HTTPREADFILE: <errcode>** 2)if failed: **OK** **+HTTPREADFILE: <errcode>** 3)if parameter format is not right or any other error occurs: **ERROR**
Parameter Saving Mode	
Max Response Time	
Reference	

Defined Values

<filename>	String type, filename. Unit: byte. The max length is 55.
<path>	The directory where the read file saved. Numeric type, range is 1-2. 1 C:/(local storage) 2 Reserved.

Examples

```
AT+HTTPREADFILE=?
+HTTPREADFILE: <filename>[,(1-2)]
```

```
OK
AT+HTTPREADFILE="readbaidu.dat"
OK

+HTTPREADFILE: 0
```

3 HTTP(S) Examples

Before all HTTP(S) related operations, we should ensure the following:
Ensure network is available:

```
AT+CSQ
+CSQ: 23,0

OK
AT+CPSI?
+CPSI:
LTE,Online,460-00,0x333C,39589680,308,EUT
RAN-BAND3,1350,5,0,0,54,0,22

OK
AT+CGACT?
+CGACT: 1,1

OK
```

3.1 Access to HTTP server

3.1.1 Send HTTP GET Request

AT+HTTPINIT	//start HTTP service, activate PDP context
OK	
AT+HTTPPARA="URL","http://opinion.people.com.cn/GB/n1/2018/0815/c1003-30228758.html"	//set the URL which will be accessed, for HTTP, the request URL begins with “HTTP://”
OK	
AT+HTTPACTION=0	//send HTTP GET request
OK	
+HTTPACTION: 0,200,22505	//22505 is the length of HTTP response information
AT+HTTPHEAD	//read HTTP response header

+HTTPHEAD: 387 **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** **CC_CACHE: TCP_REFRESH_HIT** **Accept-Ranges: bytes** **OK**	//387 is the length of response header //Content-Length indicates the length of HTTP response information is 22505 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>People's Daily Bell: Remembering History is for Better Creating the Future--Viewpoint--People's Daily </title>** **<meta name="keywords" content="" />** **<meta name="description" content=" The Japanese side should be in the correct treatment of history?** **+HTTPREAD: 0**	//read the response information from HTTP server, with a read length of 500 bytes
AT+HTTPTERM **OK**	//stop HTTP service

3.1.2 Send HTTP POST Request

```
AT+HTTPINIT                                  //start HTTP service, activate PDP context
OK
AT+HTTPPARA="URL","http://api.efxnow.com/    //set the URL which will be accessed, for HTTP, the
DEMOWebServices2.8/Service.asmx/Echo?"       request URL begins with “HTTP://”
OK
AT+HTTPDATA=18,1000                          //send data to post, the length is 18 bytes
DOWNLOAD                                     //prompt string indicating that data can be entered
Message=helloworld                           here
OK
AT+HTTPACTION=1                              //send HTTP POST request
OK

+HTTPACTION: 1,500,30
AT+HTTPHEAD                                  //read 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
AT+HTTPREAD=0,30
OK

+HTTPREAD: 30
Request format is invalid: .
+HTTPREAD: 0
AT+HTTPTERM
OK
```

3.1.3 Send HTTP HEAD Request

```
AT+HTTPINIT                                  //start HTTP service, activate PDP context
OK
AT+HTTPPARA="URL","http://opinion.people.c
```

```
om.cn/GB/n1/2018/0815/c1003-30228758.html"
OK
AT+HTTPACTION=2                          //send a HEAD request to the server to obtain only
OK                                       the header of HTTP response

+HTTPACTION: 2,200,387

+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

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

+HTTPPOSTFILE: 200,14615
AT+HTTPHEAD                              //read HTTP response header
+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=A374BCFD28DFEEAF0BA0C4EEAC7
7B0B0:FG=1; expires=Thu, 31-Dec-37 23:55:55
GMT; max-age=2147483647; path=/;
domain=.baidu.com
Set-Cookie:
BIDUPSID=A374BCFD28DFEEAF0BA0C4EEAC
77B0B0; 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
AT+HTTPREADFILE="readbaidu.dat"          //read HTTP server response content to a file
OK                                       named readbaidu.dat, saved to local storage

+HTTPREADFILE: 0
AT+HTTPTERM                              //stop HTTP service
OK
```

3.2 Access to HTTPS server

3.2.1 Send HTTPS GET Requst

AT+CSSLCFG="sslversion",0,4	//set SSL version for the first SSL context
OK	
AT+CSSLCFG="authmode",0,1	//set the authentication mode(verify server) for the first SSL context
OK	
AT+CSSLCFG="cacert",0,"server_ca.pem"	//set the server root CA for the first SSL context
OK	
AT+HTTPINIT	//start HTTPS service, activate PDP context
OK	
AT+HTTPPARA="URL","https://ss0.bdstatic.com/5aV1bjqh_Q23odCf/static/mancard/css/card_min_dee38e45.css"	
OK	
AT+HTTPPARA="SSLCFG",0	
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 header
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	
AT+HTTPREAD=0,500	//read the response information from HTTPS server, with a read length of 500 bytes
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);positio	

```
n:absolute;border:none;display:none;_height:1
86px}.sui-dialog-cardsetting{opacity:.98!import
ant;filter:alpha(opacity=98)!important;border:n
one!important}.sui-dialog-cardsetting .sui-dialo
g-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 HTTPS service
OK
```

3.2.2 Send HTTPS POST Requst

```
AT+CSSLCFG="sslversion",0,4                    //set SSL version for the first SSL context
OK
AT+CSSLCFG="authmode",0,2                      //set the authentication mode(verify server and
OK                                             client) for the first SSL context
AT+CSSLCFG="cacert",0,"ca_cert.pem"            //set the server root CA for the first SSL context
OK
AT+CSSLCFG="clientcert",0,"cert.pem"           //set the client certificate for the first SSL context
OK
AT+CSSLCFG="clientkey",0,"key_cert.pem"        //set the client key for the first SSL context
OK
AT+HTTPINIT                                    //start HTTPS service, activate PDP context
OK
AT+HTTPPARA="URL","https://pv.csdn.net/csd     //set the URL which will be accessed, for HTTPS,
nbi"                                           the request URL begins with “HTTPS://”
OK
AT+HTTPPARA="SSLCFG",0
OK
AT+HTTPDATA=465,1000                           //send data to post, the length is 465 bytes
DOWNLOAD                                       //prompt string indicating that data can be entered
[{"headers":{"component":"enterprise","datata  here
pe":"track","version":"v1"},"body":"{\"re\":\"ui
d=merry1996&ref=https%3A%2F%2Fpassport.c
sdn.net%2Faccount%2Fverify%3Bjsessionid%
3D7895A57BC64CE8616517F558940FD913.tom
cat2&pid=www&mod=&con=&ck=-&curl=https
%3A%2F%2Fwww.csdn.net%2F&session_id=1
0_1534696351647.160829&tos=12&referrer=htt
ps%3A%2F%2Fpassport.csdn.net%2Faccount
%2Fverify%3Bjsessionid%3D7895A57BC64CE8
616517F558940FD913.tomcat2&user_name=me
```

```
rry1996&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 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_37481894210-1534735230305-44
5993; 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
AT+HTTPREAD=0,10                         //read the response information from HTTPS
OK                                       server, with a read length of 10 bytes

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

3.2.3 Send HTTPS HEAD Requst

```
AT+CSSLCFG="sslversion",0,4              //set SSL version for the first SSL context
OK
AT+CSSLCFG="authmode",0,2                //set the authentication mode(verify server and
OK                                       client) for the first SSL context
AT+CSSLCFG="cacert",0,"ca_cert.pem"      //set the server root CA for the first SSL context
OK
AT+CSSLCFG="clientcert",0,"cert.pem"     //set the client certificate for the first SSL context
```

```
OK
AT+CSSLCFG="clientkey",0,"key_cert.pem"          //set the client key for the first SSL context
OK
AT+HTTPINIT                                      //start HTTPS service, activate PDP context
OK
AT+HTTPPARA="URL","https://ss0.bdstatic.co       //set the URL which will be accessed, for HTTPS,
m/5aV1bjqh_Q23odCf/static/mancard/css/card_      the request URL begins with “HTTPS://”
min_dee38e45.css"
OK
AT+HTTPPARA="SSLCFG",0
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 HTTPS service
OK
```

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

```
AT+CSSLCFG="sslversion",0,4                      //set SSL version for the first SSL context
OK
```

```
AT+CSSLCFG="authmode",0,1                    //set the authentication mode(verify server) for the
OK                                           first SSL context
AT+CSSLCFG="cacert",0,"server_ca.pem"        //set the server root CA for the first SSL context
OK
AT+HTTPINIT                                  //start HTTPS service, activate PDP context
OK
AT+HTTPPARA="URL","https://www.baidu.com     //set the server URL
"
OK
AT+HTTPPARA="SSLCFG",0
OK
AT+HTTPPOSTFILE="getbaidu.txt",1             //access the server and send the file getbaidu.txt to
OK                                           the server

+HTTPPOSTFILE: 200,14615
AT+HTTPHEAD                                  //read HTTPS response header .
+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=A374BCFD28DFEEAF0BA0C4EEAC7
7B0B0:FG=1; expires=Thu, 31-Dec-37 23:55:55
GMT; max-age=2147483647; path=/;
domain=.baidu.com
Set-Cookie:
BIDUPSID=A374BCFD28DFEEAF0BA0C4EEAC
77B0B0; 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
```

```
AT+HTTPREADFILE="readbaidu.dat"
OK
AT+HTTPTERM                                    //stop HTTPS service
OK
```

4 Appendix

4.1 Summary of <statuscode>

<statuscode>	Meaning
100	Continue
101	Switching Protocols
200	OK
201	Created
202	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
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

4.2 Summary of <errcode>

<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

4.3 Unsolicited Result Codes

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