



Welcome

Rest APIを使った PC経由での WLAN設定の読み込み



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。

The text "REST API" is displayed in a large, bold, black sans-serif font. It is enclosed within a pair of large, black curly braces. The entire graphic is centered on a light gray rectangular background.

[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_clientcent_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

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。

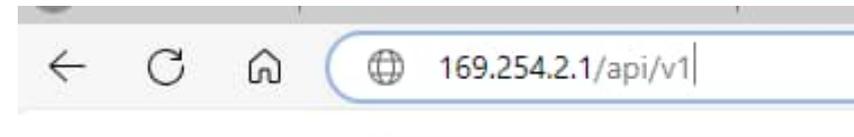
“<Device_IP_address>/api/v1”

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

サインイン

キャンセル

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 is a Swagger logo and a search bar containing "/api/static/v1/swagger.json" with an "Explore" button. Below this, the API title "PxC-Rest-Api" is displayed with a "GAS3" badge. The URL "/api/static/v1/swagger.json" and a link to "Phoenix Contact - Website" are also visible. The "Servers" section shows a dropdown menu with "{protocol}:/api/v1" selected. The "Computed URL" is "http://api/v1". The "Server variables" section shows a dropdown menu with "http" selected. At the bottom, there is 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 a `OpenAPI 3.0` badge. The URL `/api/static/v1/swagger.json` and a link to the `Phoenix Contact - Website` are also visible.

The `Servers` section shows a dropdown menu with the value `{protocol}:/api/v1`. Below it, the `Computed URL` is `http:/api/v1`.

The `Server variables` section shows a dropdown menu for `protocol` with the value `http`.

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

- `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

PHENIX 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の設定内容を取得する

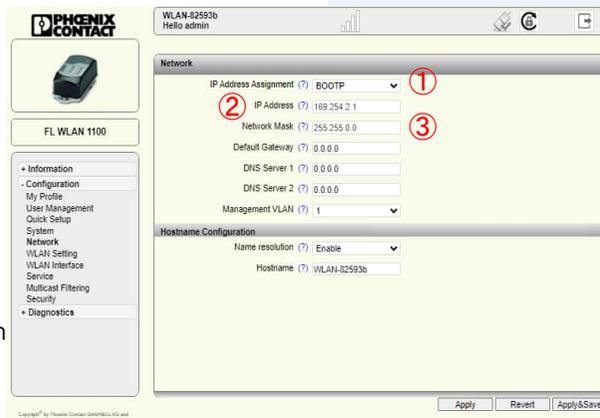
Rest API

- WebサーバよりWLANの設定を取得

例えばRest API画面にて
network > GET
/network/[interface_id]
を選択後、
“Try it out”を選択します

Response bodyの内容が返されます。

- ① IP Address Assignment
- ② IP Address
- ③ Network Mask



Rest API画面より

Pythonなどプログラミング言語から
アクセスする場合はこちらのurl部分で使します。
例: Pythonの場合
requests.get(url='http://169.254.2.1/api/v1/network/1

```
{  
  "content": {  
    "interface_id": 1,  
    "ip_assignment": "bootp",  
    "ip_address": "169.254.2.1",  
    "network_mask": "255.255.0.0",  
    "default_gateway": "0.0.0.0",  
    "management_vlan": 1,  
    "ip_parameter_state": "not modified",  
    "ip_parameter_apply": "disabled"  
  },  
  "error": []  
}
```

Rest API

- WLANの設定項目をプログラムで取得する

サンプルとして
添付項目の情報について取得します。

- ① IP Address Assignment
- ② IP Address
- ③ Network Mask

WBM画面より

WLAN-82593b
Hello admin

PHENIX 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取得する

サンプルとして
添付項目の情報について取得します。

- ④Country
- ⑤WLAN band
- ⑥Channel
- ⑦Output power

WBM画面より

The screenshot shows the WBM configuration interface for a Phoenix Contact device. The top bar displays 'WLAN-82593b' and 'Hello admin'. The main content area is titled 'WLAN Setting' and contains several configuration options:

- Country (regulatory domain) (?): Germany (marked with ④)
- Activate WLAN interface (?): Enable
- Outdoor mode (?): Enable
- Aggregation mode (?): Enable
- WLAN band (?): 2.4GHz (802.11 g/n) (marked with ⑤)
- Channel (?): 6 (marked with ⑥)
- Output power (?): 5dBm (marked with ⑦)
- Channel bandwidth (802.11n) (?): 20MHz 40MHz

At the bottom right, there are three buttons: 'Apply', 'Revert', and 'Apply&Save'. The left sidebar shows a navigation menu with 'WLAN Setting' selected.

Rest API

- WLANの設定項目をプログラムで取得する

サンプルとして
添付項目の情報について取得します。

- ⑧ Operating Mode
- ⑨ Security mode
- ⑩ Passkey

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

WLAN Interface

wlan 1 +

Setting Scan Roaming List

Port ID (?) 101

Operating Mode (?) Client(MCB) ⑧

Roaming (?) Enable

Network SSID (?) PhoenixContact

Security mode (?) WPA2_PSK_AES ⑨

Passkey (?) ⑩ Show cleartext passphrase

Apply Revert Apply&Save

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

Rest API

■ Pythonによる取得サンプルプログラム(1/4)

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)

9: 取得したいデータのURLアドレスを入力

12: GETメソッドによるAPIへの問い合わせとデータ入手

13: 入手したレスポンスデータをJSONフォーマットへ変換

19: 入手したJSONデータをテキストファイルへ書き出し

wlan_restAPI_get_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/wlan/interfaces/1/profiles/1"
10
11 # GETリクエストを送信して設定状況を取得
12 response = requests.get(url=url, auth=(username, password))
13 json_data = response.json()
14
15 # レスポンスを確認
16 if response.status_code == 200:
17     print("設定状況:")
18     print(json_data)
19     with open("reading data", "w") as f:
20         json.dump(json_data, f)
21 else:
22     print(f"エラーが発生しました: {response.status_code}")
23
```

Rest API

- Pythonによる取得サンプルプログラム(1/4)

- 本プログラムで取得したJSONデータ

```
{"content": {"port_id": 101,  
"profile_id": 1,  
"ssid": "PhoenixContact",  
"authentication": "wpa2-psk",  
"encryption": "aes",  
"preshared_key": "2bchanged",  
"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": "disabled"},  
"error": []}
```

WBM画面より



⑨ Security mode

⑩ Passkey

Rest API

- Pythonによる取得サンプルプログラム(2/4)

27: 取得したいデータのURLアドレスを入力

30: GETメソッドによるAPIへの問い合わせとデータ入手

31: 入手したレスポンスデータをJSONフォーマットへ変換

37: 入手したJSONデータをテキストファイルへ書き出し

wlan_restAPI_get_sample.py

```
26
27 url = f"http://{ip_address}/api/v1/wlan/interfaces/1"
28
29 # GETリクエストを送信して設定状況を取得
30 response = requests.get(url=url, auth=(username, password))
31 json_data = response.json()
32
33 # レスポンスを確認
34 if response.status_code == 200:
35     print("設定状況:")
36     print(json_data)
37     with open("reading data", "a") as f:
38         json.dump(json_data, f)
39 else:
40     print(f"エラーが発生しました: {response.status_code}")
41
```

Rest API

Pythonによる取得サンプルプログラム(2/4)

■本プログラムで取得したJSONデータ

```
{ "content": { "id": 1,
"port_id": 101,
"state": "ready",
"enable_state": "enabled",
"operating_mode": "mcb",
"port_forwarding": "enabled",
"wds_aging_time": "60 sec",
"802.11_mode": "gn", ⑤
"channel": 6, ⑥
"bandwidth": "20 MHz",
"max_clients": 10,
"channel_scanlist": "all",
.
.
"tx_power_conf": "5 dBm", ⑦
.
.
"apply_configuration": "disabled"}, "error": [] }
```

WBM画面より

⑧ Operating Mode

⑤ WLAN band

⑥ Channel

⑦ Output power

Rest API

■ Pythonによる取得サンプルプログラム(3/4)

44: 取得したいデータのURLアドレスを入力

47: GETメソッドによるAPIへの問い合わせとデータ入手

48: 入手したレスポンスデータをJSONフォーマットへ変換

54: 入手したJSONデータをテキストファイルへ書き出し

wlan_restAPI_get_sample.py

```
43
44 url = f"http://{ip_address}/api/v1/wlan"
45
46 # GETリクエストを送信して設定状況を取得
47 response = requests.get(url=url, auth=(username, password))
48 json_data = response.json()
49
50 # レスポンスを確認
51 if response.status_code == 200:
52     print("設定状況:")
53     print(json_data)
54     with open("reading data", "a") as f:
55         json.dump(json_data, f)
56 else:
57     print(f"エラーが発生しました: {response.status_code}")
58
```

Rest API

Pythonによる取得サンプルプログラム(3/4)

■ 本プログラムで取得したJSONデータ

```
{ "content": { "global_activation": "enabled",  
  "outdoor_mode": "disabled",  
  "country": "Germany",  
  "management_access": "enabled",  
  "wifi_individual": "enabled",  
  "ptcp_ldap_filter": "enabled",  
  "cyclic_rssi_tracking": "disabled",  
  "radio": [{"id": 1,  
    "aggregation_mode": "enabled",  
    "antenna_mask": 3}],  
  "apply_configuration_state": "ok",  
  "apply_configuration": "disabled"},  
  "error": [] }
```

④

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
Output power (?) 5dBm
Channel bandwidth (802.11n) (?) 20MHz 40MHz

④Country

Apply Revert Apply&Save

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

Rest API

■ Pythonによる取得サンプルプログラム(4/4)

61: 取得したいデータのURLアドレスを入力

64: GETメソッドによるAPIへの問い合わせとデータ入手

65: 入手したレスポンスデータをJSONフォーマットへ変換

71: 入手したJSONデータをテキストファイルへ書き出し

wlan_restAPI_get_sample.py

```
60
61 url = f"http://{ip_address}/api/v1/network/1"
62
63 # GETリクエストを送信して設定状況を取得
64 response = requests.get(url=url, auth=(username, password))
65 json_data = response.json()
66
67 # レスポンスを確認
68 if response.status_code == 200:
69     print("設定状況:")
70     print(json_data)
71     with open("reading data", "a") as f:
72         json.dump(json_data, f)
73 else:
74     print(f"エラーが発生しました: {response.status_code}")
```

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。

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

> python wlan_restAPI_get_sample.py

・実行結果

```
C:\Users\Phoenix\Documents>python wlan_restAPI_get_sample.py
設定状況:
{'content': {'port_id': 101, 'profile_id': 1, 'ssid': 'PhoenixContact', 'authentication': 'wpa2-psk', 'encryption': 'aes', 'preshared_key': '2bchanged', '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': 'disabled'}, 'error': []}
設定状況:
{'content': {'id': 1, 'port_id': 101, 'state': 'ready', 'enable_state': 'enabled', 'operating_mode': 'mcb', 'port_forwarding': 'enabled', 'wds_aging_time': '60 sec', '802.11_mode': 'gn', 'channel': 1, 'bandwidth': '20 MHz', 'max_clients': 10, 'channel_scanlist': 'all', 'start_scanning': 'disabled', 'wds_broadcast_mode': 'disabled', 'hide_ssid': 'disabled', 'fast_eapol_retry_mode': 'enabled', 'tx_power_conf': '20 dBm', 'tx_power_curr': '20 dBm', 'medium_utilization': '0 %', 'scb_manual_mac': '00:00:00:00:00:00', 'scb_mode': 'auto', 'fragmentation_threshold': 0, 'hardware_id': 1, 'network_id': 1, 'excessive_retries': 50, 'roaming_trigger_manual': 'enter ap mac-address', 'roaming_network_idle_time': '2 ms', 'roaming_rssi_threshold_force_scan': '-90 dBm', 'roaming_rssi_change_diff': '5 dB', 'roaming_rssi_change_bg_scan_diff': '4 dB', 'roaming_rssi_threshold_bg_scan': '-60 dBm', 'apply_configuration_state': 'ok', 'apply_configuration': 'disabled'}, 'error': []}
設定状況:
{'content': {'global_activation': 'enabled', 'outdoor_mode': 'disabled', 'country': 'Germany', 'management_access': 'enabled', 'wifi_individual': 'enabled', 'ptcp_lldp_filter': 'enabled', 'cyclic_rssi_tracking': 'disabled', 'radio': [{'id': 1, 'aggregation_mode': 'enabled', 'antenna_mask': 3}], 'apply_configuration_state': 'ok', 'apply_configuration': 'disabled'}, 'error': []}
設定状況:
{'content': {'interface_id': 1, 'ip_assignment': 'bootp', 'ip_address': '169.254.2.1', 'network_mask': '255.255.0.0', 'default_gateway': '0.0.0.0', 'management_vlan': 1, 'ip_parameter_state': 'not modified', 'ip_parameter_apply': 'disabled'}, 'error': []}
C:\Users\Phoenix\Documents>
```

Rest API

- Pythonによる取得サンプルプログラム(4/4)

- 本プログラムで取得したJSONデータ

```
{"content": {"interface_id": 1,  
"ip_assignment": "bootp",  
"ip_address": "169.254.2.1",  
"network_mask": "255.255.0.0",  
"default_gateway": "0.0.0.0",  
"management_vlan": 1,  
"ip_parameter_state": "not modified",  
"ip_parameter_apply": "disabled"},  
"error": []}
```

WBM画面より

The screenshot shows the WBM configuration page for 'WLAN-82593b'. The 'Network' section contains the following fields:

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

The 'Hostname Configuration' section contains:

- Name resolution Enable
- Hostname WLAN-82593b

Buttons at the bottom: Apply, Revert, Apply&Save.

Legend:

- ① IP Address Assignment
- ② IP Address
- ③ Network Mask

Thank you