

Green Revenue CFS File Workflow (CSV)

Importing libraries

```
In [1]: import os
import sys
from dotenv import dotenv_values
config = dotenv_values(".env")
```

Set RDP credentials and Initial Parameters

```
In [2]: username = config['RDP_USERNAME'] #or replacing with your RDP Username/Machine-ID
password = config['RDP_PASSWORD'] #or replacing with your RDP Password
clientId = config['RDP_APP_KEY'] #or replacing with your RDP App key

RDP_HOST= 'https://api.refinitiv.com'
access_token = None
refresh_token = None
expires_in = 0
```

RDP APIs Application Workflow

Step 1: Authentication with RDP APIs

Refinitiv Data Platform entitlement check is based on OAuth 2.0 specification. The first step of an application workflow is to get a token from RDP Auth Service, which will allow access to the protected resource, i.e. data REST API.

The API requires the following access credential information:

- Username: The username.
- Password: Password associated with the username.
- Client ID: This is also known as `AppKey`, and it is generated using an App key Generator. This unique identifier is defined for the user or application and is deemed confidential (not shared between users). The `client_id` parameter can be passed in the request body or as an "Authorization" request header that is encoded as base64.

The HTTP request for the RDP APIs Authentication service is as follows:

```
POST /auth/oauth2/v1/token HTTP/1.1
Accept: /*
Content-Type: application/x-www-form-urlencoded
Host: api.refinitiv.com:443
Content-Length: XXX

username=RDP_USERNAME
```

```
&password=RDP_PASSWORD  
&client_id=RDP_APP_KEY  
&grant_type=password  
&takeExclusiveSignOnControl=true  
&scope=trapi
```

Once the authentication success, the function gets the RDP Auth service response message and keeps the following RDP token information in the variables.

- **access_token:** The token used to invoke REST data API calls as described above. The application must keep this credential for further RDP APIs requests.
- **refresh_token:** Refresh token to be used for obtaining an updated access token before expiration. The application must keep this credential for access token renewal.
- **expires_in:** Access token validity time in seconds.

Next, after the application received the Access Token (and authorization token) from RDP Auth Service, all subsequent REST API calls will use this token to get the data. Please find more detail regarding RDP APIs workflow in the following resources:

- [RDP APIs: Introduction to the Request-Response API](#) page.
- [RDP APIs: Authorization - All about tokens](#) page.

```
In [3]: #step 1 - get RDP Access Token from RDP

import http.client
import requests
import json

# Send HTTP Request
auth_url = f'{RDP_HOST}/auth/oauth2/v1/token'
payload = f'grant_type=password&username={username}&client_id={clientId}&password={password}'
try:
    response = requests.post(auth_url,
                            headers = {'Content-Type': 'application/x-www-form-urlencoded'},
                            data = payload,
                            auth = (clientId, ''))

except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if response.status_code == 200: # HTTP Status 'OK'
    print('Authentication success')
    access_token = response.json()['access_token']
    refresh_token = response.json()['refresh_token']
    expires_in = int(response.json()['expires_in'])

if response.status_code != 200:
    print(f'RDP authentication failure: {response.status_code} {response.reason}')
    print(f'Text: {response.text}'')
```

Authentication success

Requesting Data from RDP APIs

That brings us to requesting the RDP APIs data. All subsequent REST API calls use the Access Token via the *Authorization* HTTP request message header as shown below to get the data.

- Header:
 - Authorization = Bearer <RDP Access Token>

Please notice *the space* between the Bearer and RDP Access Token values.

The application then creates a request message in a JSON message format or URL query parameter based on the interested service and sends it as an HTTP request message to the Service Endpoint. Developers can get RDP APIs the Service Endpoint, HTTP operations, and parameters from Refinitiv Data Platform's [API Playground page](#) - which is an interactive documentation site developers can access once they have a valid Refinitiv Data Platform account.

Requesting Bulk Green Revenues Data

Step 2: Listing the packageId of the Bulk Green Revenues Data

To request the Green Revenues Bulk data, the first step is to send an HTTP GET request to the RDP /file-store/v1/file-sets?bucket=bulk-greenrevenue endpoint to list all FileSets.

The HTTP Request structure is as follows:

```
GET /file-store/v1/file-sets?bucket=bulk-greenrevenue HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

```
In [4]: #step 2 - List Package IDs from bucket name

CFS_url = f'{RDP_HOST}/file-store/v1/file-sets?bucket=bulk-greenrevenue'

try:
    response = requests.get(CFS_url, headers={'Authorization': f'Bearer {access_token}'})
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if response.status_code == 200: # HTTP Status 'OK'
    print('Receive list Package IDs from RDP APIs')
else:
    print(f'RDP APIs: CFS request failure: {response.status_code} {response.reason}')
    print(f'Text: {response.text}')
```

Receive list Package IDs from RDP APIs

Example of the first entry of package IDs, the packageId is the `packageId` field.

```
In [5]: print(json.dumps(response.json()['value'][0], sort_keys=True, indent=2, separators=[

{
    "attributes": [
        {
            "name": "ContentType",
            "value": "GR Global Summary Full"
        }
    ],
    "availableFrom": "2023-10-15T21:02:50Z",
    "availableTo": "2023-11-15T21:02:50Z",
    "bucketName": "bulk-GreenRevenue",
    "contentFrom": "2023-10-08T20:55:00Z",
    "contentTo": "2023-10-15T20:55:00Z",
    "created": "2023-10-15T21:02:50Z",
    "files": [
        "4193-ab86-cdad0075-b8bb-863b8f13b68a"
    ],
    "id": "4028-3651-1e48ec4a-a823-7a6abe643b2a",
    "modified": "2023-10-15T21:02:53Z",
    "name": "Bulk-GR-Global-Summary-Full-v1-Jsonl-Delta-2023-10-15T21:01:28.093Z",
    "numFiles": 1,
    "packageId": "4e94-6d63-fa034dc-90e2-de33895bd4e9",
    "status": "READY"
}
```

The next step is choosing the package Id. Based on the [Green Revenues User Guide](#), the package Ids are the following:

- Package Name: Bulk-GR-Global-Summary-Full-v1: package Id `4e94-6d63-fa034dc-90e2-de33895bd4e9`
- Package Name: Bulk-GR-Global-Standard-Full-v1: package Id `4316-d43b-81c40763-8e6a-0dbec8162ab1`

```
In [6]: packageId = response.json()['value'][0]['packageId']
packageId
```

```
Out[6]: '4e94-6d63-fa034dc-90e2-de33895bd4e9'
```

Step 3: Listing the Filesets of the Bulk Green Revenues Data with the packageId

The next step is calling the CFS API with the bucket name and package Id to list all FileSets using **the package Id**.

API endpoint is `/file-store/v1/file-sets?bucket=bulk-greenrevenue&packageId={packageId}`

The HTTP Request structure is as follows:

```
GET /file-store/v1/file-sets?bucket=bulk-greenrevenue&packageId={packageId} HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

In [7]: *#step 3 - get file id from bucket name*

```
CFS_url = f'{RDP_HOST}/file-store/v1/file-sets?bucket=bulk-greenrevenue&packageId={packageId}'

try:
    response = requests.get(CFS_url, headers={'Authorization': f'Bearer {access_token}'})
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if response.status_code == 200: # HTTP Status 'OK'
    print('Receive FileSets list from RDP APIs')
else:
    print(f'RDP APIs: CFS request failure: {response.status_code} {response.reason}')
    print(f'Text: {response.text}'')
```

Receive FileSets list from RDP APIs

In [8]: `print(json.dumps(response.json()['value'][0], sort_keys=True, indent=2, separators=`

```
{
    "attributes": [
        {
            "name": "ContentType",
            "value": "GR Global Summary Full"
        }
    ],
    "availableFrom": "2023-10-15T21:02:50Z",
    "availableTo": "2023-11-15T21:02:50Z",
    "bucketName": "bulk-GreenRevenue",
    "contentFrom": "2023-10-08T20:55:00Z",
    "contentTo": "2023-10-15T20:55:00Z",
    "created": "2023-10-15T21:02:50Z",
    "files": [
        "4193-ab86-cdad0075-b8bb-863b8f13b68a"
    ],
    "id": "4028-3651-1e48ec4a-a823-7a6abe643b2a",
    "modified": "2023-10-15T21:02:53Z",
    "name": "Bulk-GR-Global-Summary-Full-v1-Jsonl-Delta-2023-10-15T21:01:28.093Z",
    "numFiles": 1,
    "packageId": "4e94-6d63-fa034dc-90e2-de33895bd4e9",
    "status": "READY"
}
```

The File ID is in the `files` array.

Next step is fine the files in CSV format from the return JSON 'name' attribute. This demonstration just uses the first one that has been found.

```
In [16]: index = 0
for index, package in enumerate(response.json()['value']):
    if 'Csv' in package['name']:
        print(index, package['name'])
        break
```

```
1 Bulk-GR-Global-Summary-Full-v1-Csv-Init-2023-10-01T21:01:18.831Z
```

```
In [17]: file_id = response.json()['value'][index]['files'][0]
file_id
```

```
Out[17]: '4bdb-a7e2-ed6976f-a327-613ff7bf59a8'
```

Step 4: Get the Bulk file URL on AWS S3

The last step is downloading the File using File ID with the RDP `/file-store/v1/files/{file ID}/stream` endpoint.

The HTTP Request structure is as follows:

```
GET /file-store/v1/files/{fileId}/stream?doNotRedirect=true HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

```
In [18]: #step 3 - get file stream (content) from file id

FileID_url = f'{RDP_HOST}/file-store/v1/files/{file_id}/stream?doNotRedirect=true'

try:
    response = requests.get(FileID_url, headers={'Authorization': f'Bearer {access_'
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if response.status_code == 200: # HTTP Status 'OK'
    print('Receive File URL from RDP APIs')
else:
    print(f'RDP APIs: CFS request failure: {response.status_code} {response.reason}')
    print(f'Text: {response.text}')
```

Receive File URL from RDP APIs

The File URL is in the `url` attribute.

```
In [19]: file_url = response.json()['url']
file_url
```

Step 5: Downloading the file

Based on the S3 `file_url` above, the actual file name is `Bulk-GR-Global-Summary-Full-v1-Init-2023-10-01T21_01_18.831Z.csv.gz`. So you need to replace the escape character `\` with `_` (underscore) character.

```
In [21]: #Download file
zipfilename = file_url.split("?")[0].split("/")[-1].replace("%3A","_")
print(f'Downloading File {zipfilename} ...')

try:
    response = requests.get(file_url)
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if response.status_code == 200: # HTTP Status 'OK'
    print('Receive File Successfully')
    open(zipfilename, 'wb').write(response.content)
    print(f'{zipfilename} Saved')
else:
    print(f'RDP APIs: Request file failure: {response.status_code} {response.reason}')
    print(f'Text: {response.text}')
```

And then unzip the file.

```
In [22]: #unzip file
import gzip
import shutil
try:
    unzipfilename = zipfile.split('.gz')[0]
    print(f'Unzip to {unzipfilename} ...')
    with gzip.open(zipfilename, 'rb') as f_in:
        with open(unzipfilename, 'wb') as f_out:
            shutil.copyfileobj(f_in, f_out)
    print('Done')
except Exception as e:
    print('The error is: ',e)
```

Unzip to Bulk-GR-Global-Summary-Full-v1-Init-2023-10-01T21_01_18.831Z.csv ...
Done

```
In [23]: # View some data
with open(unzipfilename) as f:
    data = f.read(2000)
    print(data)
```

_Row_Type, ObjectId, GreenRevenueFinancialPeriod_ObjectID, GreenRevenueFinancialPeriod_OrganizationId, GreenRevenueFinancialPeriod_PeriodEndDate, GreenRevenueFinancialPeriod_PeriodFiscalYear, GreenRevenueFinancialPeriod_PeriodId, GreenRevenueFinancialPeriod_PeriodStartDate, GreenRevenue_CreatedDatetime, GreenRevenue_EUTaxonomyEligibilityRevenuePercent, GreenRevenue_EstimationTypeCode, GreenRevenue_GreenRevenuePercent, GreenRevenue_LastModifiedDatetime, GreenRevenue_MaximumGreenRevenueExNuclearPercent, GreenRevenue_MaximumGreenRevenuePercent, GreenRevenue_MinimumGreenRevenueExNuclearPercent, GreenRevenue_MinimumGreenRevenuePercent, GreenRevenue_ReportedRevenueAmount, GreenRevenue_ReportedRevenueCurrencyId, GreenRevenue_GreenRevenueCurrencyIdDescription, GreenRevenue_Tier1And2GreenRevenuePercent, GreenRevenue_Tier1GreenRevenuePercent, GreenRevenue_Tier2GreenRevenuePercent, GreenRevenue_Tier3GreenRevenuePercent
","",4295911047;92,4295911047;92,4295911047,2013-12-31,2013,92,2013-01-01,2023-02-10T15:41:58.010Z,0,Disclosed,0,2023-02-09T00:00:00.000Z,null,0,0,0,698266000,500110,USD,0,0,0,0
","",4295896578;91,4295896578;91,4295896578,2018-06-30,2018,91,2017-07-01,2023-02-10T15:41:58.010Z,0,Disclosed,0,2023-02-09T00:00:00.000Z,null,0,0,0,327300000,500269,GBP,0,0,0,0
","",4295894869;43,4295894869;43,4295894869,2009-12-31,2009,43,2009-01-01,2023-02-10T15:41:58.010Z,0,Disclosed,0,2023-02-09T00:00:00.000Z,null,0,0,0,238506000,500269,GBP,0,0,0,0
","",4295913590;8,4295913590;8,4295913590,2010-12-31,2010,8,2010-01-01,2023-02-10T15:41:58.010Z,0,Disclosed,0,2023-02-09T00:00:00.000Z,null,0,0,0,13922000,500110,USD,0,0,0,0
","",5000692937;170,5000692937;170,5000692937,2021-12-31,2021,170,2021-01-01,2023-02-10T15:41:58.010Z,0,Disclosed,0,2023-02-09T00:00:00.000Z,null,0,0,0,540786000,500110,USD,0,0,0,0
","",4295904191;5,4295904191;5,4295904191,2010-10-31,2010,5,2009-11-01,2023-02-10T15:41:58.010Z,0,Disclosed,0,2023-02-09T00:00:00.000Z,null,0,0,0,1371842000,500110,USD,0,0,0
","",4295890203;281,4295890203;281,4295890203,2017

In []:

In []: