Welcome

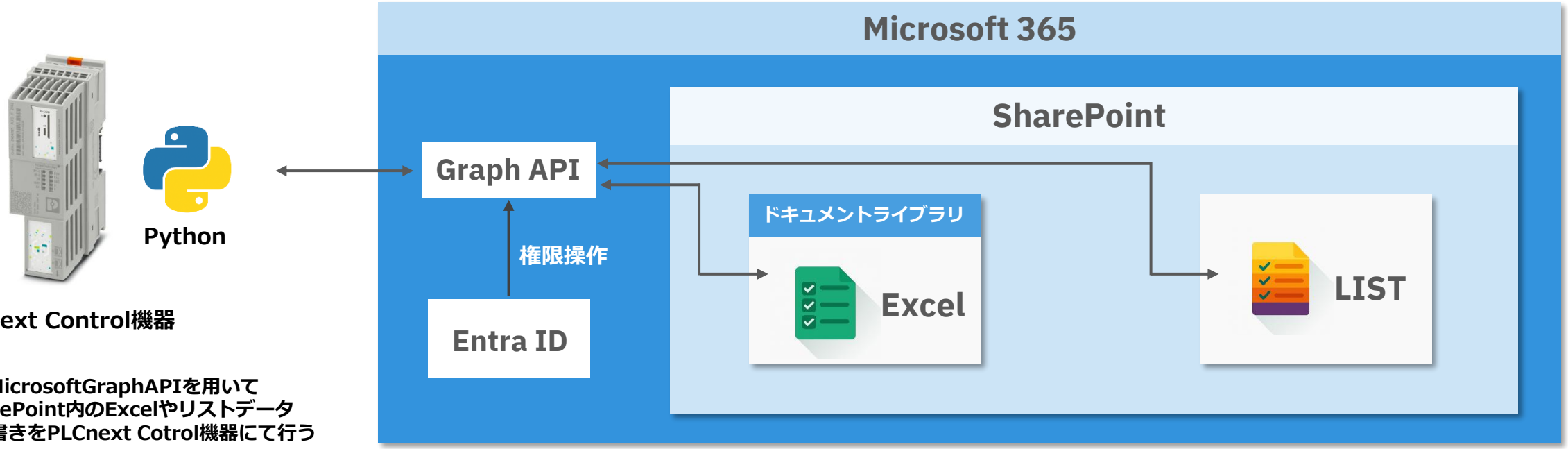
エッジデバイスから ExcelやListsとデータ連携(2/4)

-現場データをもっと身近にする仕組み-
【GraphAPI編】

- 概要 -

検証の概要

- 検証の概要は以下となります。



GraphExplorerによる SharePoint関連IDの確認

SharePoint関連IDの確認

- 以下が本項目の概要となります

PCのwebブラウザからアクセス

Graph Explorer ⟷ Graph API ⟷ Microsoft 365 / SharePoint / ドキュメント / Excel / LIST

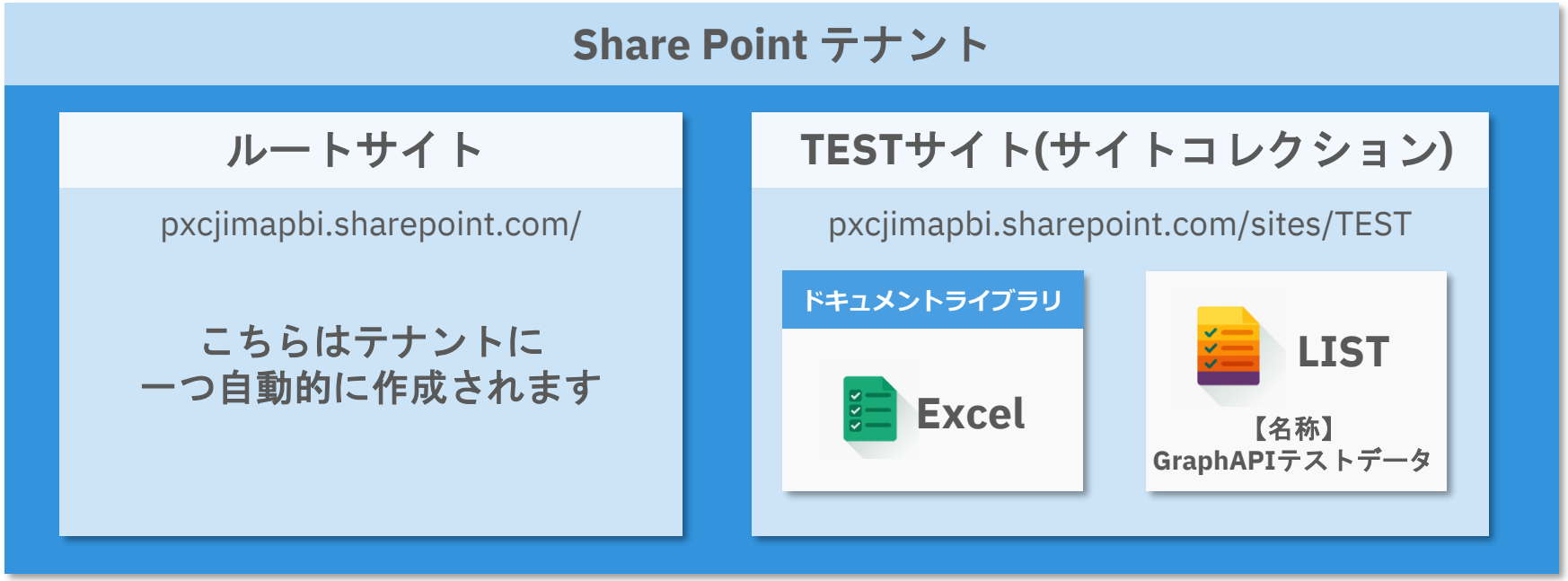
APIにて以下ID情報の取得
・対象のサイトID
・対象のリストID

GraphAPIにてデータを取得するにあたり
必要なSharePointサイトIDやリストIDなどを確認するため、
GraphExplorerというwebアプリサービスにて
API実行しID確認を行う。

【GraphExplorer サイトURL】
https://developer.microsoft.com/en-us/graph/graph-explorer

SharePoint関連IDの確認

- デモのSharePointサイト構成を記載します

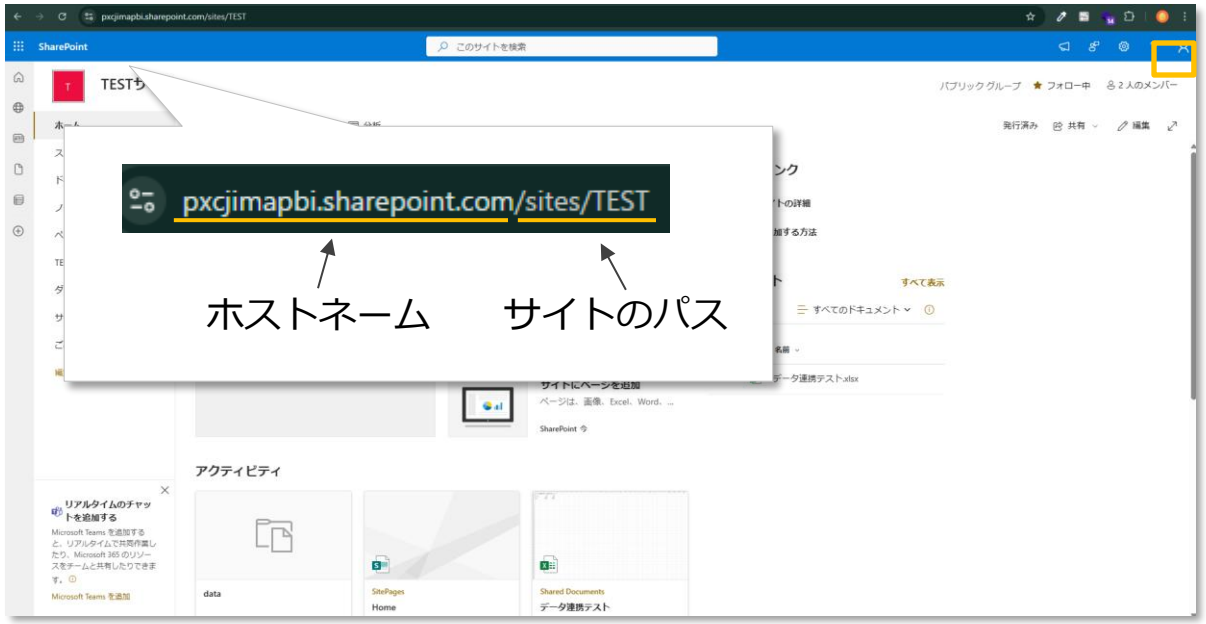


SharePoint関連IDの確認

- 対象サイトコレクションのパスを確認します

  ・対象のサイトにアクセスします。

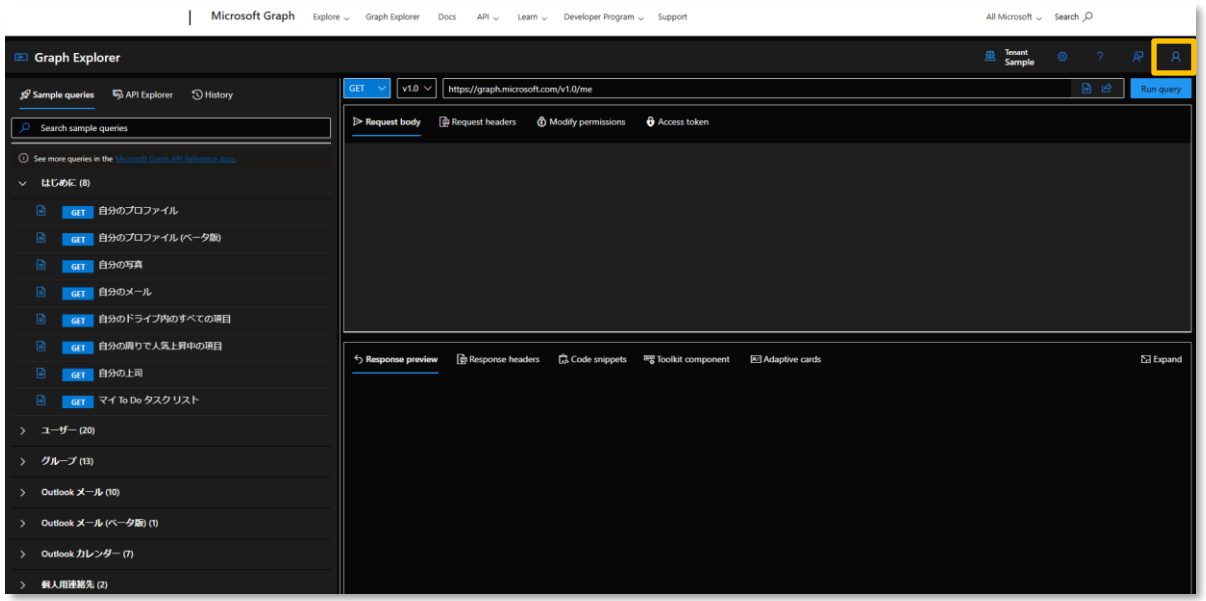
  ・URLより対象サイトの
  ホストネーム
  サイトのパス
  を確認します



SharePoint関連IDの確認

- Graph Explorerにログインします

  ・Graph Explorerにアクセスしログインを行います
  ※ログインを行っていない場合
  デモのAPIリクエストが実行されます



SharePoint関連IDの確認

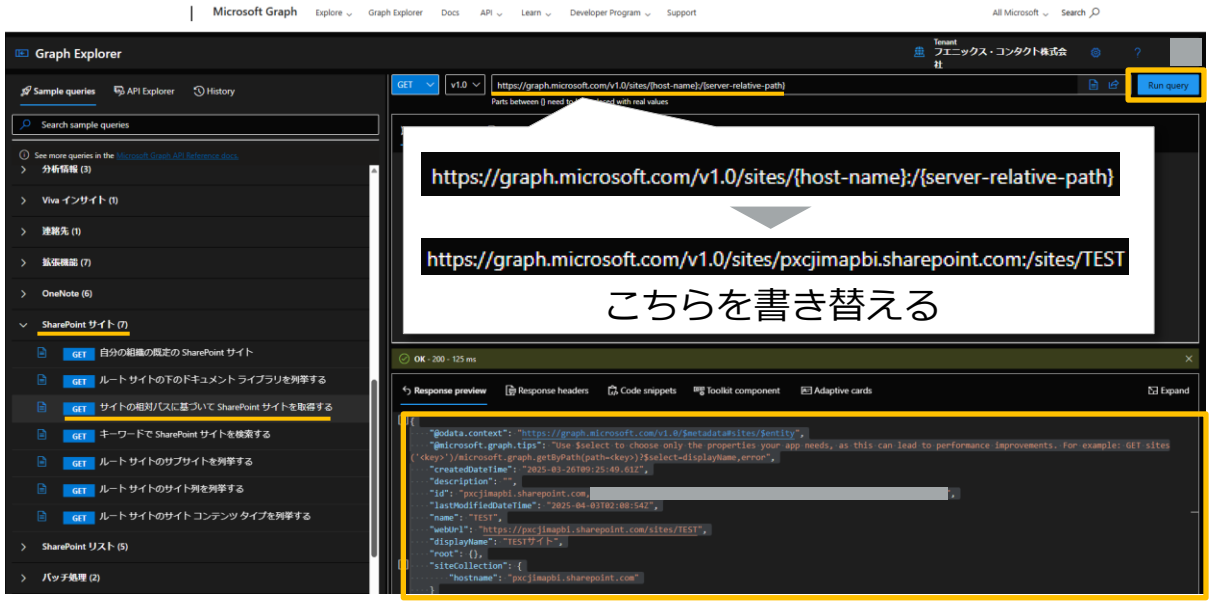
- Graph ExplorerにてサイトIDを取得します

  ・左メニューより[SharePointサイト]-
  [サイトの相対パスに基づいて
  SharePointサイトを取得する]
  をクリックします

  ・画面中央上のURLのホストネームとパスを
  さきほど確認したものに書き替えます

  ・画面右上のRun queryで対象リクエストを
  実行します

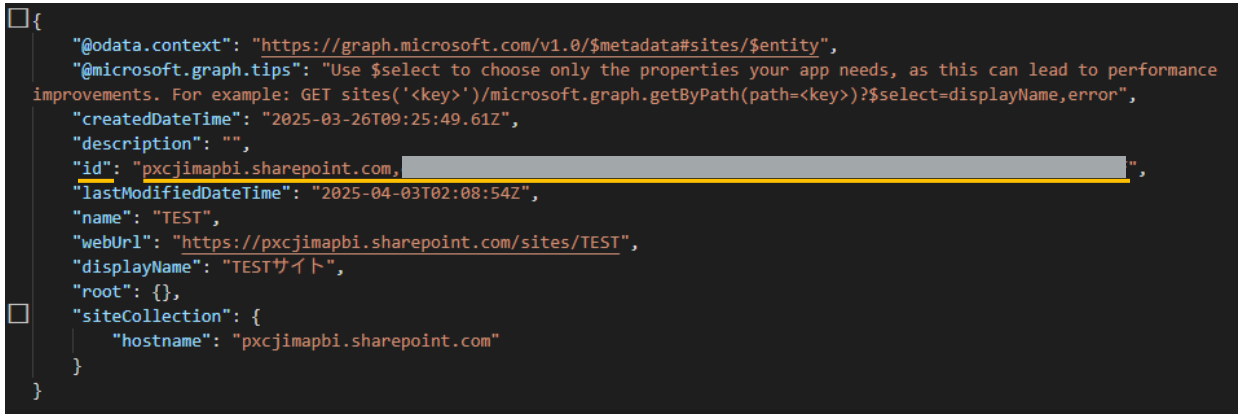
  ・画面下の[Response preview]より
  対象のレスポンスが確認できます
  このレスポンスにサイトIDが記載されています



SharePoint関連IDの確認

- Graph ExplorerにてサイトIDを取得します

  ・レスポンス内の[id]の値が
  サイトIDになります
  こちらの値を保管してください
  ※後の操作で使用します



SharePoint関連IDの確認

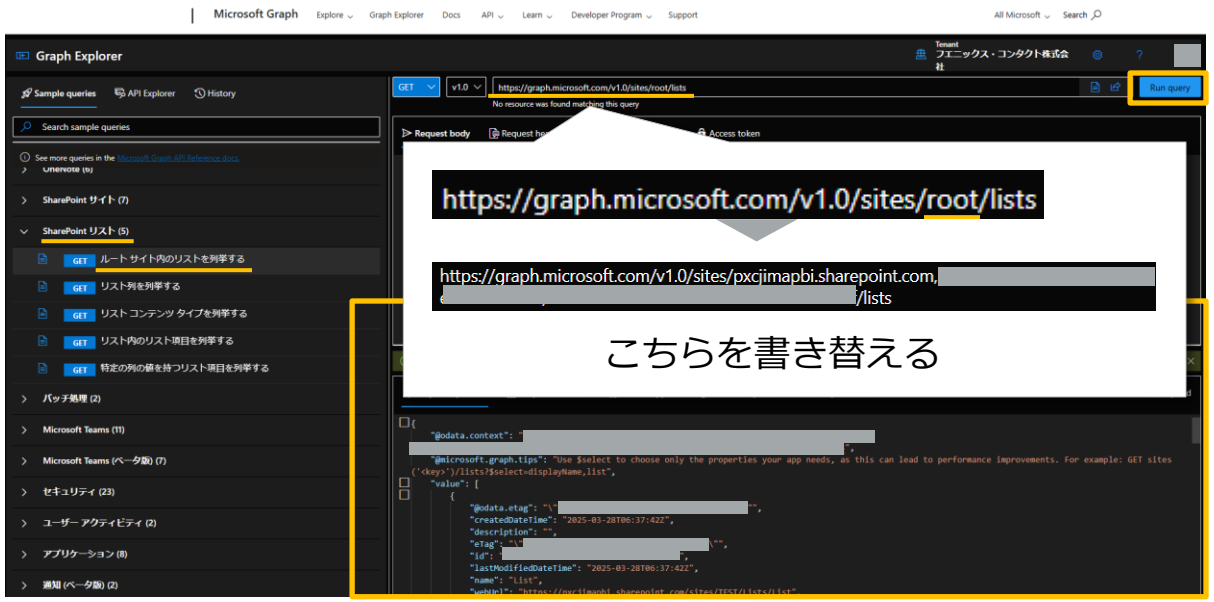
- Graph ExplorerにてリストIDを取得します

  ・左メニューより[SharePointリスト]-
  [ルートサイト内のリストを列挙する]
  をクリックします

  ・画面中央上のURLのrootの部分を先ほど
  取得したサイトIDに書き替えます

  ・画面右上のRun queryで対象リクエストを
  実行します

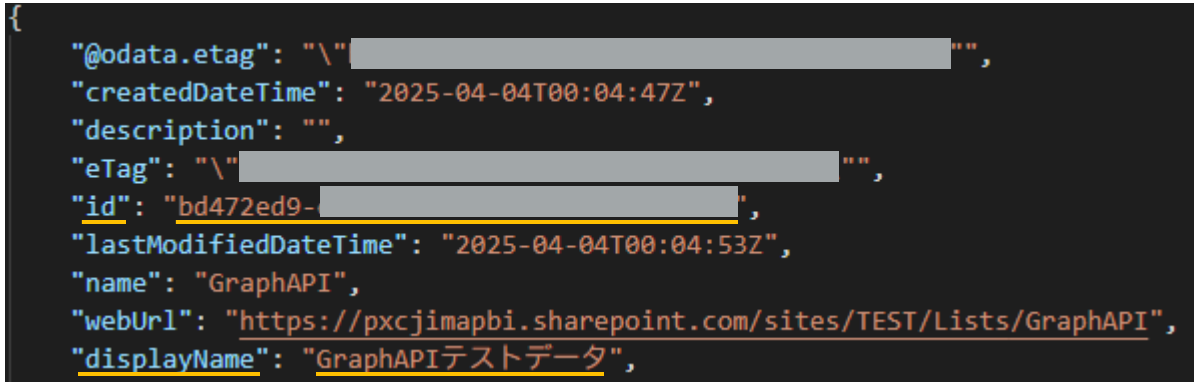
  ・画面下の[Response preview]より
  対象のレスポンスが確認できます
  このレスポンスにリストIDが記載されています



SharePoint関連IDの確認

- Graph ExplorerにてリストIDを取得します

・レスポンスにてすべてのリストのデータ
が返ってきます
この中から[displyaName]の値を確認し
対象の名前と一致しているもの探してください
一致しているもののの[id]の値がリストID
になります
こちらの値を保管してください
※後の操作で使用します



Graph APIの
アクセストークン発行

Graph APIのアクセストークン発行

- 以下が本項目の概要となります

Graph APIのアクセストークン発行

- Codeの発行を行います。

・webブラウザを開き以下のURLを入力します

https://login.microsoftonline.com/{tenant-id}/oauth2/v2.0/authorize?
client_id={**client-id**}&
response_type=code&
redirect_uri={**redirect-uri**}&
response_mode=query&
scope=**Sites.ReadWrite.All**&
state={**任意の値**}

※**黄色文字**のところはそれぞれの値を入力してください
※黄色文字の値の入力の際{}は記載しないでください
※任意の値の部分の仕組みは今回使用しないので好きな値を入れてください
※scopeの設定についてはこちらを参照してください。

・入力したらエンターで実行してください

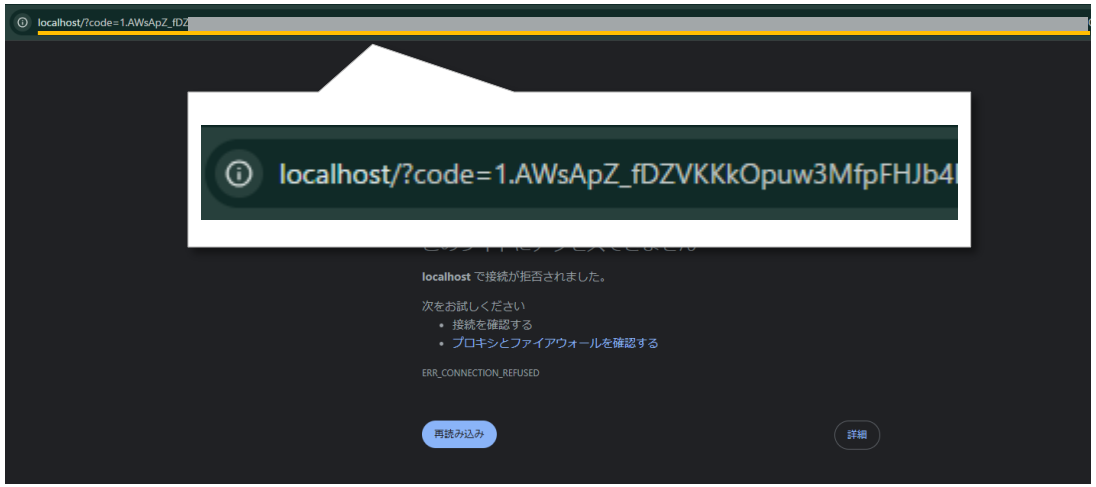
Graph APIのアクセストークン発行

- codeの発行を行います

・URLにてcodeの値が返ってきます。

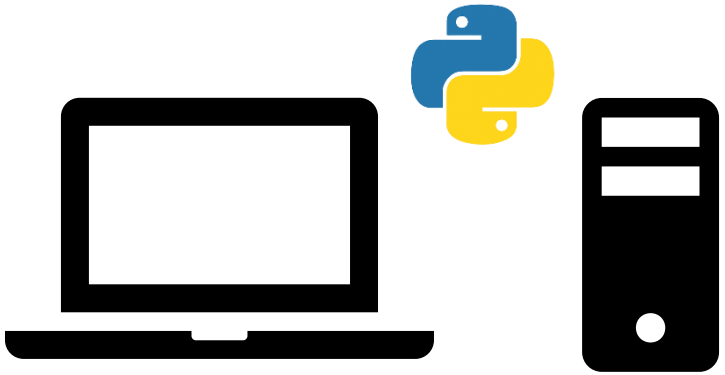
http://localhost/?code=**1.AWsApZ_{省略}&state**=%{省略}
&session_state={省略}#

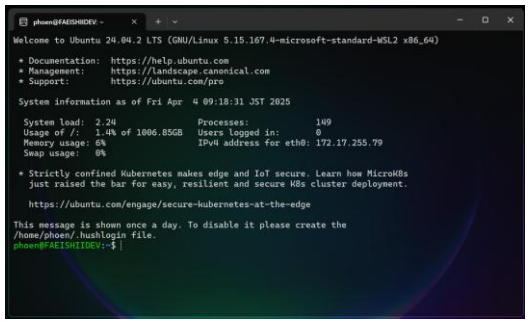
※**黄色文字**の部分がcodeとなります
※終点は**&state**が目印となります
※発行されたcodeは約10分間のみ有効となります



Graph APIのアクセストークン発行

- ここからPythonを使用した作業となります





【環境条件】

Python3.6以上がインストールされていること
pipがインストールされていること

次のページからの説明は対象機器のコマンドラインにて操作を行っております。

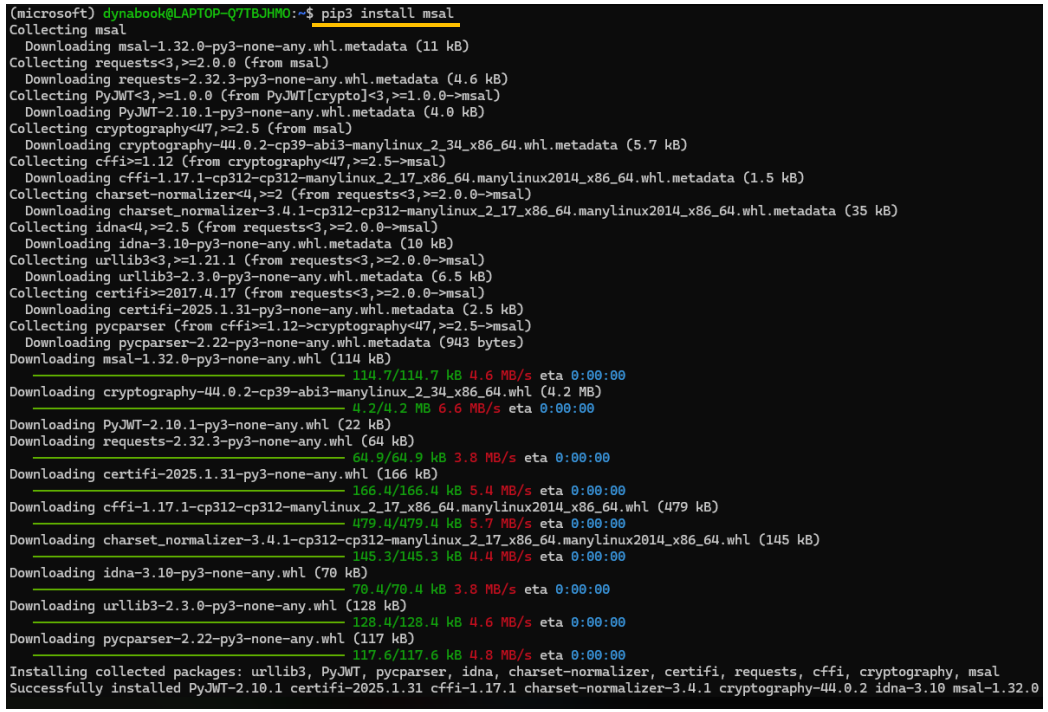
Graph APIのアクセストークン発行

- Pythonのmsalパッケージをインストールします

  ・Python用のMicrosost認証パッケージ(msal)をインストールします

  ・Python3.6以上、pipがインストールされている環境で以下のコマンドを実行してください
  pip3 install msal

  ・以下のコマンドの出力にてmsalが含まれていればインストールされています
  pip3 list



Graph APIのアクセストークン発行

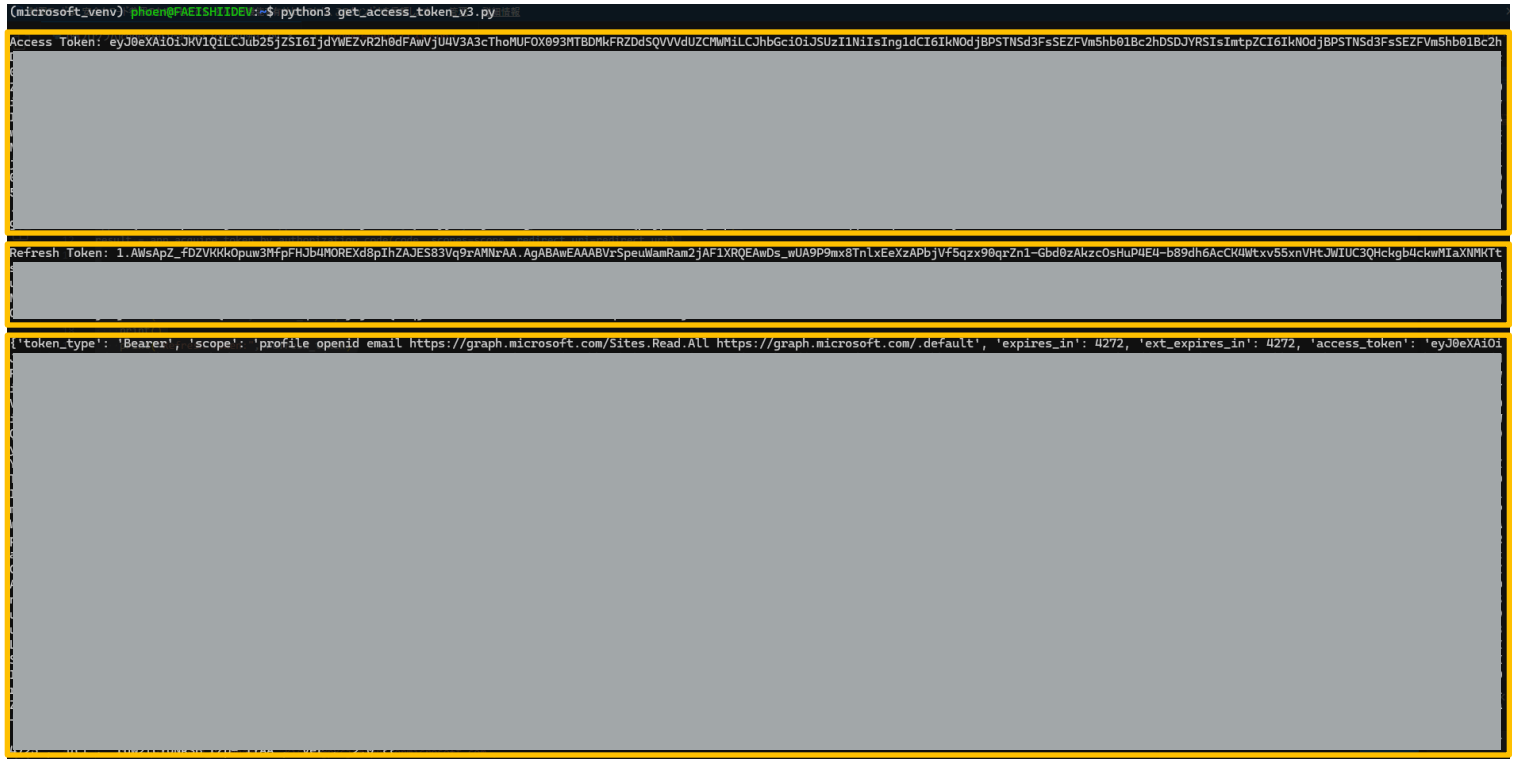
- 以下のアクセストークン発行の用pythonスクリプトを作成し実行します

```
import msal

client_id = '{client_id}'
client_secret = '{client_secret}'
tenant_id = '{tenant_id}'
authority = f"https://login.microsoftonline.com/{tenant_id}"
scope = ["Sites.ReadWrite.All"]
redirect_uri = '{redirect_uri}'
code = '{code}'

app = msal.ConfidentialClientApplication(client_id, authority=authority, client_credential=client_secret)
result = app.acquire_token_by_authorization_code(code, scopes=scope, redirect_uri=redirect_uri)

if "access_token" in result:
    access_token = result["access_token"]
    refresh_token = result["refresh_token"]
    print("Access Token:", access_token)
    print()
    print("Refresh Token:", refresh_token)
    print()
    print(result)
else:
    print("No access token in response:", result)
```

※**黄色文字**のところはそれぞれの値を入力してください
※黄色文字の値の入力の際{}は記載しないでください
※一度アクセストークンを発行する際に使ったcodeは使えません

Graph APIのアクセストークン発行

- アクセストークンの発行に成功すると以下のようなレスポンスが返ってきます



・アクセストークン
API実行時に使います

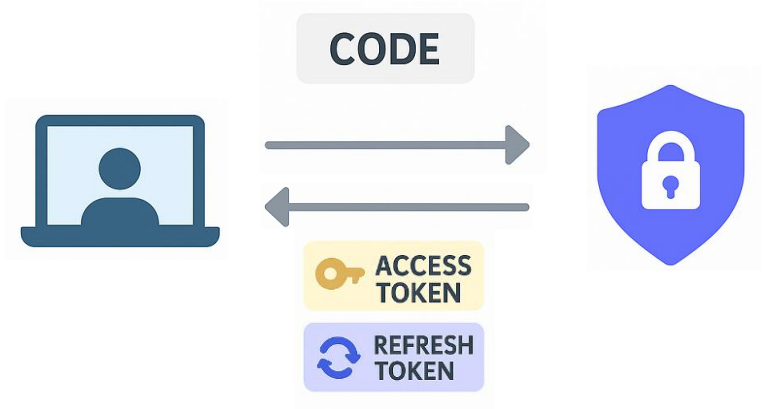
・リフレッシュトークン
アクセストークン再発行時に使います

・レスポンス全体
expires_inの値がアクセストークンの有効期限となります。
(60分~90分)

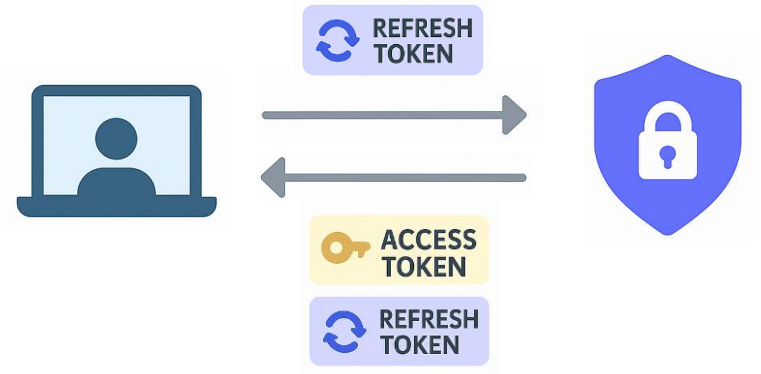
リフレッシュトークンによる アクセストークンの再発行

リフレッシュトークンによるアクセストークンの再発行

- アクセストークン発行時に作成されるリフレッシュトークンを使ってアクセストークンを再発行します



アクセストークンの発行



リフレッシュトークンによる
アクセストークンの再発行

アクセストークンを更新して、新しい有効期限のトークンを再発します。

リフレッシュトークンによるアクセストークンの再発行

- アクセストークンを再発行するpythonスクリプトを作成し実行します

```python
import requests

# 必要な情報を設定
client_id = '{client_id}'
client_secret = '{client_secret}'
tenant_id = '{tenant_id}'
refresh_token = '{refresh_token}'

# トークンエンドポイント（Microsoft共通）
token_url = f"https://login.microsoftonline.com/{tenant_id}/oauth2/v2.0/token"

# リクエストパラメータ
payload = {
    'client_id': client_id,
    'client_secret': client_secret,
    'grant_type': 'refresh_token',
    'refresh_token': refresh_token,
    'scope': 'Sites.ReadWrite.All'
}

# POSTリクエストで新しいトークンを取得
response = requests.post(token_url, data=payload)

# 結果の表示
if response.status_code == 200:
    tokens = response.json()
    print("アクセストークンが更新されました")
    print("Access Token:", tokens['access_token'])
    print()
    print("Refresh Token:", tokens.get('refresh_token'))
    print()
    print(tokens)
else:
    print("トークン更新に失敗しました")
    print(f"ステータスコード: {response.status_code}")
    print(response.text)
```

アクセストークン

リフレッシュトークン

レスポンス全体

※**黄色文字**のところは
それぞれの値を入力してください
※黄色文字の値の入力の際
{}は記載しないでください

Thank you

PHOENIX CONTACT
INSPIRING INNOVATIONS