

Ai-WB2 series module HTTP client POST request method

Content

1. Preparation of hardware and software
2. Connect gateway
3. HTTP client POST request method(Suitable for POST with a few data)
4. HTTP client POST request method(recommend)

1. Preparation of hardware and software

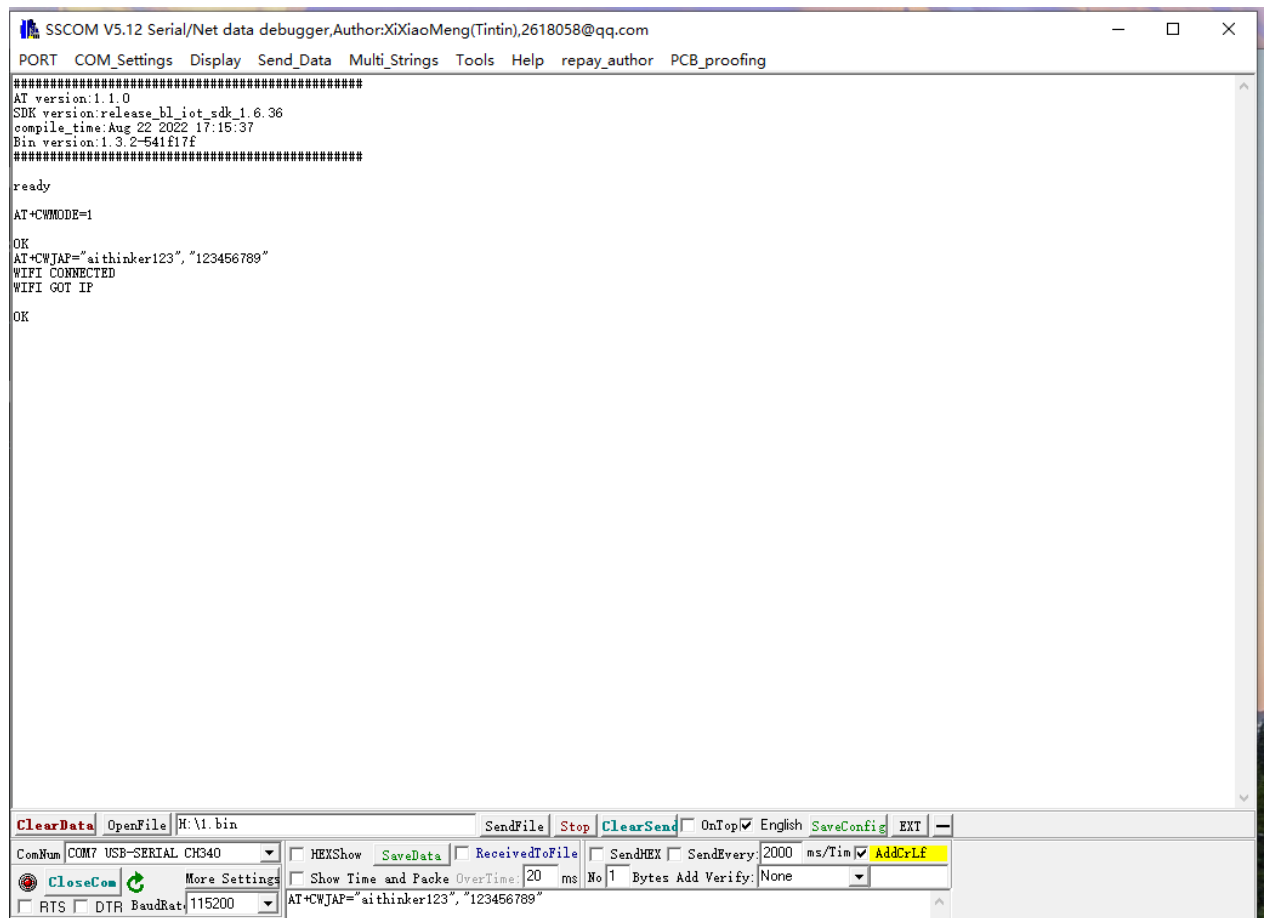
Hardware and software that need to be prepared: Ai-WB2 series module or development board, Type-C data cable

2. Connect gateway

1. First ensure the module or the development board power on and to print ready information, as shown below:



2. Send AT + CWMODE=1; AT + CWJAP="WIFI Name","WIFI password", like my current test of WIFI name is aithinker123, the password is 123456789, as shown below:



3. HTTP client POST request method(Suitable for POST with a few data)

The example uses <http://httpbin.org> as HTTP server, the data type is application/json.

1. Send a HTTP POST request.

- Set opt as 3 (POST method)
- URL as <http://httpbin.org/get>
- Content-type as 1 (application/json)
- transport_type as 1 (HTTP_TRANSPORT_OVER_TCP)

Command:

```
AT+HTTPCLIENT=3,1,"http://httpbin.org/post",1,{"form":{"purpose":"test"}}
```

as shown below:

```

SSCOM V5.12 Serial/Net data debugger,Author:XiXiaoMeng(Tintin),2618058@qq.com
PORT COM_Settings Display Send_Data Multi_Strings Tools Help repay_author PCB_proofing
*****
AT version:1.1.0
SDK version:release_bl_10t_sdk_1.6.36
compile_time:Aug 22 2022 17:16:37
Bin version:1.3.2-541f17f
*****
ready
AT+CNMODE=1
OK
AT+CNJAF="ai thinker123","123456789"
WIFI CONNECTED
WIFI GOT IP
OK
AT+HTTPCLIENT=3,1,"http://httpbin.org/post",,1,"{\form\:{\purpose\:\test\}}"
+HTTPCLIENT:262,{
  "args": {},
  "data": {},
  "files": {},
  "form": {
    "\form\:{\purpose\:\test\}": ""
  },
  "headers": {
    "Content-Length": "27",
    "Content-Type": "application/x-www-form-urlencoded",
    "Host": "httpbin.org",
    "User-Agent": "BL60X HTTP Client/1.0",
    "X-Amzn-Trace-Id": "Root=1-63213356-415b9e320b210dce622b3dc3"
  },
  "json": null,
  "origin": "112.97.59.24",
  "url": "http://httpbin.org/post"
}
OK

```

Note: The HTTP header information you get may be different from that in the above response.

4. HTTP client POST request method(recommend)

If you POST has relatively big data and already exceeds the length threshold of a single AT instruction of 256, it is recommended that you use the AT + HTTPCPOST command.

The example uses `http://httpbin.org` as the HTTP server, the data type is `application / json`. The Post specifies the length of the data. This command sets the number of HTTP header fields to 2, for the connection field and the content-type field, respectively, the connection field value is keep-alive, and the connection field value is `application / json`.

Suppose you want the JSON for post, the data is as follows, with 427 bytes in length.

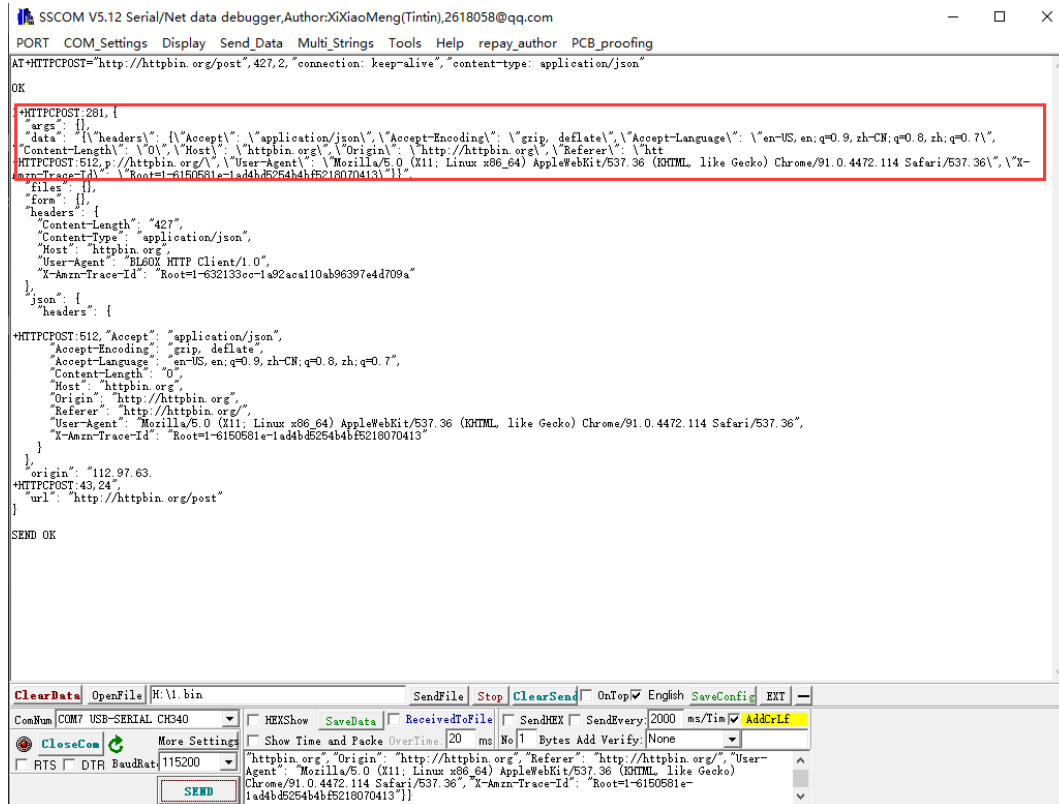
```

{"headers": {"Accept": "application/json","Accept-Encoding": "gzip, deflate","Accept-Language": "en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7","Content-Length": "0","Host": "httpbin.org","Origin": "http://httpbin.org","Referer": "http://httpbin.org/","User-Agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36","X-Amzn-Trace-Id": "Root=1-6150581e-1ad4bd5254b4bf5218070413"}}

```

Command: AT+HTTPPOST="http://httpbin.org/post",427,2,"connection: keep-alive","content-type: application/json"

As shown below:



```
SSCOM V5.12 Serial/Net data debugger,Author:XiXiaoMeng(Tintin),2618058@qq.com
PORT COM_Settings Display Send_Data Multi_Strings Tools Help repay_author PCB_proofing
AT+HTTPPOST="http://httpbin.org/post",427,2,"connection: keep-alive","content-type: application/json"
OK
+HTTPPOST:281,{
  "args": {},
  "data": {"headers": {"Accept": "application/json", "Accept-Encoding": "gzip, deflate", "Accept-Language": "en-US, en;q=0.9, zh-CN;q=0.8, zh;q=0.7",
  "Content-Length": "427", "Host": "httpbin.org", "Origin": "http://httpbin.org", "Referer": "http://httpbin.org/", "User-Agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36", "X-Amzn-Trace-Id": "Root=1-6150581e-1ad4b45254b4b5218070413"}},
  "files": {},
  "form": {},
  "headers": {
    "Content-Length": "427",
    "Content-Type": "application/json",
    "Host": "httpbin.org",
    "User-Agent": "RLGW HTTP Client/1.0",
    "X-Amzn-Trace-Id": "Root=1-632133cc-1a92aca110ab96397e4d709a"
  },
  "json": {
    "headers": {
      "Accept": "application/json",
      "Accept-Encoding": "gzip, deflate",
      "Accept-Language": "en-US, en;q=0.9, zh-CN;q=0.8, zh;q=0.7",
      "Content-Length": "0",
      "Host": "httpbin.org",
      "Origin": "http://httpbin.org",
      "Referer": "http://httpbin.org",
      "User-Agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36",
      "X-Amzn-Trace-Id": "Root=1-6150581e-1ad4b45254b4b5218070413"
    }
  },
  "origin": "112.97.63."
}
+HTTPPOST:43.24{
  "url": "http://httpbin.org/post"
}
SEND OK
```

Note:

After the AT output> characters, the special characters in the HTTP body do not need to escape the characters, or to end with a new line (CR-LF)