



Welcome

**Linux
Pythonプログラムを用いた
GoogleAppsScriptとの
データ送受信**

概要

概要

- 今回はPythonプログラムを実行することでデータの送受信が行えるプログラムを作成いたします。

今回はpythonプログラムを実行した際に以下のデータを送受信いたします。

【送信データ】

Timestamp : [現在時刻]
temp : 30
humi : 60

【受信データ】

set_data_1 : 50
set_data_2 : 80



python
プログラム

【送信データ】
timestamp : [現在時刻]
temp : 30
humi : 60



Google Apps Script

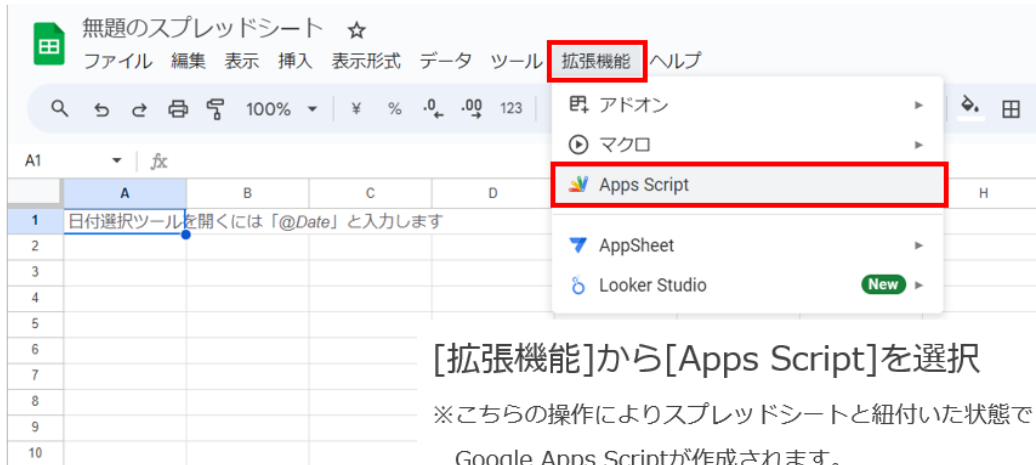
【受信データ】
set_data_1 : 50
set_data_2 : 80

今回の送受信データ

GoogleAppScript の設定

GoogleAppScript の設定

- まずはgoogleドライブにアクセスし、GoogleAppScriptのファイルを作成します。



無題のスプレッドシート ☆
ファイル 編集 表示 挿入 表示形式 データ ツール **拡張機能** ヘルプ

アドオン
マクロ
Apps Script
AppSheet
Looker Studio

[拡張機能]から[Apps Script]を選択
※こちらの操作によりスプレッドシートと紐付いた状態で
Google Apps Scriptが作成されます。

Google スプレッドシート



Apps Script 無題のプロジェクト

ファイル Az + 実行 デバッグ 関数なし 実行ログ

コード.gs

```
1 function myFunction() {  
2  
3 }  
4
```

プロジェクトが作成されます

GoogleAppScript

GoogleAppScript の設定

- プロジェクトに
右のプログラムを記載します。

```
function doPost(e) {  
    // 現在のシートデータを取得  
    const sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();  
  
    // 取得データを処理  
    const plcData = JSON.parse(e.postData.contents);  
  
    // 取得データを変数格納  
    const timestamp = plcData.timestamp;  
    const temp = plcData.temp;  
    const humi = plcData.humi;  
  
    //シートの最終行にデータを書き込む(データ取得チェック)  
    sheet.appendRow([timestamp, temp, humi]);  
  
    // JSON形式で返すデータを設定  
    const responseData = {  
        set_data_1: 50,  
        set_data_2: 80  
    };  
  
    // 設定データをレスポンスとして返す  
    return ContentService.createTextOutput(JSON.stringify(responseData)).setMimeType(ContentService.MimeType.TEXT);  
};
```

GoogleAppScript の設定

- 画面右上の[デプロイ]より[新しいデプロイ]を選択します。



GoogleAppScript の設定

- ウェブアプリとしてデプロイし、発行されたURLをコピーしてください。

新しいデプロイ

種類の選択 設定

ウェブアプリ

[種類の選択]
より
[ウェブアプリ]
を選択

説明

新しい説明文
TEST

ウェブアプリ

次のユーザーとして実行:
自分 ([redacted]@gmail.com)

このウェブアプリケーションを実行するために、あなたのアカウントデータを使用することを許可します。

アクセスできるユーザー
全員

[アクセスできるユーザー]
は[全員]を選択

キャンセル



新しいデプロイ

デプロイを更新しました。

バージョン 1 (2024/06/30 18:08)

デプロイ ID
[redacted]

ウェブアプリ

URL
[https://script.google.com/macros/s/\[redacted\]](https://script.google.com/macros/s/[redacted])

URLが発行されるので
こちらをコピーします。
(Pythonプログラムに貼り付けます)

新しいデプロイ

このウェブアプリケーションを使用するには、データへのアクセスを許可する必要があります。

データアクセスの許可を求められた場合は
[アクセスを承認]を選択してください。

キャンセル

LinuxOS機器 の設定

LinuxOS機器 の設定

- 対象機器にアクセスし、
[test.py]という名前のファイルを
右の内容にて作成します。

Google Apps ScriptのURLについては
先ほどデプロイした際に発効された
URLを入力してください。

```
import requests
import json
from datetime import datetime

# Google Apps ScriptのウェブアプリURL
url = 'https://script.google.com/macros/s/xxxxxxxxxxxxxxxxxxxxx'

# 送信するデータの準備
data = {
    'timestamp': datetime.now().strftime('%Y-%m-%d %H:%M:%S'),
    'temp': 30, # 送信するデータ値
    'humi': 60
}

# 送信データを出力
print(f'timestamp: {data["timestamp"]}')
print(f'temp: {data["temp"]}')
print(f'humi: {data["humi"]}')

# HTTP POSTリクエストを送信
response = requests.post(url, data=json.dumps(data))

# レスポンスを確認
response_data = response.json()
set_data_1 = response_data.get('set_data_1')
set_data_2 = response_data.get('set_data_2')

# レスポンスデータを出力
print(f'set_data_1: {set_data_1}')
print(f'set_data_2: {set_data_2}')
```

動作確認

動作確認 -LinuxOS機器側-

- LinuxOS機器にて[test.py]プログラムを実行してください。

実行した際に、
右図のように
・送信データ
・受信データ
が出力されます。

※機器がインターネットに繋がっていないと
正常に動作しません。

```
Pythonプログラム実行
: $ python3 test.py
timestamp: 2024-06-30 18:21:33 送信データ
temp: 30
humi: 60
set_data_1: 50 受信データ
set_data_2: 80
```

動作確認 -Google Apps Script側-

- Google Apps Scriptを作成する際に開いたスプレッドシートを確認します。

対象のスプレッドシートに

timestamp

temp

humi

のデータが最終行に追加されます。



The screenshot shows a Google Sheets interface for a spreadsheet titled "無題のスプレッドシート". The menu bar includes options like "ファイル", "編集", "表示", "挿入", "表示形式", "データ", "ツール", and "拡張". The toolbar shows search, undo, redo, print, and zoom (100%) icons, along with currency and percentage symbols. The spreadsheet grid has columns A, B, and C, and rows 1, 2, 3, and 4. Row 1 contains the values "2024-06-30 18:21:33", "30", and "60" in columns A, B, and C respectively. Row 2 contains the labels "timestamp", "temp", and "humi" in columns A, B, and C respectively. Rows 3 and 4 are empty.

	A	B	C	D
1	2024-06-30 18:21:33	30	60	
2	timestamp	temp	humi	
3				
4				

Thank you