



Welcome

Rest APIを使った PC経由での WLANの設定の書込み 設定一括変更

Rest API

- Rest APIとは

REST、すなわち「REpresentational State Transfer」とは、IT環境で広く使用されている通信アーキテクチャです。よく知られているインターネットプロトコルを使用する、ユーザーフレンドリーなインターフェースアーキテクチャです。データ伝送はHyper Text Transfer Protocol（HTTP）で行われます。

APIとは、Application Programming Interface（アプリケーションプログラミングインターフェース）の略語です。波括弧{}は、コンテンツの表記を表し、JavaScriptプログラミング言語（JSON：JavaScript Object Notation）内のオブジェクトと同じ方法でマスクされます。例えば、{"serial": "12345678"}はシリアル番号のオブジェクトです。



[RESTプログラミングインターフェース | Phoenix Contact](#)

Rest API

- Rest APIを介してPC,PLC経由で簡単な制御

WLAN 1000および2000は、PCやPLCからREST APIを介して簡単に設定したり実行中に制御を行うことができます。

したがって、複数のWLANモジュールを1つのコマンドですばやく簡単に設定することができます。

さらに、例えば、接続ごとにWLANユーザーに新しい一回限りのWLANパスワードを割り当てるために、コントローラを介してネットワーク管理タスクを自動化することができます。

これは、共有された固定WLANパスワードを使用して発生するセキュリティリスクに対するシンプルなソリューションになります。



[WLAN 1000とWLAN 2000 | Phoenix Contact](#)

Rest API

- Rest APIを介してPC,PLC経由で簡単な制御

Web APIを使用できる高級言語であれば
WLANの設定の読み込み、書き込みが可能です。

本資料ではPythonを用いた例で紹介いたします。

※PythonにてWeb APIを使用するために
外部パッケージであるrequestsモジュールを
インストールする必要があります。

本資料ではPythonによる操作を完全に保証するものでは
ございません。

PCのセキュリティや仕様によりRestAPIにアクセスでき
ない場合などございますので、各々の環境に合わせて
設定変更を実施してください。

```
import requests
import json

ip_address = "169.254.2.1"
username = "admin"
password = "private"

url = f"http://{ip_address}/api/v1/wlan/interfaces/1/profiles/1"

data = {
    "port_id": 101,
    "profile_id": 1,
    "ssid": "phoenixcontact",
    "authentication": "wpa2-psk",
    "encryption": "aes",
    "preshared_key": "passwordkakiku",
    "eap_mode": "peap",
    "eap_identity": "pxc_user",
    "eap_phase2_auth": "mschapv2",
    "eap_password": "2bchanged",
    "eap_clientcert_preshared_key": "2bchanged",
    "eap_pairwise_mode": "ccmp",
    "apply_configuration_state": "ok",
    "apply_configuration": "enabled"
}

json_data = json.dumps(data)

response = requests.patch(url=url, data=json_data, auth=(username, password))
```

■ 使用するサンプルコード

- wlan_restAPI_get_sample.py
- wlan_restAPI_patch_sample.py
- wlan_restAPI_change_ipaddress_sample.py
- wlan_restAPI_apply_save_sample.py

Rest API

- WLANのRest APIサーバにアクセスする

PCとWLANを有線のLANケーブルで接続します。
ブラウザを開き、
検索窓で以下のコマンドを実行します。

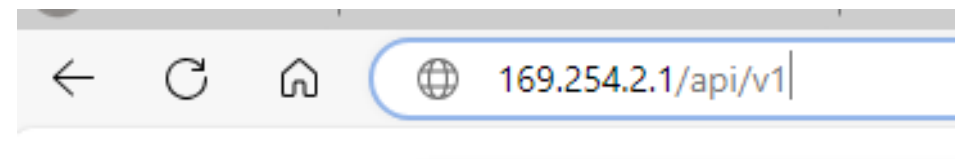
“<Device_IP_address>/api/v1”

例:

“169.254.2.1/api/v1”

※WLANの工場出荷時の
初期IPアドレス: “169.254.2.1”

PCのネットワークアドレスもWLANと同じネットワークアドレスにしてください



Rest API

- WLANのRest APIサーバにアクセスする

右記の画面に移行します。
(本資料ではRest API画面とします)

下記の画面が表示された場合はWLANに設定されたユーザ名とパスワードを入力してください

このサイトにアクセスするにはサインインしてください

http://169.254.2.1 では認証が必要となります
このサイトへの接続は安全ではありません

ユーザー名

パスワード

WLANの工場出荷時
ユーザ名:admin
パスワード:private

Rest API画面より

The screenshot shows the Swagger UI for the PxC-Rest-API. At the top, there's a Swagger logo and a search bar containing '/api/static/v1/swagger.json'. Below that, the API name 'PxC-Rest-API' is displayed with an OAS3 version indicator. The 'Servers' section shows a dropdown menu with '{protocol}:/api/v1'. The 'Computed URL' is 'http:/api/v1'. The 'Server variables' section shows a dropdown for 'protocol' set to 'http'. The main content area displays a list of API endpoints with expandable arrows:

- wlan
- network
- time synchronization
- nat
- system
- service
- port configuration
- events
- file-transfer

Rest API

- WLANのRest APIサーバにアクセスする

この画面より、get(データ取得)やpost(データの書き換え)を行うことができます。

またrequest用のurlも表示されていますので、高級言語プログラムからデータの読み書きが可能です。

Rest API画面より

The screenshot shows the Swagger UI for the PxC-Rest-API. At the top, there is a Swagger logo and a search bar containing the path `/api/static/v1/swagger.json` with an **Explore** button. Below this, the API title **PxC-Rest-API** is displayed with an OAS3 version indicator. The URL `/api/static/v1/swagger.json` and a link to the Phoenix Contact website are also visible.

The interface includes a **Servers** section with a dropdown menu showing `{protocol}:/api/v1`. Below it, the **Computed URL** is `http:/api/v1`. The **Server variables** section shows a `protocol` variable set to `http`.

The main content area displays a list of API endpoints, each with a right-pointing chevron icon:

- wlan
- network
- time synchronization
- nat
- system
- service
- port configuration
- events
- file-transfer

Rest API

- WLANの標準設定画面に入る

ブラウザの新規タブを開き、
検索窓で以下のコマンドを実行します。



“<Device_IP_address>”

例:

“169.254.2.1”

※WLANの工場出荷時の
初期IPアドレス: “169.254.2.1”

PCのネットワークアドレスもWLANと同じネットワークアド
レスにしてください

Rest API

- WLANの標準設定画面に入る

右記の画面に移行します。
(本資料ではWBM画面とします)

WLANの工場出荷時
Username:admin
Password:private

WBM画面より

WLAN-82593b

PHOENIX CONTACT
INSPIRING INNOVATIONS

FL WLAN 1100

- Information
Help & Documentation
Device Status
Local Diagnostic
Alarm & Events
Connections
Interface Status
Login

Login

Username: (?)

Password: (?)

Show cleartext passphrase

login

Rest APIでWLANの設定内容を変更する(書き込む)

Rest API

- WebサーバよりWLANの設定を書込

例えばRest API画面にて
wlan > PATCH
/wlan/interfaces/
[port_id]
を選択すると
“Request body”として
“Example Value”が表示されます。

Rest API画面より

wlan

GET /wlan

PATCH /wlan

GET /wlan/interfaces

POST /wlan/interfaces

GET /wlan/interfaces/[port_id]

PATCH /wlan/interfaces/[port_id]

JSONもしくはXMLの

どちらかのフォーマットを選択できます。

application/json

本資料内ではJSONフォーマットで説明します。

Request body

Example Value | Schema

```
{
  "enable_state": "string",
  "operating_mode": "string",
  "port_forwarding": "string",
  "wds_aging_time": 0,
  "802.11_mode": "string",
  "channel": 0,
  "bandwidth": 0,
  "max_clients": 0,
  "channel_scanlist": "string",
  "start_scanning": "string",
  "wds_broadcast_mode": "string",
  "hide_ssid": "string",
  "fast_eapol_retry_mode": "string",
  "tx_power_conf": 0,
  "scb_manual_mac": "string",
  "scb_mode": "string",
  "fragmentation_threshold": 0,
  "hardware_id": 0,
  "network_id": 0,
  "excessive_retries": 0,
  "roaming_trigger_manual": "string",
  "roaming_network_idle_time": 0,
  "roaming_rssi_threshold_force_scan": 0,
  "roaming_rssi_change_diff": 0,
  "roaming_rssi_change_bg_scan_diff": 0,
  "roaming_rssi_threshold_bg_scan": 0,
  "apply_configuration": "string"
}
```

Rest API

- WebサーバよりWLANの設定を書込

“Try it out”ボタンを選択すると
“Request body”を編集することが
出来ます。

Rest API画面より

PATCH /wlan/interfaces/{port_id}

Parameters

Try it out

Edit Value | Schema

```
{
  "enable_state": "string",
  "operating_mode": "string",
  "port_forwarding": "string",
  "wds_aging_time": 0,
  "802.11_mode": "string",
  "channel": 0,
  "bandwidth": 0,
  "max_clients": 0,
  "channel_scanlist": "string",
  "start_scanning": "string",
  "wds_broadcast_mode": "string",
  "hide_ssid": "string",
  "fast_eapol_retry_mode": "string",
  "tx_power_conf": 0,
  "scb_manual_mac": "string",
  "scb_mode": "string",
  "fragmentation_threshold": 0,
  "hardware_id": 0,
  "network_id": 0
}
```

Rest API

- WebサーバよりWLANの設定を書込

例として

下記の設定をRestAPIで書き換えます。

- ①Channel 6 -> 1
- ②Output power 5 -> 20

設定後、Executeを選択することでWLANとAPI経由で通信を行い、WLANの設定を書き換えます。

WBM画面より

この設定を1に設定したい

この設定を20に設定したい

Rest API画面より

- ①Channel 1に設定
- ②Output Power 20に設定

設定内容を反映します

Rest API

- WebサーバよりWLANの設定を書込

“Server response”が返ってきます。
Codeが200であれば無事に書き込めたこととなります。

“Response body”内の
“error”も“[]”になっていることを
確認してください。

“WBM画面”を更新すると
設定内容が反映されます。

Rest API画面より

Server response

Code	Details
200 <i>Undocumented</i>	Response body <pre>{ "content": {}, "error": [] }</pre>

WBM画面より

PHENIX CONTACT
WLAN-82593b
Hello admin

FL WLAN 1100

Information
Help & Documentation
Device Status
Local Diagnostic
Alarm & Events
Connections
Interface Status
+ Configuration
+ Diagnostics

WLAN Setting

Country (regulatory domain) (?) Germany

Activate WLAN interface (?) Enable

Outdoor mode (?) Enable

Aggregation mode (?) Enable

WLAN band (?) 2.4GHz (802.11 g/n)

Channel (?) 1 ← この設定を1に反映される

Output power (?) 20dBm ← この設定を20に反映される

Channel bandwidth (802.11n) (?) 20MHz 40MHz

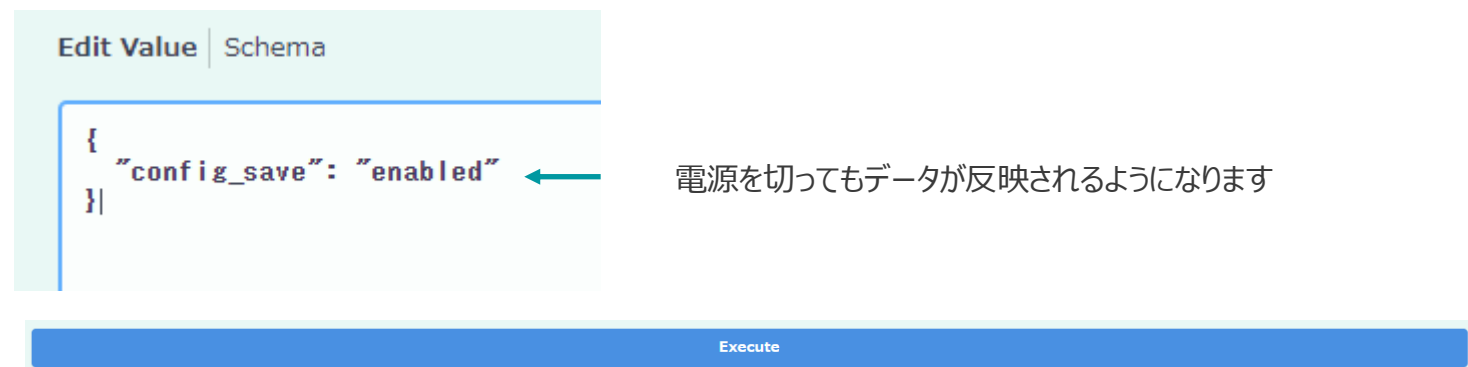
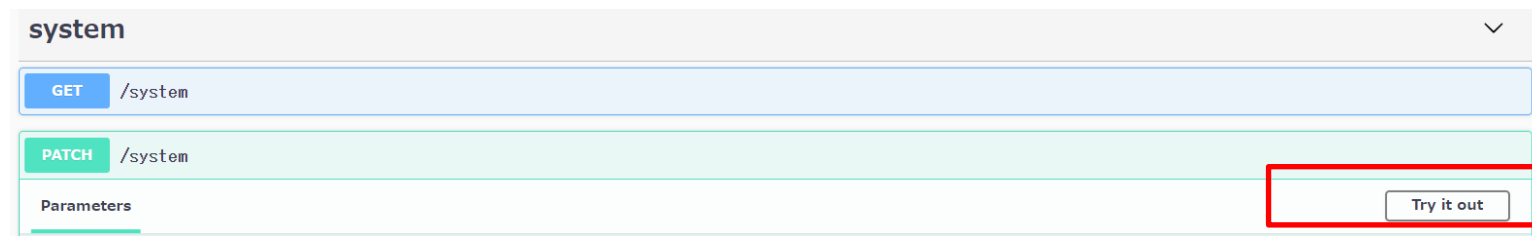
Rest API

- WLANの設定を保持する
(電源を切ってもデータを反映させる)

Rest API画面にて
system > PATCH/system/
を選択し、
“Try it out”を選択して
“Request body”を編集表示にします。

右記のように
“config_save”: “enabled”
を記載して
“Execute”を実行して書き込んでください

Rest API画面より



Rest API

- WLANの設定項目をプログラムで変更する

サンプルとして

添付項目の情報について変更(書き換え)します。

①IP Address Assignment :

“BOOTP” -> “**STATIC**”

②IP Address :

“169.254.2.1” -> “**192.168.1.30**”

③Network Mask :

“255.255.0.0” -> “**255.255.255.0**”

WBM画面より

WLAN-82593b
Hello admin

PHOENIX CONTACT

FL WLAN 1100

+ Information
- Configuration
My Profile
User Management
Quick Setup
System
Network
WLAN Setting
WLAN Interface
Service
Multicast Filtering
Security
+ Diagnostics

Network

IP Address Assignment (?) BOOTP ①
IP Address (?) 169.254.2.1 ②
Network Mask (?) 255.255.0.0 ③
Default Gateway (?) 0.0.0.0
DNS Server 1 (?) 0.0.0.0
DNS Server 2 (?) 0.0.0.0
Management VLAN (?) 1

Hostname Configuration

Name resolution (?) Enable
Hostname (?) WLAN-82593b

Apply Revert Apply&Save

Copyright © by Phoenix Contact GmbH&Co.KG and Other

Rest API

- WLANの設定項目をプログラムで変更する

サンプルとして
添付項目の情報について変更(書き換え)します。

④Country :
“Germany” -> “Japan”

⑤Channel :
6 -> 3

⑥Output power :
5 -> 20

WBM画面より

WLAN-82593b
Hello admin

PHOENIX CONTACT

FL WLAN 1100

+ Information
- Configuration
My Profile
User Management
Quick Setup
System
Network
WLAN Setting
WLAN Interface
Service
Multicast Filtering
Security
+ Diagnostics

Country (regulatory domain) (?) Germany ④
Activate WLAN interface (?) Enable
Outdoor mode (?) Enable
Aggregation mode (?) Enable
WLAN band (?) 2.4GHz (802.11 g/n)
Channel (?) 6 ⑤
Output power (?) 5dBm ⑥
Channel bandwidth (802.11n) (?) 20MHz 40MHz

Apply Revert Apply&Save

Copyright © by Phoenix Contact GmbH&Co.KG and Other

Rest API

- WLANの設定項目をプログラムで変更する

サンプルとして
添付項目の情報について変更(書き換え)します。

⑦ Operating Mode :
"Client(MCB)" -> "AP"

⑧ Network SSID :
"PhoenixContact" -> "ChangeSSID"

⑨ Passkey
"2bchanged" -> "ChangePasskey"

WBM画面より

The screenshot displays the WBM interface for a Phoenix Contact device. The top bar shows the device name 'WLAN-82593b' and the user 'Hello admin'. The main content area is titled 'WLAN Interface' and shows the configuration for 'wlan 1'. The configuration includes a 'Setting' tab, a 'Scan' button, and a 'Roaming List' button. The settings are as follows:

- Port ID (?): 101
- Operating Mode (?): Client(MCB) (highlighted with a red circle 7)
- Roaming (?): Enable
- Network SSID (?): PhoenixContact (highlighted with a red circle 8)
- Security mode (?): WPA2_PSK_AES
- Passkey (?): (highlighted with a red circle 9)

There is also a checkbox for 'Show cleartext passphrase' which is currently unchecked. At the bottom right, there are buttons for 'Apply', 'Revert', and 'Apply&Save'. The sidebar on the left contains a navigation menu with options: '+ Information', '- Configuration' (with sub-items: My Profile, User Management, Quick Setup, System, Network, WLAN Setting, WLAN Interface, Service, Multicast Filtering, Security), and '+ Diagnostics'.

Rest API

■ Pythonによる書換サンプルプログラム(1/4)

※本資料ではIPアドレスを変更するとネットワークアドレスによってはネットワークの接続が切れるため、別ファイルで運用するようにしています

1,2: requests, jsonモジュールのimport

4-6: WLANに設定されているIpAddress及びユーザ名、パスワード
※(工場出荷設定は169.254.2.1, admin, private)

8: 変更(書換)したいデータのURLアドレスを入力

11-15: 変更データ

12: Network SSID ⑧

13: PassKey ⑨

14: 設定データを適用する

17: 変更データをrequests用のJSONフォーマットへ変換

19: requests.patch経由でWLANへの書き込み

wlan_restAPI_patch_sample.py

```
1 import requests
2 import json
3
4 ip_address = "169.254.2.1"
5 username = "admin"
6 password = "private"
7
8 url = f"http://{ip_address}/api/v1/wlan/interfaces/1/profiles/1"
9
10
11 data = {
12     "ssid": "ChangeSSID",
13     "preshared_key": "ChangePasskey",
14     "apply_configuration": "enabled"
15 }
16
17 json_data = json.dumps(data)
18
19 response = requests.patch(url=url, data=json_data, auth=(username, password))
20
21 if response.status_code == 200:
22     print("preshared_key successfully updated.")
23 else :
24     print("Failed to update preshared key:", response.status_code, response.text)
25
```

Rest API

■ Pythonによる書換サンプルプログラム(2/4)

27: 変更(書換)したいデータのURLアドレスを入力

30-35: 変更データ

31: Operating mode ⑦

32: Channel ⑤

33: Output power ⑥

34: 設定データを適用する

37: 変更データをrequests用のJSONフォーマットへ変換

39: requests.patch経由でWLANへの書き込み

wlan_restAPI_patch_sample.py

```
27 url = f"http://{ip_address}/api/v1/wlan/interfaces/1"
28
29
30 data = {
31     "operating_mode": "ap",
32     "channel": 3,
33     "tx_power_conf": 20,
34     "apply_configuration": "enabled"
35 }
36
37 json_data = json.dumps(data)
38
39 response = requests.patch(url=url, data=json_data, auth=(username, password))
40
41 if response.status_code == 200:
42     print("operation mode & wlan channel, txpower successfully updated.")
43 else :
44     print("Failed to update operation mode & wlan channel, txpower:", response.status_code, response.text)
45
```

Rest API

■ Pythonによる書換サンプルプログラム(3/4)

48: 変更(書換)したいデータのURLアドレスを入力

50-53: 変更データ

51: Country ④

52: 設定データを適用する

55: 変更データをrequests用のJSONフォーマットへ変換

57: requests.patch経由でWLANへの書き込み

wlan_restAPI_patch_sample.py

```
48 url = f"http://{ip_address}/api/v1/wlan"
49
50 data = {
51     "country": "Japan",
52     "apply_configuration": "enabled"
53 }
54
55 json_data = json.dumps(data)
56
57 response = requests.patch(url=url, data=json_data, auth=(username, password))
58
59 if response.status_code == 200:
60     print("wlan country setting successfully updated.")
61 else :
62     print("Failed to update wlan country setting:", response.status_code, response.text)
63
```

Rest API

- Pythonによる書換サンプルプログラム(4/4)

67: 変更(書換)したいデータのURLアドレスを入力

69-71: 変更データ

70: 全データの保存(電源を切っても反映される)

73: 変更データをrequests用のJSONフォーマットへ変換

75: requests.patch経由でWLANへの書き込み

wlan_restAPI_patch_sample.py

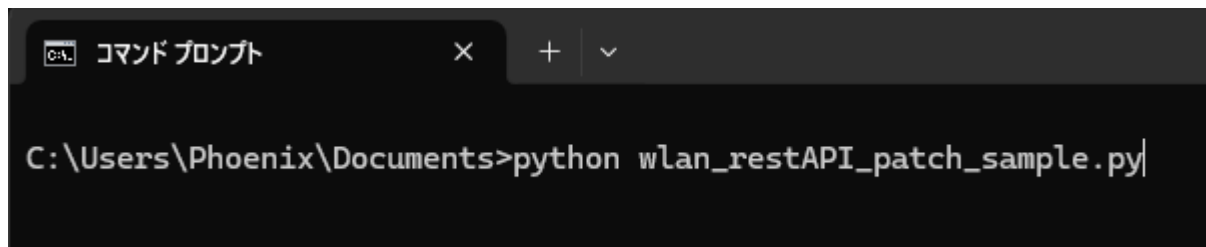
```
67 url = f"http://{ip_address}/api/v1/system"
68
69 data = {
70     "config_save": "enabled",
71 }
72
73 json_data = json.dumps(data)
74
75 response = requests.patch(url=url, data=json_data, auth=(username, password))
76
77 if response.status_code == 200:
78     print("All setting successfully updated.")
79 else :
80     print("Failed to update All setting:", response.status_code, response.text)
81
```

Rest API

- Pythonの実行

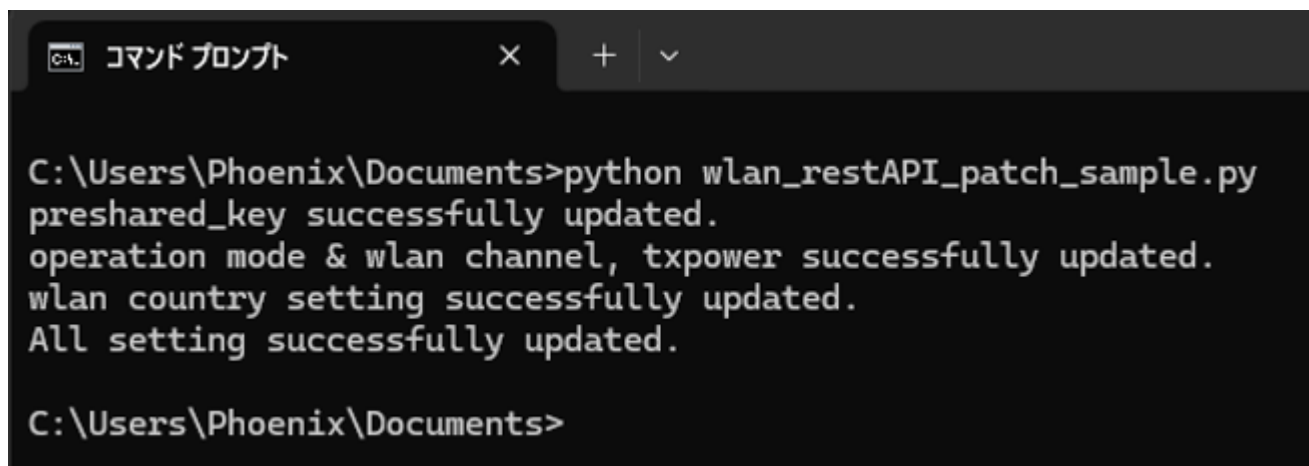
Windowsの場合はコマンドプロンプトを実行し、作成したpythonファイルの格納フォルダ(ディレクトリ)へ移動します。

```
> python wlan_restAPI_patch_sample.py
```



```
コマンド プロンプト
C:\Users\Phoenix\Documents>python wlan_restAPI_patch_sample.py
```

・実行結果



```
コマンド プロンプト
C:\Users\Phoenix\Documents>python wlan_restAPI_patch_sample.py
preshared_key successfully updated.
operation mode & wlan channel, txpower successfully updated.
wlan country setting successfully updated.
All setting successfully updated.
C:\Users\Phoenix\Documents>
```

Rest API

■ Pythonによる書換サンプルプログラム(1/2)

IPアドレスの変更

1,2: requests, jsonモジュールのimport

4-6: WLANに設定されているIpAddress及びユーザ名、パスワード
※(工場出荷設定は169.254.2.1, admin, private)

8: 変更(書換)したいデータのURLアドレスを入力

11-16: 変更データ

12: IP Address Assignment ①

13: IP Address ②

14: Network mask ③

15: 設定データを適用する

18: 変更データをrequests用のJSONフォーマットへ変換

20: requests.patch経由でWLANへの書き込み

wlan_restAPI_change_ipaddress_sample.py

```
1 import requests
2 import json
3
4 ip_address = "169.254.2.1"
5 username = "admin"
6 password = "private"
7
8
9 url = f"http://{ip_address}/api/v1/network/1"
10
11 data = {
12     "ip_assignment": "static",
13     "ip_address": "192.168.1.30",
14     "network_mask": "255.255.255.0",
15     "ip_parameter_apply": "enabled"
16 }
17
18 json_data = json.dumps(data)
19
20 response = requests.patch(url=url, data=json_data, auth=(username, password))
21
22 if response.status_code == 200:
23     print("IP address setting successfully updated.")
24 else :
25     print("Failed to update IP address setting:", response.status_code, response.text)
26
```


Rest API

- Pythonの実行

Windowsの場合はコマンドプロンプトを実行し、作成したpythonファイルの格納フォルダ(ディレクトリ)へ移動します。

> python wlan_restAPI_change_ipaddress_sample.py

※IP アドレス変更に関しては本資料のようにネットワークアドレス部を変更するとネットワークが途切れることに注意してください。

```
コマンド プロンプト
C:\Users\Phoenix\Documents>python wlan_restAPI_change_ipaddress_sample.py|
```

・実行結果

```
コマンド プロンプト
C:\Users\Phoenix\Documents>python wlan_restAPI_change_ipaddress_sample.py
IP address setting successfully updated.
C:\Users\Phoenix\Documents>
```

Rest API

■ Pythonによる書換サンプルプログラム(2/2) IPアドレスの変更

1,2: requests, jsonモジュールのimport

4-6: WLANに設定されているIpaddress及びユーザ名、パスワード
※(変更後192.168.1.30, admin, private)

8: 変更(書換)したいデータのURLアドレスを入力

10-12: 変更データ
11: 全データの保存(電源を切っても反映される)

14: 変更データをrequests用のJSONフォーマットへ変換

16: requests.patch経由でWLANへの書き込み

wlan_restAPI_apply_save_sample.py

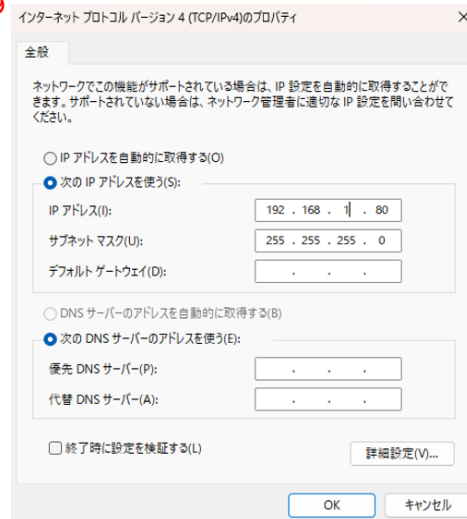
```
1 import requests
2 import json
3
4 ip_address = "192.168.1.30"
5 username = "admin"
6 password = "private"
7
8 url = f"http://{ip_address}/api/v1/system"
9
10 data = {
11     "config_save": "enabled",
12 }
13
14 json_data = json.dumps(data)
15
16 response = requests.patch(url=url, data=json_data, auth=(username, password))
17
18 if response.status_code == 200:
19     print("All setting successfully updated.")
20 else :
21     print("Failed to update All setting:", response.status_code, response.text)
22
```

Rest API

■ Pythonの実行

Windowsの場合はコマンドプロンプトを実行し、作成したpythonファイルの格納フォルダ(ディレクトリ)へ移動します。

※PCのネットワークアドレスを前項で設定したネットワークアドレスと同じにします
例:192.168.1.80



> python wlan_restAPI_apply_save_sample.py

```
コマンド プロンプト
C:\Users\Phoenix\Documents>python wlan_restAPI_apply_save.py|
```

・実行結果

```
コマンド プロンプト
C:\Users\Phoenix\Documents>python wlan_restAPI_apply_save.py
All setting successfully updated.
C:\Users\Phoenix\Documents>
```

Rest API

- 変更されたWLANの設定項目を確認する

WBM画面に入ります。

Webブラウザの検索バーにWLANのIPアドレスを入力

①IP Address Assignment :

“STATIC”

②IP Address :

“192.168.1.30”

③Network Mask :

“255.255.255.0”

WBM画面より

WLAN-82593b
Hello admin

Connected Clients : 0

PHOENIX CONTACT

FL WLAN 1100

+ Information
- Configuration
My Profile
User Management
Quick Setup
System
Network
WLAN Setting
WLAN Interface
Service
Multicast Filtering
Security
+ Diagnostics

Network

IP Address Assignment (?) STATIC ①
IP Address (?) 192.168.1.30 ②
Network Mask (?) 255.255.255.0 ③
Default Gateway (?) 0.0.0.0
DNS Server 1 (?) 0.0.0.0
DNS Server 2 (?) 0.0.0.0
Management VLAN (?) 1
DHCP Configuration (?) [DHCP Services](#)

Hostname Configuration

Name resolution (?) Enable
Hostname (?) WLAN-82593b

Apply Revert Apply&Save

Copyright © by Phoenix Contact GmbH&Co.KG and

Rest API

- 変更されたWLANの設定項目を確認する

④Country :
"Japan"

⑤Channel :
3

⑥Output power :
20

WBM画面より

← ↻ 🏠 ⚠ セキュリティ保護なし | 192.168.1.30

PHOENIX CONTACT

WLAN-82593b
Hello admin

Connected Clients : 0

FL WLAN 1100

+ Information
- Configuration
My Profile
User Management
Quick Setup
System
Network
WLAN Setting
WLAN Interface
Service
Multicast Filtering
Security
+ Diagnostics

WLAN Setting

Country (regulatory domain) (?) Japan* ④

Activate WLAN interface (?) Enable

Outdoor mode (?) Enable

Aggregation mode (?) Enable

WLAN band (?) 2.4GHz (802.11 g/n) ⑤

Channel (?) 3 ⑤

Output power (?) 20dBm ⑥

Channel bandwidth (802.11n) (?) 20MHz 40MHz

Apply Revert Apply&Save

Copyright © by Phoenix Contact GmbH&Co.KG and

Rest API

- 変更されたWLANの設定項目を確認する

⑦ Operating Mode :

“Access Point”

⑧ Network SSID :

“ChangeSSID”

⑨ Passkey

“ChangePasskey”

WBM画面より

WLAN-82593b
Hello admin
Connected Clients : 0

PHOENIX CONTACT

FL WLAN 1100

+ Information
- Configuration
My Profile
User Management
Quick Setup
System
Network
WLAN Setting
WLAN Interface
Service
Multicast Filtering
Security
+ Diagnostics

WLAN Interface

wlan 1 +

Setting

Port ID (?) 101

Operating Mode (?) Access Point ⑦

Network SSID (?) ChangeSSID ⑧ Hide SSID

Security mode (?) WPA2_PSK_AES

Passkey (?) ⑨ Show cleartext passphrase

Apply Revert Apply&Save

Copyright © by Phoenix Contact GmbH&Co.KG and

Batファイル・Pythonプログラムを用いた REST APIによる設定一括変更例

Rest API

- pythonでAPIのみ実行する場合と、一括設定する場合の違いは以下となります。

	操作内容	設定・操作
1	PCのIPアドレス設定変更 (WLANネットワークにIPを合わせる)	PCのネットワーク設定
2	WLANのIPアドレス変更API Pythonプログラム実行	CLI
3	PCのIPアドレス設定変更 (WLANネットワークにIPを合わせる)	PCのネットワーク設定
4	WLANの設定保存API Pythonプログラム実行	CLI
5	PCのIPアドレス設定変更 (元の設定に戻す)	PCのネットワーク設定

これらの操作が
一括設定プログラムへ変更すると
batファイルを右クリックして
実行するのみ
となります

PythonにてAPIのみ実行する場合のフロー

Rest API

- batファイルとpythonプログラムを組み合わせ、WLANの設定変更から設定変更保存まで一括で処理する方法例を記載します。

【プログラム実行方法】

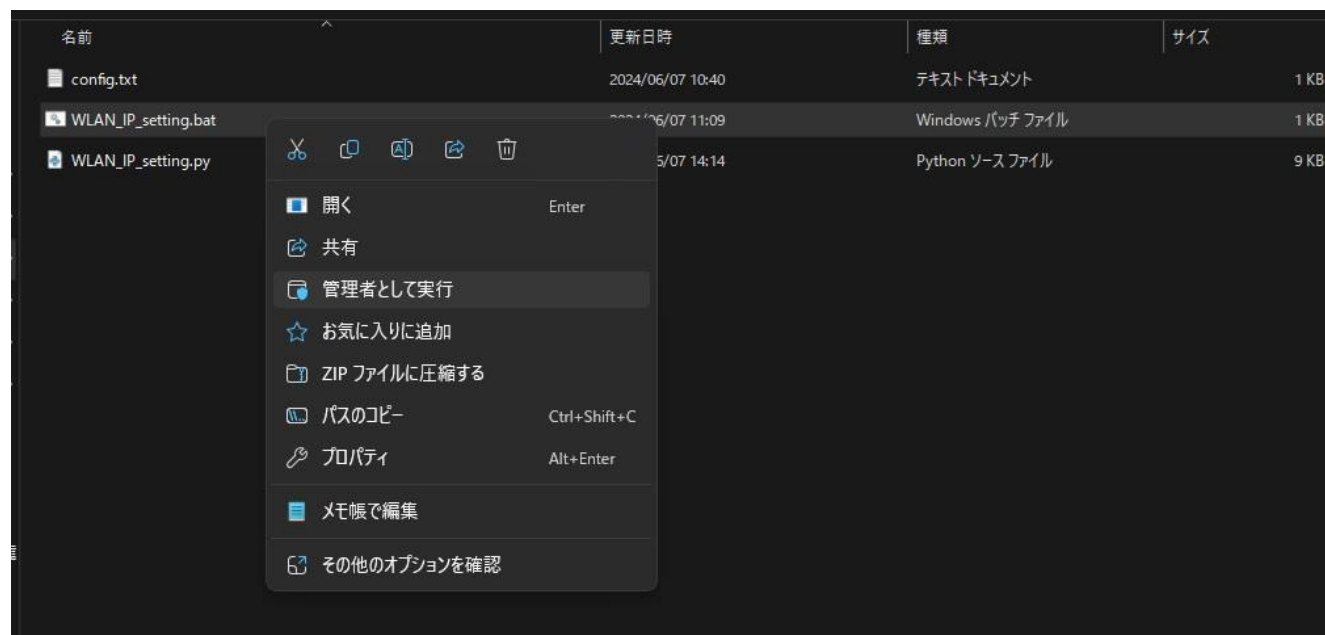
以下の3つのファイルを同じフォルダ内に保存します。

- config.txt
- WLAN_IP_setting.bat
- WLAN_IP_setting.py

batファイルを右クリックして、「管理者として実行」を選択してプログラムを実行します。

【メリット】

- CLIによるpythonプログラム実行操作不要
- pythonプログラムの変更不要
- windowsの設定画面による設定変更不要



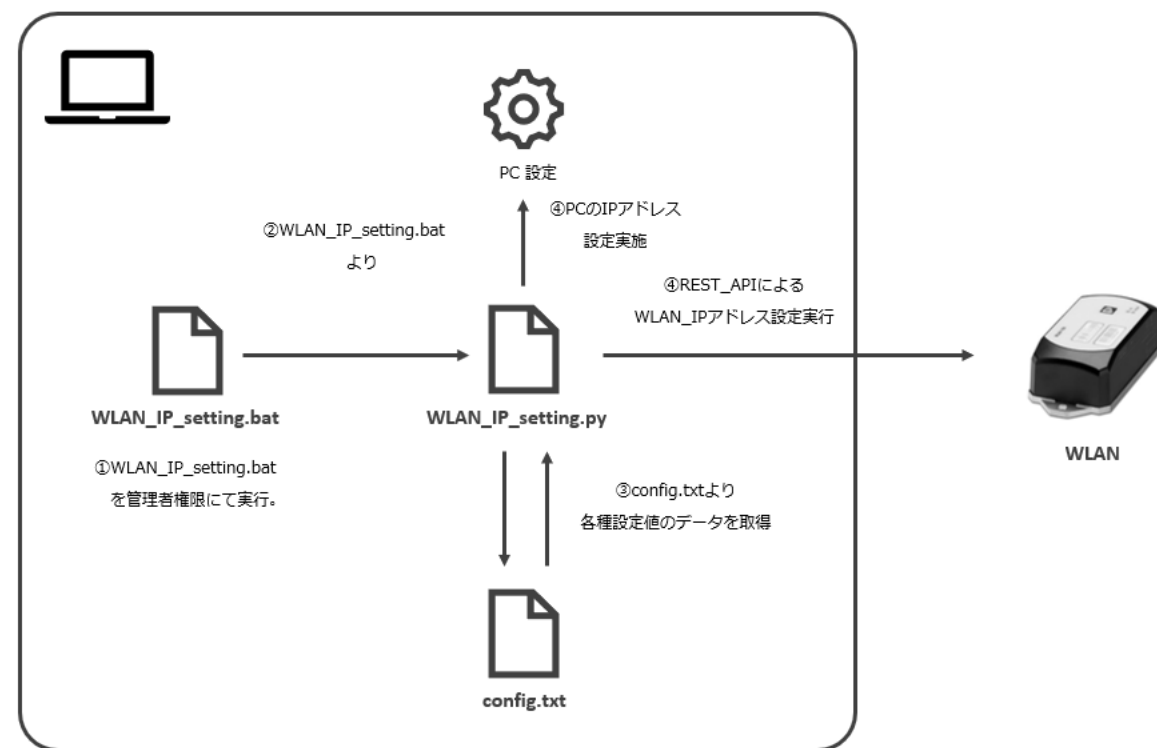
プログラム実行イメージ

Rest API

- プログラムの大まかな実行イメージは右図のようになります。

【内容】

- ①WLAN_IP_setting.batを管理者権限にて実行
- ②WLAN_IP_setting.batからWLAN_IP_setting.pyを実行
- ③config.txtより各種設定データを取得
- ④REST_APIを用いたWLAN_IPアドレス設定及びPCのIPアドレス設定変更を実行



プログラム実行イメージ

Rest API

- WLANやPCのIPアドレス設定については config.txtファイルをテキストエディタにて開き 各種設定値を入力します。

【設定方法】

ファイル内の各設定内容については、ファイル内に説明記載。
(プログラムはこちらの設定ファイルを読み込んで設定実施)

```
# PCのIP設定(WLANのIP設定変更前と同じネットワークにする)
[PC_WLAN_SETTING]
# 正しいアダプター名
adapter_name = イーサネット
# IPアドレス
set_ip = 169.254.2.150
# サブネットマスク
set_subnet_mask = 255.255.255.0
# デフォルトゲートウェイ
set_gateway = 0.0.0.0

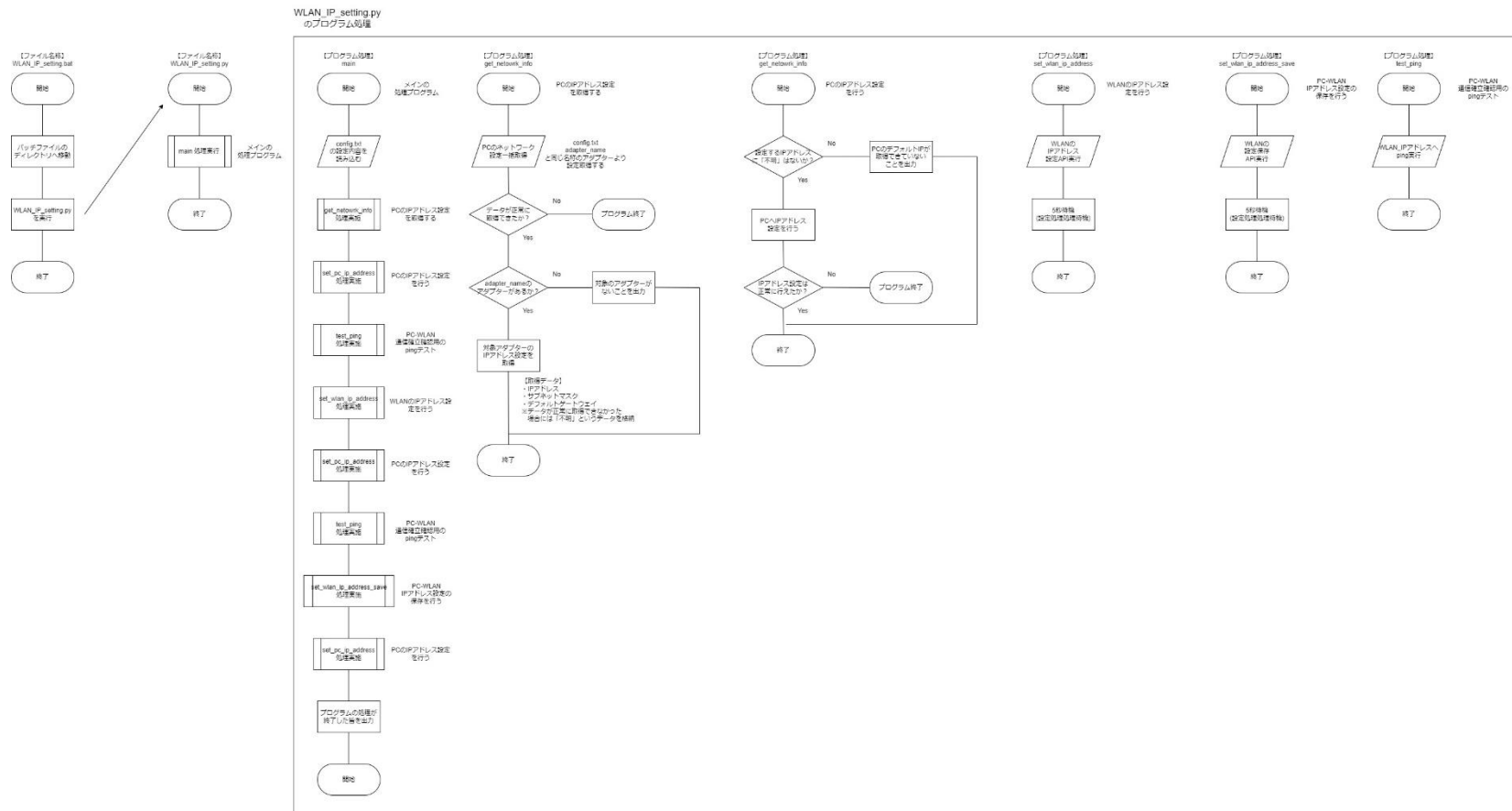
# PCのIP設定(WLANのIP設定変更後と同じネットワークにする)
[PC_WLAN_SAVE]
# IPアドレス
set_ip_wlan_network = 192.168.2.150

# WLANのログイン情報
[WLAN_LOGIN]
# wlanのログインアカウント
username = admin
# wlanのログインパスワード
password = private

# WLANの変更IPアドレスを記載
[WLAN_SETTING]
# wlan本体の設定変更前のIPアドレス
default_wlan_ip_address = 169.254.2.1
# wlanの設定変更後のIPアドレス
set_WLAN_ip_address = 192.168.2.30
```

Config.txtの内容

Rest API



プログラムのフローチャート

Thank you