



Welcome

エッジデバイスから

ExcelやListsとデータ連携(4/4)

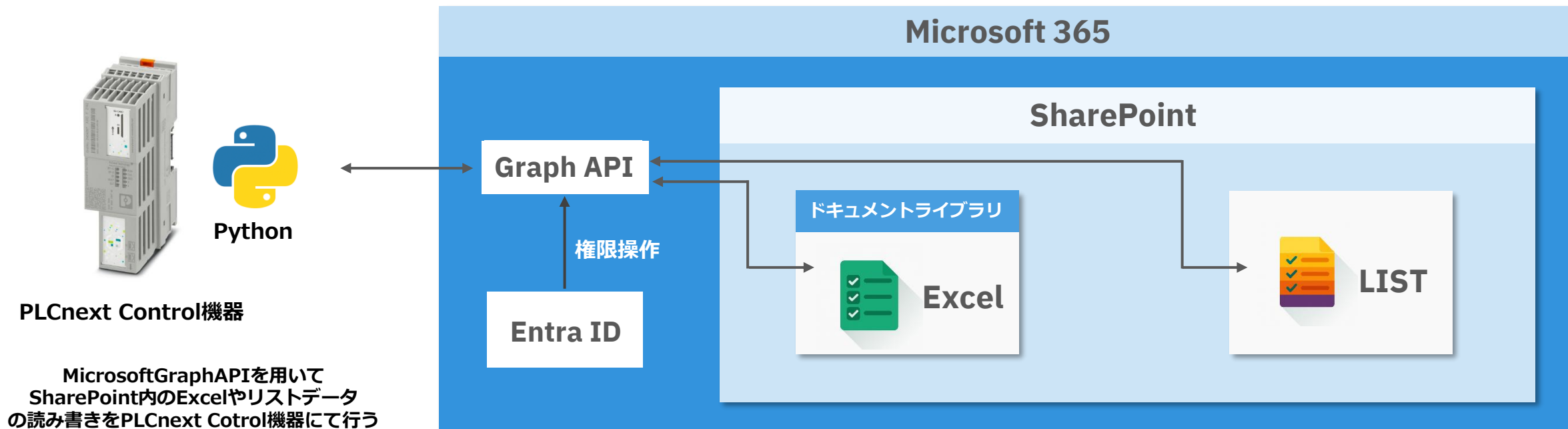
-現場データをもっと身近にする仕組み-

【Excel編】

- 概要 -

検証の概要

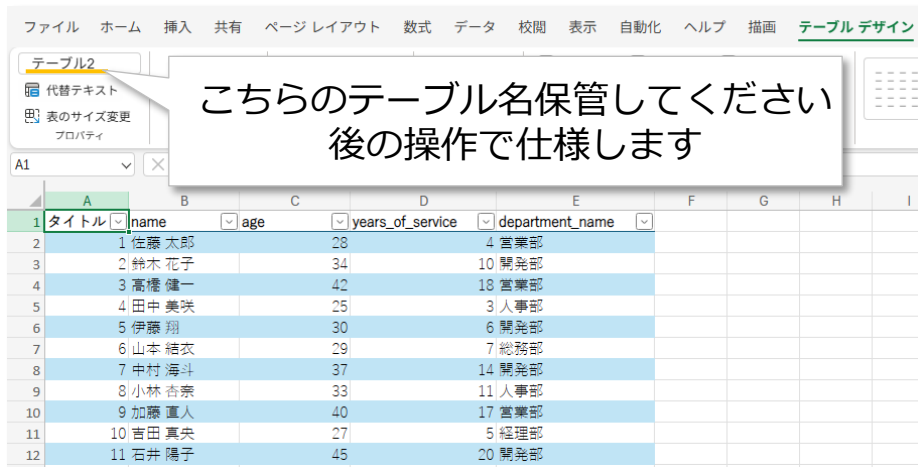
- 検証の概要は以下となります。



- Excelとのデータ連携について -

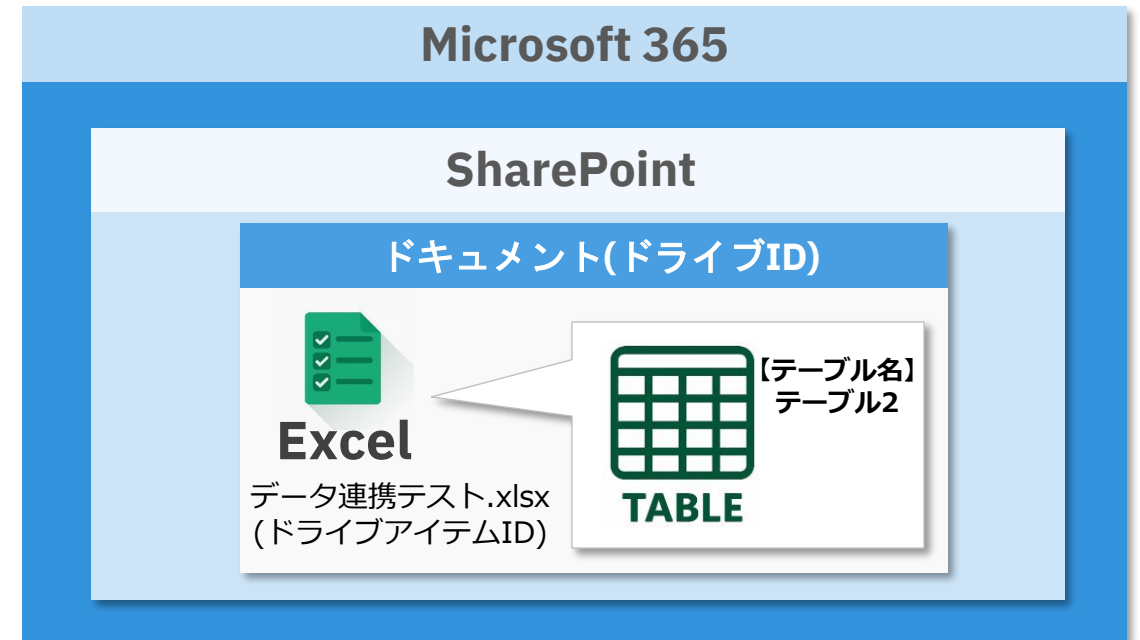
SharePointExcelの事前準備

- 事前準備及び確認を行います



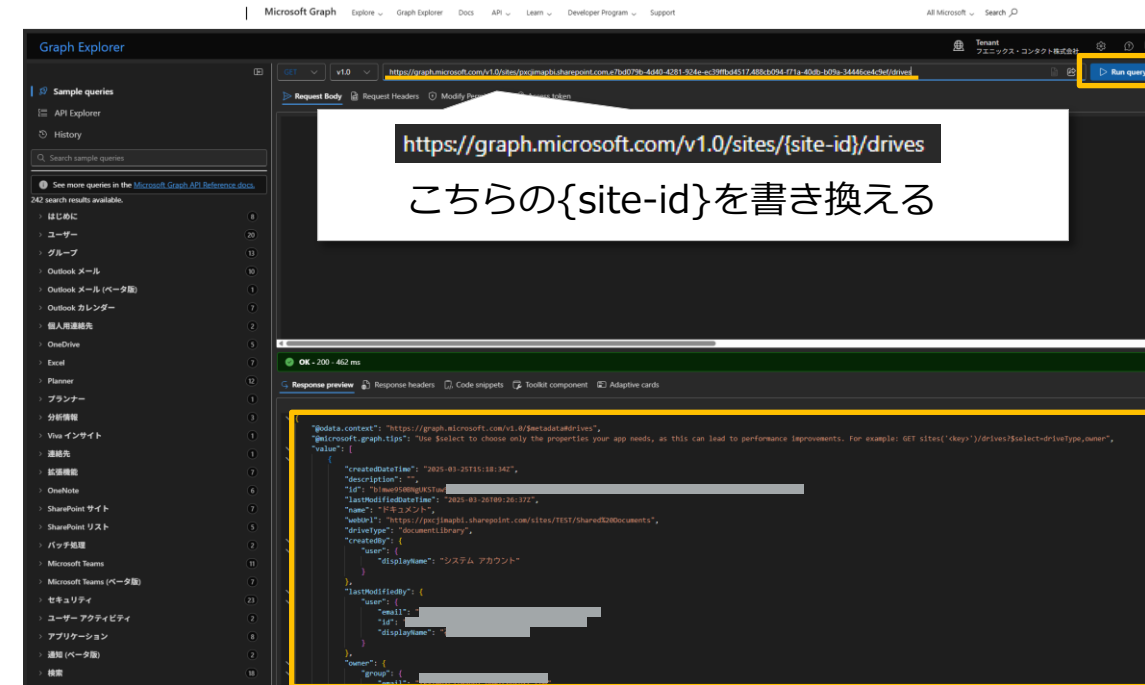
タイトル	name	age	years_of_service	department_name
1	佐藤 太郎	28	4	営業部
2	鈴木 花子	34	10	開発部
3	高橋 健一	42	18	営業部
4	田中 美咲	25	3	人事部
5	伊藤 翔	30	6	開発部
6	山本 結衣	29	7	総務部
7	中村 海斗	37	14	開発部
8	小林 杏奈	33	11	人事部
9	加藤 直人	40	17	営業部
10	吉田 真央	27	5	経理部
11	石井 陽子	45	20	開発部

対象Excelデータをテーブル化します
(対象範囲を選んで[挿入]-[テーブル])



SharePoint関連IDの確認

- Graph ExplorerにてドライブIDを取得します
 - Graph Explorerにてユーザーログインしてください
 - 画面中央上のURLのホストネームとパスを右のような形に書き換えてください。
(`{site-id}`は対象の値を記入してください)
`https://graph.microsoft.com/v1.0/sites/{site-id}/drives`
 - 画面右上のRun queryで対象リクエストを実行します(GETメソッド)
 - 画面下の[Response preview]より対象のレスポンスが確認できます
このレスポンスにドライブIDが記載されています



SharePoint関連IDの確認

- Graph ExplorerにてドライブIDを取得します
 - ・レスポンスにてすべてのドライブのデータが返ってきます
この中から[name]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[id]の値がドライブIDになります
こちらの値を保管してください
※後の操作で使します

```
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#drives",
  "@microsoft.graph.tips": "Use $select to choose only the properties your app needs, as this can lead to performance",
  "value": [
    {
      "createdDateTime": "2025-03-25T15:18:34Z",
      "description": "",
      "id": "b1mwe950BNwUKSTuw5",
      "lastModifiedDateTime": "2025-03-26T09:26:37Z",
      "name": "ドキュメント",
      "webUrl": "https://pxcjimapbi.sharepoint.com/sites/TEST/Shared%20Documents",
      "driveType": "documentLibrary",
      "createdBy": {
        "user": {
          "displayName": "システム アカウント"
        }
      },
      "lastModifiedBy": {
        "user": {
          "email": "",
          "id": "",
          "displayName": ""
        }
      },
      "owner": {
        "group": {

```

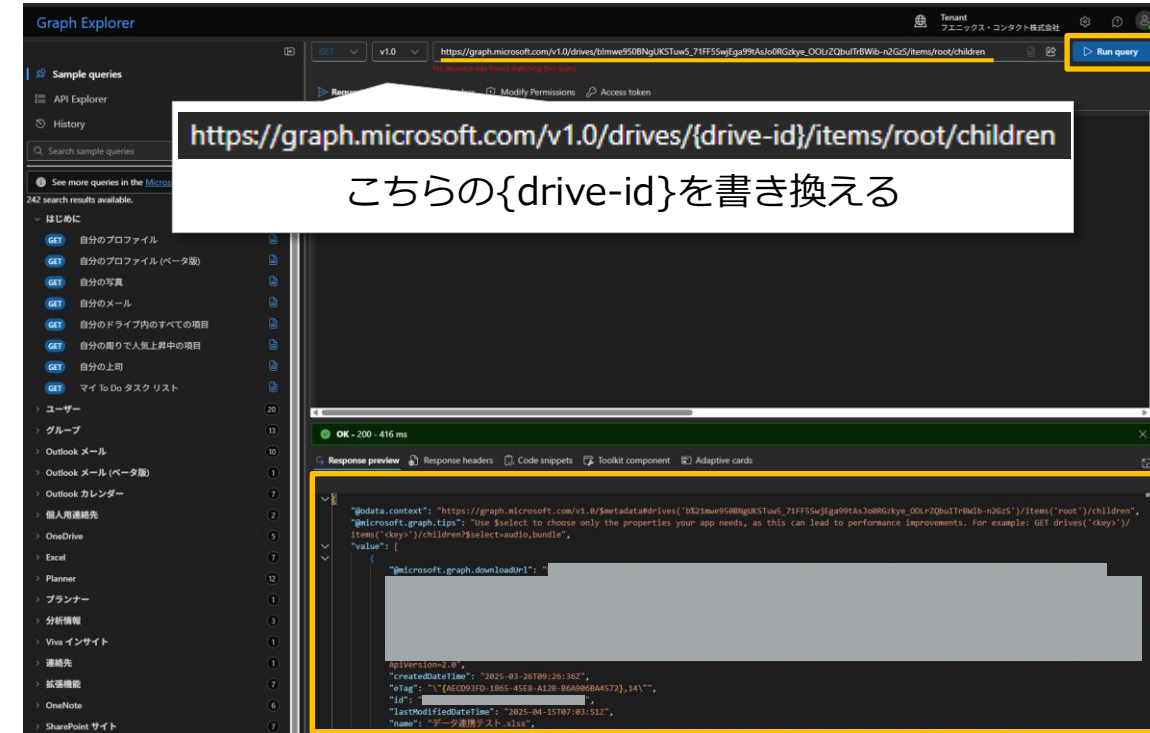
SharePoint関連IDの確認

- Graph ExplorerにてドライブアイテムIDを取得します

- Graph Explorerにてユーザーログインしてください
- 画面中央上のURLのホストネームとパスを右のような形に書き換えてください。
(`{drive-id}`は対象の値を記入してください)

<https://graph.microsoft.com/v1.0/drives/{drive-id}/items/root/children>

- 画面右上のRun queryで対象リクエストを実行します(GETメソッド)
- 画面下の[Response preview]より対象のレスポンスが確認できます
このレスポンスにドライブアイテムIDが記載されています



SharePoint関連IDの確認

- Graph ExplorerにてドライブアイテムIDを取得します

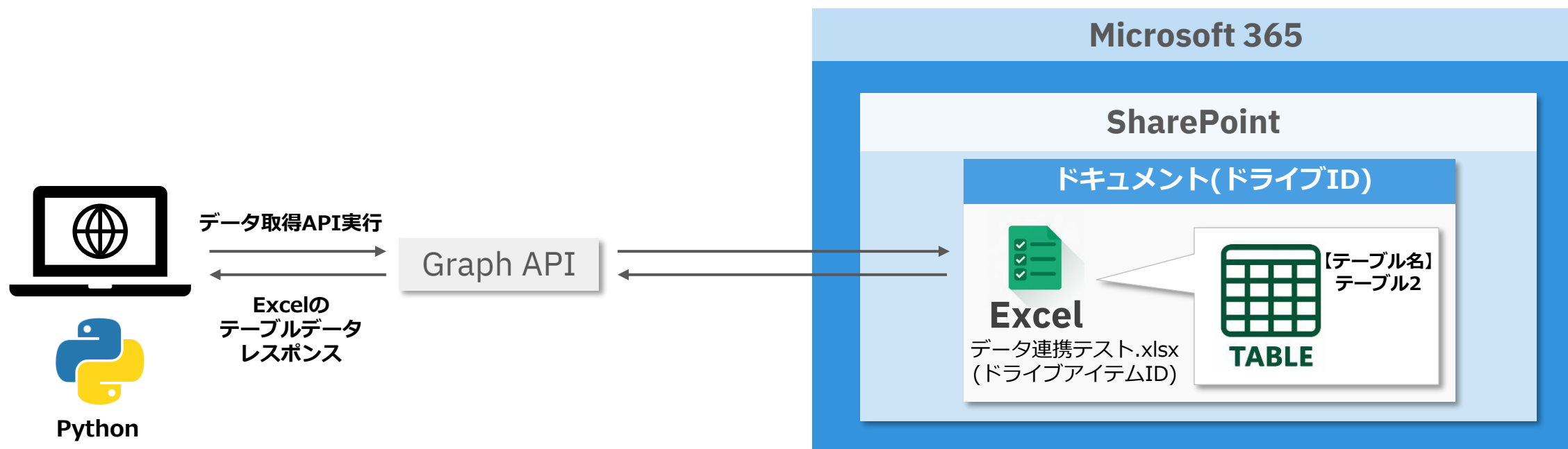
- ・ レスポンスにてすべてのファイルのデータが返ってきます。
この中から[name]の値を確認し
対象の名前と一致しているもの探して
ください一致しているものの[id]の値が
ドライブアイテムIDになります
こちらの値を保管してください
※後の操作で使します

```
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#drives('b%21mwe950BNgUKSTuw5_71FF5SwjEg...')/children?$select=audio,bundle",
  "@microsoft.graph.tips": "Use $select to choose only the properties your app needs, as this can lead to better performance.",
  "value": [
    {
      "@microsoft.graph.downloadUrl": "https://pxcjimapbi.sharepoint.com/sites/TEST/_layouts/15/Doc.aspx?sourcedoc=%7BAE...",
      "ApiVersion=2.0",
      "createdDateTime": "2025-03-26T09:26:36Z",
      "eTag": "\"",
      "id": "1",
      "lastModifiedDateTime": "2025-04-15T07:03:51Z",
      "name": "データ連携テスト.xlsx",
      "webUrl": "https://pxcjimapbi.sharepoint.com/sites/TEST/_layouts/15/Doc.aspx?sourcedoc=%7BAE..."
    }
  ]
}
```

SharePointExcelの データ取得API実行

SharePointExcelのデータ取得API実行

- 以下が本項目の概要となります



SharePointExcelのデータ取得API実行

- 以下がデータ連携テスト.xlsxの構成となります



タイトル	name	age	years_of_service	department_name
1	佐藤 太郎	28	4	営業部
2	鈴木 花子	34	10	開発部
3	高橋 健一	42	18	営業部
4	田中 美咲	25	3	人事部
5	伊藤 翔	30	6	開発部
6	山本 結衣	29	7	総務部
7	中村 海斗	37	14	開発部
8	小林 杏奈	33	11	人事部
9	加藤 直人	40	17	営業部
10	吉田 真央	27	5	経理部
11	石井 陽子	45	20	開発部
12	斎藤 隆	50	25	総務部
13	渡辺 優	26	4	営業部
14	井上 舞	32	9	人事部
15	松本 勇気	38	15	開発部
16	森 彩	24	2	経理部
17	橋本 拓也	31	7	営業部
18	清水 優子	36	13	開発部
19	山田 和也	29	6	人事部

【ヘッダー】

タイトル

name

age

Years_of_service

department_name

レコード数:50

※これらのデータはダミー生成されたデータとなります

※Excelデータの場合カラム情報は取得できません。

SharePointExcelのデータ取得API実行

- 以下がAPI実行時のレスポンスとなります

```
{
  '@odata.context': 'https://graph.microsoft.com/v1.0/$metadata#drives('b%21mwe950BNGUKStun5_71FF5SwjEga99tAsJo0RGzkY00LrZQbuIT
  'value': [
    {
      '@odata.id': '/drives('b%21mwe950BNGUKStun5_71FF5SwjEga99tAsJo0RGzkY00LrZQbuIT
      'index': 0,
      'values': [
        1,
        '佐藤 太郎',
        28,
        4,
        '営業部'
      ]
    },
    {
      '@odata.id': '/drives('b%21mwe950BNGUKStun5_71FF5SwjEga99tAsJo0RGzkY00LrZQbuIT
      'index': 1,
      'values': [
        2,
        '鈴木 花子',
        34,
        10,
        '開発部'
      ]
    },
    {
      '@odata.id': '/drives('b%21mwe950BNGUKStun5_71FF5SwjEga99tAsJo0RGzkY00LrZQbuIT
      'index': 2,
      'values': [
        3,
        '高橋 健一',
        42,
        18,
        '営業部'
      ]
    }
  ]
}
```

```
{
  'index': 0,
  'values': [
    1,
    '佐藤 太郎',
    28,
    4,
    '営業部'
  ]
}
```

タイトル	name	age	years_of_service	department_name
1	佐藤 太郎	28	4	営業部

1レコードずつ二次元配列でデータが返ってきます。

SharePointExcelのデータ取得API実行

- 各レコードのデータ部分のみとindexを出力するpythonスクリプトを作成し実行します

```
import json
import requests
# 各種設定項目
drive_id = "{drive-id}"
item_id = "{item-id}"
table_name = "{テーブル名}"
access_token = "{access_token}"
# Graph API base URL
base_url = "https://graph.microsoft.com/v1.0"
# テーブルデータ取得関数
def get_table_rows(access_token, drive_id, item_id, table_name):
    url = f"{base_url}/drives/{drive_id}/items/{item_id}/workbook/tables/{table_name}/rows"
    headers = {"Authorization": f"Bearer {access_token}"}
    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        result = response.json()
        print(f"{len(result['value'])} 件のリストアイテムを取得しました。")
        return result["value"]
    else:
        print(f"API リクエスト失敗: {response.status_code}")
        print(response.json())
        return []
# メイン処理
if __name__ == "__main__":
    table_rows = get_table_rows(access_token, drive_id, item_id, table_name)
    if not table_rows:
        exit()
    for row in table_rows:
        index = row.get("index", "N/A")
        values = row.get("values", [[]])[0] # values は 2次元配列になっている
        values_str = [str(v) for v in values]
        print(f"[index: {index}] " + " / ".join(values_str))
```



```
(microsoft_venv) phoen@FAEISHIDEV:~/microsoft_API$ python3 get_excel.py
50 件のリストアイテムを取得しました。
[index: 0] 1 / 佐藤 太郎 / 28 / 4 / 営業部
[index: 1] 2 / 鈴木 花子 / 34 / 10 / 開発部
[index: 2] 3 / 高橋 健一 / 42 / 18 / 営業部
[index: 3] 4 / 田中 美咲 / 25 / 3 / 人事部
[index: 4] 5 / 伊藤 翔 / 30 / 6 / 開発部
[index: 5] 6 / 山本 結衣 / 29 / 7 / 総務部
[index: 6] 7 / 中村 海斗 / 37 / 14 / 開発部
[index: 7] 8 / 小林 杏奈 / 33 / 11 / 人事部
[index: 8] 9 / 加藤 直人 / 40 / 17 / 営業部
[index: 9] 10 / 吉田 真央 / 27 / 5 / 経理部
[index: 10] 11 / 石井 陽子 / 45 / 20 / 開発部
```

データ取得APIのPythonスクリプトのレスポンス

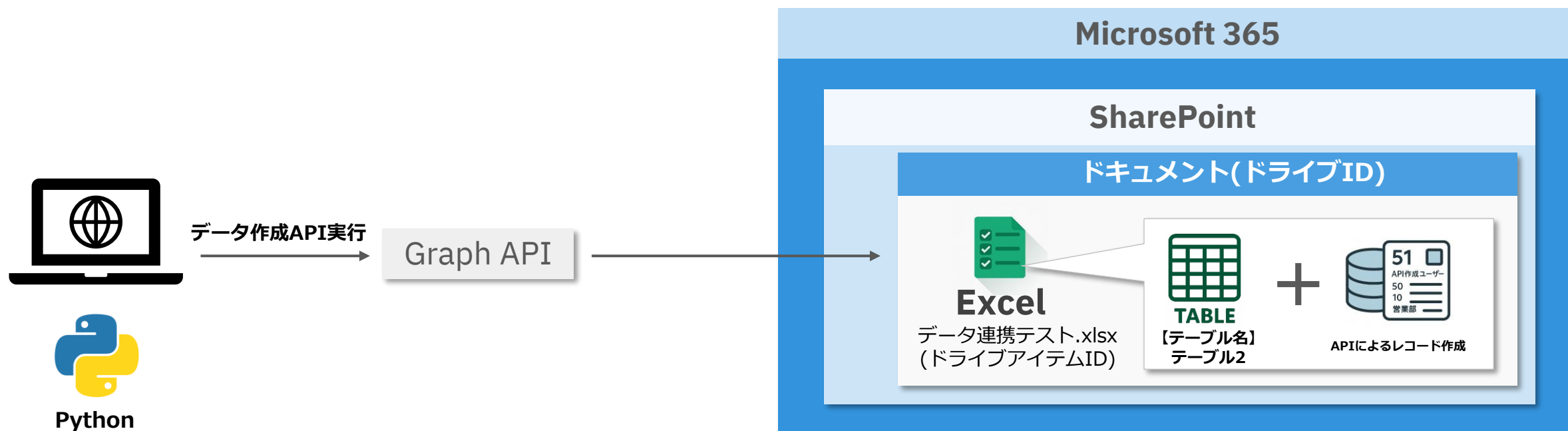
- ①こちらに対象リストのデータレコードがカラムごとに
分かれてすべて表示されます。
最初のIDの値については対象レコードの{index}が
表示されています。
(レコードを検索する際などにindexが使用されます)

※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

SharePointExcelの データ作成API実行

SharePointExcelのデータ作成API実行

- 以下が本項目の概要となります



SharePointExcelのデータ作成API実行

- 以下が作成するレコードの内容になります。

ヘッダー名	値
タイトル	51
name	API作成ユーザー
age	50
years_of_service	10
department_name	営業部

レコード内容

SharePointExcelのデータ作成API実行

- 対象レコードを作成するpythonスクリプトを作成し実行します。

```
import requests
import json

# アクセストークンとファイル情報
drive_id = "{drive-id}"
item_id = "{item-id}"
table_name = "{テーブル名}"
access_token = "{access_token}"

# APIエンドポイント
url = f"https://graph.microsoft.com/v1.0/drives/{drive_id}/items/{item_id}/workbook/tables/{table_name}/rows/add"

# ヘッダー
headers = {
    'Authorization': f'Bearer {access_token}',
    'Content-Type': 'application/json'
}

# 作成するレコード情報記載 (Excelの列順に値を並べます)
payload = {
    "values": [
        ["51", "API作成ユーザー", 50, 10, "営業部"]
    ]
}

# POST リクエストでデータ作成
response = requests.post(url, headers=headers, data=json.dumps(payload))

# 結果確認
if response.status_code == 201:
    print("新しいExcelテーブル行が作成されました")
    print(json.dumps(response.json(), indent=2, ensure_ascii=False))
else:
    print("API リクエスト失敗:", response.status_code)
    print(response.text)
```



45	田村 結城	29	6	開発部
46	黒田 舞	33	10	営業部
47	長谷川 亮	39	16	総務部
48	近藤 悠真	28	4	人事部
49	岩田 愛美	26	3	開発部
50	上野 拓真	45	22	経理部
51	API作成ユーザー	50	10	営業部

sharepointのExcelデータに
作成したレコードが追加されます

※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