SIM7070_SIM7080_SIM7090 Series_HTTP(S)_Application Note

LPWA Module
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 WIRELESS SOLUTIONS LIMITED. COPYING, TO OTHERS AND USING THIS DOCUMENT, ARE FORBIDDEN WITHOUT EXPRESS AUTHORITY BY SIMCOM. OFFENDERS ARE LIABLE TO THE PAYMENT OF INDEMNIFICATIONS. ALL RIGHTS RESERVED BY SIMCOM IN THE PROPRIETARY TECHNICAL INFORMATION, INCLUDING BUT NOT LIMITED TO REGISTRATION GRANTING OF A PATENT, A UTILITY MODEL OR DESIGN. ALL SPECIFICATIONS SUPPLIED HEREIN ARE SUBJECT TO CHANGE WITHOUT NOTICE AT ANY TIME.

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633 Jinzhong Road, Changning District, Shanghai P.R. China
Tel: +86 21 31575100
Email: simcom@simcom.com

For more information, please visit:
https://www.simcom.com/download/list-863-en.html

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

Copyright © 2020 SIMCom Wireless Solutions Limited All Rights Reserved.
About Document

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Owner</th>
<th>What is new</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.00</td>
<td>2019.9.16</td>
<td>Jiangting.Ding</td>
<td>First Release</td>
</tr>
<tr>
<td>V1.01</td>
<td>2019.11.7</td>
<td>Jiangting.Ding</td>
<td>Change AT+SHBOD</td>
</tr>
<tr>
<td>V1.02</td>
<td>2020.7.8</td>
<td>Wenjie.Lai</td>
<td>All</td>
</tr>
</tbody>
</table>

Scope

This document applies to the following products

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size(mm)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM7080G</td>
<td>CAT-M/NB</td>
<td>17.6<em>15.7</em>2.3</td>
<td>N/A</td>
</tr>
<tr>
<td>SIM7070G/SIM7070E</td>
<td>CAT-M/NB/GPRS</td>
<td>24<em>24</em>2.4</td>
<td>N/A</td>
</tr>
<tr>
<td>SIM7070G-NG</td>
<td>NB/GPRS</td>
<td>24<em>24</em>2.4</td>
<td>N/A</td>
</tr>
<tr>
<td>SIM7090G</td>
<td>CAT-M/NB</td>
<td>14.8<em>12.8</em>2.0</td>
<td>N/A</td>
</tr>
</tbody>
</table>
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Document</td>
<td>3</td>
</tr>
<tr>
<td>Version History</td>
<td>3</td>
</tr>
<tr>
<td>Scope</td>
<td>3</td>
</tr>
<tr>
<td>Contents</td>
<td>4</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>5</td>
</tr>
<tr>
<td>1.1 Purpose of the document</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Related documents</td>
<td>5</td>
</tr>
<tr>
<td>1.3 Conventions and abbreviations</td>
<td>5</td>
</tr>
<tr>
<td><strong>2 HTTP Introduction</strong></td>
<td>6</td>
</tr>
<tr>
<td>2.1 Characteristic</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Request Method</td>
<td>7</td>
</tr>
<tr>
<td><strong>3 AT Commands for HTTP(S)</strong></td>
<td>8</td>
</tr>
<tr>
<td><strong>4 Bearer Configuration</strong></td>
<td>9</td>
</tr>
<tr>
<td>4.1 PDN Auto-activation</td>
<td>9</td>
</tr>
<tr>
<td>4.2 APN Manual Configuration</td>
<td>10</td>
</tr>
<tr>
<td><strong>5 HTTP(S) Examples</strong></td>
<td>12</td>
</tr>
<tr>
<td>5.1 HTTP Function</td>
<td>12</td>
</tr>
<tr>
<td>5.1.1 HTTP GET</td>
<td>12</td>
</tr>
<tr>
<td>5.1.2 HTTP POST</td>
<td>13</td>
</tr>
<tr>
<td>5.2 HTTPS Function</td>
<td>16</td>
</tr>
<tr>
<td>5.2.1 HTTPS download and convert SSL certificate</td>
<td>16</td>
</tr>
<tr>
<td>5.2.2 HTTPS GET</td>
<td>17</td>
</tr>
<tr>
<td>5.2.3 HTTPS POST</td>
<td>18</td>
</tr>
</tbody>
</table>
### 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

[2] SIM7070_SIM7080_SIM7090 Series_SSL_Application Note  
[3] RFC2616

#### 1.3 Conventions and abbreviations

In this document, the GSM engines are referred to as following term:
- ME (Mobile Equipment);
- MS (Mobile Station);
- TA (Terminal Adapter);
- DCE (Data Communication Equipment) or facsimile DCE (FAX modem, FAX board);

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

HTTP (HyperText Transfer Protocol) is an application layer protocol. When you browse a web page, the browser and the web server will send and receive data on the Internet through the HTTP protocol. HTTP is a stateless protocol based on request and response patterns. That is what we usually call Request/Response.

2.1 Characteristic

- Support client/server mode;
  - Simple and fast

When a client requests a service from a server, it only needs to pass the request method and path. Because the HTTP protocol is simple, the program size of the HTTP server is small, and the communication speed is fast.

- Flexible

HTTP allows the transfer of any type of data object. The type being transferred is marked by Content-Type.

- No connection

No connection means limiting the processing of only one request per link. After the server processes the client's request and receives the customer's response, the server disconnects the link. This way, the transmission time can be saved.

- Stateless

The HTTP protocol is a stateless protocol. Stateless means that the protocol has no memory for transaction processing. A lack of state means that if subsequent processing requires the previous information, it must be retransmitted, which may result in an increase in the amount of data transferred per connection. On the other hand, it responds faster when the server does not need previous information.
2.2 Request Method

According to the HTTP standard, HTTP requests can use a variety of request methods. HTTP 1.0 defines three request methods: the GET, POST, and HEAD methods. HTTP1.1 adds six new request methods: OPTIONS, PUT, PATCH, DELETE, TRACE, and CONNECT methods.

<table>
<thead>
<tr>
<th>No</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GET</td>
<td>Make a request to a specific resource.</td>
</tr>
<tr>
<td>2</td>
<td>HEAD</td>
<td>Ask the server for a response that is consistent with the GET request, except that the response body will not be returned. This method can obtain the meta information contained in the response message header without having to transmit the entire response content.</td>
</tr>
<tr>
<td>3</td>
<td>POST</td>
<td>Submit data to a specified resource for processing requests (such as submitting a form or uploading a file). The data is included in the request body. POST requests may result in the creation of new resources and/or modifications to existing resources.</td>
</tr>
<tr>
<td>4</td>
<td>PUT</td>
<td>Uploads its latest content to a specified resource location.</td>
</tr>
<tr>
<td>5</td>
<td>DELETE</td>
<td>Requests the server to delete the resource identified by the Request-URI.</td>
</tr>
<tr>
<td>6</td>
<td>CONNECT</td>
<td>The HTTP/1.1 protocol is reserved for proxy servers that can connect connections to pipes.</td>
</tr>
<tr>
<td>7</td>
<td>OPTIONS</td>
<td>Returns the HTTP request method supported by the server for a particular resource. You can also test the functionality of the server by sending a &quot;*&quot; request to the web server.</td>
</tr>
<tr>
<td>8</td>
<td>TRACE</td>
<td>Echoes requests received by the server, primarily for testing or diagnostics.</td>
</tr>
<tr>
<td>9</td>
<td>PATCH</td>
<td>It is a supplement to the PUT method for local updating of known resources.</td>
</tr>
</tbody>
</table>

The SIM7070_SIM7080_SIM7090 Series supports several methods: GET, POST, PUT, PATCH and HEAD.
# 3 AT Commands for HTTP(S)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CSSLCFG</td>
<td>Configure SSL parameters of a context identifier</td>
</tr>
<tr>
<td>AT+SHCONF</td>
<td>Set HTTP(S) Parameter</td>
</tr>
<tr>
<td>AT+SHSSL</td>
<td>Select SSL Configure</td>
</tr>
<tr>
<td>AT+SHCONN</td>
<td>HTTP(S) Connection</td>
</tr>
<tr>
<td>AT+SHBOD</td>
<td>Set Body</td>
</tr>
<tr>
<td>AT+SHAHEAD</td>
<td>Add Head</td>
</tr>
<tr>
<td>AT+SHPARA</td>
<td>Set HTTP(S) Para</td>
</tr>
<tr>
<td>AT+SHCPARA</td>
<td>Clear HTTP(S) Para</td>
</tr>
<tr>
<td>AT+SHHEAD</td>
<td>Clear Head</td>
</tr>
<tr>
<td>AT+SHSTATE</td>
<td>Query HTTP(S) Connection Status</td>
</tr>
<tr>
<td>AT+SHREQ</td>
<td>Set Request Type</td>
</tr>
<tr>
<td>AT+SHREAD</td>
<td>Read Response Value</td>
</tr>
<tr>
<td>AT+SHDISC</td>
<td>Disconnect HTTP(S)</td>
</tr>
<tr>
<td>AT+HTTPTOFS</td>
<td>Download file to ap file system</td>
</tr>
<tr>
<td>AT+HTTPTOFSRL</td>
<td>State of download file to ap file system</td>
</tr>
</tbody>
</table>

For detail information, please refer to “SIM7070_SIM7080_SIM7090 Series_AT Command Manual”.
4 Bearer Configuration

Usually module will register PS service automatically.

4.1 PDN Auto-activation

//Example of PDN Auto-activation.

```
AT+CPIN?                          //Check SIM card status
+CPIN:READY
OK
AT+CSQ                           //Check RF signal
+CSQ: 20,0
OK
AT+CGATT?                       //Check PS service. 1 indicates PS has attached.
+CGATT: 1
OK
AT+COPS?                        //Query Network information, operator and network.
+COPS: 0,0,"CHN-CT",9           //Mode 9 means NB-IOT network.
OK
AT+CGNAPN                       //Query the APN delivered by the network after the CAT-M or NB-IOT network is successfully registered.
+CGNAPN: 1,"ctnb"
"ctnb" is APN delivered by the CAT-M or NB-IOT network. APN is empty under the GSM network.
OK
AT+CNCFG=0,1,"ctnb"             //Before activation please use AT+CNCFG to set APN\user name\password if needed.
OK
AT+CNACT=0,1                    //Activate network, Activate 0th PDP.
OK
+APP PDP: 0,ACTIVE
```
4.2 APN Manual Configuration

If not attached automatically, could configure correct APN setting.

//Example of APN Manual configuration.

AT+CFUN=0 //Disable RF
+CPIN: NOT READY
OK

AT+CGDCONT=1,"IP","ctnb" //Set the APN manually. Some operators need to set APN first when registering the network.
OK

AT+CFUN=1 //Enable RF
OK

+CPIN: READY
AT+CGATT? //Check PS service. 1 indicates PS has attached.
+CGATT: 1
OK

AT+CGNAPN //Query the APN delivered by the network after the CAT-M or NB-IOT network is successfully registered.
+CGNAPN: 1,"ctnb" //"ctnb" is APN delivered by the CAT-M or NB-IOT network. APN is empty under the GSM network.
OK

AT+CNCFG=0,1,"ctnb" //Before activation please use AT+CNCFG to set APN\user name\password if needed.
OK

AT+CNCACT=0,1 //Activate network, Activate 0th PDP.
OK
<table>
<thead>
<tr>
<th>Command</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>+APP PDP: 0,ACTIVE</td>
<td>//Get local IP</td>
</tr>
<tr>
<td>AT+CNACT?</td>
<td></td>
</tr>
<tr>
<td>+CNACT: 0,1,&quot;10.94.36.44&quot;</td>
<td></td>
</tr>
<tr>
<td>+CNACT: 1,0,&quot;0.0.0.0&quot;</td>
<td></td>
</tr>
<tr>
<td>+CNACT: 2,0,&quot;0.0.0.0&quot;</td>
<td></td>
</tr>
<tr>
<td>+CNACT: 3,0,&quot;0.0.0.0&quot;</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>
## 5 HTTP(S) Examples

### 5.1 HTTP Function

#### 5.1.1 HTTP GET

```plaintext
// Example of HTTP GET.

AT+SHCONF="URL","http://httpbin.org"  // Set up server URL
OK
AT+SHCONF="BODYLEN",1024                     // Set HTTP body length, for range of max body length
OK
AT+SHCONF="HEADERLEN",350                   // Set HTTP head length, for range of max head length
OK
AT+SHCONN                                    // HTTP build
OK
AT+SHSTATE?                                   // Get HTTP status
+SHSTATE: 1                                   // +SHSTATE: 1": connected
+SHSTATE: 0                                   // +SHSTATE: 0": disconnected
OK
AT+SHCHEAD                                    // Clear HTTP header, because of http header is appended
OK
AT+SHAHEAD="User-Agent","curl/7.47.0"      // Add header content
OK
AT+SHAHEAD="Cache-control","no-cache"      // For detail, please refer to document "rfc2616"
OK
AT+SHAHEAD="Connection","keep-alive"       // Add header content
OK
AT+SHAHEAD="Accept","*/*"                  // For detail, please refer to document "rfc2616"
OK
AT+SHREQ="/get?user=jack&password=123",1      // Set request type is GET.
OK
+SHREQ: "GET",200,387                        // Get data size is 387.
AT+SHREAD=0,387                               // Read data length is 387.
```
OK

+SHREAD: 387
{
    "args": {
        "password": "123",
        "user": "jack"
    },
    "headers": {
        "Accept": "*/*",
        "Cache-Control": "no-cache",
        "Content-Length": "0",
        "Host": "httpbin.org",
        "User-Agent": "curl/7.47.0",
        "X-Amzn-Trace-Id": "Root=1-5ed706c8-99b97372ae6f043f805cf243"
    },
    "origin": "117.132.195.245",
    "url": "http://httpbin.org/get?user=jack&password=123"
}

AT+SHDISC
OK

//Disconnect HTTP connect

5.1.2 HTTP POST

//Example 1 of HTTP POST.

AT+SHCONF="URL","http://httpbin.org"
OK

AT+SHCONF="BODYLEN",1024
OK

AT+SHCONF="HEADERLEN",350
OK

AT+SHCONN
OK

AT+SHSTATE?
+SHSTATE: 1

OK

//Set up server URL

//Set HTTP body length

//Set HTTP head length

//HTTP build

//Get HTTP status
AT+SHCHEAD
OK
AT+SHAHEAD="Content-Type","application/x-www-form-urlencoded"
OK
AT+SHAHEAD="Cache-control","no-cache"
OK
AT+SHAHEAD="Connection","keep-alive"
OK
AT+SHAHEAD="Accept","*/*
OK
AT+SHCPARA
OK
AT+SHPARA="product","apple"
OK
AT+SHPARA="price","1"
OK
AT+SHREQ="/post",3
OK

+SHREQ: "POST",200,452
AT+SHREAD=0,452
OK

+SHREAD: 452
{
   "args": {},
   "data": "",
   "files": {},
   "form": {
      "price": "1",
      "product": "apple"
   },
   "headers": {
      "Accept": "*/*",
      "Cache-Control": "no-cache",
      "Content-Length": "21",
      "Content-Type": "application/x-www-form-urlencoded",
      "Host": "httpbin.org",
      "X-Amzn-Trace-Id": "Root=1-5ed633df-058feb6412204392e95333b2"
   },
   "json": null,
   "origin": "218.204.252.187",
   "args": {},
   "data": "",
   "files": {},
   "form": {
      "price": "1",
      "product": "apple"
   },
   "headers": {
      "Accept": "*/*",
      "Cache-Control": "no-cache",
      "Content-Length": "21",
      "Content-Type": "application/x-www-form-urlencoded",
      "Host": "httpbin.org",
      "X-Amzn-Trace-Id": "Root=1-5ed633df-058feb6412204392e95333b2"
   },
   "json": null,
   "origin": "218.204.252.187", //Clear HTTP header
   "Add header content
   "Add header content
   "Add header content
   "Add header content
   "Clear body content parameter
   "Add body content parameter
   "Add body content parameter
   "Set request type is POST
   "Get data size is 452.
   "Read data length is 452
   "The data content is follow “+SHREAD: 452"
"url": "http://httpbin.org/post"}

AT+SHDISC
OK  //Disconnect HTTP connect

//Example 2 of HTTP POST.

AT+SHCONF="URL","http://httpbin.org"
OK  //Set up server URL
AT+SHCONF="BODYLEN",1024
OK  //Set HTTP body length
AT+SHCONF="HEADERLEN",350
OK  //Set HTTP head length
AT+SHCONN
OK  //HTTP build
AT+SHSTATE?
+SHSTATE: 1
OK  //Get HTTP status
AT+SHCHE
OK  //Clear HTTP header
AT+SHAHEAD="Content-Type","application/x-www-form-urlencoded"
OK  //Add header content
AT+SHAHEAD="Cache-control","no-cache"
OK  //Add header content
AT+SHAHEAD="Connection","keep-alive"
OK  //Add header content
AT+SHAHEAD="Accept","*/*
OK  //Add header content
AT+SHBOD=29,10000
> 
{"title":"Hello http server"}
OK
AT+SHREQ="/post",3
OK  //Set request type is POST
+SHREQ: "POST",200,457
AT+SHREAD=0,457
OK  //Get data size is 457.
+SHREAD: 457
OK  //Read data length is 457
{  //The data content is follow "+SHREAD: 457"

"args": {},
"data": "\{"title\":\"Hello http server\"\}",
"files": {},
"form": {},
"headers": {
  "Accept": "*/*",
  "Cache-Control": "no-cache",
  "Content-Length": "29",
  "Content-Type": "application/json",
  "Host": "httpbin.org",
  "X-Amzn-Trace-Id": "Root=1-5ed63fa7-3dda07707b3f2ea63e092a3a"
},
"json": {
  "title": "Hello http server"
},
"origin": "218.204.252.187",
"url": "http://httpbin.org/post"

AT+SHDISC
OK //Disconnect HTTP connect

5.2 HTTPS Function

5.2.1 HTTPS download and convert SSL certificate

//Example of HTTPS download and convert SSL Certificate.

AT+CFSINIT
OK //Init FS AT command
AT+CFSWF=3,"httpbin_root_ca.cer",0,1492,1000
DOWNLOAD OK //After download, sent certificate file through the serial port. 1492 is certificate size.
AT+CFSTERM
OK //Free data buffer
AT+CSSLCFG="convert",2,"httpbin_root_ca.cer" //Conversion CA certificate format.
5.2.2 HTTPS GET

//Example of HTTPS GET.

AT+CSSLCFG="sslversion",1,3 OK  //Configure SSL/TLS version
AT+SHSSL=1,"httpbin_root_ca.cer" OK  //Set HTTP SSL Configure
AT+SHCONF="URL","https://httpbin.org" OK  //Set connect server parameter
AT+SHCONF="BODYLEN",1024 OK  //Set max body length
AT+SHCONF="HEADERLEN",350 OK  //Set max header length
AT+SHCONN OK  //Connect HTTPS server
AT+SHSTATE? +SHSTATE: 1 OK  //Get HTTP status
AT+SHCHEAD OK  //Clear HTTP header content
AT+SHAHEAD="User-Agent","curl/7.47.0" OK  //Add header content
AT+SHAHEAD="Cache-control","no-cache" OK  //Add header content
AT+SHAHEAD="Connection","keep-alive" OK  //Add header content
AT+SHAHEAD="Accept","*/*" OK  //Add header content
AT+SHREQ="/get?user=jack&password=123",1 OK  //Set request type is GET.
 AT+SHREAD=0,388 OK  //Get data size is 388
+SHREQ: "GET",200,388 AT+SHREAD=0,388 OK  //Read data length is 388
 //The data content is follow “+SHREAD: 388”
+SHREAD: 388
{
   "args": {
      "password": "123",
      "user": "jack"
   },
   "headers": {
      "Accept": "*/*",
      "Cache-Control": "no-cache",
      "Content-Length": "0",
      "Host": "httpbin.org",
      "User-Agent": "curl/7.47.0",
      "X-Amzn-Trace-Id": 
      "Root=1-5ed706c8-99b97372ae6f043f805cf243"
   },
   "origin": "117.132.195.245",
   "url": "https://httpbin.org/get?user=jack&password=123"
}

AT+SHDISC
OK  //Disconnect HTTP connect

5.2.3 HTTPS POST

//Example 1 of HTTPS POST.

AT+CSSLCFG="sslversion",1,3
OK  //Configure SSL/TLS version

AT+SHSSL=1,"baidu_root_ca.cer"
OK  //Set HTTP SSL Configure
     //if you would skip certificate verify, use
     AT+SHSSL=1,""
     instead

AT+SHCONF="URL","https://httpbin.org"
OK  //Set connect server parameter

AT+SHCONF="BODYLEN",1024
OK  //Set max body length

AT+SHCONF="HEADERLEN",350
OK  //Set max header length

AT+SHCONN
OK  //Connect HTTPS server
AT+SHSTATE?
+SHSTATE: 1

OK
AT+SHCHEAD
OK
AT+SHAHEAD="Content-Type","application/x-www-form-urlencoded"
OK
AT+SHAHEAD="Cache-control","no-cache"
OK
AT+SHAHEAD="Connection","keep-alive"
OK
AT+SHAHEAD="Accept","*/*"
OK
AT+SHCPARA
OK
AT+SHPARA="product","apple"
OK
AT+SHPARA="price","1"
OK
AT+SHREQ="/post",3
OK

+SHREQ: "POST",200,453
AT+SHREAD=0,453
OK

+SHREAD: 453
{
  "args": {},
  "data": "",
  "files": {},
  "form": {
    "price": "1",
    "product": "apple"
  },
  "headers": {
    "Accept": "*/*",
    "Cache-Control": "no-cache",
    "Content-Length": "21",
    "Content-Type": "application/x-www-form-urlencoded",
    "Host": "httpbin.org",
    "X-Amzn-Trace-Id": "Root=1-5ed633df-058feb6412204392e95333b2"
  }

//Set request type is POST
//Get data size is 453.

//Get HTTP status
//Clear HTTP header
//Add header content
//Add header content
//Add header content
//Add header content
//Clear body content para
//Add body content para
//Add body content para
//Read data length is 453
//The data content is follow "+SHREAD: 453"
"}
"json": null,
"origin": "218.204.252.187",
"url": "https://httpbin.org/post"
}

AT+SHDISC
OK
//Disconnect HTTP connect

//Example 2 of HTTPS POST.

AT+CSSLCFG="sslversion",1,3
OK
//Configure SSL/TLS version

AT+SHSSL=1,"baidu_root_ca.cer"
OK
//Set HTTP SSL Configure

AT+SHCONF="URL","https://httpbin.org"
OK
//if you would skip certificate verify, use
AT+SHSSL=1,""

AT+SHCONF="BODYLEN",1024
OK
//Set connect server parameter

AT+SHCONF="HEADERLEN",350
OK
//Set max body length

AT+SHCONN
OK
//Set max header length

AT+SHSTATE?

+SHSTATE: 1
OK
//Connect HTTPS server

//Get HTTP status

AT+SHCHEAD
OK
//Clear HTTP header

AT+SHAHEAD="Content-Type","application/json"
OK
//Add header content

AT+SHAHEAD="Cache-control","no-cache"
OK
//Add header content

AT+SHAHEAD="Connection","keep-alive"
OK
//Add header content

AT+SHAHEAD="Accept","*/*
OK
//Add header content

AT+SHBOD=29,10000
> "{"title":"Hello http server"}"
OK
//Set body content
AT+SHREQ="/post",3
OK

//Set request type is POST

+SHREQ: "POST",200,458
AT+SHREAD=0,458
OK

+SHREAD: 458
{
    "args": {},
    "data": "{"title":"Hello http server"}",
    "files": {},
    "form": {},
    "headers": {
        "Accept": "*/*",
        "Cache-Control": "no-cache",
        "Content-Length": "29",
        "Content-Type": "application/json",
        "Host": "httpbin.org",
        "X-Amzn-Trace-Id": "Root=1-5ed63fa7-3dda07707b3f2ea63e092a3a"
    },
    "json": {
        "title": "Hello http server"
    },
    "origin": "218.204.252.187",
    "url": "https://httpbin.org/post"
}

//Read data length is 458

//The data content is follow "+SHREAD: 458"

AT+SHDISC
OK

//Disconnect HTTP connect