HTTP Client

# Introduction

This document provides details on using the HTTP client APIs provided by the HTTP module under components. This application demonstrates using these APIs to connect to HTTP servers in secured (HTTPS) and non-secured way.

The document also discusses about the GET and POST of HTTP methods and ways of executing it.

# Features and Limitations

The HTTP/S implementation supports:

1. HTTP 1.1 version. Both HTTP and HTTPS
2. HTTP client connection to the server specified by the IP address or Host name)
3. HTTP GET and POST methods
4. Setting of headers by the application. Only the Host header is implicitly set
5. Supports receiving the response with Transfer encoding Chunked.

Following are the limitations:

1. CA certificate must be in the PEM format.
2. HTTP redirection is not supported

# HTTP/S APIs

## HTTP/S Open:

This is performed using the following API:

|  |
| --- |
| http\_client\_handle\_t  http\_client\_open(http\_client\_config\_t \*cfg) |

This API connects to the remote HTTP server. The configuration needed for the connection is passed using the following data structure:

|  |
| --- |
| typedef struct {  char \*hostname;/\*\*<Host name or the IP address of the server\*/  int port;/\*\*<Server port\*/  int secured; /\*\* 0 – HTTP (non secure),  1 - HTTPS without server verification  2 – HTTPS with server certificate validation\*/  ssl\_wrap\_cfg\_t ssl\_cfg;/\*\*<SSL configuration\*/  int time\_out;/\*\*<Connect timeout in seconds\*/  } http\_client\_config\_t; |

## HTTP/S Get:

Once the connection is established using http\_client\_open(), GET is performed using the following API:

|  |
| --- |
| int  http\_client\_get(http\_client\_handle\_t handle, char \*uri,  http\_client\_resp\_cb cb, void \*cb\_ctx,  int time\_out) |

The HTTP response is provided through the call back. The call back is called multiple times until the response. Following is the call back definition and the data structure passed to call back:

|  |
| --- |
| typedef struct {  int status\_code;/\*\*< HTTP response status code\*/  char \*\*resp\_hdrs;/\*< Response headers. Array of strings\*/  char \*resp\_body; /\*\*< Response body len\*/  int resp\_len; /\*\*< Resp len, currently availabe in the resp\_body\*/  unsigned int resp\_total\_len;/\*\*< Total length of the response body. If 0,  No total length available before hand as the body  may be sent using chunked or multipart encoding\*/  int more\_data;  } http\_client\_resp\_info\_t;  /\*This is the call back called when the response is received\*/  typedef void http\_client\_resp\_cb(void \*ctx, http\_client\_resp\_info\_t \*resp); |

## HTTP/S Post:

Once the connection is performed using http\_client\_open(), data can be posted to HTTP server using the following API:

|  |
| --- |
| int  http\_client\_post(http\_client\_handle\_t handle, char \*uri,  char \*buff, int buff\_len,  http\_client\_resp\_cb cb, void \*cb\_ctx,  int time\_out); |

The response is provided using the call back. Setting content length header is a must before using this API.

## Setting the header

User application can set the header using the following API:

|  |
| --- |
| int  http\_client\_set\_req\_hdr(http\_client\_handle\_t handle,  const char \*hdrname, const char \*hdrval); |

## Closing the connection

The connection can be closed using the following API:

|  |
| --- |
| int  http\_client\_close(http\_client\_handle\_t handle); |

# Running the Application

## Programming Talaria TWO board with certificates

For HTTPS secure applications, CA certificates need to be provided. For Testing with httpbin.org, the certificate - httpbin\_ca.pem which is present in the location: examples/http\_client/cert can be used. This file can be written into the Talaria TWO File System using the Download Tool as mentioned in the subsequent sections. The default path for httpbin\_ca.pem in the File System should be: /data/certs/https\_client/app.

### Show File System Contents

Click on Show File System Contents to see the current available files in the file system.

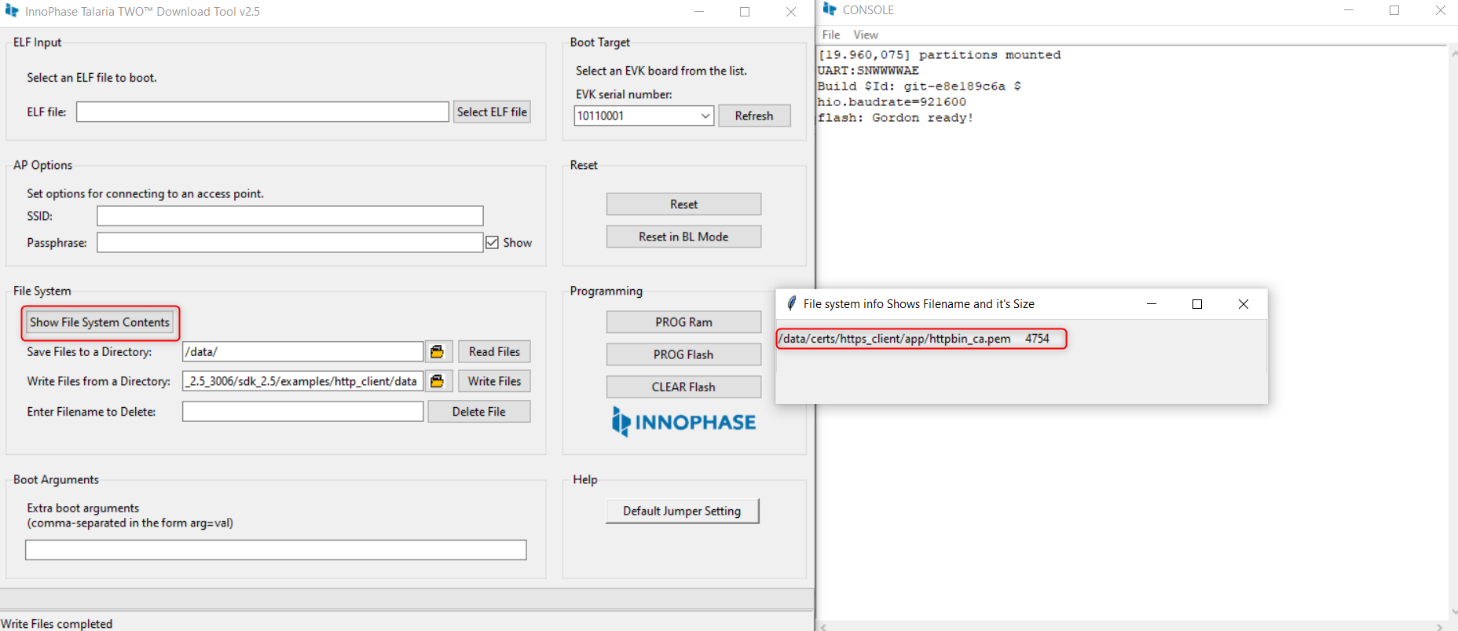


Figure : Show File System Content

### Write Files

Before writing the file(s) into Talaria TWO, user must create a folder with the name data and place the certificate either directly into the data or they can create multiple subfolders (for example: data/ or data/certs/https\_client/app ) and place the certificates inside the sub-directory. The certificate shall be present in the data fs with the same name as in the boot arguments.

To write files into Talaria TWO File System, use the Download Tool as shown in Figure 2. After clicking on “Select Path to Write Files”, select the data directory from the host in which the certificate is stored and then click on “Write Files” to write the certificate into the File System.

**Note:** When trying to access a secured web server, keep only the CA certificate specific to that server in data fs. Do not use bundle of CA certificates intended for a browser. Use use\_ca\_bundle=1 if bundled certificates are used.

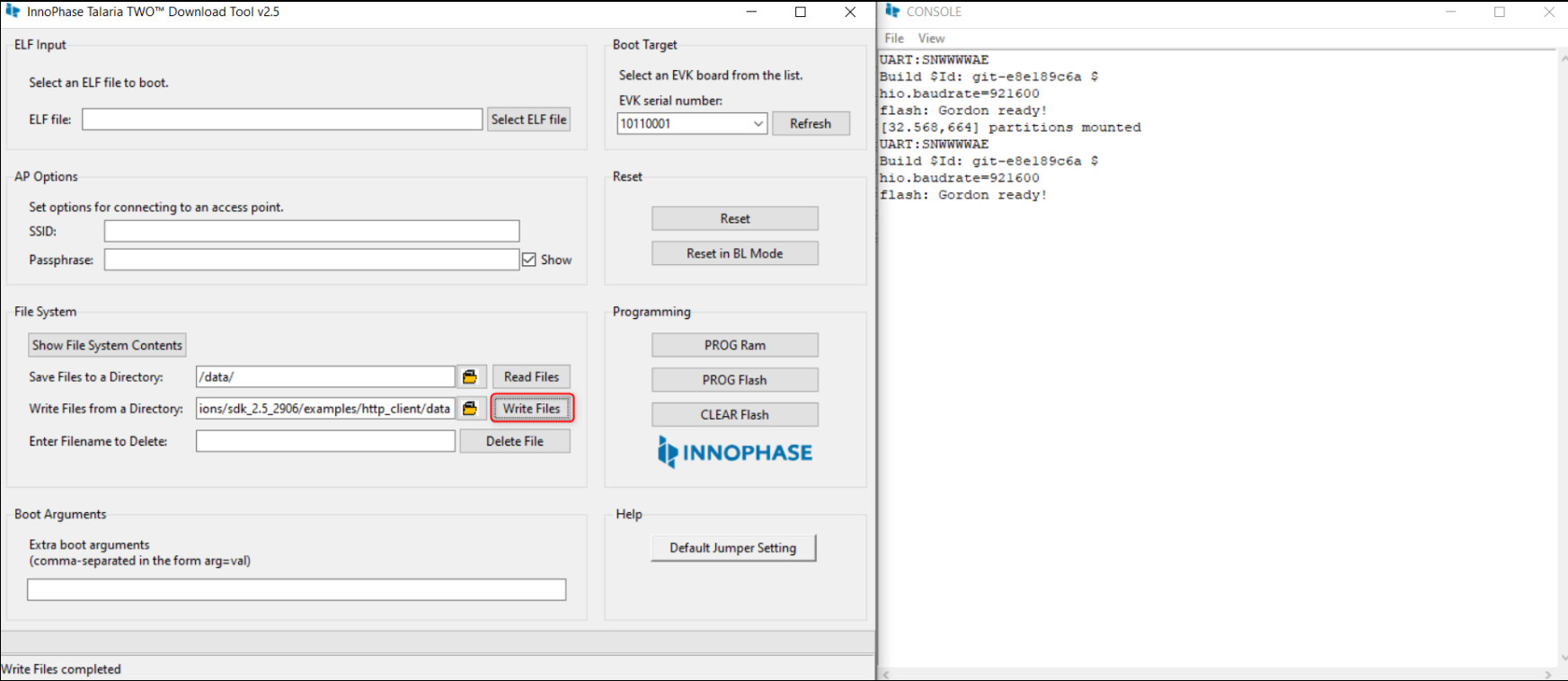


Figure : Write certificates to Talaria TWO

## Programming Talaria TWO board with ELF

Program http\_client.elf(sdk\_x.y\examples\http\_client\bin) using the Download tool:

1. Launch the Download tool provided with InnoPhase Talaria TWO SDK.
2. In the GUI window:
   1. Boot Target: Select the appropriate EVK from the drop-down
   2. ELF Input: Load the http\_client.elf by clicking on Select ELF File.
   3. AP Options: Provide the SSID and Passphrase under AP Options to connect to an Access Point.
   4. Boot arguments: Pass the following boot arguments:

|  |
| --- |
| host=httpbin.org,path=/json,port=80,secured=0,method=get,post\_len=3000 |

* 1. Programming: Prog RAM or Prog Flash as per requirement.

For more details on using the Download tool, refer to the document: UG\_Download\_Tool.pdf (path: *sdk\_x.y/pc\_tools/Download\_Tool/doc*).

**Note**: x and y refer to the SDK release version. For example: sdk\_2.5/doc.

# Using the Application

The list of boot arguments are as follows:

1. ssid: SSID of the Wi-Fi network to connect to
2. passphrase: Passphrase of the network
3. host: Domain name of the server or the IP address (dotted decimal format)
4. path: path of the file
5. port: Server port. 80 in case of non-secured (HTTP), 443 in case of secured (HTTPS)
6. secured:

0 - HTTP,

1. HTTPS without server verification
2. HTTPS with server certificate validation
3. method: Get/Post
4. ca\_cert: Certificate path in data FS in case of HTTPS
5. post\_len: Number of bytes to be sent as part of post data. The post data is internally generated

## Example boot args for HTTP Get (non-secure)

|  |
| --- |
| ssid=<ssid>,passphrase=<passphrase>,host=httpbin.org,path=/json,port=80,secured=0,method=get |

Console output:

|  |
| --- |
| UART:SNWWWWWAEBuild $Id: git-e8e189c6a $  host=httpbin.org path=/json port=80 secured=0 method=get np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE\_AP passphrase=innophaseae  $App:git-24f563c7  SDK Ver: sdk\_2.5  Http Client Demo App  Application Information:  ------------------------  Name : HTTP application  Version : 2.0  Build Date : Jun 30 2022  Build Time : 03:57:06  Heap Available: 333 KB (341400 Bytes)  [APP]Bootparams :  --------------------  url=<null>  host= httpbin.org  port=80  path= /json  secured= 0  method= get  ca\_cert=<null>  post\_len=<null>  test\_iterations = <null>  use\_ca\_bundle = <null>  hdr1\_name= <null> hdr1\_val= <null>  hdr2\_name= <null> hdr2\_val= <null>  hdr3\_name= <null> hdr3\_val= <null>  post\_data= <null>  post\_data\_file= <null>  [APP]Bootparams end here....  [APP]Bootparams check done....ret = 0  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE\_AP  [2.268,551] CONNECT:98:da:c4:73:b7:76 Channel:2 rssi:-34 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [4.141,896] MYIP 192.168.0.164  [4.142,290] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE\_AP  \*\* Test Iterations = 1 \*\*  [APP]Calling http\_client\_open(). heap size = 267064  [APP]Succes: HTTP connection done  [APP]HTTP Get  [APP]Response:  429 ----------------------  200  Date: Thu, 30 Jun 2022 11:00:36 GMT  Content-Type: application/json  Server: gunicorn/19.9.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Credentials: true  Accept-Ranges: none  Content-Length: 429  Via: HTTP/1.1 forward.http.proxy:3128  Connection: keep-alive  [APP]Body:  {  "slideshow": {  "author": "Yours Truly",  "date": "date of publication",  "slides": [  {  "title": "Wake up to WonderWidgets!",  "type": "all"  },  {  "items": [  "Why <em>WonderWidgets</em> are great",  "Who <em>buys</em> WonderWidgets"  ],  "title": "Overview",  "type": "all"  }  ],  "title": "Sample Slide Show"  }  }  [APP]Success: http\_client\_get(), rval = 0  [APP]------ Program Exit------------- |

## Example boot args for HTTPS Get (without server verification)

|  |
| --- |
| ssid=<ssid>,passphrase=<passphrase>,host=httpbin.org,path=/json,port=443,secured=1,method=get,ca\_cert= /data/httpbin\_ca.pem |

Console output:

|  |
| --- |
| UART:SNWWWWWAEBuild $Id: git-e8e189c6a $  host=httpbin.org path=/json port=443 secured=1 method=get ca\_cert=/data/httpbin\_ca.pem np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE\_AP passphrase=innophaseae  $App:git-24f563c7  SDK Ver: sdk\_2.5  Http Client Demo App  Application Information:  ------------------------  Name : HTTP application  Version : 2.0  Build Date : Jun 30 2022  Build Time : 03:57:06  Heap Available: 333 KB (341400 Bytes)  [APP]Bootparams :  --------------------  url=<null>  host= httpbin.org  port=443  path= /json  secured= 1  method= get  ca\_cert=/data/httpbin\_ca.pem  post\_len=<null>  test\_iterations = <null>  use\_ca\_bundle = <null>  hdr1\_name= <null> hdr1\_val= <null>  hdr2\_name= <null> hdr2\_val= <null>  hdr3\_name= <null> hdr3\_val= <null>  post\_data= <null>  post\_data\_file= <null>  [APP]Bootparams end here....  [APP]Bootparams check done....ret = 0  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE\_AP  [2.257,011] CONNECT:98:da:c4:73:b7:76 Channel:2 rssi:-35 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [4.167,747] MYIP 192.168.0.164  [4.168,026] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE\_AP  \*\* Test Iterations = 1 \*\*  [APP]Calling http\_client\_open(). heap size = 267480  . [SSL\_WRAP]Checking input configurations...  . [SSL\_WRAP]Seeding the random number generator...  . [SSL\_WRAP]Connecting to tcp httpbin.org:443...  . [SSL\_WRAP]Setting up the SSL/TLS structure...  . [SSL\_WRAP]setting configurations..  >auth mode = 0 (0- skip, 1- optional, 2- required  >max fragment len = 0  >Handshake timeout = 30 Sec  . [SSL\_WRAP]Performing the SSL/TLS handshake...  . [SSL\_WRAP] Handshake done. ok  . [SSL\_WRAP]Verifying peer X.509 certificate.  [APP]Succes: HTTP connection done  [APP]HTTP Get  [APP]Response:  429 ----------------------  200  Date: Thu, 30 Jun 2022 11:03:39 GMT  Content-Type: application/json  Content-Length: 429  Connection: keep-alive  Server: gunicorn/19.9.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Credentials: true  [APP]Body:  {  "slideshow": {  "author": "Yours Truly",  "date": "date of publication",  "slides": [  {  "title": "Wake up to WonderWidgets!",  "type": "all"  },  {  "items": [  "Why <em>WonderWidgets</em> are great",  "Who <em>buys</em> WonderWidgets"  ],  "title": "Overview",  "type": "all"  }  ],  "title": "Sample Slide Show"  }  }  [APP]Success: http\_client\_get(), rval = 0  [APP]------ Program Exit------------- |

## Example boot args for HTTPS Get (with server certificate validation)

|  |
| --- |
| ssid=<ssid>,passphrase=<passphrase>, host=httpbin.org,path=/json,port=443,secured=2,method=get,ca\_cert= /data/certs/https\_client/app/httpbin\_ca.pem, post\_len=3000 |

Console output:

|  |
| --- |
| UART:SNWWWWAE  Build $Id: git-e8e189c6a $  hio.baudrate=921600  flash: Gordon ready!  UART:SNWWWWWAEBuild $Id: git-e8e189c6a $  host=httpbin.org path=/json port=443 secured=2 method=get ca\_cert=/data/certs/https\_client/app/httpbin\_ca.pem post\_len=3000 np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE\_AP passphrase=innophaseae  $App:git-24f563c7  SDK Ver: sdk\_2.5  Http Client Demo App  Application Information:  ------------------------  Name : HTTP application  Version : 2.0  Build Date : Jun 30 2022  Build Time : 03:57:06  Heap Available: 333 KB (341400 Bytes)  [APP]Bootparams :  --------------------  url=<null>  host= httpbin.org  port=443  path= /json  secured= 2  method= get  ca\_cert=/data/certs/http\_client/app/httpbin\_ca.pem  post\_len=3000  test\_iterations = <null>  use\_ca\_bundle = <null>  hdr1\_name= <null> hdr1\_val= <null>  hdr2\_name= <null> hdr2\_val= <null>  hdr3\_name= <null> hdr3\_val= <null>  post\_data= <null>  post\_data\_file= <null>  [APP]Bootparams end here....  [APP]Bootparams check done....ret = 0  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE\_AP  [2.243,095] CONNECT:98:da:c4:73:b7:76 Channel:2 rssi:-28 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [2.292,167] MYIP 192.168.0.164  [2.292,445] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE\_AP  \*\* Test Iterations = 1 \*\*  [APP]Calling http\_client\_open(). heap size = 262352  . [SSL\_WRAP]Checking input configurations...  . [SSL\_WRAP]Seeding the random number generator...  . [SSL\_WRAP]Loading the CA root certificate ...Cert Len = 4755  . [SSL\_WRAP]Connecting to tcp httpbin.org:443...  . [SSL\_WRAP]Setting up the SSL/TLS structure...  . [SSL\_WRAP]setting configurations..  >auth mode = 2 (0- skip, 1- optional, 2- required  >max fragment len = 0  >Handshake timeout = 30 Sec  . [SSL\_WRAP]Performing the SSL/TLS handshake...  . [SSL\_WRAP] Handshake done. ok  . [SSL\_WRAP]Verifying peer X.509 certificate.  [APP]Succes: HTTP connection done  [APP]HTTP Get  [APP]Response:  429 ----------------------  200  Date: Thu, 30 Jun 2022 11:10:28 GMT  Content-Type: application/json  Content-Length: 429  Connection: keep-alive  Server: gunicorn/19.9.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Credentials: true  [APP]Body:  {  "slideshow": {  "author": "Yours Truly",  "date": "date of publication",  "slides": [  {  "title": "Wake up to WonderWidgets!",  "type": "all"  },  {  "items": [  "Why <em>WonderWidgets</em> are great",  "Who <em>buys</em> WonderWidgets"  ],  "title": "Overview",  "type": "all"  }  ],  "title": "Sample Slide Show"  }  }  [APP]Success: http\_client\_get(), rval = 0  [APP]------ Program Exit------------- |

## Example boot args for HTTP Post (non-secure)

|  |
| --- |
| ssid=<ssid>,passphrase=<passphrase>,host=httpbin.org,path=/anything,port=80,secured=0,method=post, post\_len=100 |

**Note**:

1. If only post\_len is provided, internally generated data of specified length is posted.
2. Use post\_data boot param to send a specific data. This data length is limited by the boot param max length.
3. To send a file content as post data, place the file in the data fs and provide the file name using post\_data\_file boot param. Example, /data/postdata.txt. postdata.txt mist be part of the data fs

Console output:

|  |
| --- |
| UART:SNWWWWAE  Build $Id: git-e8e189c6a $  hio.baudrate=921600  flash: Gordon ready!  UART:SNWWWWWAEBuild $Id: git-e8e189c6a $  host=httpbin.org path=/anything port=80 secured=0 method=post post\_len=100 np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE\_AP passphrase=innophaseae  $App:git-24f563c7  SDK Ver: sdk\_2.5  Http Client Demo App  Application Information:  ------------------------  Name : HTTP application  Version : 2.0  Build Date : Jun 30 2022  Build Time : 03:57:06  Heap Available: 333 KB (341400 Bytes)  [APP]Bootparams :  --------------------  url=<null>  host= httpbin.org  port=80  path= /anything  secured= 0  method= post  ca\_cert=<null>  post\_len=100  test\_iterations = <null>  use\_ca\_bundle = <null>  hdr1\_name= <null> hdr1\_val= <null>  hdr2\_name= <null> hdr2\_val= <null>  hdr3\_name= <null> hdr3\_val= <null>  post\_data= <null>  post\_data\_file= <null>  [APP]Bootparams end here....  [APP]Bootparams check done....ret = 0  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE\_AP  [2.227,701] CONNECT:98:da:c4:73:b7:76 Channel:2 rssi:-22 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [4.159,469] MYIP 192.168.0.164  [4.159,632] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE\_AP  \*\* Test Iterations = 1 \*\*  [APP]Calling http\_client\_open(). heap size = 266264  [APP]Succes: HTTP connection done  [APP]HTTP Post  [APP]Response:  417 ----------------------  200  Date: Thu, 30 Jun 2022 11:16:21 GMT  Content-Type: application/json  Server: gunicorn/19.9.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Credentials: true  Accept-Ranges: none  Content-Length: 417  Via: HTTP/1.1 forward.http.proxy:3128  Connection: keep-alive  [APP]Body:  {  "args": {},  "data": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",  "files": {},  "form": {},  "headers": {  "Content-Length": "100",  "Host": "httpbin.org",  "X-Amzn-Trace-Id": "Root=1-62bd8605-3b2457fc69b14f6a4a57b854"  },  "json": null,  "method": "POST",  "origin": "106.51.65.237",  "url": "http://httpbin.org/anything"  }  [APP]Success: http\_client\_post(), rval = 0  [APP]------ Program Exit------------- |

## Example boot args for HTTPS Post (without server verification)

|  |
| --- |
| ssid=<ssid>,passphrase=<passphrase>,host=httpbin.org,path=/anything,port=443,secured=1,method=post,ca\_cert=/data/certs/https\_client/app/httpbin\_ca.pem, post\_len=100 |

Console output:

|  |
| --- |
| UART:SNWWWWAE  Build $Id: git-e8e189c6a $  hio.baudrate=921600  flash: Gordon ready!  UART:SNWWWWWAEBuild $Id: git-e8e189c6a $  host=httpbin.org path=/anything port=443 secured=1 method=post ca\_cert=/data/certs/https\_client/app/httpbin\_ca.pem post\_len=100 np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE\_AP passphrase=innophaseae  $App:git-24f563c7  SDK Ver: sdk\_2.5  Http Client Demo App  Application Information:  ------------------------  Name : HTTP application  Version : 2.0  Build Date : Jun 30 2022  Build Time : 03:57:06  Heap Available: 333 KB (341400 Bytes)  [APP]Bootparams :  --------------------  url=<null>  host= httpbin.org  port=443  path= /anything  secured= 1  method= post  ca\_cert=/data/certs/https\_client/app/httpbin\_ca.pem  post\_len=100  test\_iterations = <null>  use\_ca\_bundle = <null>  hdr1\_name= <null> hdr1\_val= <null>  hdr2\_name= <null> hdr2\_val= <null>  hdr3\_name= <null> hdr3\_val= <null>  post\_data= <null>  post\_data\_file= <null>  [APP]Bootparams end here....  [APP]Bootparams check done....ret = 0  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE\_AP  [2.202,306] CONNECT:98:da:c4:73:b7:76 Channel:2 rssi:-23 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [2.256,650] MYIP 192.168.0.164  [2.256,929] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE\_AP  \*\* Test Iterations = 1 \*\*  [APP]Calling http\_client\_open(). heap size = 267480  . [SSL\_WRAP]Checking input configurations...  . [SSL\_WRAP]Seeding the random number generator...  . [SSL\_WRAP]Connecting to tcp httpbin.org:443...  . [SSL\_WRAP]Setting up the SSL/TLS structure...  . [SSL\_WRAP]setting configurations..  >auth mode = 0 (0- skip, 1- optional, 2- required  >max fragment len = 0  >Handshake timeout = 30 Sec  . [SSL\_WRAP]Performing the SSL/TLS handshake...  . [SSL\_WRAP] Handshake done. ok  . [SSL\_WRAP]Verifying peer X.509 certificate.  [APP]Succes: HTTP connection done  [APP]HTTP Post  [APP]Response:  418 ----------------------  200  Date: Thu, 30 Jun 2022 11:20:33 GMT  Content-Type: application/json  Content-Length: 418  Connection: keep-alive  Server: gunicorn/19.9.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Credentials: true  [APP]Body:  {  "args": {},  "data": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",  "files": {},  "form": {},  "headers": {  "Content-Length": "100",  "Host": "httpbin.org",  "X-Amzn-Trace-Id": "Root=1-62bd8700-447fdf8419d52d3223976937"  },  "json": null,  "method": "POST",  "origin": "106.51.65.237",  "url": "https://httpbin.org/anything"  }  [APP]Success: http\_client\_post(), rval = 0  [APP]------ Program Exit------------- |

## Example boot args for HTTPS Post (with server certificate validation)

|  |
| --- |
| ssid=<ssid>,passphrase=<passphrase>,host=httpbin.org,path=/anything,port=443,secured=2,method=post,ca\_cert=/data/certs/https\_client/app/httpbin\_ca.pem, post\_len=100 |

Console output:

|  |
| --- |
| UART:SNWWWWAE  Build $Id: git-e8e189c6a $  hio.baudrate=921600  flash: Gordon ready!  UART:SNWWWWWAEBuild $Id: git-e8e189c6a $  host=httpbin.org path=/anything port=443 secured=2 method=post ca\_cert=/data/certs/http\_client/app/httpbin\_ca.pem post\_len=100 np\_conf\_path=/data/nprofile.json ssid=InnoPhase\_AE\_AP passphrase=innophaseae  $App:git-24f563c7  SDK Ver: sdk\_2.5  Http Client Demo App  Application Information:  ------------------------  Name : HTTP application  Version : 2.0  Build Date : Jun 30 2022  Build Time : 03:57:06  Heap Available: 333 KB (341400 Bytes)  [APP]Bootparams :  --------------------  url=<null>  host= httpbin.org  port=443  path= /anything  secured= 2  method= post  ca\_cert=/data/certs/http\_client/app/httpbin\_ca.pem  post\_len=100  test\_iterations = <null>  use\_ca\_bundle = <null>  hdr1\_name= <null> hdr1\_val= <null>  hdr2\_name= <null> hdr2\_val= <null>  hdr3\_name= <null> hdr3\_val= <null>  post\_data= <null>  post\_data\_file= <null>  [APP]Bootparams end here....  [APP]Bootparams check done....ret = 0  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase\_AE\_AP  [2.234,811] CONNECT:98:da:c4:73:b7:76 Channel:2 rssi:-32 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [2.279,255] MYIP 192.168.0.164  [2.279,534] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase\_AE\_AP  \*\* Test Iterations = 1 \*\*  [APP]Calling http\_client\_open(). heap size = 262416  . [SSL\_WRAP]Checking input configurations...  . [SSL\_WRAP]Seeding the random number generator...  . [SSL\_WRAP]Loading the CA root certificate ...Cert Len = 4755  . [SSL\_WRAP]Connecting to tcp httpbin.org:443...  . [SSL\_WRAP]Setting up the SSL/TLS structure...  . [SSL\_WRAP]setting configurations..  >auth mode = 2 (0- skip, 1- optional, 2- required  >max fragment len = 0  >Handshake timeout = 30 Sec  . [SSL\_WRAP]Performing the SSL/TLS handshake...  . [SSL\_WRAP] Handshake done. ok  . [SSL\_WRAP]Verifying peer X.509 certificate.  [APP]Succes: HTTP connection done  [APP]HTTP Post  [APP]Response:  418 ----------------------  200  Date: Thu, 30 Jun 2022 11:58:20 GMT  Content-Type: application/json  Content-Length: 418  Connection: keep-alive  Server: gunicorn/19.9.0  Access-Control-Allow-Origin: \*  Access-Control-Allow-Credentials: true  [APP]Body:  {  "args": {},  "data": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",  "files": {},  "form": {},  "headers": {  "Content-Length": "100",  "Host": "httpbin.org",  "X-Amzn-Trace-Id": "Root=1-62bd8fdc-05a5d68d133b62c8333f28b7"  },  "json": null,  "method": "POST",  "origin": "106.51.65.237",  "url": "https://httpbin.org/anything"  }  [APP]Success: http\_client\_post(), rval = 0  [APP]------ Program Exit------------- |