

SAFE Python Package Documentation

version

Philip Schwedler

March 26, 2023

Contents

Welcome to SAFE Research Data Center Python Package's documentation!	1
safe_datacenter	1
safe_datacenter package	1
Subpackages	1
safe_datacenter.dropbox_safe package	1
Submodules	1
safe_datacenter.dropbox_safe.connector module	1
Module contents	2
safe_datacenter.onedrive_safe package	2
Submodules	2
safe_datacenter.onedrive_safe.connector module	2
Module contents	5
Module contents	5
Indices and tables	5
Index	7
Python Module Index	9

Welcome to SAFE Research Data Center Python Package's documentation!

safe_datacenter

safe_datacenter package

Subpackages

safe_datacenter.dropbox_safe package

Submodules

safe_datacenter.dropbox_safe.connector module

Contains the Dropbox object class for interaction with your personal Dropbox account using the Python package `safe_datacenter.dropbox_safe`.

`class safe_datacenter.dropbox_safe.connector.Connector` (access_token: str)

Bases: `dropbox.dropbox_client.Dropbox`

Creates an instance to interact with Dropbox API.

classmethod `connect` (access_token: str) → [safe_datacenter.dropbox_safe.connector.Connector](#)

Create an instance of the Dropbox class.

Returns: OneDrive object instance

Return type: [Connector](#)

`download` (local_file_path: str, dbx_file_path: str)

Download a file from a user's Dropbox.

Parameters:

- **local_file_path** (str) – Path on local machine to save file.
- **dbx_file_path** (str) – Dropbox path of the file to download.

`list_directory` (dbx_file_path: str | None = None) → None

This method lists the files and folders contained in the given directory (folder_id). If the method is called without arguments, it returns the content of the root folder.

Parameters: **dbx_file_path** (str | None, optional) – A unique identifier for the file., defaults to None

`open_csv` (dbx_file_path: str) → `pandas.core.frame.DataFrame`

Read a comma-separated values (csv) file from Dropbox into DataFrame.

Parameters: **dbx_file_path** (str) – Dropbox path of the file to download.

Returns: A comma-separated values (csv) file is returned as two-dimensional data structure with labeled axes.

Return type: DataFrame

`upload` (local_file_path: str, dbx_file_path: str) → None

This method uploads a local file based on the full path to dropbox given a full path. Do not use this to upload a file larger than 150 MB.

Parameters:

- **local_file_path** (str) – Path on local machine of the file to be uploaded.
- **dbx_file_path** (str) – Dropbox path to which the file should be uploaded.

upload_csv (dataframe: pandas.core.frame.DataFrame, dbx_file_path: str) → None

This method uploads a Pandas DataFrame to a csv to Dropbox given the full path. Do not use this to upload a file larger than 150 MB.

Parameters:

- **dataframe** (*DataFrame*) – Pandas Dataframe to be uploaded
- **dbx_file_path** (*str*) – Dropbox path to which the file should be uploaded.

Module contents

safe_datacenter.onedrive_safe package

Submodules

safe_datacenter.onedrive_safe.connector module

Contains the OneDrive object class for interaction via the Graph API using the Python package `safe_datacenter.onedrive_safe`. The Graph API provides hooks to interact with a number of drives associated with an account, including SharePoint. However, the SAFE Data Center Python package currently only connects to a user's personal OneDrive, including a user's personal OneDrive at work or school.

```
class safe_datacenter.onedrive_safe.connector.Connector (client_id: str, client_secret: str, tenant: str = 'common', redirect_url: str = 'http://localhost:8080', refresh_token: str | None = None)
```

Bases: **object**

Creates an instance to interact with Microsoft's OneDrive platform through their Graph API.

Parameters:

- **client_id** (*str*) – Azure app client id
- **client_secret** (*str*) – Azure app client secret
- **tenant** (*str, optional*) – Azure app org tenant id number, use default if multi-tenant, defaults to "common"
- **redirect_url** (*str, optional*) – Authentication redirection url, defaults to "<http://localhost:8080>"
- **refresh_token** (*str | None, optional*) – optional token from previous session, defaults to None

classmethod connect () → [safe_datacenter.onedrive_safe.connector.Connector](#)

Create an instance of the OneDrive class for SAFE default configuration.

Returns: OneDrive object instance

Return type: [Connector](#)

detail_item (item_path: str) → dict[str, Any]

Retrieves the metadata for an item from a web path.

Parameters: **item_path** (*str*) – web path of the drive folder or file

Returns: metadata of the requested item

Return type: dict[str, Any]

detail_item_by_id (item_id: str) → dict[str, Any]

Retrieves the metadata for an item.

Parameters: **item_id** (*str*) – item id of the folder or file

Returns: metadata of the requested item

Return type: dict[str, Any]

download_file (item_path: str, max_connections: int = 8, dest_dir: str | Path | None = None, verbose: bool = False) → str

Downloads a file by file system path to the current working directory asynchronously. More details find at **safe_datacenter.onedrive.connector.Connector.download_file_by_id()**.

Parameters:

- **item_path** (str) – item path from root of the file to be downloaded
- **max_connections** (int, optional) – max concurrent open http requests, refer Docs regarding throttling limits, defaults to 8
- **dest_dir** (str | Path | None, optional) – destination directory for the downloaded file, defaults to None
- **verbose** (bool, optional) – prints status message during the download process, defaults to False

Returns: returns the path of the file including extension

Return type: str

download_file_by_dict (item_dict: dict, max_connections: int = 8, dest_dir: str | Path | None = None, verbose: bool = False) → str

Downloads a file by a dict retrieved from Get Item Graph API request to the current working directory asynchronously. More details find at

safe_datacenter.onedrive.connector.Connector.download_file_by_id().

Parameters:

- **item_dict** (dict) – item dict retrieved from Get Item Graph API request of the file to be downloaded
- **max_connections** (int, optional) – max concurrent open http requests, refer Docs regarding throttling limits, defaults to 8
- **dest_dir** (str | Path | None, optional) – destination directory for the downloaded file, defaults to None
- **verbose** (bool, optional) – prints status message during the download process, defaults to False

Returns: returns the path of the file including extension

Return type: str

download_file_by_id (item_id: str, max_connections: int = 8, dest_dir: str | Path | None = None, verbose: bool = False) → str

Downloads a file by driveItem unique identifier to the current working directory asynchronously. Note folders cannot be downloaded, you need to use a loop instead. For file transfer methods keep max_connections ≤ 16. Depending on your internet connection and the sizes of the files that you and transferring may see increased transfer speeds with more connections than the default. Graph-OneDrive currently will create a new connection for every 1MiB if not limited by the max_connections. If you have a really fast connection having many connections may slow the overall performance. A word of caution on too many connections - the Graph API may throttle requests. This can be a hard throttle where all existing connections are cut and a cool-down period is enforced. Throttling can be applied at the user level or the whole organization. Details on throttling are available, however the exact limits are not provided. We recommended to not exceed 16 connections for performance and to avoid throttling.

Parameters:

- **item_id** (str) – item id of the file to be downloaded
- **max_connections** (int, optional) – max concurrent open http requests, refer Docs regarding throttling limits, defaults to 8
- **dest_dir** (str | Path | None, optional) – destination directory for the downloaded file, defaults to None
- **verbose** (bool, optional) – prints status message during the download process, defaults to False

Returns: returns the path of the file including extension

Return type: str

classmethod **from_dict** (config: dict[str, Any]) → [safe_datacenter.onedrive_safe.connector.Connector](#)
Create an instance of the OneDrive class from a dictionary.

Parameters: **config** (dict[str, Any]) – dictionary containing at minimum tenant_id, client_id, client_secret_value

Returns: OneDrive object instance

Return type: [Connector](#)

classmethod **from_file** (file_path: str | Path = 'C:\\Users\\PhilipSchwedler\\anaconda3\\lib\\site-packages\\safe_datacenter\\config.json', config_key: str = 'onedrive', save_refresh_token: bool = False) → [Connector](#)
Create an instance of the OneDrive class from a config file.

Parameters:

- **file_path** (str | Path, optional) – path to configuration file, defaults to “config.json”
- **config_key** (str, optional) – key of the item storing the configuration, defaults to “onedrive”
- **save_refresh_token** (bool, optional) – save the refresh token back to the config file during instance initiation, defaults to False

Returns: OneDrive object instance

Return type: [Connector](#)

list_directory (folder_path: str | None = None, verbose: bool = False) → list[dict[str, object]]
List the files and folders within the input folder/root of the connected OneDrive by using file system path.

Parameters:

- **folder_path** (str | None, optional) – The folder path from root to look into, None being the root directory, defaults to None
- **verbose** (bool, optional) – Print the items along with their ids, defaults to False

Returns: Details of all the items within the requested directory

Return type: list[dict[str, object]]

list_directory_by_id (folder_id: str | None = None, verbose: bool = False) → list[dict[str, object]]
List the files and folders within the input folder/root of the connected OneDrive by using driveItem unique identifier.

Parameters:

- **folder_id** (str | None, optional) – The folder id of the folder to look into, None being the root directory, defaults to None
- **verbose** (bool, optional) – Print the items along with their ids, defaults to False

Returns: Details of all the items within the requested directory

Return type: list[dict[str, object]]

to_file (file_path: str | Path = 'C:\\Users\\PhilipSchwedler\\anaconda3\\lib\\site-packages\\safe_datacenter\\config.json', config_key: str = 'onedrive') → None
Save the configuration to a config file.

Parameters:

- **file_path** (str | Path, optional) – path to configuration file, defaults to “config.json”
- **config_key** (str, optional) – key of the item storing the configuration, defaults to “onedrive”

upload_file (file_path: str | Path, new_file_name: str | None = None, parent_folder_id: str | None = None, if_exists: str = 'rename', verbose: bool = False) → str
Uploads a file to a particular folder with a provided file name.

Parameters:

- **file_path** (*str* | *Path*) – path of the local source file to upload
- **new_file_name** (*str* | *None*, *optional*) – new name of the file as it should appear on OneDrive, with extension, defaults to None
- **parent_folder_id** (*str* | *None*, *optional*) – item id of the folder to put the file within, if None then root, defaults to None
- **if_exists** (*str*, *optional*) – action to take if the new folder already exists [fail, replace, rename], defaults to “rename”
- **verbose** (*bool*, *optional*) – prints status message during the download process, defaults to False

Returns: item id of the newly uploaded file

Return type: str

exception safe_datacenter.onedrive_safe.connector.**GraphAPIError**

Bases: **Exception**

Exception raised when Graph API returns an error status.

Module contents

Module contents

Indices and tables

- **genindex**
- **modindex**
- **search**

Index

C

`connect()`
(`safe_datacenter.dropbox_safe.connector.Connector`
class method)

(`safe_datacenter.onedrive_safe.connector.Connector`
class method)

`Connector` (class in
`safe_datacenter.dropbox_safe.connector`)

(class in
`safe_datacenter.onedrive_safe.connector`)

D

`detail_item()`
(`safe_datacenter.onedrive_safe.connector.Connector`
method)

`detail_item_by_id()`
(`safe_datacenter.onedrive_safe.connector.Connector`
method)

`download()`
(`safe_datacenter.dropbox_safe.connector.Connector`
method)

`download_file()`
(`safe_datacenter.onedrive_safe.connector.Connector`
method)

`download_file_by_dict()`
(`safe_datacenter.onedrive_safe.connector.Connector`
method)

`download_file_by_id()`
(`safe_datacenter.onedrive_safe.connector.Connector`
method)

F

`from_dict()`
(`safe_datacenter.onedrive_safe.connector.Connector`
class method)

`from_file()`
(`safe_datacenter.onedrive_safe.connector.Connector`
class method)

G

`GraphAPIError`

L

`list_directory()`
(`safe_datacenter.dropbox_safe.connector.Connector`
method)

(`safe_datacenter.onedrive_safe.connector.Connector`
method)

`list_directory_by_id()`
(`safe_datacenter.onedrive_safe.connector.Connector`
method)

M

module

`safe_datacenter`

`safe_datacenter.dropbox_safe`

`safe_datacenter.dropbox_safe.connector`

`safe_datacenter.onedrive_safe`

`safe_datacenter.onedrive_safe.connector`

O

`open_csv()`
(`safe_datacenter.dropbox_safe.connector.Connector`
method)

S

safe_datacenter

module

safe_datacenter.dropbox_safe

module

safe_datacenter.dropbox_safe.connector

module

safe_datacenter.onedrive_safe

module

safe_datacenter.onedrive_safe.connector

module

T

`to_file()`
(`safe_datacenter.onedrive_safe.connector.Connector`
method)

U

`upload()`
(`safe_datacenter.dropbox_safe.connector.Connector`
method)

`upload_csv()`
(`safe_datacenter.dropbox_safe.connector.Connector`
method)

`upload_file()`
(`safe_datacenter.onedrive_safe.connector.Connector`
method)

Python Module Index

s

[safe_datacenter](#)

[safe_datacenter.dropbox_safe](#)

[safe_datacenter.dropbox_safe.connector](#)

[safe_datacenter.onedrive_safe](#)

[safe_datacenter.onedrive_safe.connector](#)