R2DMS Uploader
使用手引書

0.1 版

2025.09.29

データ管理システム開発ユニット）林　寛生

改訂履歴

版数	改訂日	改訂者	対象	改訂内容
0	2025.09.10	林寬生	全体	新規作成
0.1	2025.09.29	林寬生	全体	Poetry 版から変更

目次

1　はじめに

このソフトウェア（R2DMS Uploader）は、GakuNin RDM に新規のプロジェクトを作成し、そのプロジェクトのストレージにローカルの研究データをアップロードするための Python スクリプトです。

以下の２つのツールの機能を順次呼び出して実行します。

1） GakuNin RDM 上で、既存プロジェクトの下に新規の子プロジェクトを作成［create_grdm］

2） GakuNin RDM のプロジェクトにマウントされたストレージへローカルのデータをアップロード［wbclient］

*上記のツールは、R2DMS_Uploader のパッケージに含まれています。

2　前提条件

Python（Version 3.10 以上）および Python のサードパーティ製モジュールである以下の３つが利用できる環境であることが前提です。

- pandas
- requests
- pydantic-settings

3　インストール

① Python のアップデート

本ソフトウェアでは、**バージョン 3.10 以上**の Python が必要です。以下のように作業環境のバージョンを確認してください。

例）

```
$ python --version
```

もし、Python のバージョンが低い場合はアップデートしてください。（※環境によっては管理者権限が必要かもしれません）

② 依存ライブラリのインストール

本ソフトウェアでは、Python の標準ライブラリの他に、サードパーティ製のモジ

ュールである pandas、requests および pydantic-settings が必要です。

Linux では以下のようにすると、これらのモジュールが作業環境で利用できる場合は、その名称とバージョンが表示されます。

例）

```
$ pip list | grep -e pandas -e requests -e pydantic-settings
```

Windows の場合は、Powershell を立ち上げて、以下のように確認してください。

例）

```
$ pip list | findstr "pandas requests pydantic-settings"
```

もし、これらのモジュールが無い場合は、例えば以下のようにインストールしてください。（※環境によっては管理者権限が必要かもしれません）

例）

```
$ pip install pandas
$ pip install requests
$ pip install pydantic-settings
```

③ 本ソフトウェアの取得

wegt コマンドや Web ブラウザ等で以下の URL よりダウンロードしてください。

https://dmsutil.riken.jp/tool/r2dms_uploader-0.0.3.zip

④ 本ソフトウェアのインストール

ダウンロードした本ソフトウェアのパッケージ（ZIP ファイル）を、適当なディレクトリに解凍・展開してください。

Linux では以下のようにします。

例）

```
$ unzip -d {展開先のディレクトリパス} r2dms_uploader-0.0.3.zip
```

Windows の場合は、エクスプローラーで ZIP ファイルを右クリックして、表示されたメニュー内から「すべて展開」を選択します。その後、表示されるダイアログで展開先のフォルダを指定して「展開」ボタンをクリックしてください。

4　設定ファイルの作成（.env_control）

本ソフトウェアの展開先のディレクトリに行き、適当なエディタを使って、以下のような .env_control ファイルを作成してください。

【.env_control ファイルの例】

```
GRDM_TOKEN="ABCDEFGHIJKLM0123456789NOPQRSTUVWXYZ"
GRDM_OSFAPI="https://api.rdm.nii.ac.jp/v2"
GRDM_WBAPI="https://files.rdm.nii.ac.jp"
```

- ファイル名（.env_control）は変更しないでください。
- 以下の３つの環境変数を設定してください。

環境変数名	説明
GRDM_TOKEN	ユーザが GakuNin RDM で設定したパーソナルアクセストークン
GRDM_OSFAPI	GakuNin RDM の OSF API のベース URL
GRDM_WBAPI	GakuNin RDM の WaterButler API のベース URL

✓ パーソナルアクセストークンの取得方法については、付録 A. を参照してください。

✓ 上記の .env_control の例に示す OSF API および Waterbutler API の URL は、NII の GakuNin RDM のものです。NII 以外の GakuNin RDM を利用する場合は、それぞれのシステム管理者にお問い合わせください。

5　入力パラメタの準備

適当なエディタを使って、以下に示すような入力ファイル（CSV フォーマット）を作成してください。

【入力ファイルの例】

```
"parent_id","prj_title","data_path","replace_flag"
"abcde","A sample child project 001","./sample_data","0"
,"A sample new project XYZ","./sample_data2","0"
"xyz01",,"./sample_data3","1"
```

- UTF-8 でエンコードされた CSV ファイルを使ってください。
- 上の例で示した通りのヘッダー行（先頭行）を最初の行に置き、その後にデータ行を記述してください。
- 各データ行は、以下に定義する４個の要素の値（カンマ区切り）から構成されます。

要素名	説明

parent_id	既存プロジェクト ID（このプロジェクトの子プロジェクトとして新規のプロジェクトが作成される）
prj_title	新規プロジェクトのタイトル
data_path	アップロードするデータ（ディレクトリ or ファイル）のパス
replace_flag	上書きフラグ（0：アップロード先に同じファイルパスが存在するケースでは、そのファイルを上書きしない / 1：同ケースにて上書きする）

- ✓ プロジェクト ID の確認方法は、付録 B. を参照してください
- ✓ 既存プロジェクト ID（parent_id）をブランクにすると、新規の独立したプロジェクトが作成され、そこにデータがアップロードされます。
- ✓ 既に作成済みのプロジェクトにデータをアップロードする場合は、そのプロジェクトの ID を既存プロジェクト ID（parent_id）に設定し、新規プロジェクトのタイトル（prj_title）をブランクにしてください。
- ✓ プロジェクトを作成するケースでは、上書きフラグはどちらの値でも構いません。

- 要素の順番は変えないでください。
- データ行の各要素の値は二重引用符（″）で括ってください。
- シャープ記号(#)で始まる行は無視されます。
- 要素の値に二重引用符（″）を含めたい場合は、もう一つの二重引用符（″）を直前に置いてエスケープしてください。
- プロジェクトのタイトルに「<」や「>」を使えますが、それらは GakuNin RDM 上では「¥<」や「¥>」のように表示されます。
- サンプルの入力ファイル（sample_input.csv）を参照してください。

6 R2DMS Uploader の使用方法

本ソフトウェアの展開先のディレクトリで、以下のコマンドを実行します。

例）

```
$ python -m r2dms_uploader {入力 CSV ファイルパス} -o/--output {出力 JSON ファイルパス}
```

注意：

- .env_control ファイルも同じディレクトリに置いてください。
- 入力ファイルパスは必須です。
- 処理のサマリ(JSON フォーマット)が返されます。
- 出力ファイルパスを指定しない場合は標準出力に処理のサマリが表示されます。
- サンプルの出力ファイル（sample_output.json）を参照してください。

処理のサマリとして出力される内容は、以下の通りです。

```
{
  "summary": {
    "message": 処理全体に関するメッセージ
    "start_date": 処理全体の開始時刻(YYYY-mm-ddTHH:MM:SS.ffffff+HH:MM　; 時刻に関してはすべて同じ形式)
    "stop_date": 処理全体の終了時刻
    "work_host": 実行ホスト名
    "work_dir": 実行ディレクトリパス
    "exec_user_id": 実行ユーザ ID
    "input_filepath": 入力ファイルパス
    "output_filepath": 出力ファイルパス
    "num_csvline_valid": 入力ファイルの有効 CSV 行数
    "num_proc_total": 全処理数
    "num_proc_succeeded": 成功処理数
    "num_proc_failed": 失敗処理数
    "num_proc_skipped": スキップ処理数
    "processes": [ ※以下の要素を CSV ファイルの有効行ごとの配列データとして格納
      {
        "proc_id": 処理番号(1 スタートの連番)
        "status": ステータス(OK/NG/SKIPPED)
        "message": 処理に関するメッセージ
        "start_date": 処理開始時刻
        "stop_date": 処理終了時刻
        "csv_input": [ 処理対象の入力 CSV 行の内容
          "`parent_id`の値",
          "`prj_title`の値",
          "`data_path`の値",
          "`replace_flag`の値"
        ],
        "create_grdmprj": { ※「create_grdmprj」機能の実行結果として以下の要素を格納
          "status": ステータス(OK/NG/SKIPPED)
          "start_date": 開始時刻
          "stop_date": 終了時刻
          "message": この関数からのメッセージ
          "internal_rc": 内部リターンコード(0:正常終了/0 以外：異常終了)
          "parent_guid": 作成したプロジェクトの親プロジェクトの ID
          "root_guid": 作成したプロジェクトの一番上の親プロジェクトの ID
          "guid_created": 作成したプロジェクトの ID
          "url_created": 作成したプロジェクトの URL
          "title_created": 作成したプロジェクトのタイトル
          "creator_guid": プロジェクト作成者の ID
          "creator_name": プロジェクト作成者の名前
        },
        "wbclient": { ※「wbclient」機能の実行結果として以下の要素を格納
          "status": ステータス(OK/NG/SKIPPED)
          "start_date": 開始時刻
          "stop_date": 終了時刻
          "message": この関数からのメッセージ
          "local_path": アップロードするデータ(ディレクトリ/ファイル)のパス
          "target_project_id": アップロード先の GakuNinRDM のプロジェクト ID
          "storage_provider": アップロード先のストレージ名
          "remote_dir": アップロード先のディレクトリ
          "replace_flag": 上書きフラグの値
          "upload_result": { ※アップロードされたディレクトリ/ファイルとアップロード結果のリストを以下に格納
            “リモート/ディレクトリ/パス/": "created.” または “already exists.",
            ...
            "リモート/ファイル/パス": "uploaded.” または “replaced.” または ”found (not replaced)",
            ...
          }
        }
      },
      ... CSV ファイルの次の有効行に関する処理結果が格納される
      ...
    ]
  }
}
```

付録

A. パーソナルアクセストークン

環境変数の GRDM_TOKEN で指定するパーソナルアクセストークンは、GakuNin RDM にて以下の手順で設定します。

1) GakuNin RDM にログイン後、画面右上のユーザ名をクリックし、「設定」を選択します。

2) 左メニューから「パーソナルアクセストークン」を選択し、「新規トークン」ボタンをクリックします。

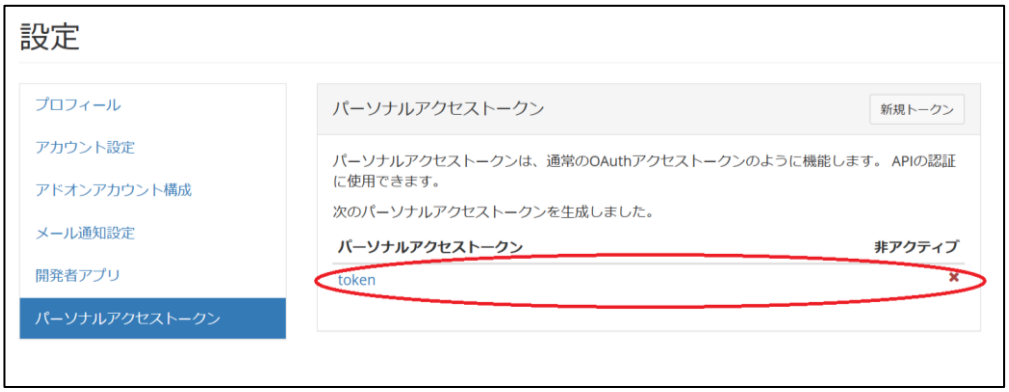
3) 以下のようにトークン名（任意）とスコープを設定し、「作成」ボタンをクリックします。

4) トークン ID が作成されますが、この時点では保存されていないため、「保存」ボ

タンをクリックします。

5) 「クリップボードにコピー」ボタンをクリックしてトークン ID をコピーし、ファイル等に保存します。（※この文字列を GRDM_ACCESS_TOKEN に指定してください）

6) パーソナルアクセストークンの画面に戻り、作成したトークン名が表示されていることを確認します。



B. <u>プロジェクト ID</u>

入力 CSV ファイルの parent_id で指定するプロジェクト ID は、GakuNin RDM のプロジェクトに付与されている ID（英数５文字）のことです。プロジェクトの URL の末尾の５文字（/を除く）が該当しますので、ブラウザでプロジェクトのページにアクセスし、アドレスバーから確認してください。