

※重要：2022/10/10にパスワードの有効期限について説明を追記

SharePoint Online REST API

2022/10/21...v8

第1電動先行開発部

大路

[本資料の最新版はこちら](#)

まえがき

- 本資料はSharePoint Online(以降SPO)の[REST API v1](#)の基本的な操作をPythonプログラムを用いて実現する方法を示したものです。
 - 本資料で示す手順や情報は、あくまでも自己責任でご使用ください(SPO窓口からも機能使用は自己責任でと回答あり)。
 - 情報には誤りのないように努めていますが、100%正確ではない可能性があります。
 - プログラムに一定レベルの知見がある方向けの資料となっています。
 - 同封のプログラムとセットでご使用ください。
- [ファイル一式はココをクリックしてアクセス](#)

まえがき

● REST APIの嬉しさ・できること

- ・ユーザー依存なしにSPOとデータをやり取りすることができる。
- ・任意のプログラムからSPOとデータをやり取りすることができる。
- ・PowerAutomateではプレミアムコネクタとなっている機能やPowerPlatformアプリ群だけでは実現できない機能を外部プログラム(本資料ではPython)で補完することで実現することができる。
- ・活用例:社内LANサーバにあるファイル又はそれを元に加工したデータをSPOへ自動的にアップロードする、ログ収集システム、データ拠点としての柔軟なアクセスサービス、等。

改訂履歴

バージョン	改訂内容	改訂日
v1	・ なし(初版)	2021/11/19
v2	・ <u>8-7項 フォルダ・ファイル操作のプログラム例</u> を追加	2021/11/22
v3	・ <u>3-2①項</u> にアクセストークンの有効期限のヒント情報を追記	2021/11/23
v4	・ <u>8-2項に対処その4</u> を追加	2021/11/23
v5	・ バッチ処理実装方法攻略により各所ヒントの文言を小変更	2022/1/10
v6	・ <u>改訂履歴のページ</u> を追加 ・ コピー展開対策として最初のページにセルフリンクを追加 ・ <u>8-7項</u> にて操作には事前に権限付与が必要である旨を追加 ・ <u>2-2項</u> にて権限付与に関する注意喚起を追加	2022/5/18
V7	・ SPO アドインIDに対するパスワード(クライアントシークレット)の有効期限について説明 <u>7項</u> と <u>2-1②項</u> に注意書きを追加	2022/10/10
v8	・ 随所にページリンクを挿入 ・ <u>2-3②項</u> のヒントにIDの削除ができると説明していたが、正しくは権限の削除であるため修正 ・ 各所の文言・レイアウト小修正	2022/10/21

目次

1. 前準備

1-1. 使用ファイルの確認

1-2. 対象SPOサイトの設定

1-3. 動作確認用リストの設定

2. REST API用IDの設定

2-1. アドインの登録

2-2. アドインへの権限付与

2-3. 権限付与状況の確認

目次

3.リストアイテムの取得

3-1.リストアイテムの取得

3-2.内容解説

3-3.クエリを用いたリストアイテムの取得

4.リストアイテムの作成

4-1.内部リスト名の確認・設定

4-2.リストアイテムの作成

4-3.内容解説

目次

5.リストアイテムの更新

5-1.アイテムIDの確認・設定

5-2.リストアイテムの更新

5-3.内容解説

6.リストアイテムの削除

6-1.アイテムIDの設定

6-2.リストアイテムの削除

6-3.内容解説

目次

7.REST API用IDのパスワード管理

7-1.パスワードの有効期限

7-2.パスワード有効期限の確認

7-3.パスワード有効期限切れ対応

目次

8.おまけ情報

8-1.権限付与XMLのポイント

8-2.プロキシサーバエラーへの対処

8-3.リストアイテムデータの送信フォーマット

8-4.リストの列名に日本語を使用した際の留意点

8-5.リストアイテムの5000/20000件問題

8-6.リストアイテムデータ更新手法の一例

8-7.フォルダ・ファイル操作のプログラム例



1.前準備

1-1.使用ファイルの確認

1-2.対象SPOサイトの設定

1-3.動作確認用リストの設定



1-1.使用ファイルの確認

① 下記3つのファイルを確認して開く

- apl_info.txt
- SPO_RESTAPI_Basic.py
- SPO_RESTAPI_ListOpe.py

※本資料ではエディターにVisual Studio Codeを使用します

ヒント：各ファイルの役割は以下の通りです。

「apl_info.txt」・・・各種の補助的情報が載っています。

「SPO_RESTAPI_Basic.py」・・・REST API導入動作確認プログラムです。

「SPO_RESTAPI_ListOpe.py」・・・リストアイテムの作成・更新・削除プログラムです。

※プログラムの関数化・モジュール化はしていません。各自で頑張ってください。

ファイル一式はココをクリックしてアクセスしてください。



1-2.対象SPOサイトの設定

①対象のSPOサイトをブラウザで開く

← → ↻ 🔒 https://toyotajp.sharepoint.com/sites/msspo_pss ★ ☆ 📁 大路 ...

SharePoint このサイトを検索 🔍 📣 ⚙️ ? 大路

🏠 🌐 📄 📁

計画書集計システム ホームが分かりやすいためホームを推奨 共有

ホーム + 新規 ▾ 🗑️ 変更の破棄 ⚙️ ページの詳細 ... 下書きの保存日時 2021/9/2 ✎ 編集 📄 再発行 ↗️

ドキュメント お知らせ リンク

試験計画書集計システム

サイト コンテンツ

ごみ箱

編集

お知らせ すべて表示 **リンク** すべて表示

+ 新規 ▾ ... ≡ すべてのアイテム ▾ ⓘ ... ≡ すべてのリンク ▾ ⓘ

タイトル ▾	更新日時 ▾	📄 URL ▾
SPOホームページ	9月2日	SharePointOnline窓口

ヒント：本資料ではSPO単体の払い出しサイトを利用していますが、Teams付随のSPOでも利用可能です。



1-2.対象SPOサイトの設定

②URLのサイト固有部分をコピーする

https://toyotajp.sharepoint.com/sites/msspo_pss

SharePoint このサイトを検索

計画書集計システム

ホーム + 新規 変更の破棄 ページの詳細 下書きの保存日時 2021/9/2 編集 再発行

ドキュメント

お知らせ

リンク

試験計画書集計システム

サイト コンテンツ

ごみ箱

編集

タイトル 更新日時

SPOホームページ... 9月2日

URL

SharePointOnline窓口

sites/の後ろにある部分をコピーする。
本資料の例ではmsspo_pssとなる。
Teams付随の場合はmsteams_xxxxなど。



1-2.対象SPOサイトの設定

③各使用ファイルのサイトURL部分を置換する

1. `apl_info.txt`を開き、置換箇所テキストの
<ここに対象のSPOサイト名を入力>で検索をかける。

2. 前項のコピーしたURL部分を貼り付けて、
3つのファイル全てで全置換を行う。

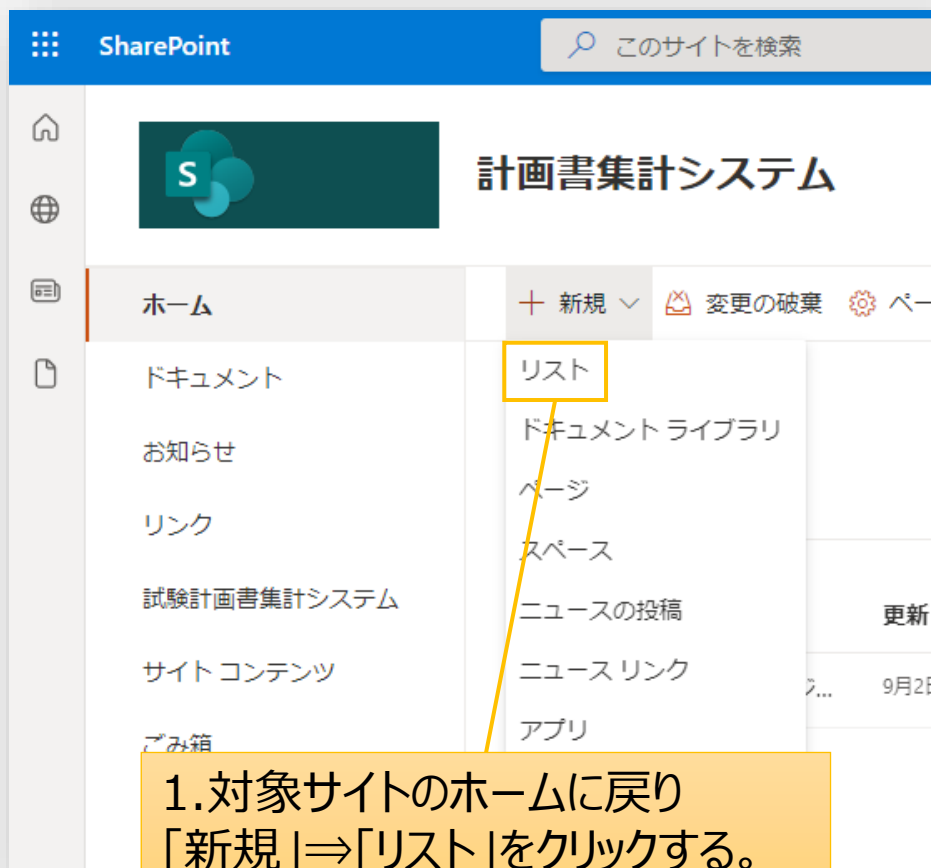
ヒント：動作確認をスムーズに行えるように、手順に沿って情報を
置換していくと、すぐに動作可能な構成になっています。

18 ■アプリ登録ページ
19 `https://toyotajp.sharepoint.com/sites/<ここに対象のSPOサイト名を入力>/_layouts/15/appregnew.aspx`



1-3.動作確認用リストの設定

①空白のリストを作成する





1-3.動作確認用リストの設定

①空白のリストを作成する つづき

名前 *

RESTAPI_TEST

説明

リストの内容は何ですか?

サイトナビゲーション

☐ サイトナビゲーションに表示

戻る

作成

キャンセル

3.任意のリスト名を入力する。
本資料の例では「RESTAPI_TEST」と入力。
※日本語・全角文字は非推奨。
理由はおまけ情報8-4項を参考にしてください。

4.「作成」をクリックする。



1-3.動作確認用リストの設定

②動作確認用のリストアイテムを作成する

SharePoint

このリストを検索

計画書集計システム

★ フォロー中 共有

ホーム

ドキュメント

お知らせ

リンク

試験計画書集計システム

サイト コンテンツ

ごみ箱

+ 新規

グリッドビューでの編集 共有 ...

すべてのアイテム

RESTAPI_TEST ☆

タイトル

+ 列の追加

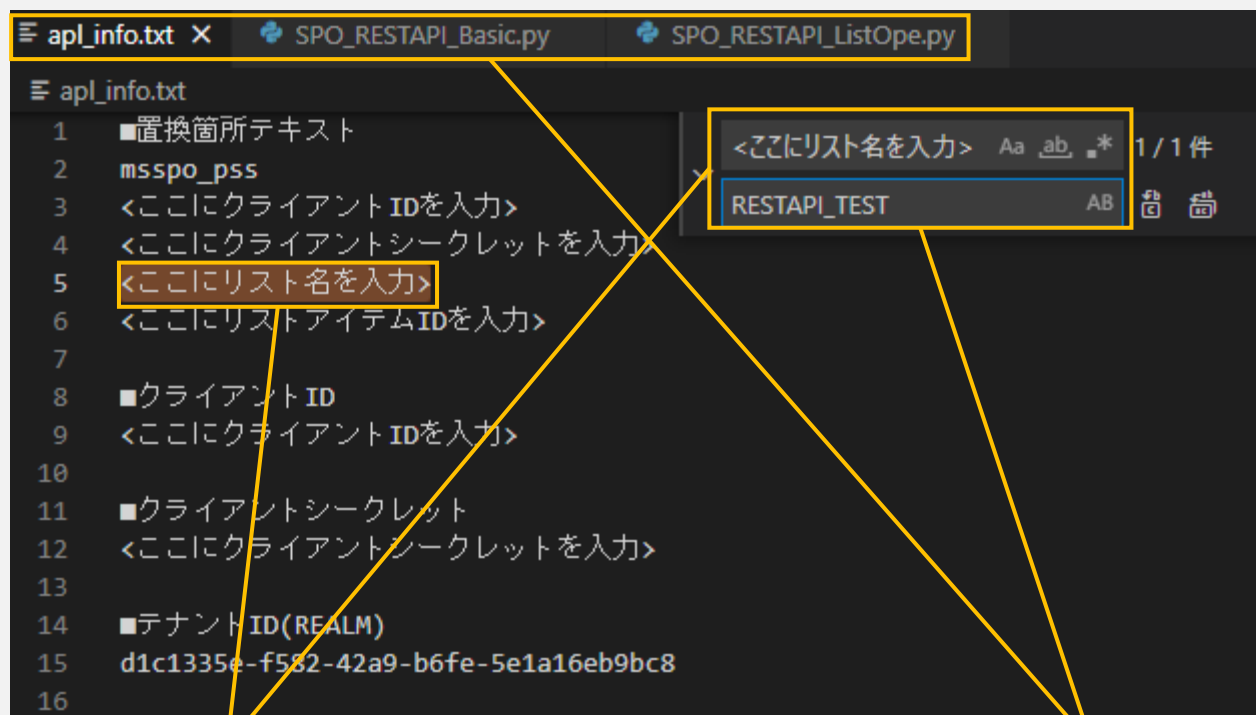
Hello REST API!

「新規」をクリックして任意のアイテムを追加する。
本資料の例ではタイトル列に「Hello REST API!」を追加。
※動作確認をスムーズに行うため、タイトル以外の列の追加は非推奨。



1-3.動作確認用リストの設定

③リスト名を置換する



1.apl_info.txtを開き、置換箇所テキストの
<ここにリスト名を入力>で検索をかける。

2.前項で作成したリスト名を入力して、
3つのファイル全てで全置換を行う。



2.REST API用IDの設定

2-1.アドインの登録

2-2.アドインへの権限付与

2-3.権限付与状況の確認



2-1. アドインの登録

① アドイン登録ページを開く

```
apl_info.txt x SPO_RESTAPI_Basic.py SPO_RESTAPI_ListOpe.py
apl_info.txt
14
15 ※サイトに対するユーザーの権限によっては、アドイン登録の操作を実施できません。
16 サイトやTeamに対する所有者や管理者の権限を獲得してから実施してください。
17
18
19 ■アドイン登録ページ
20 https://toyotajp.sharepoint.com/sites/msspo\_pss/\_layouts/15/appregnew.aspx
21
22 ■権限付与ページ
23 https://toyotajp.sharepoint.com/sites/msspo\_pss/\_layouts/15/appinv.aspx
24
25 ▼権限XML
26 <AppPermissionRequests AllowAppOnlyPolicy="true">
27 |   <AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web/list" Right="FullControl" />
28 </AppPermissionRequests>
29
30 ■権限確認ページ
31 https://toyotajp.sharepoint.com/sites/msspo\_pss/\_layouts/15/appprincipals.aspx
32
```

ヒント：リンク先のページでREST API用のIDを登録することができます。



2-1. アドインの登録

② 必要情報を入力する

1. クライアントIDとクライアントシークレットは「生成」をクリックする。
※手動入力も可能かもしれませんが、自動生成の方が手堅いと思われます。

アプリ情報

アプリ ID、シークレット、タイトル、ホスティング URL、リダイレクト URL などのアプリ情報です。

クライアント ID:

3403e028-2612-4fcc-be2a-8d883597c11a

生成

クライアント シークレット:

Vq...

生成

タイトル:

REST_APL

アプリドメイン:

localhost

例: "www.contoso.com"

リダイレクト先の URI:

https://localhost

ヒント：クライアントシークレットは
パスワードと同じです。
取り扱いには注意してください。

2. その他の項目は以下のように入力
「タイトル」・・・任意のアプリ名。本資料の例では「REST_APL」と入力。
「アプリドメイン」・・・localhost
「リダイレクト先のURI」・・・https://localhost

作成

キャンセル

3. 「作成」をクリックする。

ヒント：その他、アドイン登録の詳細は下記リンクやおまけ情報8-1項を参考にしてください。

[SharePoint アドインを登録する | Microsoft Docs](#)



2-1. アドインの登録

③IDとシークレットを取得して置換する



2-2. アドインへの権限付与

① 権限付与ページを開く

```
apl_info.txt x SPO_RESTAPI_Basic.py SPO_RESTAPI_ListOpe.py
apl_info.txt
14
15 ■テナント ID (REALM)
16
17 apl_info.txtの権限付与ページのURLをブラウザで開く。
18
19 ■アドイン登録ページ
20 https://toyotajp.sharepoint.com/sites/msspo_pss/_layouts/15/appregnew.aspx
21
22 ■権限付与ページ
23 https://toyotajp.sharepoint.com/sites/msspo_pss/_layouts/15/appinv.aspx
24
25 ▼権限XML
26 <AppPermissionRequests AllowAppOnlyPolicy="true">
27 |   <AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web/list" Right="FullControl" />
28 | </AppPermissionRequest>
29 </AppPermissionRequests>
30
31 ■権限確認ページ
32 https://toyotajp.sharepoint.com/sites/msspo_pss/_layouts/15/appinv.aspx
```

ヒント：リンク先のページでREST API用のIDに対して各種の権限を付与することができます。ここで権限を付与しなければクライアントIDを利用しても何もできません。



2-2. アドインへの権限付与

② 必要情報を入力する

アプリ ID およびタイトル

このアプリの ID とタイトルです。

1. アプリIDの欄へ前項のクライアントIDを
コピーして「参照」をクリックする。

アプリ ID:

3403e028-2612-4fcc-be2a-8

参照

タイトル:

REST_APL

アプリドメイン:

localhost

例: "www.contoso.com"

リダイレクト先の URL

https://localhost/

例: "https://www.contoso.com/default.aspx"

アプリの権限要求 XML

アプリに必要な権限です。

権限の要求 XML:

2. タイトル、アプリドメイン、リダイレクト先のURLが
自動入力されるため、対象のIDのものか確認する。

作成

キャンセル



2-2. アドインへの権限付与

② 必要情報を入力する つづき

The screenshot shows a code editor with three tabs: `apl_info.txt`, `SPO_RESTAPI_Basic.py`, and `SPO_RESTAPI_ListOpe.py`. The `apl_info.txt` tab is active and contains the following content:

```
22  ■権限付与 3.apl_info.txtを開き、権限XMLの箇所にあるXMLをコピーする。  
23  https://toyota.jp.sharepoint.com/sites/msspo_pss/_layouts/15/appinv.aspx  
24  
25  ▼権限XML  
26  <AppPermissionRequests AllowAppOnlyPolicy="true">  
27  |    <AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web/list" Right="FullControl" />  
28  </AppPermissionRequests>
```

A yellow box highlights the XML code in lines 26-28. A yellow callout box points to the URL in line 23, stating: "3.apl_info.txtを開き、権限XMLの箇所にあるXMLをコピーする。"

アプリの権限要求 XML

アプリに必要な権限です。

権限の要求 XML:

```
<AppPermissionRequests  
AllowAppOnlyPolicy="true"  
<AppPermissionRequest  
Scope="http://sharepoint/  
eb/list" Right="FullControl" />  
</AppPermissionRequests>
```

4. 権限の要求XMLの欄に貼り付ける。
本資料の例ではリストに対するフルコントロールの
権限を要求している。

5. 「作成」をクリックする。

作成

キャンセル

ヒント：本資料では手順を簡素化するためフルコントロールの権限を付与していますが、実際には必要最低限の権限付与を心掛けてください。

その他、権限要求XMLの詳細は下記リンクやおまけ情報8-1項を参考にしてください。

[SharePoint でのアドインのアクセス許可 | Microsoft Docs](#)



2-2. アドインへの権限付与

③ 対象のリストを選択する

1. セレクトボックスの中から操作対象のリストを選択する。
本資料の例では前項でRESTAPI_TESTのリスト名で作成したため、これを選択する。



2. 「信頼する」をクリックする。

ヒント：他の権限も付与したい場合は、2-2①項から作業を繰り返してください。
※セキュリティ対策として必要最低限の権限付与を心掛けてください。



2-3. 権限付与状況の確認

① 権限確認ページを開く

```
apl_info.txt x SPO_RESTAPI_Basic.py SPO_RESTAPI_ListOpe.py
apl_info.txt
14
15 ■テナント ID (REALM)
16 d1c1335e-f582-42a9-b6fe-5e1a16eb9bc8
17
18
19 ■アドイン登録ページ
20 https://toyotajp.sharepoint.com/sites/msspo\_pss/\_layouts/15/appregnew.aspx
21
22 ■権限付与ページ
23 https://toyotajp.sharepoint.com/sites/msspo\_pss/\_layouts/15/appinv.aspx
24
25 ▼権 apl_info.txtの権限確認ページのURLをブラウザで開く。
26 <AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web/list" Right="FullControl" />
27 </AppPermissionRequests>
28
29
30 ■権限確認ページ
31 https://toyotajp.sharepoint.com/sites/msspo\_pss/\_layouts/15/appprincipals.aspx
32
```

ヒント：リンク先のページでREST API用のIDに対する権限付与の状況を確認することができます。具体的にどのような権限が付与されているかまでは確認できません。



2-3. 権限付与状況の確認

② 権限付与状況を確認する



前項で登録したアドインアプリの名前やIDが表示されていることを確認する。

ヒント：各「アプリの表示名」の左側にある「×」ボタンをクリックすると、権限を削除することができます。セキュリティや情報漏洩対策として不要なアドインの権限は削除しましょう。



3.リストアイテムの取得

3-1.リストアイテムの取得

3-2.内容解説

3-3.クエリを用いたリストアイテムの取得



3-1. リストアイテムの取得

① プログラムを実行する

1. SPO_RESTAPI_Basic.pyを開き、▶マークをクリックしてプログラムを実行する。

ヒント：Pythonライブラリの「requests」とVisual Studio CodeのPython拡張機能を利用しています。実行前にインストールしておいてください。

ヒント：プログラムの実行やライブラリのインストールでプロキシサーバのエラーが出る場合は次ページやおまけ情報8-2項のページを参考にしてください。



3-1. リストアイテムの取得

① プログラムを実行する つづき

問題 出力 デバッグ コンソール ターミナル

```
return session.request(method=method, url=url, **kwargs)
File "C:\Users\1562033\AppData\Local\Programs\Python\Python39\lib\site-packages\requests\sessions.py", lin
e 542, in request
    resp = self.send(prepare_request(method=method, url=url, **kwargs))
File "C:\Users\1562033\AppData\Local\Programs\Python\Python39\lib\site-packages\requests\sessions.py", lin
e 655, in send
    r = adapter.send(request, **kwargs)
File "C:\Users\1562033\AppData\Local\Programs\Python\Python39\lib\site-packages\requests\adapters.py", lin
e 414, in send
    raise InvalidURL(e, request=request)
requests.exceptions.InvalidURL: Proxy URL had no scheme, should start with http:// or https://
PS C:\Users\1562033\Desktop\SPO_RESTAPI> |
```

ヒント：使用環境によりプロキシサーバのエラー解消方法は異なる場合があります。ここであげた対策は一例です。

2. 図のようなプロキシサーバ関係のエラーが出る場合は、
下記のコマンド2つをターミナル上にコピーして再度実行する。

```
$env:HTTP_PROXY="http://proxy1000.adm.toyota.co.jp:15520"
$env:HTTPS_PROXY="http://proxy1000.adm.toyota.co.jp:15520"
```

問題 出力 デバッグ コンソール

```
e 655, in send
    r = adapter.send(request)
File "C:\Users\1562033\AppData\Local\Programs\Python\Python39\lib\site-packages\requests\adapters.py", lin
e 414, in send
    raise InvalidURL(e, request=request)
requests.exceptions.InvalidURL: Proxy URL had no scheme, should start with http:// or https://
PS C:\Users\1562033\Desktop\SPO_RESTAPI> $env:HTTP_PROXY="http://proxy1000.adm.toyota.co.jp:15520"
PS C:\Users\1562033\Desktop\SPO_RESTAPI> $env:HTTPS_PROXY="http://proxy1000.adm.toyota.co.jp:15520"
PS C:\Users\1562033\Desktop\SPO_RESTAPI> |
```



3-1. リストアイテムの取得

② 実行結果を確認する

```
問題 出力 デバッグ コンソール ターミナル

PS C:\Users\1562033\Desktop\SPO_RESTAPI> 1. 実行が上手くいくと結果が図のようにjson形式で表示される。
PS C:\Users\1562033\Desktop\SPO_RESTAPI> & C:/Users/1562033/AppData/Local/Programs/Python/Python39/python.exe
e c:/Users/1562033/Desktop/SPO_RESTAPI/SPO_RESTAPI_Basic.py
-----ココから中身-----
{"d":{"results":[{"__metadata":{"id":"7c1fc8e8-cbde-4486-a4cb-953a4fab6cf2","uri":"https://toyotajp.sharepoi
nt.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)","etag":"\"1\"","
type":"SP.Data.RESTAPI_x005f_TESTListItem"},"FirstUniqueAncestorSecurableObject":{"__deferred":{"uri":"https
//toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)
/File"}}, "Folder":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Folder"}}, "Like
dByInformation":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Like
dByInformation"}}, "ParentList":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/ParentList"}}, "Properties":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Properties"}}, "Versions":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Versions"}}, "FileSystemObject
Type":0,"Id":1,"ServerRedirectedEmbedUri":null,"ServerRedirectedEmbedUrl":"","ID":1,"ContentTypeId":"0x01000FCC5A03F589FFF41A136BFD246F9FC6400A04C360CBE55F54BA18023D9119CD08A","Title":"Hello REST API!","Modified":"2021-11-11T07:15:23Z","Created":"2021-11-11T07:15:23Z","AuthorId":14,"EditorId":14,"OData_UIVersionString":1}
```

2. 本資料の例ではリストのタイトル列に「Hello REST API!」というテキストでアイテムを1つ追加したため、ターミナル上で検索をかける。対象のリストアイテムを取得できていることが確認できる。

```
6')/Items(1)/File"}}, "Folder":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Folder"}}, "Liked
ByInformation":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Like
dByInformation"}}, "ParentList":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/ParentList"}}, "Properties":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Properties"}}, "Versions":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Versions"}}, "FileSystemObject
Type":0,"Id":1,"ServerRedirectedEmbedUri":null,"ServerRedirectedEmbedUrl":"","ID":1,"ContentTypeId":"0x01000FCC5A03F589FFF41A136BFD246F9FC6400A04C360CBE55F54BA18023D9119CD08A","Title":"Hello REST API!","Modified":"2021-11-11T07:15:23Z","Created":"2021-11-11T07:15:23Z","AuthorId":14,"EditorId":14,"OData_UIVersionString":1}
```




3-2.内容解説

①アクセストークンの取得

SPO_RESTAPI_Basic.pyの処理の流れは以下の通り。

- ①アクセストークンの取得
- ②リスト操作(リストアイテムの一覧読み出し)

アクセストークンを取得することができる接続先URL。

```
13 #####
14 ### アクセストークン取得 ###
15 url = 'https://accounts.accesscontrol.windows.net/d1c1335e-f582-42a9-b6fe-5e1a16eb9bc8/tokens/OAuth/2'
16 data = {
17     'grant_type': 'client_credentials',
18     'client_id': client_id,
19     'client_secret': client_secret,
20     'resource': '00000003-0000-0ff1-ce00-000000000000',
21 }
```

アクセストークンの取得には設定したクライアントIDとクライアントシークレットの情報が必要。
データに載せて送信している。他の情報は定型。

指定URLへの接続と結果の受け取り。
結果はjson形式で扱えるように読み込み。

```
26 t = requests.post(url, data=data, headers=headers)
27 json_token = json.loads(t.text)
28 #####
```

ヒント：アクセストークンの有効期限は8時間です。
8時間は同じアクセストークンでSPOと接続することができます。



3-2.内容解説

②リストアイテム一覧の読み出し

取得したアクセストークンをヘッダーに挿入。
アクセストークンで認証を通す仕組み。

操作内容に沿ったURL形式を利用。
今回はリストアイテムの一覧取得を行いたいため、
対象のリスト名を指定し、図のような形式となる。

```
SPO_RESTAPI_Basic.py > ...
31 #####
32 ### リスト操作 ###
33 headers = {
34     'Authorization': 'Bearer ' + json_token['access_token'],
35     'Accept': 'application/json;odata=verbose',
36 }
37
38 url = "https://toyotajp.sharepoint.com/sites/" + target_SiteName + "/_api/web/lists/GetByTitle('" + target_ListName + ")/items"
39 # url = "https://toyotajp.sharepoint.com/sites/" + target_SiteName + "/_api/web/lists/GetByTitle('" + target_ListName + "')?$select=*"
40 l = requests.get(url, headers=headers)
41 print('-----ココから中身-----')
42 print(l.text)
43 #####
```

ヒント：操作内容によってURL形式やヘッダー情報、データ構成は様々です。
詳しくは下記リンクを参考にしてください。

[SharePoint REST サービス エンドポイント URI を決定する | Microsoft Docs](#)



3-3.クエリを用いたリストアイテムの取得

● 概要説明

前項の手法ではリストアイテムを取得する際に全てのアイテムを取得することになります。場合によっては全てのアイテムは必要なく、特定のアイテムだけ取得できればよいことがあります。そのような時に、条件を指定してアイテム取得をすることができるクエリというものがあります。利用方法としてはURLに条件文を追記していただけます。この条件指定の部分がクエリに当たります。

クエリ操作でできる条件指定の一例としては、取得する列の指定、昇順・降順の並び替え指定、取得件数の指定、日付データを元にした範囲指定、数値データを元にした範囲指定、任意の文字列を含んでいるかの検索、等があります。

クエリ操作の一例を示します。この例ではアイテムの更新が2021年1月1日以降のものを更新日時で降順に並び替え、上から5件分のデータを抽出する、となっています。

```
url = "https://toyotajp.sharepoint.com/sites/msspo_pss/_api/web/lists/GetByTitle('RESTAPI_TEST')?$filter=Modified ge '2021-01-01'&$orderby=Modified desc&$top=5"
```

ヒント：その他、クエリ操作に関する情報は下記リンクを参考にしてください。

[SharePoint REST 要求で OData クエリ操作を使用する | Microsoft Docs](#)

[SharePoint REST サンプル集 - SharePoint Developer \(orivers.jp\)](#)

[【#PowerAutomate Tip's】 フィルタークエリー \(OData クエリ\) メモ - Qiita](#)



4.リストアイテムの作成

4-1.内部リスト名の確認・設定

4-2.リストアイテムの作成

4-3.内容解説



4-1.内部リスト名の確認・設定

①プログラムを実行する

```
apl_info.txt  SPO_RESTAPI_Basic.py x  SPO_RESTAPI_ListOpe.py
SPO_RESTAPI_Basic.py > ...
30
31 #####
32 ### リスト操作 ###
33 headers = {
34     'Authorization': 'Bearer ' + json_token['access_token'],
35     'Accept': 'application/json;odata=verbose',
36 }
37
38 # url = "https://toyotajp.sharepoint.com/sites/" + target_SiteName + "/_api/web/lists/GetByTitle('" + target_ListName + ")/items"
39 url = "https://toyotajp.sharepoint.com/sites/" + target_SiteName + "/_api/web/lists/GetByTitle('" + target_ListName + "')?$select=ListItemEntit
40 l = requests.get(url, headers=headers)
41 print('-----ココから中身-----')
42 print(l.text)
43
```

1.SPO_RESTAPI_Basic.pyを開き、38行目をコメントアウトして39行目のコメントアウトを外す。

2.▶マークをクリックしてプログラムを実行する。

ヒント：内部リスト名とはリストのListItemEntityTypeFullNameプロパティのデータのことを指しています。リストアイテムの作成・更新・削除をする際に必要な情報になります。確認方法はいくつかありますが今回はその一例を示します。



4-1.内部リスト名の確認・設定

②実行結果を確認して設定する

```
問題 出力 デバッグ コンソール ターミナル
"1.0","Attachments":false,"GUID":"0620a9b2-c8ca-45a7-b19e-912821490ef2","ComplianceAssetId":null}}}}
PS C:\Users\1562033\Desktop\SPO_RESTAPI> & C:/Users/1562033/AppData/Local/Programs/Python/Python39/python.exe SPO_RESTAPI/SPO_RESTAPI_Basic.py
-----ココから中身-----
{"d":{"__metadata":{"id":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')","uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')","etag":"\\\"5\\\"","type":"SP.List"},"listItemEntityTypeFullName":"SP.Data.RESTAPI_x005f_TESTListItem"}}
PS C:\Users\1562033\Desktop\SPO_RESTAPI>
```

1.「ListItemEntityTypeFullName」で検索をかけて、その次にあるデータをコピーする。
本資料の例では「SP.Data.RESTAPI_x005f_TESTListItem」となる。

2.SPO_RESTAPI_ListOpe.pyを開き、
<ここに内部リスト名を入力>のテキストに
対して前項のコピーしたデータで置換をかける。

```
apl_info.txt SPO_RESTAPI_Basic.py SPO_RESTAPI_ListOpe.py X
SPO_RESTAPI_ListOpe.py > ...
4 #####
<ここに内部リスト名を入力>
SP.Data.RESTAPI_x005f_TESTListAB
c1335e-f582-42a9-b6fe-5e1a16eb9bc8'
ットを入力>

10 target_ListEntityName = '<ここに内部リスト名を入力>'
11 target_ListItemID = '<ここにリストアイテムIDを入力>'
12 #####
```



4-2. リストアイテムの作成

① プログラムを実行して結果を確認する

問題 出力 デバッグ コンソール ターミナル

-----リスト作成-----

```
{
  "d": {
    "__metadata": {
      "id": "f1b64652-762f-448b-b01b-27114b228014",
      "uri": "https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(2)",
      "etag": "\"1\"",
      "type": "SP.Data.RESTAPI_x005f_TESTListItem"
    },
    "FirstUniqueAncestorSecurableObject": {
      "__deferred": {
        "uri": "https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(2)/FirstUniqueAncestorSecurableObject"
      }
    },
    "RoleAssignments": {
      "__deferred": {
        "uri": "https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(2)/RoleAssignments"
      }
    },
    "AttachmentFiles": {
      "__deferred": {
        "uri": "https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(2)/AttachmentFiles"
      }
    },
    "Content": "Hello REST API!!",
    "FieldValuesAsHtml": {
      "Title by python": "Title by python"
    },
    "GetDlpPolicyTip": {
      "uri": "https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(2)/GetDlpPolicyTip"
    }
  }
}
```

1. SPO_RESTAPI_ListOpe.pyを実行する。
正常に実行されると図のような結果が表示される。

2. ブラウザで対象のリストを開き、「Title by python」のデータが追加されていること確認する。

お知らせ

リンク

試験計画書集計システム

サイトコンテンツ

ごみ箱

RESTAPI_TEST ☆

タイトル ▾

Hello REST API!!

Title by python

ヒント：本資料の手法では、リストアイテムは1アクセスにつき、1アイテムずつしか作成できません。1度に複数の操作をする方法の文書化は未着手です。別途大路まで問合せください。トライしてみたい方は下記リンク等を参考にしてください。

[REST API によりバッチ要求を発行する | Microsoft Docs](#)



4-3.内容解説

①リストアイテム作成

SPO_RESTAPI_ListOpe.pyのリストアイテム作成処理の流れは以下の通り。

①アクセストークンの取得(前項で説明したため詳細は割愛)

②リストアイテム作成

```
34  ### リストアイテム作成 ###
35  url = "https://toyotajp.sharepoint.com/sites/"+ target_SiteName +"/_api/web/lists/GetByTitle('"+ target_ListName +')/items"
36  data = '''{
37      "_metadata": {
38          "type": "type"+ target_ListEntityName +""
39      },
40      "Title": "Title by python"
41  }'''
42  headers = {
43      'Authorization': 'Bearer ' + json_token['access_token'],
44      'Accept': 'application/json;odata=verbose',
45      'Content-Type': 'application/json;odata=verbose',
46      'Content-Length': str(len(data)),
47  }
48
49  l = requests.post(url, data=data, headers=headers)
50  print('-----リストアイテム作成-----')
```

リストアイテム作成用のURL形式。

json形式で列名とデータの組合せで任意の情報を入力。
本資料の例では「Title by python」とする。

ヘッダーにデータ長を設定。
ただし、なくても動作する。

ヒント：その他、リストアイテム作成に関する詳しい情報は下記リンクやおまけ情報のページを参考にしてください。

[REST を使用してリストとリスト アイテムを操作する | Microsoft Docs](#)



5.リストアイテムの更新

5-1.アイテムIDの確認・設定

5-2.リストアイテムの更新

5-3.内容解説



5-1. アイテムIDの確認・設定

① アイテムIDを確認して設定する

問題 出力 デバッグコンソール ターミナル

```
ikedByInformation":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/LikedByInformation"}}, "ParentList":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/ParentList"}}, "Properties":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Properties"}}, "Versions":{"__deferred":{"uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(1)/Versions"}}, "FileSystemObjectType":0, "Id":1, "ServerRedirectedEmbedUri":null, "ServerRedirectedEmbedUrl":null, "ID":1, "ContentTypeId":"0x0100FCC5A03F589FFF41A136BFD246F9FC6400A04C360CBE55F54BA18023D9119CD08A", "Title":"Hello REST API!", "Modified":"2021-11-11T07:15:23Z", "Created":"2021-11-11T07:15:23Z", "AuthorId":14, "EditorId":14, "OData__UIVersionString":"1.0", "Attachments":false, "GUID":"0620a9b2-c8ca-45a7-b19e-912821490ef2", "ComplianceAssetId":null}, {"__metadata":{"id":"6acbaeda-4700-416c-a2c6-f1877397af31", "uri":"https://toyotajp.sharepoint.com/sites/msspo_pss/_api/Web/Lists(guid'7c8ab36c-faa5-4ef6-b83c-c8f40ff545a6')/Items(2)", "etag":"\\\"1\\\"", "type":"SP.Data.RESTAPI_x005f_TESTListItem"},"FirstUniqueAncestorSecurableObject":
```

1. リストアイテムのIDを確認する。アイテム取得した際に送られてくるデータの中にID等の名前が入っている。本資料の例では一番最初に追加したアイテムを対象とし、IDは1となる。

ヒント：リストアイテムを更新・削除する際は、アイテム固有のIDを指定する必要があります。このIDはSPOのシステムが自動で割り付けを行う数字となっていて、重複する値がありません。



5-1. アイテムIDの確認・設定

①アイテムIDを確認して設定する つづき

2.SPO_RESTAPI_ListOpe.pyを開き、35~51行目のリストアイテム作成の処理をコメントアウトして、57~75行目のリストアイテム更新の処理のコメントアウトを外す。

The screenshot shows a code editor with the following Python code:

```

1 import json
2 import requests
3
4 #####
5 ### 基本情報 ###
6 target_SiteName = 'msspo_pss'
7 client_id = '3403e028-2612-4fcc-be2a-8d883597c11a@01c1335e-f582-42a9-b6fe-5e1a16eb9bc8'
8 client_secret = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
9 target_ListName = 'RESTAPI_TEST'
10 target_ListEntityName = 'SP.Data.RESTAPI_x005f_TESTListItem'
11 target_ListItemID = '<ここにリストアイテムIDを入力>'
12 #####

```

A yellow box highlights the input field for 'target_ListItemID' in the code, which contains the placeholder text '<ここにリストアイテムIDを入力>'. Another yellow box highlights a search bar above the code, containing the text '<ここにリストアイテムIDを;'.

3.<ここにリストアイテムIDを入力>のテキストに対して前項のID値で置換をかける。

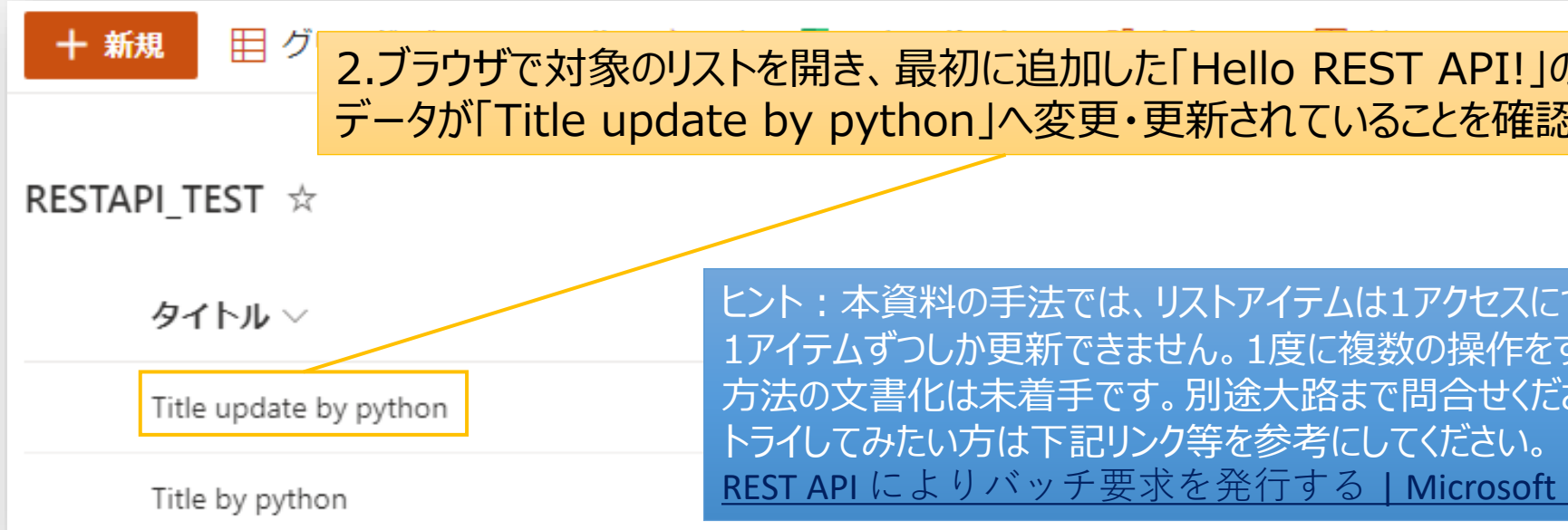


5-2. リストアイテムの更新

① プログラムを実行して結果を確認する

```
問題 出力 デバッグ コンソール ターミナル
PS C:\Users\1562033\Desktop\SPO_RESTAPI> & C:/Users/1562033/AppData/Local/Programs/Python/Python39/python.exe c:/Users/1562033/Desktop/SPO_RESTAPI/SPO_RESTAPI_ListOpe.py
-----リストアイテム更新-----
PS C:\Users\1562033\Desktop\SPO_RESTAPI> 
```

1. SPO_RESTAPI_ListOpe.pyを実行する。
正常に実行されると図のような結果となり、特に表示されるものはない。



2. ブラウザで対象のリストを開き、最初に追加した「Hello REST API!」のデータが「Title update by python」へ変更・更新されていることを確認する。

ヒント：本資料の手法では、リストアイテムは1アクセスにつき、1アイテムずつしか更新できません。1度に複数の操作をする方法の文書化は未着手です。別途大路まで問合せください。トライしてみたい方は下記リンク等を参考にしてください。

[REST API によりバッチ要求を発行する | Microsoft Docs](#)



5-3.内容解説

①リストアイテム更新

SPO_RESTAPI_ListOpe.pyのリストアイテム更新処理の流れは以下の通り。

- ①アクセストークンの取得(前項で説明したため詳細は割愛)
- ②リストアイテム更新

```
56  ### リストアイテム更新 ###
```

```
57  url = "https://toyotajp.sharepoint.com/sites/" + target_SiteName + "/_api/web/lists/GetByTitle('"+ target_ListName +"')/items("+ target_ListItemID +")"
```

リストアイテム更新用のURL形式。最後にIDの指定が必要。

```
58  type = "application/json"
59  data = {
60      "Title": "Title update by python"
61  }
62  headers = {
63      'Authorization': 'Bearer ' + json_token['access_token'],
64      'Accept': 'application/json;odata=verbose',
65      'Content-Type': 'application/json;odata=verbose',
66      'Content-Length': str(len(data)),
67      'If-Match': '*',
68      'X-HTTP-Method': 'MERGE'
69  }
70
71  l = requests.post(url, data=data, headers=headers)
72  print('-----リストアイテム更新-----')
73  print(l.text)
```

json形式で列名とデータの組合せで任意の更新情報を入力。
本資料の例では「Title update by python」とする。

ヘッダーのIf-Matchプロパティに「*」を入力。更新時にアイテムのバージョン情報を活用したい場合はアイテム取得した際に送られてくるデータの中に入っているetag値を利用する。「*」を入力した場合はバージョンに関わらず強制的に上書きする。

ヘッダーのX-HTTP-Methodプロパティに「MERGE」を入力。

ヒント：その他、リストアイテム更新に関する詳しい情報は下記リンクやおまけ情報のページを参考にしてください。

[REST を使用してリストとリスト アイテムを操作する | Microsoft Docs](#)



6.リストアイテムの削除

6-1.アイテムIDの設定

6-2.リストアイテムの削除

6-3.内容解説



6-1. アイテムIDの設定

①アイテムIDを設定する

1.SPO_RESTAPI_ListOpe.pyを開き、57~75行目のリストアイテム更新の処理をコメントアウトして、81~92行目のリストアイテム削除の処理のコメントアウトを外す。

The screenshot shows the VS Code editor interface with three files open: `apl_info.txt`, `SPO_RESTAPI_Basic.py`, and `SPO_RESTAPI_ListOpe.py`. The active file is `SPO_RESTAPI_ListOpe.py`, which contains the following Python code:

```

1 import json
2 import requests
3
4 #####
5 ### 基本情報 ###
6 target_SiteName = 'msspo_pss'
7 client_id = '3403e028-2612-4fcc-be2a-8d883597c11a@01c1335e-f582-42a9-b6fe-5e1a16eb9bc8'
8 client_secret = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
9 target_ListName = 'RESTAPI_TEST'
10 target_ListEntityName = 'SP.Data.RESTAPI_x005f_TESTListItem'
11 target_ListItemID = '<ここにリストアイテムIDを入力>'
12 #####

```

Annotations include:

- A yellow box highlights the input field for the list item ID, containing the placeholder text "<ここにリストアイテムIDを".
- A red box highlights the `client_secret` value, which has been replaced by a series of X's for security.
- A yellow arrow points from the input field annotation to the `target_ListItemID` assignment in the code.

2.<ここにリストアイテムIDを入力>のテキストに対して前項のID値を入力する。
本資料の例では既に入力されている1とする。



6-2.リストアイテムの削除

①プログラムを実行して結果を確認する

```
問題 出力 デバッグコンソール ターミナル
PS C:\Users\1562033\Desktop\SPO_RESTAPI> python SPO_RESTAPI_ListOpe.py
-----リストアイテム削除-----
PS C:\Users\1562033\Desktop\SPO_RESTAPI>
```

1.SPO_RESTAPI_ListOpe.pyを実行する。
正常に実行されると図のような結果となり、特に表示されるものはない。

+ 新規 グリッドビューで

RESTAPI_TEST ☆

タイトル ▼ + 列の追加 ▼

Title by python

2.ブラウザで対象のリストを開き、最初に追加し、前項で更新をかけた「Title update by python」のデータが削除されていることを確認する。

ヒント：本資料の手法では、リストアイテムは1アクセスにつき、1アイテムずつしか削除できません。1度に複数の操作をする方法の文書化は未着手です。別途大路まで問合せください。トライしてみたい方は下記リンク等を参考にしてください。
[REST API によりバッチ要求を発行する | Microsoft Docs](#)



6-3.内容解説

①リストアイテム削除

SPO_RESTAPI_ListOpe.pyのリストアイテム削除処理の流れは以下の通り。

- ①アクセストークンの取得(前項で説明したため詳細は割愛)
- ②リストアイテム削除

リストアイテム削除用のURL形式。
最後にIDの指定が必要。

```
#####
80  ### リストアイテム削除 ###
81  url = "https://toyotajp.sharepoint.com/sites/"+ target_SiteName +"/_api/web/lists/GetByTitle('"+ target_ListName +"')/items('"+ target_ListItemID +"")"
82  headers = {
83      'Authorization': 'Bearer ' + json_token['access_token'],
84      'Accept': 'application/json;odata=verbose',
85      'Content-Type': 'application/json',
86      'If-Match': '*',
87      'X-HTTP-Method': 'DELETE'
88  }
89
90  l = requests.post(url, headers=headers)
91  print('-----リストアイテム削除-----')
92  print(l.text)
93  #####
```

ヘッダーのIf-Matchプロパティに「*」を入力。「*」を入力した場合はバージョンに関わらず強制的に削除する。

ヘッダーのX-HTTP-Methodプロパティに「DELETE」を入力。

リストアイテムの削除にデータ部分の情報は不要。

ヒント：その他、リストアイテム削除に関する詳しい情報は下記リンクやおまけ情報のページを参考にしてください。
[REST を使用してリストとリスト アイテムを操作する | Microsoft Docs](#)



7.REST API用IDのパスワード管理

7-1.パスワードの有効期限

7-2.パスワード有効期限の確認

7-3.パスワード有効期限れ対応



7-1. パスワードの有効期限

● 留意点

本資料2項の手順により登録されたREST API用ID(アドイン)のパスワード(クライアントシークレット)は、**1年後に有効期限が切れます**。

有効期限が切れた場合、そのIDを使用してAPIにアクセスすること(アクセストークンを取得すること)はできなくなります。

次項の手順で確認・対応をしてください。

また、ID・パスワード更新への運用課題の対策は、各自計画的に用意しておくことを推奨します。後述の本資料で紹介している対応方法でも、1年ごとにパスワードの有効期限は切れます。



7-2. パスワード有効期限の確認

※コマンドをコピペの際は一部文字が消えたり、不要な改行が入るためご注意ください

①MSOnlineモジュールをインストールする

1.PowerShellを管理者権限で起動する。

2.「Install-Module MSOnline」を実行する。

※既にMSOnlineモジュールをインストール済みの場合はこの作業は不要です。
実施済みの方は③の手順へ進んでください。

```
管理者: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください http://aka.ms/powershell

PS C:\Windows\system32> Install-Module MSOnline

続行するには NuGet プロバイダーが必要です
PowerShellGet で NuGet ベースのリポジトリを操作するには、'2.8.5.201' 以降のバージョンの NuGet
プロバイダーが必要です。NuGet プロバイダーは 'C:\Program Files\PackageManagement\ProviderAssemblies' または
'C:\Users\%user%\AppData\Local\PackageManagement\ProviderAssemblies'
に配置する必要があります。'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force' を実行して NuGet
プロバイダーをインストールすることもできます。今すぐ PowerShellGet で NuGet
プロバイダーをインストールしてインポートしますか?
[Y] はい(Y) [N] いいえ(N) [S] 中断(S) [?] ヘルプ (既定値は "Y") : Y
```

3.下記のようなメッセージが出てきた場合は「Y」と入力してEnterキーを押す。
エラーが出る場合は②の手順へ進む。

ヒント : MSOnlineモジュールとはPowerShell用Microsoft Azure Active Directoryモジュールのことです。



7-2. パスワード有効期限の確認

① MSOnlineモジュールをインストールする つづき

```
管理者: Windows PowerShell
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.
新しいバージョンの PowerShell Core 6.0.0 がリリースされました。//aka.ms/pscore6

PS C:\Windows\system32> Install-Module MSOnline

続行するには NuGet プロバイダーが必要です。PowerShellGet で NuGet ベースのリポジトリを操作するには、'2.8.5.2'
プロバイダーが必要です。NuGet プロバイダーは 'C:\Program Files\PowerShell\PowerShellGet\NuGetProvider\NuGetProvider.dll'
'C:\Users\user\AppData\Local\PackageManagement\ProviderAssemblies'
に配置する必要があります。'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.2'
プロバイダーをインストールすることもできます。今すぐ PowerShellGet
プロバイダーをインストールしてインポートしますか?
[Y] はい(Y) [N] いいえ(N) [S] 中断(S) [?] ヘルプ (既定値は "Y"): Y

信頼されていないリポジトリ
信頼されていないリポジトリからモジュールをインストールしようとしています。このリポジトリを信頼する場合は、Set-PSRepository コマンドレットを実行して、リポジトリの InstallationPolicy の値を変更してください。'PSGallery'
からモジュールをインストールしますか?
[Y] はい(Y) [A] すべて続行(A) [N] いいえ(N) [L] すべて無視(L) [S] 中断(S) [?] ヘルプ (既定値は "N"): Y

PS C:\Windows\system32>
```

4. 下記のようなメッセージが出てきた場合は「Y」と入力してEnterキーを押す。

5. 2~4の手順で特にエラーが出ない場合はインストール完了。④の手順へ進む。
エラーが出る場合は②の手順へ進む。



7-2. パスワード有効期限の確認

※コマンドをコピペの際は一部文字が消えたり、不要な改行が入るためご注意ください

②モジュールインストールエラーへの対処 その1

モジュールインストール時にPSGalleryがないという旨のエラーが出た場合の対処方法を示します。

1.「Register-PSRepository -Default」を実行する。

```
管理者: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Windows\system32> Register-PSRepository -Default
PS C:\Windows\system32> Get-PSRepository

Name                InstallationPolicy SourceLocation
-----
PSGallery            Untrusted         https://www.powershellgallery.com/api/v2

PS C:\Windows\system32>
```

2.「Get-PSRepository」を実行して、上記のような表示が出てくることを確認。
表示が出てきた場合は①の手順からやり直す。
「パッケージソースが見つかりません」と出る場合は③の手順へ進む。



7-2. パスワード有効期限の確認

※コマンドをコピペの際は一部文字が消えたり、不要な改行が入るためご注意ください

③モジュールインストールエラーへの対処 その2

「Get-PSRepository」を実行後「パッケージソースが見つかりません」と出た場合の対処方法を示します。この方法はZscalerではない従来プロキシを使用した場合の手順です。

```
管理者: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Windows\system32> [system.net.webrequest]::defaultwebproxy = new-object system.net.webproxy('http://proxy1000.adm.toyota.co.jp:15520')
PS C:\Windows\system32> [system.net.webrequest]::defaultwebproxy.credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials
PS C:\Windows\system32> [system.net.webrequest]::defaultwebproxy.BypassProxyOnLocal = $true
PS C:\Windows\system32> Register-PSRepository -Default -Verbose
詳細: 対象 "プロバイダー 'PowerShellGet' のパッケージ ソース 'PSGallery' ()。" に対して操作 "パッケージ ソースの登録。" を実行しています。
詳細: リポジトリの詳細 Name = 'PSGallery' Location = 'https://www.powershellgallery.com/api/v2' IsTrusted = False、
```

1. 以下3つのコマンドを順番に実行する(1行1コマンド)。

```
[system.net.webrequest]::defaultwebproxy = new-object system.net.webproxy('http://proxy1000.adm.toyota.co.jp:15520')
[system.net.webrequest]::defaultwebproxy.credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials
[system.net.webrequest]::defaultwebproxy.BypassProxyOnLocal = $true
```



7-2. パスワード有効期限の確認

※コマンドをコピペの際は一部文字が消えたり、不要な改行が入るためご注意ください

③モジュールインストールエラーへの対処 その2 つづき

管理者: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォーム

2. 「Register-PSRepository -Default -Verbose」を実行する。

```
PS C:\Windows\system32> [system.net.webrequest]::defaultwebproxy = new-object system.net.webproxy('http://proxy1000.adm.toyota.co.jp:15520')
PS C:\Windows\system32> [system.net.webrequest]::defaultwebproxy.credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials
PS C:\Windows\system32> [system.net.webrequest]::defaultwebproxy.BypassProxyOnLocal = $true
PS C:\Windows\system32> Register-PSRepository -Default -Verbose
```

詳細: 対象 "プロバイダー 'PowerShellGet' のパッケージ ソース 'PSGallery' ()。" に対して操作 "パッケージ ソースの登録。" を実行しています。

詳細: リポジトリの詳細、Name = 'PSGallery'、Location = 'https://www.powershellgallery.com/api/v2'、IsTrusted = 'False'、IsRegistered = 'True'。

```
PS C:\Windows\system32> Get-PSRepository
```

Name	InstallationPolicy	SourceLocation
PSGallery	Untrusted	https://www.powershellgallery.com/api/v2

3. 「Get-PSRepository」を実行して、上記のような表示が出てくることを確認。
表示が出てきた場合は①の手順からやり直す。



7-2. パスワード有効期限の確認

※コマンドをコピペの際は一部文字が消えたり、不要な改行が入るためご注意ください

④ サービスにログインする

The screenshot shows a Windows PowerShell window with the command `Connect-MsolService` entered. Below the command line, a Microsoft login window is displayed. The login window has the title 'アカウントにサインイン' (Sign in to account), the Microsoft logo, and the word 'サインイン' (Sign in). A text input field contains the email address '1562033@tmc.twfr.toyota.co.jp'. Below the input field, there is a link that says 'アカウントにアクセスできない場合' (If you can't access your account). At the bottom right of the login window is a blue button labeled '次へ' (Next).

1. 「Connect-MsolService」を実行する。

2. サインイン画面が出てくるため自分のID「従業員コード@tmc.twfr.toyota.co.jp」を入力する。

3. 「次へ」をクリック。



7-2.パスワード有効期限の確認

④サービスにログインする つづき

管理者: Windows PowerShell

サインイン

4. 再びサインイン画面が出てくるため
自分のPCログイン時のパスワードを入力する。

TMC

組織アカウントを使用してサインインしてください

1562033@tmc.twfr.toyota.co.jp

●●●●●●

5. 「サインイン」をクリック。

サインイン



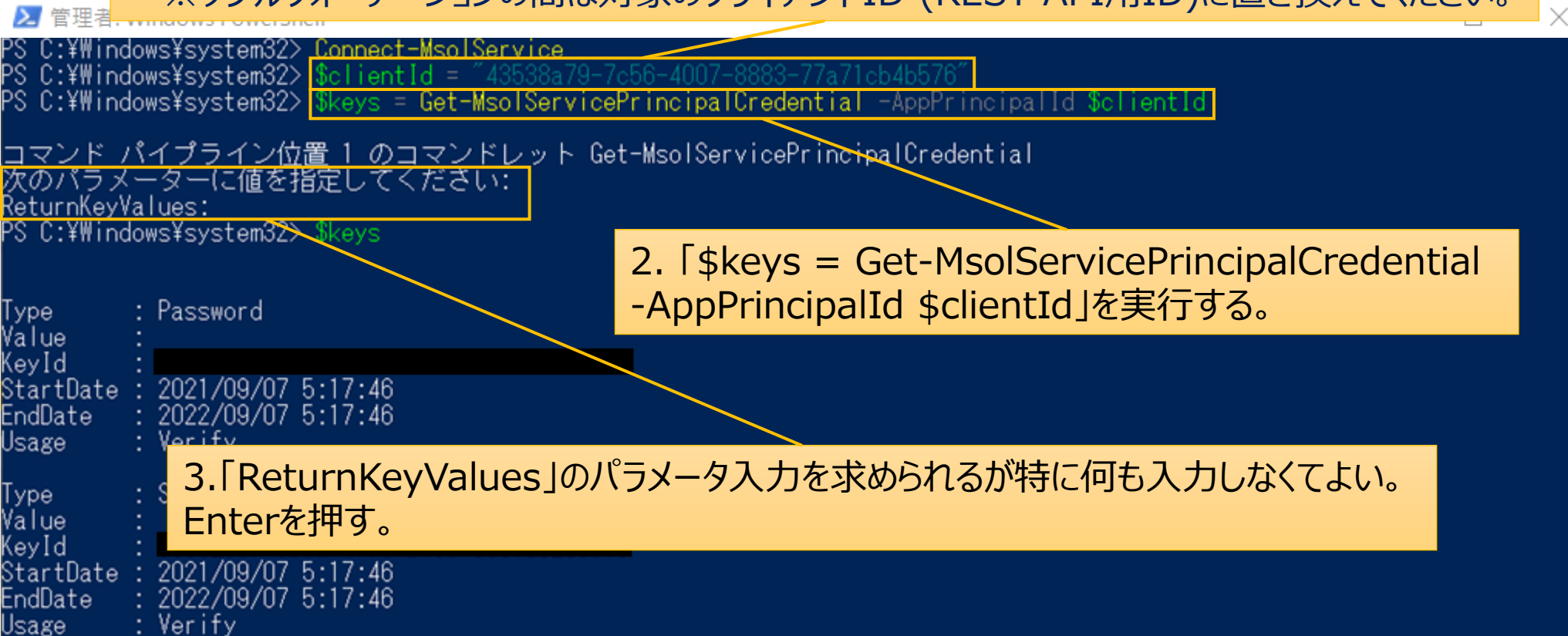
7-2. パスワード有効期限の確認

※コマンドをコピペの際は一部文字が消えたり、不要な改行が入るためご注意ください

⑤ 有効期限を確認する

1. 「\$clientId = "対象のクライアントID(REST API用ID)"」を実行する。

※ダブルクォーテーションの間は対象のクライアントID (REST API用ID)に置き換えてください。



```
PS C:\Windows\system32> Connect-MsolService
PS C:\Windows\system32> $clientId = "43538a79-7c56-4007-8883-77a71cb4b576"
PS C:\Windows\system32> $keys = Get-MsolServicePrincipalCredential -AppPrincipalId $clientId

コマンド パイプライン位置 1 のコマンドレット Get-MsolServicePrincipalCredential
次のパラメーターに値を指定してください:
ReturnKeyValues:
PS C:\Windows\system32> $keys

Type      : Password
Value     : 
KeyId     : 
StartDate : 2021/09/07 5:17:46
EndDate   : 2022/09/07 5:17:46
Usage     : Verify

Type      : $
Value     : 
KeyId     : 
StartDate : 2021/09/07 5:17:46
EndDate   : 2022/09/07 5:17:46
Usage     : Verify
```

2. 「\$keys = Get-MsolServicePrincipalCredential -AppPrincipalId \$clientId」を実行する。

3. 「ReturnKeyValues」のパラメータ入力を求められるが特に何も入力しなくてよい。
Enterを押す。



7-2. パスワード有効期限の確認

⑤ 有効期限を確認する つづき

管理者: Windows PowerShell

```
ReturnKeyValues:  
PS C:\Windows\system32> $keys
```

Type : Password
Value :
KeyId :
StartDate : 2021/09/07 5:17:46
EndDate : 2022/09/07 5:17:46
Usage : Verify

Type : Symmetric
Value :
KeyId :
StartDate : 2021/09/07 5:17:46
EndDate : 2022/09/07 5:17:46
Usage : Verify

Type : Symmetric
Value :
KeyId :
StartDate : 2021/09/07 5:17:46
EndDate : 2022/09/07 5:17:46
Usage : Sign

4.「\$keys」を実行する。
3種類ほど表示が出てくるが、いずれかの「EndDate」を確認する。
「EndDate」の日付がパスワードの有効期限となる。



7-3.パスワード有効期限切れ対応

● 対応方針

パスワード有効期限切れの対応としては
本資料2項の手順より別IDの生成から始め、再度IDとパスワード(クライアントシークレット)の両方を付与し直す方法が簡単かと思われます。

ヒント：IDはそのままにパスワードのみを更新する方が自然ですが、2022/10/10現在パスワードのみを更新するために必要な各種の操作権限は付与されていないようです。また、仕様としてはパスワードの有効期限延長も可能ですが、こちらも操作権限がないようです。

参考までにパスワード更新方法を紹介しているサイトを掲載します。

ただし上記の通り、現在操作権限はありません。

クイック スタート: Microsoft ID プラットフォームでアプリを登録する - Microsoft Entra | Microsoft Learn(←Azure portalでの更新方法)

SharePoint アドインで期限が切れたクライアント シークレットを置換する | Microsoft Learn(←PowerShellでの更新方法)



8.おまけ情報

8-1.権限付与XMLのポイント

8-2.プロキシサーバエラーへの対処

8-3.リストアイテムデータの送信フォーマット

8-4.リストの列名に日本語を使用した際の留意点

8-5.リストアイテムの5000/20000件問題

8-6.リストアイテムデータ更新手法の一例

8-7.フォルダ・ファイル操作のプログラム例



8-1. 権限付与XMLのポイント

● ポイント

アドインへの権限付与で使用する下記のようなXMLですが、Teams付随や単体払い出しのSPOサイトでは、1行目の「**AllowAppOnlyPolicy="true"**」の部分がないと上手く権限付与できません。ご注意ください。

```
<AppPermissionRequests AllowAppOnlyPolicy="true">  
  <AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web/list" Right="FullControl" />  
</AppPermissionRequests>
```

ScopeのURIとRightの権限には以下のようなものがあります。それぞれできることに違いが発生します。詳しくは下記リンクを参考にしてください。
[SharePoint でのアドインのアクセス許可 | Microsoft Docs](#)

スコープ URI	使用可能な権限
http://sharepoint/content/sitecollection	Read、Write、Manage、FullControl
http://sharepoint/content/sitecollection/web	Read、Write、Manage、FullControl
http://sharepoint/content/sitecollection/web/list	Read、Write、Manage、FullControl
http://sharepoint/content/tenant	Read、Write、Manage、FullControl



8-2. プロキシサーバエラーへの対処

● 対処その1

Pythonのライブラリを追加でインストールする時や、本資料のプログラムを実行する時等、インターネット接続を使用する操作でプロキシサーバのエラーが出る場合があります。ここでは対処方法を4例ほど紹介します。

対処例1) 環境変数へ以下のようにプロキシサーバの情報を2つ追加する。

http_proxy⇒http://proxy1000.adm.toyota.co.jp:15520

https_proxy⇒http://proxy1000.adm.toyota.co.jp:15520

※ 以下のように設定しておくと、この値を自動的に読み取り、別途プロキシサーバの設定をしなくても済む場合があります。ただし、効いてくれないこともあります。

変数	値
http_proxy	http://proxy1000.adm.toyota.co.jp:15520
https_proxy	http://proxy1000.adm.toyota.co.jp:15520



8-2. プロキシサーバエラーへの対処

● 対処その2

コマンドプロンプト上でインターネット接続を必要とする操作を実行した時などにプロキシサーバのエラーが出る場合があります。コマンドプロンプトを扱う際にお試ください。

対処例2)コマンドプロンプト上で以下のコマンドを2つ実行する。

```
set HTTP_PROXY=http://proxy1000.adm.toyota.co.jp:15520
```

```
set HTTPS_PROXY=http://proxy1000.adm.toyota.co.jp:15520
```

※このコマンドでプロキシサーバの設定ができます。

ただし、設定が有効であるのはコマンドプロンプトを閉じるまでの間です。

```
コマンド プロンプト
C:\Users¥1562033¥Desktop¥SPO_RESTAPI>set HTTP_PROXY=http://proxy1000.adm.toyota.co.jp:15520
C:\Users¥1562033¥Desktop¥SPO_RESTAPI>set HTTPS_PROXY=http://proxy1000.adm.toyota.co.jp:15520
C:\Users¥1562033¥Desktop¥SPO_RESTAPI>
```



8-2. プロキシサーバエラーへの対処

● 対処その3

PowerShell上でインターネット接続を必要とする操作を実行した時などにプロキシサーバのエラーが出る場合があります。PowerShellを扱う際にお試ください。

対処例3) PowerShell上で以下のコマンドを2つ実行する。

```
$env:HTTP_PROXY="http://proxy1000.adm.toyota.co.jp:15520"
```

```
$env:HTTPS_PROXY="http://proxy1000.adm.toyota.co.jp:15520"
```

※このコマンドでプロキシサーバの設定ができます。

ただし、設定が有効であるのはPowerShellを閉じるまでの間です。

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Users\¥1562033> $env:HTTP_PROXY="http://proxy1000.adm.toyota.co.jp:15520"
PS C:\Users\¥1562033> $env:HTTPS_PROXY="http://proxy1000.adm.toyota.co.jp:15520"
PS C:\Users\¥1562033>
```



8-2. プロキシサーバエラーへの対処

● 対処その4

本資料の例ではPythonのrequestsライブラリを利用しています。
アクセス時の引数に設定情報を載せることができます。

対処例4) アクセス処理実行前に以下の文を追記し、引数に追加する。

```
proxies = {  
    'http': 'http://proxy1000.adm.toyota.co.jp:15520',  
    'https': 'http://proxy1000.adm.toyota.co.jp:15520'  
}
```

```
proxies = {  
    'http': 'http://proxy1000.adm.toyota.co.jp:15520',  
    'https': 'http://proxy1000.adm.toyota.co.jp:15520'  
}
```

```
t = requests.post(url, data=data, headers=headers, proxies=proxies)
```



8-3. リストアイテムデータの送信フォーマット

● ポイント

リストアイテム作成等の際に送るアイテムデータ(ボディ部分)ですが、フォーマットにいくつかポイントがあります。下記のPythonサンプルコードを元に説明します。使用するプログラミング言語で情報は異なる可能性があります。

```
strjson_PSSdata = json.dumps(json_PSSdata[i], indent=2, ensure_ascii=False)
strjson_PSSdata = strjson_PSSdata.lstrip('{')
strjson_PSSdata = strjson_PSSdata.rstrip('}')
strjson_PSSdata = strjson_PSSdata.replace('""', 'null')

data = ('''{
  "_metadata": {
    "type": "SP.Data.Test_x005f_plan_x005f_allListItem"
  },
  ''' + strjson_PSSdata + '''
}''').encode('utf-8')

l = requests.post(url, data=data, headers=hea
```

データは文字列に変換して格納してください。
ヘッダー部分は直接json形式で格納できますが、
データ部分はjson形式の文字列にする必要があります。

Python上でのjson形式データを、サンプルコードのように結合する際は、最初と最後の波かっこ{ }は不要です。結合前に取り除いてください。



8-3.リストアイテムデータの送信フォーマット

●ポイント つづき

文字コードはUTF-8にしてください。日本語を含むファイル等から情報を読み取った際にUTF-8になっていない場合があります。

リストの列データの中に日付形式や数値形式のデータ等がある場合、引用符で囲まれた空データ("")は空と認識されずにデータ型エラーとなります。空の場合はnullに置き換えてください。

```
strjson_PSSdata = json.dumps(json_PSSdata[i], indent=2, ensure_ascii=False)
strjson_PSSdata = strjson_PSSdata.lstrip('{')
strjson_PSSdata = strjson_PSSdata.rstrip('}')
strjson_PSSdata = strjson_PSSdata.replace('""', 'null')

data = ('''{
    "_metadata": {
        "type": "SP.Data.Test_x005f_plan_x005f_allListItem"
    },
    ''' + strjson_PSSdata + '''
}''').encode('utf-8')

l = requests.post(url, data=data, headers=headers)
```

json形式データの最後の要素の後ろにカンマが付いているとエラーになります。最後のカンマは取り除いてください。



8-4.リストの列名に日本語を使用した際の留意点

● 留意点

リストの列を追加していく際に、列名を日本語にして追加することもあると思います。しかし、REST APIを利用する上では、日本語(全角文字)で列名を設定すると少々厄介なことになります。理由は内部的に日本語(全角文字)の列名は文字コードの数値等を組み合わせたもので置き換えられてしまうからです。

例えば「書類ID」という列名が存在するリストのアイテムを作成する場合です。左図のように書類IDと入力しても認識されず、右図のような置き換え文字で入力する必要があります。

```
data = '''{
  "_metadata": {
    "type": "SP.Data.RESTAPI_x005f_TESTListItem"
  },
  "Title": "Title update by python",
  "書類ID": "123456"
}'''
```

```
data = '''{
  "_metadata": {
    "type": "SP.Data.RESTAPI_x005f_TESTListItem"
  },
  "Title": "Title update by python",
  "OData__x66f8__x985e_ID": "123456"
}'''
```



8-4.リストの列名に日本語を使用した際の留意点

● 留意点 つづき

この問題を回避する方法を以下に3つ示します。

①列名が半角英数字で問題なければ半角英数字で設定する。

最も簡単な対策です。しかし、半角英数字では他のM365アプリケーション上で使いづらい場面も出てきます。

②列を作成する最初だけ半角英数字で設定し、後から日本語で上書きする。

リストの仕様上、内部的な名前は最初に設定した名前になります。そのため、こう対策することで他のM365アプリケーションからは日本語で扱え、内部的な名前が必要な時は最初に指定した半角英数字を利用できます。しかし、二重に設定する手間と使い分ける手間が発生します。

③潔く内部的な名前が必要な時は置き換え文字を利用する。

ヘッダーが日本語で大量の列があるExcelからリスト作成する時や、逆にデータが少ない時、内部的な名前を利用しない時は潔くそのまま日本語を利用するのも良いです。置き換え文字を調べる手間や、大量データの場合は置き換え処理を作る必要があるかもしれません。

8-4.リストの列名に日本語を使用した際の留意点

● 留意点 つづき

最初から日本語(全角文字)で設定した場合、内部的な名前が必要な時はその名前がどうなっているか知っている必要があります。ここでは内部名を調べる方法の一例を示します。置き換え文字は規則性のある置き換えとなっているためプログラムによる変換も可能ですが、本資料では取り扱いません。

例)PowerAutomateで対象のリストに対してアイテムを作成するフローを作り、実行結果の確認画面で出ている「未加工入力の表示」で確認することができます。

確認時に相対が取りやすいようにデータの中身は順番に数字を入れる等くふうした方が良いでしょう。

※あくまで一例です。

項目の作成 1秒

入力

未加工入力の表示

サイトのアドレス

計画書集計システム - https://toyotajp.sharepoint.com/sites/msspo_pss

```
{
  "host": {
    "connectionReferenceName": "shared_sharepointonline",
    "operationId": "PostItem"
  },
  "parameters": {
    "dataset": "https://toyotajp.sharepoint.com/sites/msspo_pss",
    "table": "ef0acb1c-104d-47db-af0e-e2c94fc79529",
    "item/Title": "1",
    "item/OData_x66f8_x985e_ID": 2,
    "item/OData_x66f8_x985e_x30d0_x30fc_x30": "3",
    "item/OData_x6d0fb_x4ed8_x6709_x7121": true,
    "item/OData_x30b3_x30e1_x30f3_x30c8_x67": true,
    "item/OData_x95a2_x9023_x66f8_x985e_x67": "4",
    "item/OData_x30d5_x30a9_x30fc_x30e0_x30": "5",
    "item/OData_x30d5_x30a9_x30fc_x30e0_x54": "6",
    "item/OData_x30d5_x30a9_x30fc_x30e0_x300": 7,
    "item/OData_x56de_x4ed8_x30eb_x30fc_x30": "8",
    "item/OData_x56de_x4ed8_x30eb_x30fc_x300": "9",
    "item/OData_x56de_x4ed8_x30eb_x30fc_x301": 10,
    "item/OData_x66f8_x985e_x72b6_x614b": "11",
    "item/OData_x73fe_x5728_x306e_x30b9_x30": "12",
    "item/OData_x57fa_x6e96_x65e5": "2021-11-02",
    "item/OData_x627f_x8a8d_x5b8c_x4e86_x65": "2021-11-02",
    "item/OData_x66f8_x985e_x30ea_x30fc_x30": "15",
    "item/OData_x4f5c_x6210_x8005_x7d44_x7e": "16",
    "item/OData_x4f5c_x6210_x8005_x7d44_x7e0": "17",
    "item/OData_x4f5c_x6210_x8005_x30ed_x30": "18",
    "item/OData_x4f5c_x6210_x8005_x30ed_x300": "19",
    "item/OData_x4f5c_x6210_x8005_x30b3_x30": "20",
    "item/OData_x4f5c_x6210_x8005_x540d": "21",
    "item/OData_x4f5c_x6210_x65e5_x6642": "2021-11-02"
  }
}
```

リスト名	APITest2
書類管理番号	1
書類ID	2
書類バージョン	3



8-5.リストアイテムの5000/20000件問題

● 概要説明

リストに保存できるアイテムの最大数は3000万件です。しかし、REST APIには一度に5000件を超えるアイテムを扱う操作はできないという制約があります。例えば5000件を超えるアイテムが入っているリストのタイトル列に対して、昇順並び替えの条件でアイテム取得をした場合、この操作は拒否されます。この問題の回避方法の一例として、filterクエリを利用して扱うデータ件数が5000件を超えないように範囲を限定するという方法があります。

REST APIとは関係ないですが、ブラウザ等の通常使用では2万件を超えるアイテムが入っていると並び替え等の操作ができなくなる場合があります。これは列インデックスが生成されていない列に対して操作を行った際に起こります。そのため、並び替えをしたい列に手動で列インデックスを追加することで解決できます。

ヒント：手動で列インデックスを追加する方法やその他の5000/20000件問題に関する詳しい情報は下記リンクを参考にしてください。分かりやすくまとめられています。

[SharePoint Online モダン リストの 5,000 件問題対応と 20,000 件問題の回避方法 – idea.toString\(\);](#)



8-6.リストアイテムデータ更新手法の一例

● 概要説明

【背景】ある社内LAN上のファイルサーバに、定期的にcsv形式でデータファイルを吐き出すシステムがあったとします。csvには元システム側でのデータ一意性を示すID値が入っています。1行目には列名も入っています。データは日々新しいものが追加されたり、既存のデータが更新されたりしています。

【やりたいこと】このcsvデータを自動的にSPOのリストにアップロードして、その後、Power BI等を活用してデータの分析・視える化を実現したい。

【実現の流れ】Pythonから社内LANのcsvファイルにアクセスして、SPOのリストにアップロードできるようにデータを抽出・加工する。SPO側でのデータ一意性の確保は元システム側のID値を利用する。このID値で検索をかけて、既にデータが存在していれば取得データの中に入っているSPO側のID値を利用して更新処理を実施、データが存在していなければ新規作成処理を実施する。そして、このプログラムを常時起動のPCでタスクスケジューラ登録する。

ヒント：リストの設定で元システム側のID値の列に、必須の入力と固有の値を適用し、列インデックスを設定した方がより手堅いシステムになります。また、同封の「[omake.py](#)」に上記の処理を実現したサンプルコードを入れています。
※上記の本題とは無関係な処理も入っていますがそれでも良ければ参考までにご覧ください。



8-7.フォルダ・ファイル操作のプログラム例

● 概要説明

これまでの説明では、主にリストに関する情報を元に説明してきましたが、REST APIを利用してSPOへフォルダの作成、ファイルのアップロード、ダウンロードを実施することも可能です。

同封の「[SPO RESTAPI FileOpe.py](#)」を参考にしてください。

こちらのプログラムは他の方から共有していただいたものです。
詳細な説明は割愛させていただきます。手法はプログラムを読んでご理解ください。

また、フォルダ・ファイル操作をする際は2-2②項に当たる作業にて
「Scope=http://sharepoint/content/sitecollection/web/list」ではなく、
「Scope=http://sharepoint/content/sitecollection/web」にして権限付与をしておく
必要があります。
権限レベルに関してはFullControlではなく、ReadやWrite等、最低限にすることを推奨します。