Table of Contents

1 API Documentation

- Get Device ID
- Get Network Status
- Get Device Activation Status
- Get WiFi Access Point List
- Put WiFi Client SSID/PSK.
- Get WiFi Client Connection Status. (I'm testing this now but is seems likely I can do both STA can AP at the same time)
- Put WiFi Access Point Disable.
- Put NewCPID Set new cloud system identifier/username/password
- Put CloudAttach Start SDK and configure wifi
- Get IOTGetIOTConnectConf get the current SDK configuration file
- Put IOTSetIOTConnectConf set the current SDK configuration file
- Get WiFiGetWPAConf get the current wpa_supplicant.conf file
- Put WiFiPutWPAConf set the current wpa_supplicant.conf file.
- Get IOTGetIOTConnectConfItem get the current item from the SDK configuration file
- Put IOTSetIOTConnectConfItem set the item in SDK configuration file
- Put IOTSetAPNConf – set contents of /etc/apn.conf for cell modem configuration.

1. Title: Get Device ID

Description: Using this API we should get device id in response.

Command: curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET http://192.168.2.1:8080/DeviceId

Possible Response:

```
{
    "DeviceID": {deviceid}
}
```

2. Title: Get Network Status

Description: Using this API we should get network (internet) status, it is connected or not.

Command: curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET http://192.168.2.1:8080/NetworkStatus

Possible Response:

```
{
    "IsConnected": {true/false}
}
```

3. Title: Get Device Activation Status

Description: Using this API we will know the device is activated into IoT connect or not. When SDK done with initialization process this flag needs to be set it true.

Command: curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET http://192.168.2.1:8080/DeviceActivationStatus

Possible Response:

```
{
    "IsActive": {true/false}
}
```

4.Title: Get WiFi Access Point List

Description: Using this API we should get WiFi Access Point List

Command: curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET http://192.168.2.1:8080/WiFiAccessPointList

Possible Response:

```
{ “None Found”: “-99 dBm”}
```

- Or -

```
{
 "AccessPointList": [
        {
            “SSID”: “value”,
            “SignalStrength”: “value”
```

},

{…}

]

}

5.Title: Put Client WiFi SSID/PSK

Description: Using this API we should get the response from the REST command after SSID/PSK setup is completed.

Command:

curl -X PUT -H "Content-Type: application/json" -d '{"SSID":"PSK"}' http://192.168.2.1:8080/WiFiClientSSID_PSK

Possible Response:

{

"Response": { true/false}

}

6.Title: Get WiFi Client Connection Status

Description: Using this API we should get WiFi client connection status. Note: Wait at least 30 second after posting SSID/PSK for this to become valid.

Command: curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET http://192.168.2.1:8080/WiFiClientConnectionStatus

Possible Response:

{

"IsActive": { true/false}

}

7.Title: Put WiFi Access Point Disable.

Description: Using this API we should disable the WiFi Access Point Mode. Note: Response will be given before we disconnect 30 seconds later.

Command:

curl -X PUT -H "Content-Type: application/json" -d '{"SSID":"1", "PSK":"2"}' http://0.0.0.0:8080/WiFiAccessPointDisable

Possible Response:

{

“IsActive”: { true/false}

}

8.Title: Put IOTNewCPID Set new cloud system identifier.

Description: Using this API you can set your system identifier/username/password. Note this will restart the example python code if it is running.

Command:

curl -X PUT -H "Content-Type: application/json" -d '{"cpid":"123dddd4567","username":"crash1xxxx0@gmail.com","password":"1xxxx0"}' http://192.168.1.134:8080/IOTNewCPID

Possible Response:

{

“Response”: { 1, -1}

}

9.Title: Put CloudAttach

Description: Using this API will cause the gateway to configure the wifi connection and start the SDK.

Command:

curl -X PUT -H "Content-Type: application/json" -d '{""}' http://192.168.2.1:8080/CloudAttach

Possible Response:

{

“Response”: { 1/0}

}

10.Title: Get the current IoTConnectSDK.conf file

Description: Using this API returns the current IOTConnectSDK.conf file

curl -X GET -H "Content-Type: application/json" -d '{""}' http://192.168.2.1:8080/IOTGetIOTConnectSDKConf

Possible Response:

{

“IOTConnectSDKConf”: { 1/0}

}

11.Title: Set the current IoTConnectSDK.conf file

Description: Using this API sets the current IOTConnectSDK.conf file

curl -X PUT -H "Content-Type: application/json" -d '{"<contents of file"}' http://192.168.1.134:8080/IOTSetIOTConnectConf

Possible Response:

{

“IOTConnectSDKConf”: {1/0}

}

12.Title: Get the current wpa_supplicant.conf file

Description: Using this API returns the current wpa_supplicant.conf file

curl -X GET -H "Content-Type: application/json" -d '{""}' http://192.168.2.1:8080/

Possible Response:

{

“WiFiWAPConf”: {<contents of file>}

}

13.Title: Set the current wpa_supplicant.conf file

Description: Using this API sets the current wpa_supplicant.conf file

curl -X PUT -H "Content-Type: application/json" -d '{"<contents of file"}' http://192.168.1.134:8080/WiFiSetWPAConf

Possible Response:

{

“Response”: {1/0}

}

14.Title: Get the current IoTConnectSDK.conf file item

Description: Using this API returns the current IOTConnectSDK.conf file item

curl -X GET -H "Accept: apapplication/json" -d '{"SectionName":"CloudSDKConfiguration","ValueName":"cpid"}' http://192.168.1.118:8080/IOTGetIOTConnectSDKConfItem

Possible Response:

{

“IOTGetConnectSDKConfItem”: {"IOTGetConnectSDKConfItem": {"cpid": "B3A7D54220AD4397ABF35D7EC539FBA6", "SectionName": "CloudSDKConfiguration"} }

}

15.Title: Set the current /etc/apn.conf file IOTSetAPNConf

Description: Using this API sets the current /etc/apn.conf file

curl -X PUT -H "Content-Type: application/json" -d '{"<contents of file"}' http://192.168.1.134:8080/IOTSetAPNConf

Possible Response:

{

“IOTSetAPNConf”: {1/0}

}

2 Open Questions/Concerns

User Case: How mobile application will know that the user is using Ethernet or Wifi, and based on that it will showcase wifi page.