



Welcome

エッジデバイスから

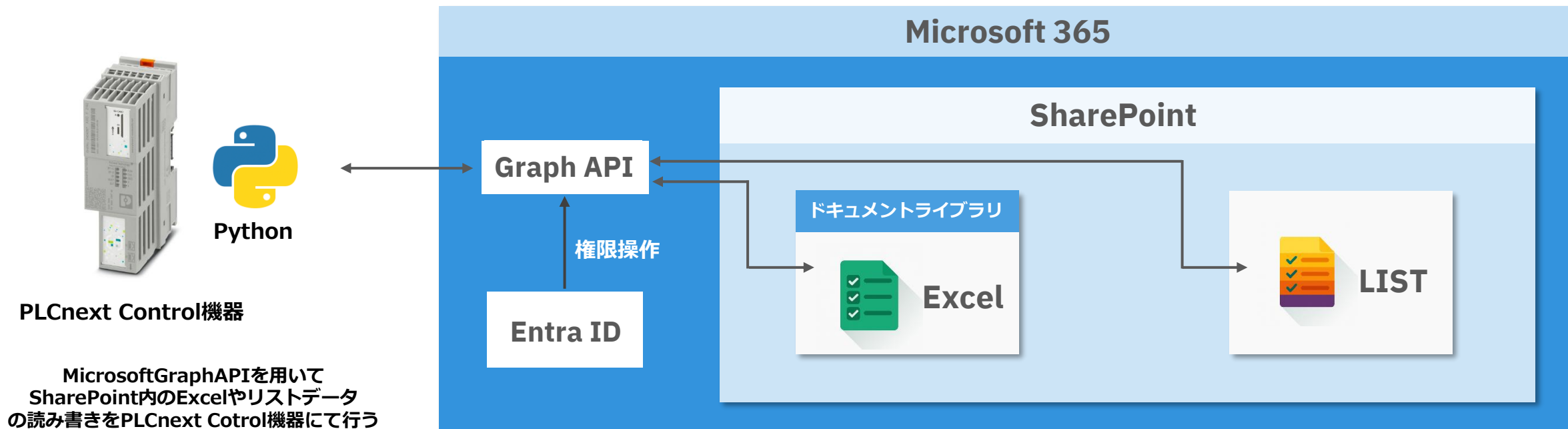
ExcelやListsとデータ連携(3/4)

-現場データをもっと身近にする仕組み-
【Lists編】

- 概要 -

検証の概要

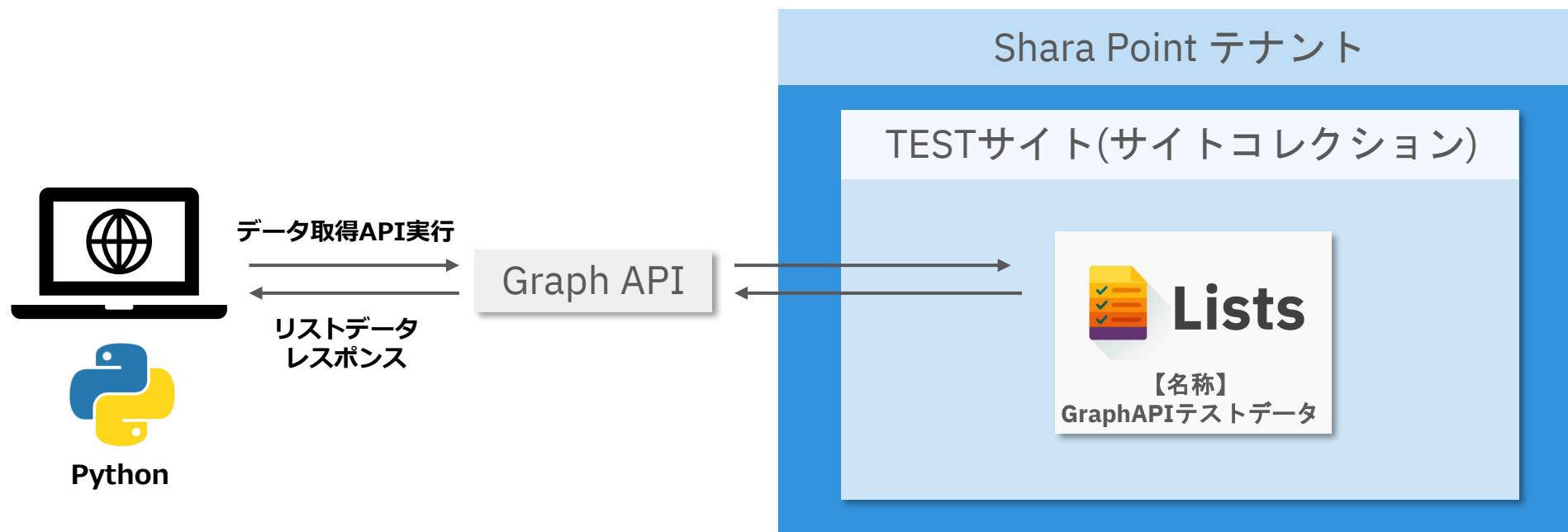
- 検証の概要は以下となります。



SharePointリストの データ取得API実行

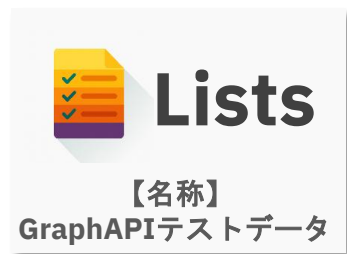
SharePointリストのデータ取得API実行

- 以下が本項目の概要となります



SharePointリストのデータ取得API実行

- 以下がGpaphAPIテストデータの構成となります



GraphAPIテストデータ ☆				
📄 タイトル	📄 name	① age	① 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

(カラム名称 / データ型)

タイトル / テキスト

name / 数値

age / 数値

Years_of_service / 数値

department_name / テキスト

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

SharePointリストのデータ取得API実行

- 以下がAPI実行時のレスポンスとなります

The screenshot displays the JSON response from a SharePoint API call. The response is a list of items, with the first item highlighted. The highlighted item's fields are shown in a table view to the right.

```
{
  '@odata.context': 'https://graph.microsoft.com/v1.0/$metadata#sites('pcxjimapi.sharepoint.c
  'value': [
    {
      'fields': {
        '@odata.etag': '"f1a9ac87-514c-4b22-a46d-11cdc0c084c3,1"',
        'Title': '1',
        'field_1': '佐藤 太郎',
        'field_2': 28.0,
        'field_3': 4.0,
        'field_4': '営業部',
        'id': '1',
        'ContentType': 'アイテム',
        'Modified': '2025-04-04T00:04:48Z',
        'Created': '2025-04-04T00:04:48Z',
        'AuthorLookupId': '7',
        'EditorLookupId': '7',
        '_UIVersionString': '1.0',
        'Attachments': False,
        'Edit': '',
        'LinkTitleNoMenu': '1',
        'LinkTitle': '1',
        'ItemChildCount': '0',
        'FolderChildCount': '0',
        '_ComplianceFlags': '',
        '_ComplianceTag': '',
        '_ComplianceTagWrittenTime': '',
        '_ComplianceTagUserId': ''
      }
    }
  ]
}
```

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

1レコードずつデータが返ってきます。

詳細は以下ページなどを参照ください

<https://learn.microsoft.com/ja-jp/graph/api/listitem-list?view=graph-rest-1.0&tabs=http>

SharePointリストのデータ取得API実行

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

```
import json
import requests

# 各種設定項目
site_id = "{site_id}"
list_id = "{list_id}"
access_token = "{access_token}"

# Graph API base URL
base_url = "https://graph.microsoft.com/v1.0"

# SharePoint リストアイテム取得関数
def get_list_items(access_token, site_id, list_id):
    """SharePoint リストのアイテムを取得"""
    url = f'{base_url}/sites/{site_id}/lists/{list_id}/items?$expand=fields'
    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 []

# 右へ続く→
```

```
# カスタムカラム名だけを抽出 (@odata.etag は無視、id が来たら終了)
def extract_user_defined_keys(fields_dict):
    keys = []
    for key in fields_dict.keys():
        if key == "@odata.etag":
            continue
        if key == "id":
            break
        keys.append(key)
    return keys

# メイン処理
if __name__ == "__main__":
    list_items = get_list_items(access_token, site_id, list_id)

    if not list_items:
        exit()

    # 最初のレコードで列構造を解析
    sample_fields = list_items[0]["fields"]
    column_keys = extract_user_defined_keys(sample_fields)

    print("対象リストのデータ構造は以下になっています。")
    print(" / ".join(column_keys))

    # 各レコードのデータを出力 (item-id付き)
    for item in list_items:
        fields = item["fields"]
        values = [str(fields.get(key, "")) for key in column_keys]
        item_id = fields.get("id", "N/A")
        print(f'[ID: {item_id}] ' + " / ".join(values))
```

```
(microsoft_venv) phoen@FAEISHIIDEV:~/microsoft_API$ python3 get_list.py
51 件のリストアイテムを取得しました。
対象リストのデータ構造は以下になっています。
Title / field_1 / field_2 / field_3 / field_4 ①
[ID: 1] 1 / 佐藤 太郎 / 28.0 / 4.0 / 営業部
[ID: 2] 2 / 鈴木 花子 / 34.0 / 10.0 / 開発部
[ID: 3] 3 / 高橋 健一 / 42.0 / 18.0 / 営業部
[ID: 4] 4 / 田中 美咲 / 25.0 / 3.0 / 人事部
[ID: 5] 5 / 伊藤 翔 / 30.0 / 6.0 / 開発部 ②
```

データ取得APIのPythonスクリプトのレスポンス

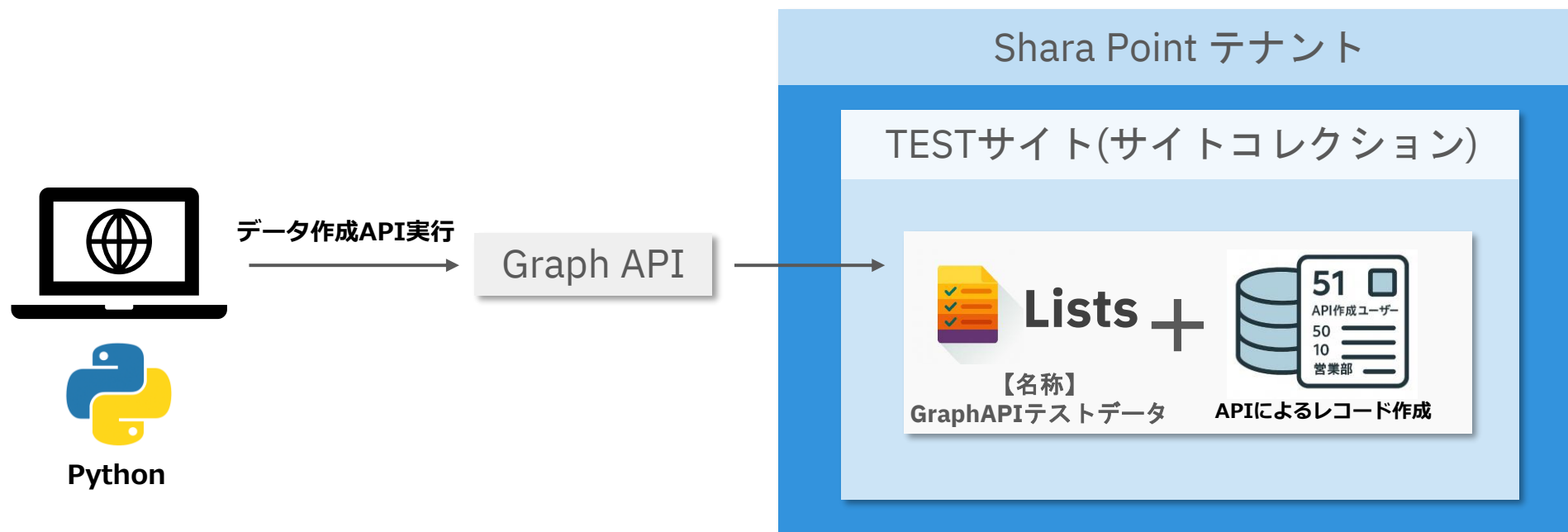
- ①:こちらにカラムごとのキーの名称が表示されます。
このキーの値はデータ作成などで使用します。
- ②:こちらに対象リストのデータレコードがカラムごとに
分かれてすべて表示されます。
最初のIDの値については対象レコードの{item-id}が
表示されています。
(レコードを検索する際などにitem-idが使用されます)

※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

SharePointリストの データ作成API実行

SharePointリストのデータ作成API実行

- 以下が本項目の概要となります



SharePointリストのデータ作成API実行

- 以下が作成するレコードの内容になります。

カラム名称	キー名称	値
タイトル	Title	51
name	field_1	API作成ユーザー
age	field_2	50
years_of_service	field_3	10
department_name	field_4	営業部

レコード内容

SharePointリストのデータ作成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サイト情報
site_id = "{site_id}"
list_id = "{list_id}"
access_token = "{access_token}"
# APIエンドポイント
url = f"https://graph.microsoft.com/v1.0/sites/{site_id}/lists/{list_id}/items"
# 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 ※こちらに作成したいレコードの情報を記載してください。
payload = {
    "fields": {
        "Title": "51",
        "field_1": "API作成ユーザー",
        "field_2": 50,
        "field_3": 10,
        "field_4": "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("新しいリストアイテムが作成されました")
    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)
```



○ 田	田 タイトル	田 name	① age	① years_of_service	田 department_name
	41	菅原 美月		32	9 営業部
	42	大西 大輔		27	5 開発部
	43	柴田 結衣		35	12 人事部
	44	内田 優		31	7 経理部
	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

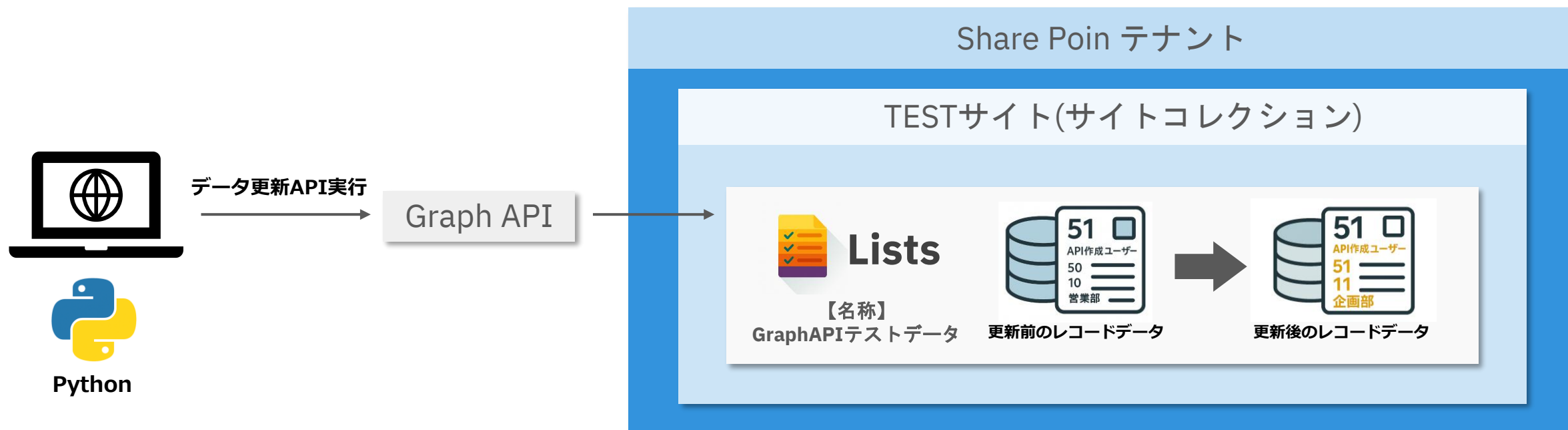
sharepointのリストデータに
作成したレコードが追加されます

※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

SharePointリストの データ更新API実行

SharePointリストのデータ更新API実行

- 以下が本項目の概要となります



SharePointリストのデータ更新API実行

- 以下が更新するレコードの内容になります。

カラム名称	キー名称	値
タイトル	Title	51
name	field_1	API作成ユーザー
age	field_2	50
years_of_service	field_3	10
department_name	field_4	営業部

更新前のレコード内容



カラム名称	キー名称	値
タイトル	Title	51
name	field_1	API作成ユーザー
age	field_2	51
years_of_service	field_3	11
department_name	field_4	企画部

変更後のレコード内容

SharePointリストのデータ更新API実行

- 対象レコードを更新するpythonスクリプトを作成し実行します。

```
import requests
import json

# 各種設定項目
site_id = "{site_id}"
list_id = "{list_id}"
item_id = "{item_id}" # ここを更新したいレコードの ID に書き換えてください
access_token = "{access_token}"
# Microsoft Graph API エンドポイント
endpoint = f"https://graph.microsoft.com/v1.0/sites/{site_id}/lists/{list_id}/items/{item_id}/fields"
# ヘッダー
headers = {
    "Authorization": f"Bearer {access_token}",
    "Content-Type": "application/json"
}
# 更新するデータ (変更したいキーとバリューを記載してください)
payload = {
    "Title": "51",
    "field_1": "API作成ユーザー",
    "field_2": 51,
    "field_3": 11,
    "field_4": "企画部"
}
# PATCH リクエストでレコード更新
response = requests.patch(endpoint, headers=headers, data=json.dumps(payload))
# 結果確認
if response.status_code == 200:
    print("レコードが正常に更新されました")
    print(json.dumps(response.json(), indent=2, ensure_ascii=False))
else:
    print("エラーが発生しました")
    print(f"Status Code: {response.status_code}")
    print(response.text)
```



○ ③ タイトル	⑦ name	① age	① years_of_service	③ department_name
42	八幡 八幡	29	7	開発部
43	柴田 結衣	35	12	人事部
44	内田 優	31	7	経理部
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	51	11	企画部

sharepointのリストデータが
指定した内容に更新されます

※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