



# SIM7000系列\_ HTTP(S)\_应用文档

LPWA 模组

芯讯通无线科技(上海)有限公司  
上海市长宁区金钟路633号晨讯科技大楼B座6楼  
电话: 86-21-31575100  
技术支持邮箱: support@simcom.com  
官网: www.simcom.com

名称:	SIM7000系列_HTTP(S)_应用文档
版本:	1.03
日期:	2020.07.28
状态:	已发布

## 版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

### 芯讯通无线科技(上海)有限公司

上海市长宁区金钟路 633 号晨讯科技大楼 B 座 6 楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

### 了解更多资料，请点击以下链接：

<http://cn.simcom.com/download/list-230-cn.html>

### 技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 [support@simcom.com](mailto:support@simcom.com)

版权所有 © 芯讯通无线科技(上海)有限公司 2020，保留一切权利。

## 关于文档

### 版本历史

版本	日期	作者	备注
1.00	2018-8-28	Xiaobao.qu	First Release
1.01	2019-01-23	Xiaobao.qu	Modified Charpter 3.3 and max length of some parameters
1.02	2019-12-25	Jiangting.ding	Added HTTP and HTTPS Samples
1.03	2019-07-28	Wenjie.Lai	All

### 适用范围

本文档适用于以下产品型号：

型号	类别	尺寸(mm)	备注
SIM7000E/C/A/G	Cat-M1(/NB1/EGPRS)	24*24	
SIM7000E-N SIM7000C-N	NB1	24*24	

# 目录

关于文档.....	3
版本历史.....	3
适用范围.....	3
目录.....	4
<b>1 介绍.....</b>	<b>5</b>
1.1 本文目的.....	5
1.2 参考文档.....	5
1.3 术语和缩写.....	5
<b>2 HTTP(S)功能描述.....</b>	<b>6</b>
2.1 特性.....	6
2.2 HTTP 请求方法.....	6
<b>3 HTTP(S)相关 AT 指令.....</b>	<b>8</b>
<b>4 Bearer 配置.....</b>	<b>9</b>
4.1 PDN 自激活.....	9
4.2 手动改变 APN 配置.....	10
<b>5 HTTP(S)示例.....</b>	<b>12</b>
5.1 HTTP 功能.....	12
5.1.1 HTTP GET 请求.....	12
5.1.2 HTTP POST 请求.....	13
5.2 HTTPS 功能.....	16
5.2.1 HTTPS 证书下载/转换/设置.....	16
5.2.2 HTTPS GET 请求.....	17
5.2.3 HTTPS POST 请求.....	18

# 1 介绍

## 1.1 本文目的

基于 AT 指令手册扩展，本文主要介绍 HTTP(S)的业务流程。  
参考此应用文档，开发者可以很快理解并快速开发相关业务。

## 1.2 参考文档

- [1] SIM7000 Series\_AT Command Manual
- [2] RFC 2616

## 1.3 术语和缩写

## 2 HTTP(S)功能描述

HTTP (HyperText Transfer Protocol)--超文本传输协议，HTTP 是应用层协议，当你浏览网页的时候，浏览器和 web 服务器之间就会通过 HTTP 协议在 Internet 上进行数据的发送和接收。HTTP 是一个基于请求、响应模式的，无状态的协议。即我们通常所说的 Request/Response。

### 2.1 特性

➤ 支持客户端/服务器模式：

✧ 简单快速

客户向服务器请求服务时，只需传送请求方法和路径。由于 HTTP 协议简单，使得 HTTP 服务器的程序规模小，因而通信速度很快；

✧ 灵活

HTTP 允许传输任意类型的数据对象。正在传输的类型由 Content-Type 加以标记；

✧ 无连接

无连接的含义是限制每次链接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开链接，采用这种方式可以节省传输时间

✧ 无状态

HTTP 协议是无状态协议。无状态是指协议对于事物处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能会导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

### 2.2 HTTP 请求方法

根据 HTTP 标准，HTTP 请求可以使用多种请求方法。

HTTP1.0 定义了三种请求方法：GET, POST 和 HEAD 方法。

HTTP1.1 新增了六种请求方法：OPTIONS、PUT、PATCH、DELETE、TRACE 和 CONNECT 方法。

序号	方法	描述
1	GET	请求指定的页面信息，并返回实体主体。
2	HEAD	类似于 GET 请求，只不过返回的响应中没有具体的内容，用于获取报头
3	POST	向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含

		在请求体中。POST 请求可能会导致新的资源的建立和/或已有资源的修改。
4	PUT	从客户端向服务器传送的数据取代指定的文档的内容。
5	DELETE	请求服务器删除指定的页面。
6	CONNECT	HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。
7	OPTIONS	允许客户端查看服务器的性能。
8	TRACE	回显服务器收到的请求，主要用于测试或诊断。
9	PATCH	是对 PUT 方法的补充，用来对已知资源进行局部更新 。

SIM7000 系列支持如下请求方式：GET、POST、PUT、PATCH、HEAD。

SIMCom  
Confidential

## 3 HTTP(S)相关 AT 指令

命令	描述
<b>AT+SHSSL</b>	Select SSL Configure
<b>AT+SHCONF</b>	Set HTTP(S) Parameter
<b>AT+SHCONN</b>	HTTP(S) Connection
<b>AT+SHBOD</b>	Set Body
<b>AT+SHBODEXT</b>	Set Extension Body
<b>AT+SHAHEAD</b>	Add Head
<b>AT+SHPARA</b>	Set HTTP(S) Para
<b>AT+SHCHEAD</b>	Clear Head
<b>AT+SHSTATE</b>	Query HTTP(S) Connection Status
<b>AT+SHREQ</b>	Set Request Type
<b>AT+SHREAD</b>	Read Response Value
<b>AT+SHDISC</b>	Disconnect HTTP(S)
<b>AT+HTTPTOFS</b>	Download file to ap file system
<b>AT+HTTPTOFSRL</b>	State of download file to ap file system

更多详情，请参考“SIM7000 Series\_AT Command Manual”文档。



## 4 Bearer 配置

模块开机机会自动激活 PDN 并获取 PS 业务地址。前提是数据卡和天线正常。

### 4.1 PDN 自激活

模块开机机会自动激活 PDN 并获取 PS 业务地址。前提是数据卡和天线正常。

//PDN 自动激活示例.

AT+CPIN?

//检查 SIM 卡状态

+CPIN:READY

OK

AT+CGDCONT=1,"IP",""

//在 CAT-M 或 NB-IOT 网络，如果需要可以在网络注册前配置 APN

//APN 设置为空，模块将采用基站下发的 APN（推荐用此方式）。使用 AT+CGAPN 可查看具体下发的 APN。

OK

AT+CSQ

//检查射频信号

+CSQ: 13,99

OK

AT+CGATT?

//检查是否成功注册 PS 服务.

+CGATT: 1

//1 表示已经注册成功

OK

AT+COPS?

//查询网络信息，运营商及网络制式

+COPS: 0,0,"CHN-CT",9

//9 即 NB-IOT 网络

OK

AT+CGNAPN

//查询网络下发 APN 参数。

+CGNAPN: 1,"ctnb"

OK

AT+CNCFG=1,"ctnb","cdma","1234"

//如果需要的话激活之前请使用 AT+CNCFG 设置

OK

AT+CNACT=1

OK

+APP PDP: ACTIVE

AT+CNACT?

+CNACT: 1,"10.94.36.44"

OK

APN\用户名\密码等。

在 CAT-M 或 NB-IOT 网络，如果设置的 APN 和 CGNAPN 查询的 APN 相同或者 APN 设置为空,将使用默认的 PDN，否则将会向网络激活一路新的 PDN。

//在 GSM 网络下，此 APN 必须向运营商查询

//激活网络承载

//查询注册网络成功后分配的 IP 地址

## 4.2 手动改变 APN 配置

若有需要改变 APN 配置的情景，请参照如下步骤。

//APN 配置示例.

AT+CFUN=0

+CPIN: NOT READY

OK

AT+CGDCONT=1,"IP","ctnb"

//关闭 RF

//在 CAT-M 或 NB-IOT 网络，如果需要可以在网络注册前配置 APN。

//使用 AT+CGAPN 可查看网络下发的 APN。

OK

AT+CFUN=1

OK

//打开 RF

+CPIN: READY

AT+CGATT?

+CGATT: 1

//检查是否成功注册 PS 服务

//1 表示已经注册成功

OK

AT+CGNAPN

+CGNAPN: 1,"ctnb"

//在 CAT-M 或 NB-IOT 网络注册成功后查询网络下发的 APN。GSM 网络下 APN 为空。

OK

**AT+CNCFG=1,"ctnb","cdma","1234"**

//如果需要的话激活之前请使用 AT+CNCFG 设置 APN\用户名\密码等。

//在 CAT-M 或 NB-IOT 网络，如果设置的 APN 和 CGNAPN 查询的 APN 相同或者 APN 设置为空，将使用默认的 PDN，否则将会向网络激活一路新的 PDN。

OK

**AT+CNACT=1**

//激活网络承载。

OK

**+APP PDP: ACTIVE**

**AT+CNACT?**

//查询注册网络成功后分配的 IP 地址

**+CNACT: 0,1,"10.94.36.44"**

OK

## 5 HTTP(S)示例

### 5.1 HTTP 功能

#### 5.1.1 HTTP GET 请求

//HTTP GET 请求示例

```
AT+SHCONF="URL","http://httpbin.org" //设置 HTTP(S)服务器地址
OK
AT+SHCONF="BODYLEN",1024 //设置 body 的最大长度
OK
AT+SHCONF="HEADERLEN",350 //设置 header 的最大长度
OK
AT+SHCONN //连接 HTTP(S)服务器
OK
AT+SHSTATE?
+SHSTATE: 1 //获取 HTTP(S)状态
OK
AT+SHCHEAD //清除 HTTP 请求信息头
OK
AT+SHAHEAD="User-Agent","curl/7.47.0" //增加 HTTP 请求信息头内容
OK
AT+SHAHEAD="Cache-control","no-cache" //增加 HTTP 请求信息头内容
OK
AT+SHAHEAD="Connection","keep-alive" //增加 HTTP 请求信息头内容
OK
AT+SHAHEAD="Accept","*/*" //增加 HTTP 请求信息头内容
OK
AT+SHREQ="/get?user=jack&password=123", //发送 GET 请求。
1
OK
+SHREQ: "GET",200,387 //获取的数据长度为 387.
```

**AT+SHREAD=0,387**

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=1
23"
}
```

//读取数据的长度为 387

//数据内容紧随“+SHREAD: 387”之后

**AT+SHDISC**

OK

//断开 HTTP(S)连接

### 5.1.2 HTTP POST 请求

// HTTP POST 请求示例 1

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

OK

//设置 HTTP(S)服务器地址

**AT+SHCONF="BODYLEN",1024**

OK

//设置 body 的最大长度

**AT+SHCONF="HEADERLEN",350**

OK

//设置 header 的最大长度

**AT+SHCONN**

OK

//连接 HTTP(S)服务器

**AT+SHSTATE?**

**+SHSTATE: 1**

//获取 HTTP(S)状态

```

OK
AT+SHCHEAD                                     //清除 HTTP 请求信息头
OK
AT+SHAHEAD="Content-Type","application/x-      //增加 HTTP 请求信息头内容
www-form-urlencoded"
OK
AT+SHAHEAD="Cache-control","no-cache"         //增加 HTTP 请求信息头内容
OK
AT+SHAHEAD="Connection","keep-alive"         //增加 HTTP 请求信息头内容
OK
AT+SHAHEAD="Accept","/*/*"                   //增加 HTTP 请求信息头内容
OK
AT+SHCPARA                                     //清除 HTTP 请求信息 para 内容
OK
AT+SHPARA="product","apple"                  //增加 HTTP 请求信息 para 内容
OK
AT+SHPARA="price","1"                        //增加 HTTP 请求信息 para 内容
OK
AT+SHREQ="/post",3
OK
//发送 POST 请求
//获取的数据长度为 452

+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
"
  },
}

```

//读取数据的长度为 452  
//数据内容紧随 “+SHREAD: 452” 之后

```
"json": null,
"origin": "218.204.252.187",
"url": "http://httpbin.org/post"
}
```

**AT+SHDISC**

OK

//断开 HTTP(S)连接

// HTTP POST 请求示例 2

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

OK

//设置 HTTP(S)服务器地址

**AT+SHCONF="BODYLEN",1024**

OK

//设置 body 的最大长度

**AT+SHCONF="HEADERLEN",350**

OK

//设置 header 的最大长度

**AT+SHCONN**

OK

//连接 HTTP(S)服务器

**AT+SHSTATE?**

**+SHSTATE: 1**

//获取 HTTP(S)状态

OK

**AT+SHCHEAD**

OK

//清除 HTTP 请求信息头

**AT+SHAHEAD="Content-Type","application/js  
on"**

OK

//增加 HTTP 请求信息头内容

**AT+SHAHEAD="Cache-control","no-cache"**

OK

//增加 HTTP 请求信息头内容

**AT+SHAHEAD="Connection","keep-alive"**

OK

//增加 HTTP 请求信息头内容

**AT+SHAHEAD="Accept","/\*/\*"**

OK

//增加 HTTP 请求信息头内容

**AT+SHBOD="{\"title\":\"Hello http server\"}",29**

OK

//设置 HTTP 请求信息 body 内容

**AT+SHREQ="/post",3**

OK

//发送 POST 请求

//获取的数据长度为 457

**+SHREQ: "POST",200,457**

**AT+SHREAD=0,457**

OK

//读取数据的长度为 457

//数据内容紧随 “**+SHREAD: 457**” 之后

**+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

//断开 HTTP(S)连接

## 5.2 HTTPS 功能

### 5.2.1 HTTPS 证书下载/转换/设置

//HTTPS 下载转换证书示例

**AT+CFSINIT**  
OK

//初始化文件系统

**AT+CFSWFILE=3,"httpbin\_root\_ca.cer",0,1492**  
**,1000**

//"DOWNLOAD"提示之后，通过串口发送证书文件。

//1492 是证书文件长度。

**DOWNLOAD**

//发送证书文件成功。

OK

**AT+CFSTERM**

//释放数据缓冲区



OK	
AT+CSSLCFG="convert",2,"httpbin_root_ca.cer"	//转换证书格式。 //2 表示证书格式类型。
OK	//httpbin_root_ca.cer 是证书文件的名字。

## 5.2.2 HTTPS GET 请求

### //HTTPS GET 请求示例

AT+CSSLCFG="sslversion",1,3	
OK	//设置 SSL/TLS 版本
AT+SHSSL=1,"httpbin_root_ca.cer"	//设置 SSL 的 CA 证书
OK	//如果想跳过服务端证书合法性检查，可以使用 <b>AT+SHSSL=1,""</b>
AT+SHCONF="URL","https://httpbin.org"	
OK	//设置 HTTP(S)服务器地址
AT+SHCONF="BODYLEN",1024	
OK	//设置 body 的最大长度
AT+SHCONF="HEADERLEN",350	
OK	//设置 header 的最大长度
AT+SHCONN	
OK	//连接 HTTP(S)服务器
AT+SHSTATE?	
+SHSTATE: 1	//获取 HTTP(S)状态
OK	
AT+SHCHEAD	
OK	//清除 HTTP 请求信息头
AT+SHAHEAD="User-Agent","curl/7.47.0"	
OK	//增加 HTTP 请求信息头内容
AT+SHAHEAD="Cache-control","no-cache"	
OK	//增加 HTTP 请求信息头内容
AT+SHAHEAD="Connection","keep-alive"	
OK	//增加 HTTP 请求信息头内容
AT+SHAHEAD="Accept","*/*"	
OK	//增加 HTTP 请求信息头内容
AT+SHREQ="/get?user=jack&password=123",1	
OK	//发送 GET 请求。 //获取的数据长度为 388。
+SHREQ: "GET",200,388	
AT+SHREAD=0,388	//读取数据的长度为 388

OK

//数据内容紧随“+SHREAD: 388”之后

**+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

//断开 HTTP(S)连接

### 5.2.3 HTTPS POST 请求

// HTTPS POST 请求示例 1

**AT+CSSLCFG="sslversion",1,3**

OK

//设置 SSL/TLS 版本

**AT+SHSSL=1,"httpbin\_root\_ca.cer"**

OK

//设置 SSL 的 CA 证书

//如果想跳过服务端证书合法性检查，可以使用

**AT+SHSSL=1,""****AT+SHCONF="URL","https://httpbin.org"**

OK

//设置 HTTP(S)服务器地址

**AT+SHCONF="BODYLEN",1024**

OK

//设置 body 的最大长度

**AT+SHCONF="HEADERLEN",350**

OK

//设置 header 的最大长度

**AT+SHCONN**

//连接 HTTP(S)服务器

OK

AT+SHSTATE?

+SHSTATE: 1

//获取 HTTP(S)状态

OK

AT+SHCHEAD

OK

//清除 HTTP 请求信息头

AT+SHAHEAD="Content-Type","application/x-www-form-urlencoded"

//增加 HTTP 请求信息头内容

OK

AT+SHAHEAD="Cache-control","no-cache"

//增加 HTTP 请求信息头内容

OK

AT+SHAHEAD="Connection","keep-alive"

//增加 HTTP 请求信息头内容

OK

AT+SHAHEAD="Accept","\*/\*"

//增加 HTTP 请求信息头内容

OK

AT+SHCPARA

//清除 HTTP 请求信息 para 内容

OK

AT+SHPARA="product","apple"

//增加 HTTP 请求信息 para 内容

OK

AT+SHPARA="price","1"

//增加 HTTP 请求信息 para 内容

OK

AT+SHREQ="/post",3

OK

//发送 POST 请求

//获取的数据长度为 453

+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",

//读取数据的长度为 453

//数据内容紧随 “+SHREAD: 453” 之后

```
"X-Amzn-Trace-Id":
"Root=1-5ed633df-058feb6412204392e95333b2
"
},
"json": null,
"origin": "218.204.252.187",
"url": "https://httpbin.org/post"
}
```

AT+SHDISC

OK

//断开 HTTP(S)连接

// HTTPS POST 请求示例 2

AT+CSSLCFG="sslversion",1,3

OK

//设置 SSL/TLS 版本

AT+SHSSL=1,"httpbin\_root\_ca.cer"

OK

//设置 SSL 的 CA 证书

//如果想跳过服务端证书合法性检查，可以使用

AT+SHSSL=1,""

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

OK

//设置 HTTP(S)服务器地址

AT+SHCONF="BODYLEN",1024

OK

//设置 body 的最大长度

AT+SHCONF="HEADERLEN",350

OK

//设置 header 的最大长度

AT+SHCONN

OK

//连接 HTTP(S)服务器

AT+SHSTATE?

+SHSTATE: 1

//获取 HTTP(S)状态

OK

AT+SHCHEAD

OK

//清除 HTTP 请求信息头

AT+SHAHEAD="Content-Type","application/js  
on"

OK

//增加 HTTP 请求信息头内容

AT+SHAHEAD="Cache-control","no-cache"

OK

//增加 HTTP 请求信息头内容

AT+SHAHEAD="Connection","keep-alive"

OK

//增加 HTTP 请求信息头内容

AT+SHAHEAD="Accept","/\*/\*"

OK

//增加 HTTP 请求信息头内容

AT+SHBOD="{\"title\":\"Hello http server\"}";29

OK

//设置 HTTP 请求信息 body 内容

AT+SHREQ="/post",3

OK

//发送 POST 请求

//获取的数据长度为 458

+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"
}
```

//读取数据的长度为 458

//数据内容紧随“+SHREAD: 458”之后

AT+SHDISC

OK

//断开 HTTP(S)连接