

TM3 CFS File Workflow using Data Library for Python

Last Updated: Aug 2025

Example Code Disclaimer: ALL EXAMPLE CODE IS PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS FOR ILLUSTRATIVE PURPOSES ONLY. LSEG MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AS TO THE OPERATION OF THE EXAMPLE CODE, OR THE INFORMATION, CONTENT, OR MATERIALS USED IN CONNECTION WITH THE EXAMPLE CODE. YOU EXPRESSLY AGREE THAT YOUR USE OF THE EXAMPLE CODE IS AT YOUR SOLE RISK.

Import Libraries

```
In [1]: import lseg.data as ld
import lseg.data.delivery.cfs as cfs
from datetime import timedelta
```

Step 1: Open the Session

```
In [ ]: ld.open_session('platform.ldp')
#ld.open_session(config_name='./lseg-data.devrel.config.json')
```

```
Out[ ]: <lseg.data.session.Definition object at 0x7c2c5921e0d0 {name='ldp'}>
```

The `bucket-name` for the TM3 (Municipal Market Monitor) feed data is **bulk-Custom**.

Please contact your LSEG representative to help you with the `packageId`.

```
In [ ]: package_id = 'Your Package ID'
bucket_name = 'bulk-Custom'
```

Step 2: Query for specific Date using `modified_since` and `page_size=100` to limit data return<-- Recommended One

Now we come to getting the `FileSets` information. The application needs to send a request to RDP to list **the first** 100 FileSets under the input `bucket` and `package_id` that has been modified since `modified_since` as follows.

The `modified_since` parameter can help an application to limit the returned File-Set only for the File-Set that has been modified after a specified time.

```
In [ ]: response = cfs.file_sets.Definition(
    bucket = bucket_name,
```

```

package_id = package_id,
page_size = 100,
modified_since="2025-07-25T12:00:00Z"
).get_data()
response.data.df

```

Get the file URL detail from the File Set

The `file set` information is in the `id` column. Example with the first one.

```

In [ ]: first_file_set_id = response.data.df.at[0, 'id']

response_fileset = cfs.files.Definition(first_file_set_id).get_data()
response_fileset.data.df

```

Step 2.5: Or you can listing the 100 records of the FileSets using the Bucket Name and Package ID and page_size=100

```

In [ ]: response = cfs.file_sets.Definition(
    bucket = bucket_name,
    package_id = package_id,
    page_size = 100).get_data()
response.data.df

```

The FileSets can be more than 100!!

Please note that the FileSets can be more than 100 records. The API returns data maximum 100 records per one query.

You can do the paging to turn on the next page as follows:

```

In [ ]: page1_response = cfs.file_sets.Definition(
    bucket = bucket_name,
    package_id = package_id,
    page_size = 100).get_data()
display(page1_response.data.df)

token = page1_response.data.raw['skip_token']

page2_response = cfs.file_sets.Definition(
    bucket='bulk-Custom',
    page_size=100,
    skip_token=token,
).get_data()
display(page2_response.data.df)

```

Step 3: Get the Cloud file URL using FileSet

The next step is getting the file URL on the Cloud service.

```
In [ ]: file_url = response_fileset.data.files[0]
        print(file_url)
```

Step 4: Downloading the actual file from the Cloud

Our beloved Zscaler blocks my download :p.

```
In [9]: file_downloader = cfs.file_downloader.Definition(file_url).retrieve()
        file_downloader.download()
        file_downloader.extract()
```

```
Out[9]: 'TM3_SIFMAIndex2025-07-29'
```

Step 5: Close the Session

This close session mechanism allows an application to invalidate its tokens if the end-user logs out, changes identity, or exits the respective application.

```
In [10]: ld.close_session()
```

```
In [ ]:
```