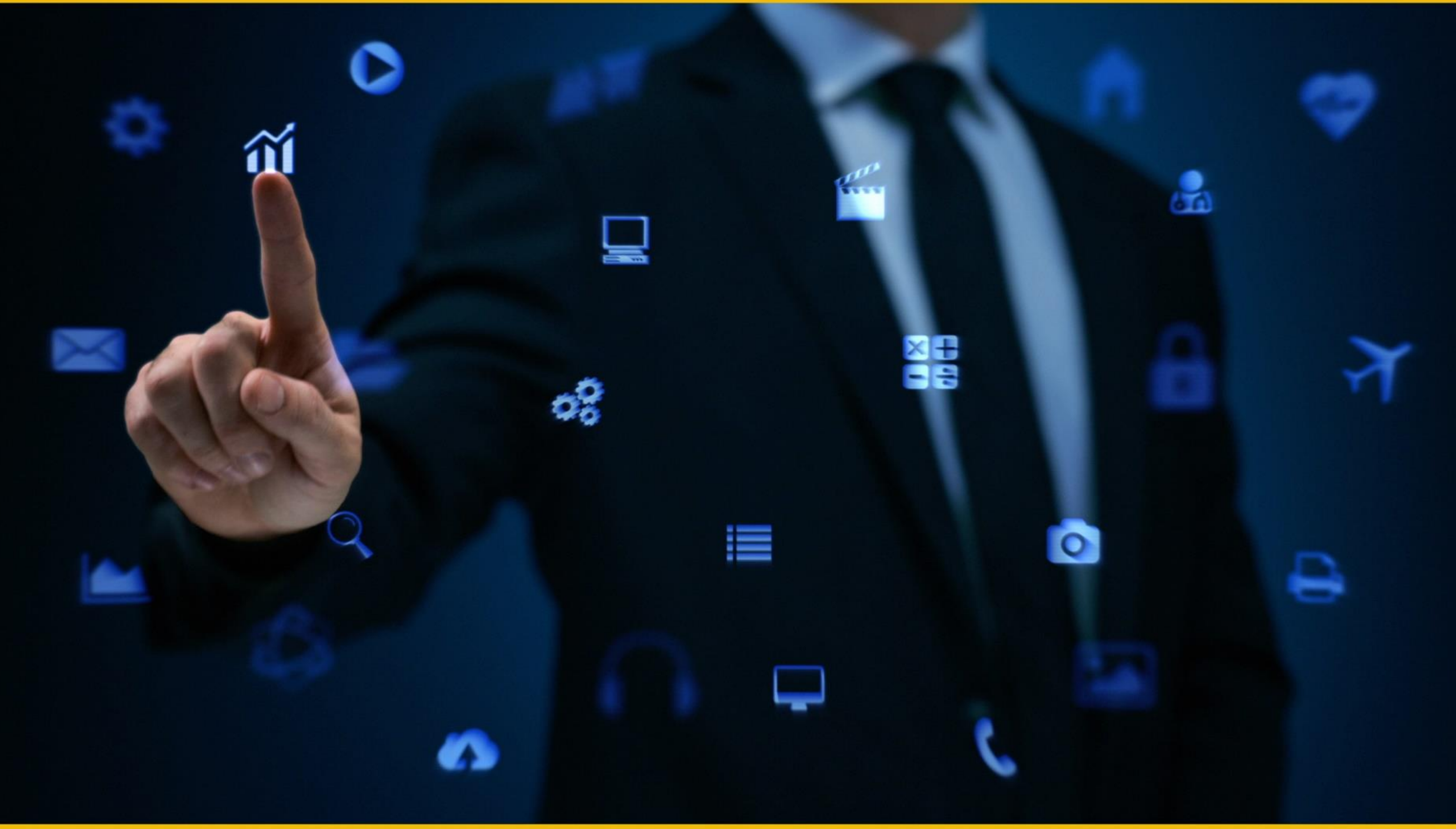
BRASIL
BOLSA
BALCÃO



UP2DATA

INSTALLATION GUIDE

Sumary

Figure Index

1. UP2DATA

UP2DATA is a Windows service that provides, in an organized, consolidated and standardized end-of-day data from B3 markets. The data will be transmitted through software and will be available according to the closing of each market or at the end of the day, with monitored delivery.

1.1 UP2DATA Cloud

There are two ways of accessing UP2DATA: in the Client version the files can be accessed through the contractor's directory, while in the Cloud version the files can be accessed in the cloud through a Cloud reader such as Microsoft Azure.

The Cloud version is a solution for storing large amounts of data in the cloud, in a staggered and secure way, with high-performance technological platforms. The data is organized in Blobs and made available in folders within a cloud tool like Microsoft Azure.

1.2 UP2DATA Cloud characteristics

The UP2DATA Cloud joins 30 consecutive days of historical data that are stored in the cloud. The files will be available in CSV, TXT, JSON and XML and the customer chooses the files he wants to download. In addition, it can be accessed on more than one machine.

The sharing of the Cloud Keys (SAS) is now allowed and must not be used at the same time. The SAS Keys are valid for three months and are non-transferable.

2. Necessary Informations

When installing, is important to have some information to access the API. Are they:

Access Kit – The Access Kit is sent at the contracting step, in a txt document, containing the Contractor's credentials (Client ID, Client Secret and Certificate Password.

- **Certificate –** Pfx file sent by the contracting team. It is what guarantees the access to the product through the platform. Valid for two years.
- **Tool to access the API –** The API (UP2DATA Cloud) can be accessed through code implementations or apps that make the use more intuitive, such as Postman Collection.
- **Cloud Tool –** To download the files it is necessary to have a cloud tool installed, such as Microsoft Azure.

3. Access the API via Postman Collection

After getting the information from the previous topic, it is necessary to access the API. In case you have your own programming mechanism or already use the Postman Collection, please, disregard this step. In case you don't, keep going.

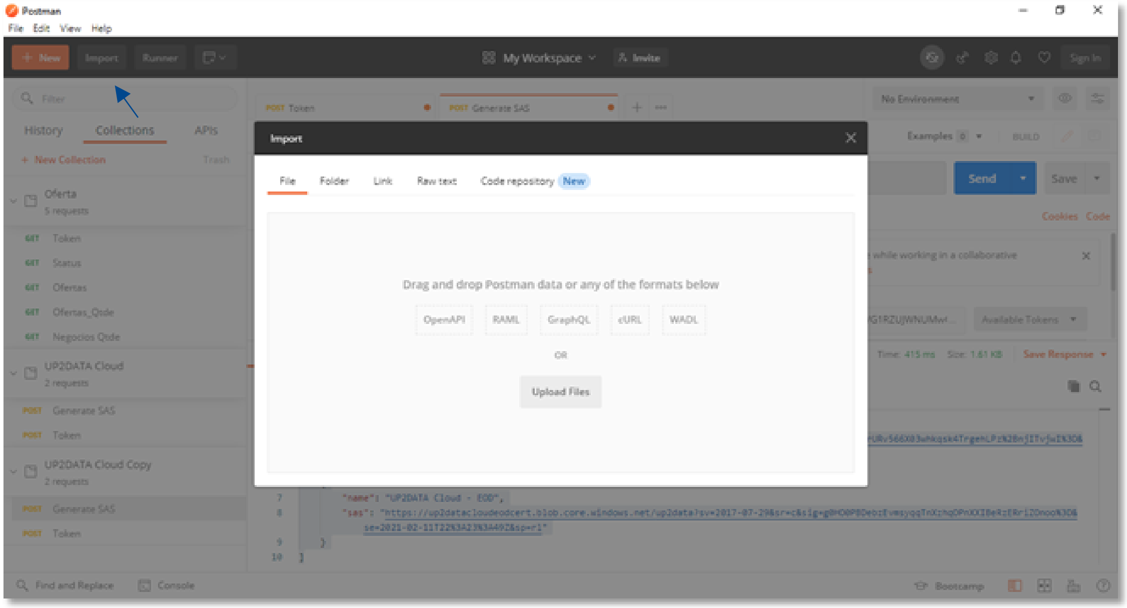
Here's the step by step to generate the Postman Collection:

1. Check the application's default settings.

2. Download the Postman Collection in your computer

3. Then, download the file from the UP2DATA Cloud to the Postman Collection through this link.

4. Click on the "Import" menu in the upper left corner of the Postman Collection platform and upload the library you just downloaded to your computer through the previous step.



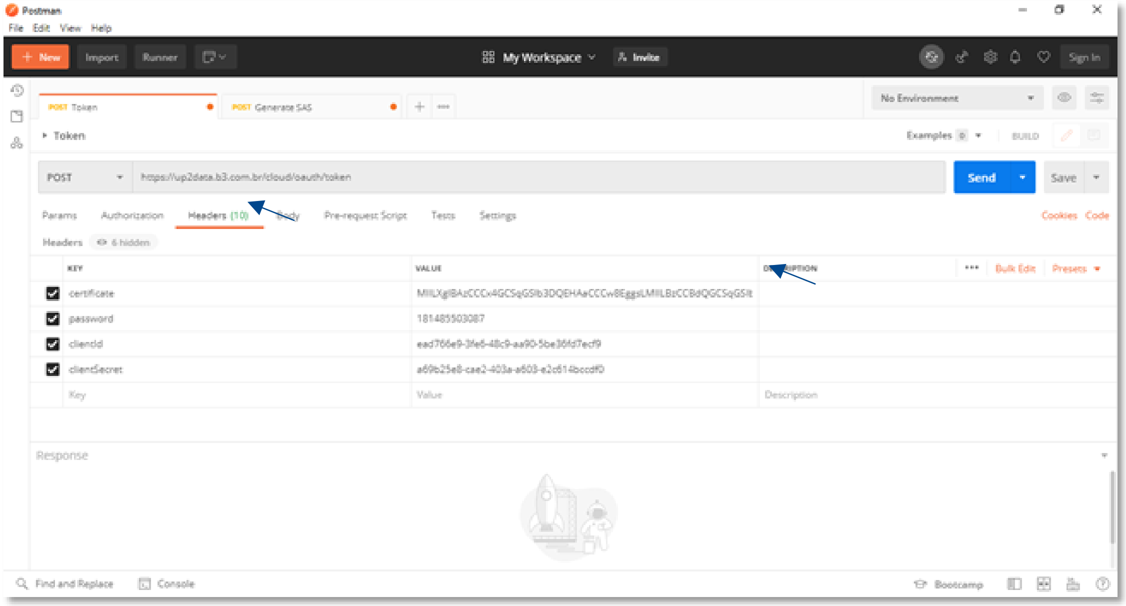
```
        "name": "UP2DATA Cloud - EOD",
        "sas": "https://up2datacloudeodcert.blob.core.windows.net/up2data?sv=2017-07-29&sr=c&sig=g0HO0PBDebzEvmsyqqTnXzhqOPnXXI8eRzERr1ZOnoo%3D&
            se=2021-02-11T22%3A23%3A49Z&sp=rl"
    }
}
```

Figure 1- View of the Postman Collection Import Menu

Now that you have access to the Postman Collection, the certificate and the Access Kit, you will be able to generate the SAS keys that will be used inside a cloud reader (such as Microsoft Azure) to access the contracted channels.

5. Go to the **Headers** menu (Figure 2) and fill the fields with the information that was sent in the access kit (certificate*; password; clientId; clientSecret).

IMPORTANT*: The "certificate" field must be filled in with your certificate ("certificate.pfx") encoded to base 64 through the website https://www.base64encode.org/ or any other platform you prefer. In this link, insert the certificate sent at the time of contracting in the formatting field for base 64 and click on "encode". After obtaining the certificate, click on "Click or Tap Here to download the encoded file" to access the file with the base 64 code and copy it into the Postman field assigned to it (Figure 2)



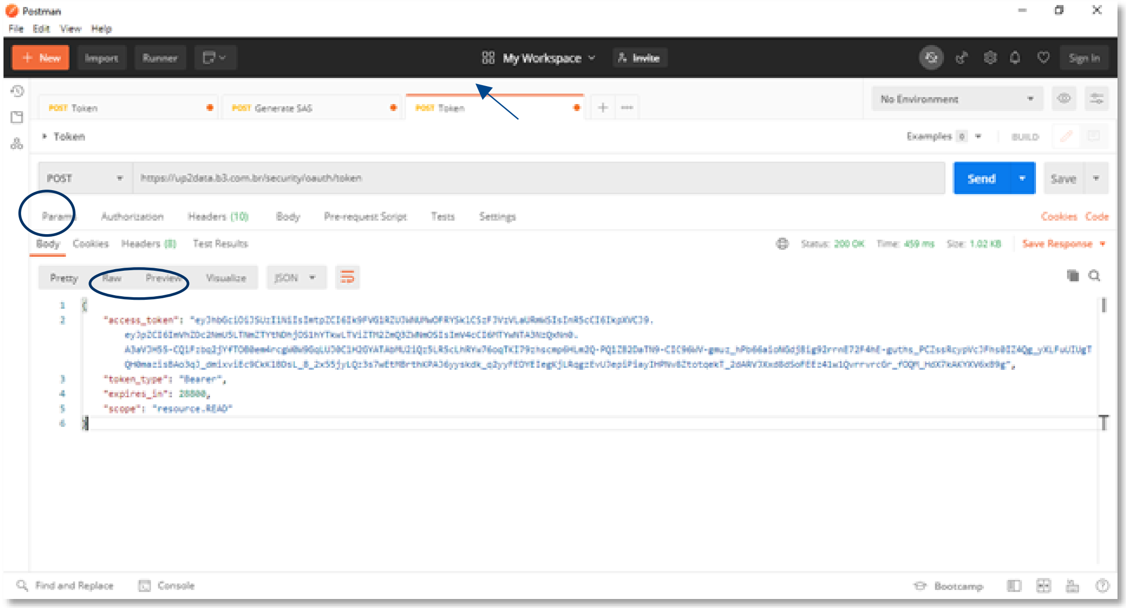
POST https://up2data.b3.com.br/cloud/oauth/token

	KEY	VALUE	DESCRIPTION
☑	certificate	MIILXglBAzCCCx4GCSqGSIb3DQEHAaCCCw8EggsLMIILBzCCBdQGCSqGSIt	
☑	password	181485503087	
☑	clientId	ead766e9-3fe6-48c9-aa90-5be36fd7ecf9	
☑	clientSecret	a69b25e8-cae2-403a-a603-e2c614bccdf0	
	Key	Value	Description

Figure 2 – View of Headers Menu, from the Token tab, inside Postman Collection

6. Press “send” after filling in all the data. Then, you will have the token. This token will need to be informed in the API (UP2DATA Cloud) to generate the SAS keys (Generate SAS).

7. Copy the key that are between the “ ” of the access_token of the Token's Body field, as shown in the image below.

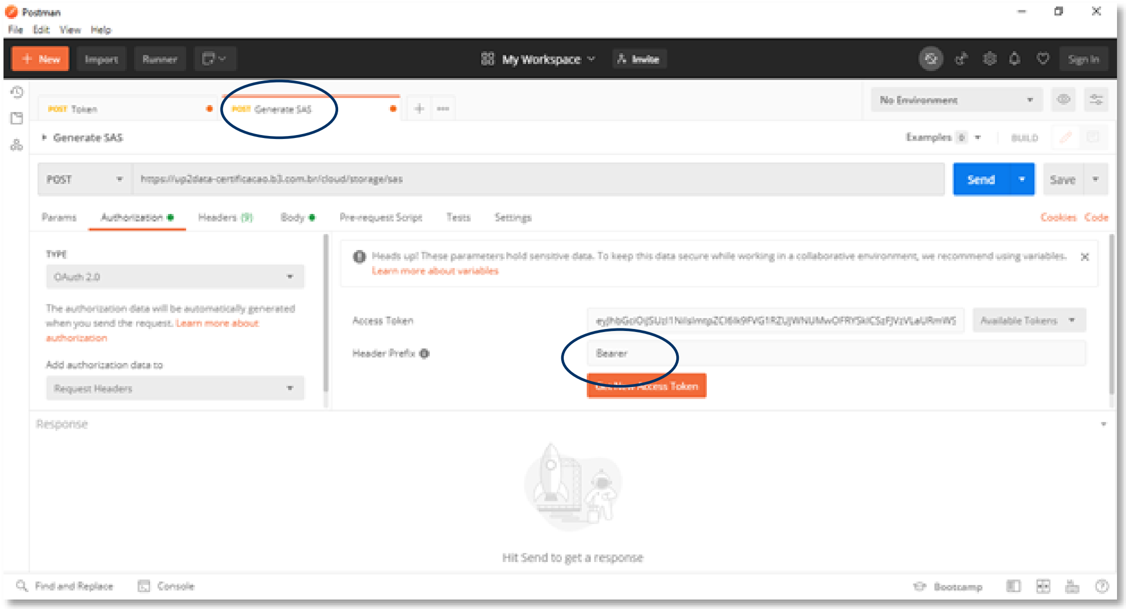


```
{
    "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6Ik9FVG1RZUJWNUMwOFRYSk1CSzF3VzVLaURmdSIsInR5cCI6IkpXVCJ9.[illegible]",
    "token_type": "Bearer",
    "expires_in": 28800,
    "scope": "resource.READ"
}
```

Figure 2 - View of the Body tap, from menu Token, inside Postman Collection Platform

8. Open the “Generate SAS” tab (as shown below), and paste the key copied from the Token tab (step above) in the Access Token field of the Authorization menu

9. Fill in the “Heard Prefix” field of the “Generate SAS” tab with the word “Bearer” (as shown in Figure 4)



Generate SAS

POST https://up2data-certificacao.b3.com.br/cloud/storage/sas

TYPE: OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about authorization

Add authorization data to: Request Headers

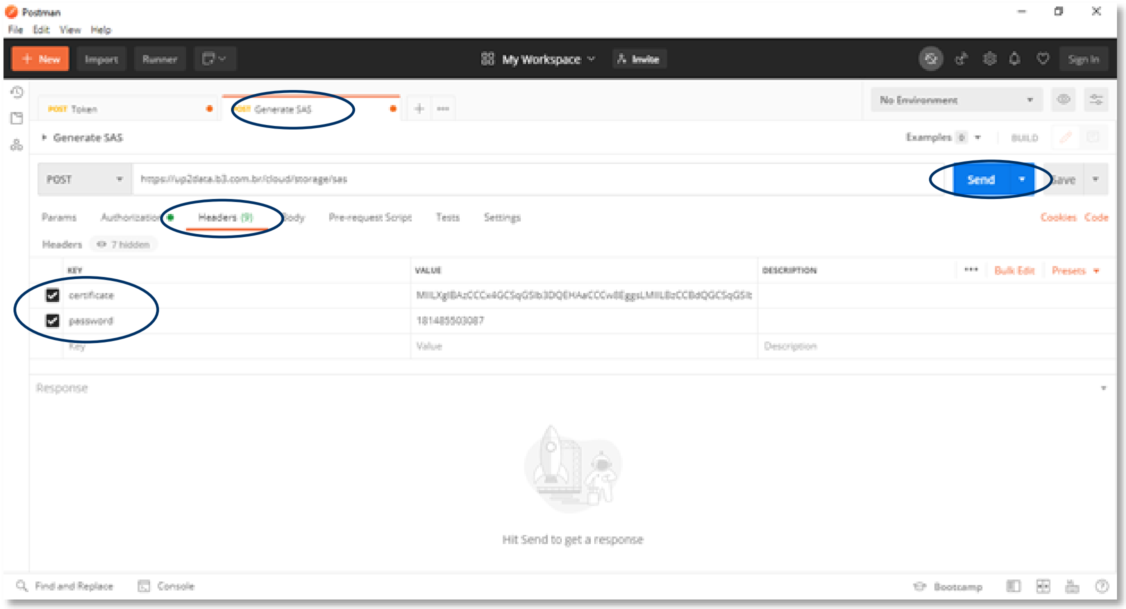
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables

Access Token: eyJhbGciOiJSUzI1NiIsImtpZCI6Ik9FVG1RZUJWNUMwOFRYSkICSzFJVzVLaURmWS

Header Prefix: Bearer

Figure 3- Vision of Authorization tab, from Generate SAS menu, inside Postman Collection Platform

10. In the Headers menu on the same tab as in step 7 and 8 (Generate SAS), fill the certificate and password fields (image below) with the information that was sent in the access kit and click on send.

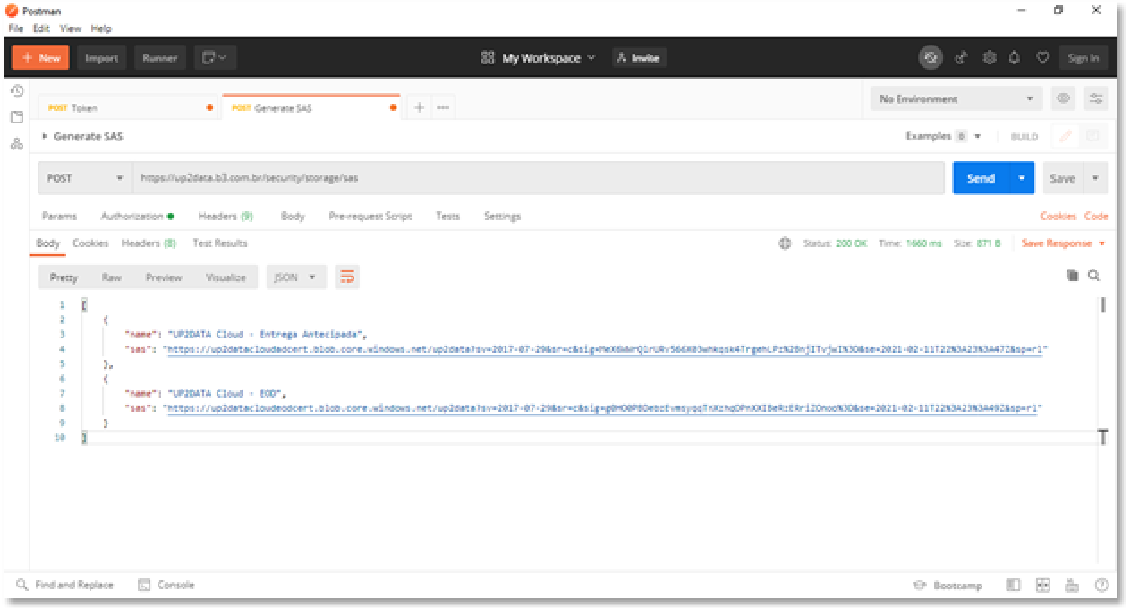


POST https://up2data.b3.com.br/cloud/storage/sas

	KEY	VALUE	DESCRIPTION
☑	certificate	MIILXgIBAzCCCx4GCSqGSIb3DQEHAaCCCw8EggsLMIILBzCCBdQGCSqGSIt	
☑	password	181485503087	
	Key	Value	Description

Figure 4- Vision of Headers tab, from Generate SAS menu, inside Postman Collection Platform

11. The SAS keys (contracted in the body tab from the Generate SAS menu) will appear on the screen. These keys are valid for 90 days and this process can be automated through the implementation of a command line according to item 5 of this manual.



POST https://up2data.b3.com.br/security/storage/sas

Status: 200 OK Time: 1660 ms Size: 871 B

```
[
    {
        "name": "UP2DATA Cloud - Entrega Antecipada",
        "sas": "https://up2datacloudadcert.blob.core.windows.net/up2data?sv=2017-07-29&sr=c&sig=MeX6WArQ1rURv566X83whkqsk4TrgehLPz%2BnjITvjwI%3D&se=2021-02-11T22%3A23%3A47Z&sp=rl"
    },
    {
        "name": "UP2DATA Cloud - EOD",
        "sas": "https://up2datacloudeodcert.blob.core.windows.net/up2data?sv=2017-07-29&sr=c&sig=p0HO0PBDebzEvmsyqq7nXzhqDPnXXIBeRzERr1ZOnoo%3D&se=2021-02-11T22%3A23%3A49Z&sp=rl"
    }
]
```

Figura 5- Vision of Body tab, from Generate Sas Menu, inside Postman Collection Platform

4. Access to UP2DATA Cloud via Microsoft Azure

To access the UP2DATA Cloud files it is necessary to apply thekey in a cloud tool. One of these tools is Microsoft Azure Storage Explorer.

Here’s how to use Microsoft’s web cloud reader (Microsoft Azure Storage Explorer):

1. Install Microsoft Azure Storage Explorer.

2. Open the application.

3. With the right mouse button, click on ***Storage Accounts***, then select the item ***Connect to azure storage*** (image below)

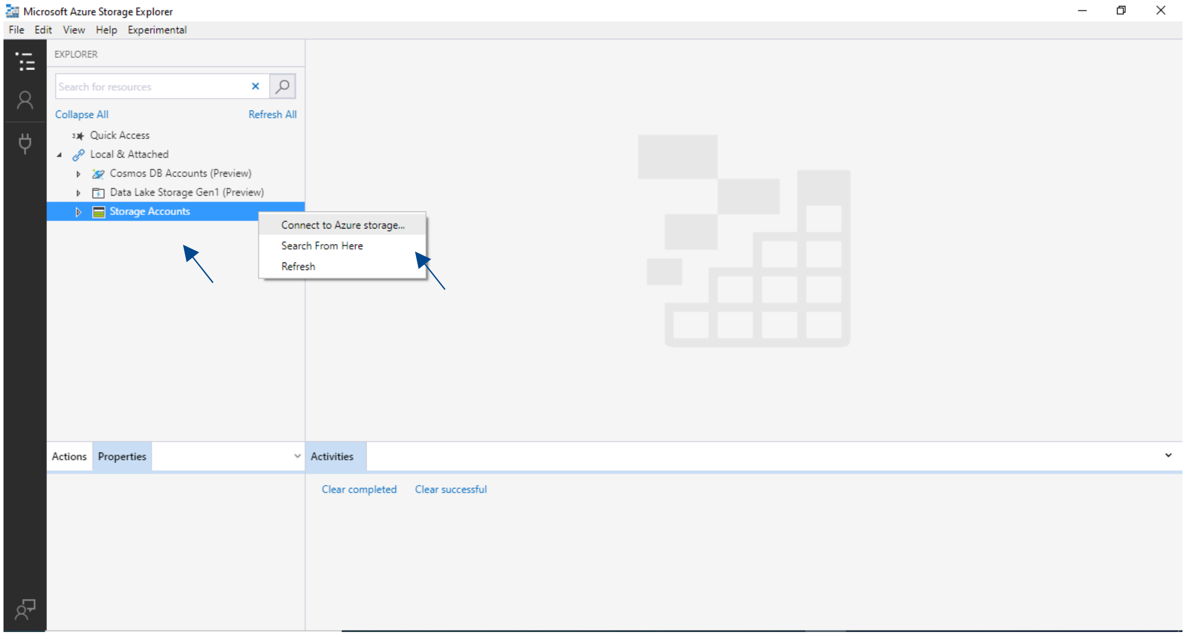
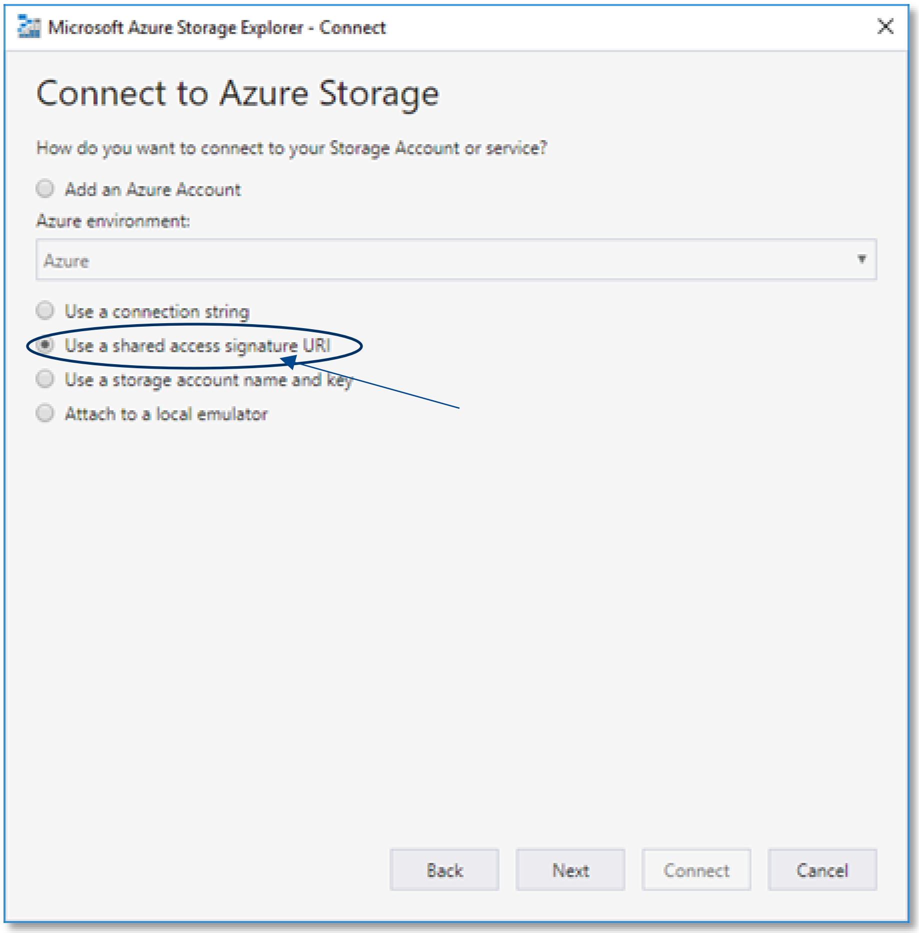


Figure 6- Microsoft Azure Storage Accounts Options view

4. Click on ***Use a shared access signature URI*** and then on **Next**. (image below)



Microsoft Azure Storage Explorer - Connect

Connect to Azure Storage

How do you want to connect to your Storage Account or service?

○ Add an Azure Account

Azure environment: Azure

○ Use a connection string

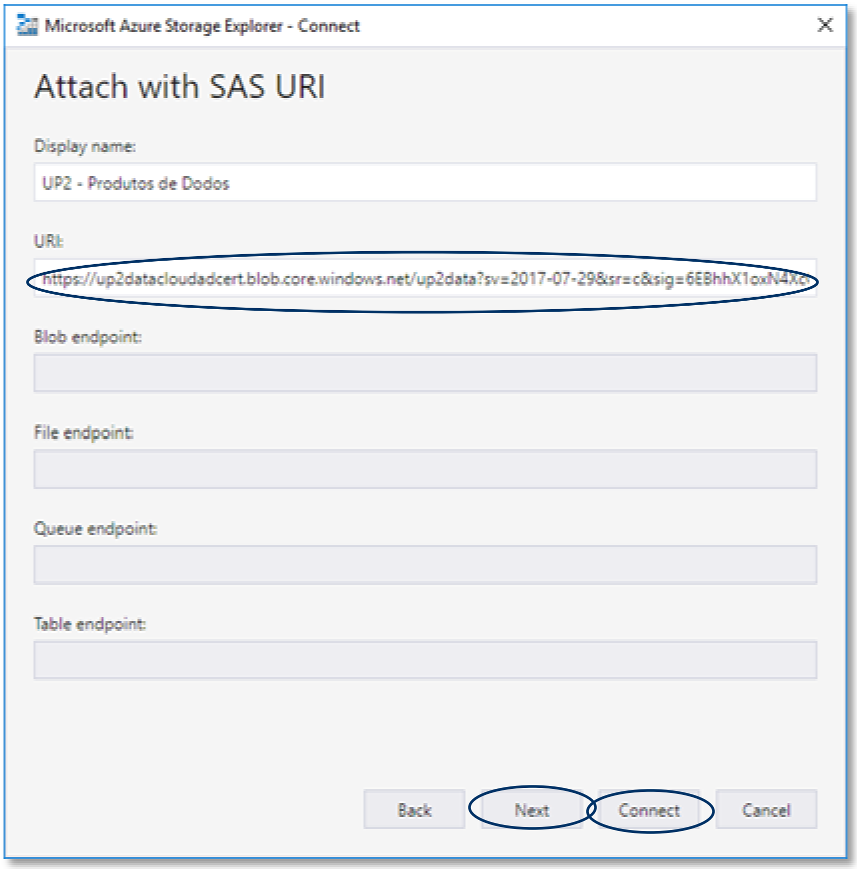
● Use a shared access signature URI

○ Use a storage account name and key

○ Attach to a local emulator

Figure 7- Connect to Azure Storage view from Microsoft Azure

5. Enter the SAS Key acquired by the "Generate SAS" API in the **URI Field.** Change the **display name** field, if necessary. Then, click **Next** and **Connect**. (image below)



Microsoft Azure Storage Explorer - Connect

Attach with SAS URI

Display name:

UP2 - Produtos de Dodos

URI:

https://up2datacloudadcert.blob.core.windows.net/up2data?sv=2017-07-29&sr=c&sig=6EBhhX1oxN4Xc

Blob endpoint:

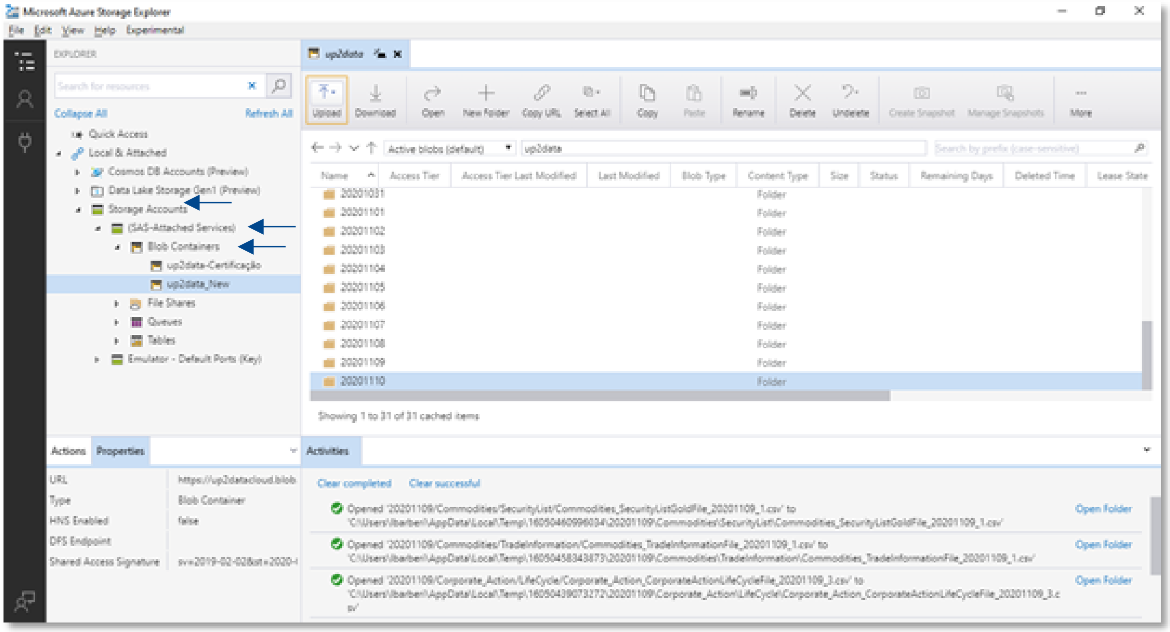
File endpoint:

Queue endpoint:

Table endpoint:

Figure 8- Microsoft Azure Tool Attach with SAS URI View

6. To access the files: click on **Storage Accounts**, then **(SAS-Attached Services)**, **Blob Containers** and choose the channel where you want to access the files (image below)



Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Status	Remaining Days	Deleted Time	Lease State
20201031					Folder					
20201101					Folder					
20201102					Folder					
20201103					Folder					
20201104					Folder					
20201105					Folder					
20201106					Folder					
20201107					Folder					
20201108					Folder					
20201109					Folder					
20201110					Folder					

Showing 1 to 31 of 31 cached items

URL	https://up2datacloud.blob
Type	Blob Container
HNS Enabled	false
DFS Endpoint	
Shared Access Signature	sv=2019-02-02&st=2020-(

```
Opened '20201109/Commodities/SecurityList/Commodities_SecurityListGoldFile_20201109_1.csv' to
'C:\Users\barberi\AppData\Local\Temp\16050460996034\20201109\Commodities\SecurityList\Commodities_SecurityListGoldFile_20201109_1.csv'
Opened '20201109/Commodities/TradeInformation/Commodities_TradeInformationFile_20201109_1.csv' to
'C:\Users\barberi\AppData\Local\Temp\16050458343873\20201109\Commodities\TradeInformation\Commodities_TradeInformationFile_20201109_1.csv'
Opened '20201109/Corporate_Action/LifeCycle/Corporate_Action_CorporateActionLifeCycleFile_20201109_3.csv' to
'C:\Users\barberi\AppData\Local\Temp\16050439073272\20201109\Corporate_Action\LifeCycle\Corporate_Action_CorporateActionLifeCycleFile_20201109_3.c
sv'
```

Figure 9- View of file folders separated by day in Microsoft Azure

7. In case the files do not download automatically, check the proxy settings in the Edit > Configure Proxy menu. These data are institution specific. If you do not have this information, contact your technical support.

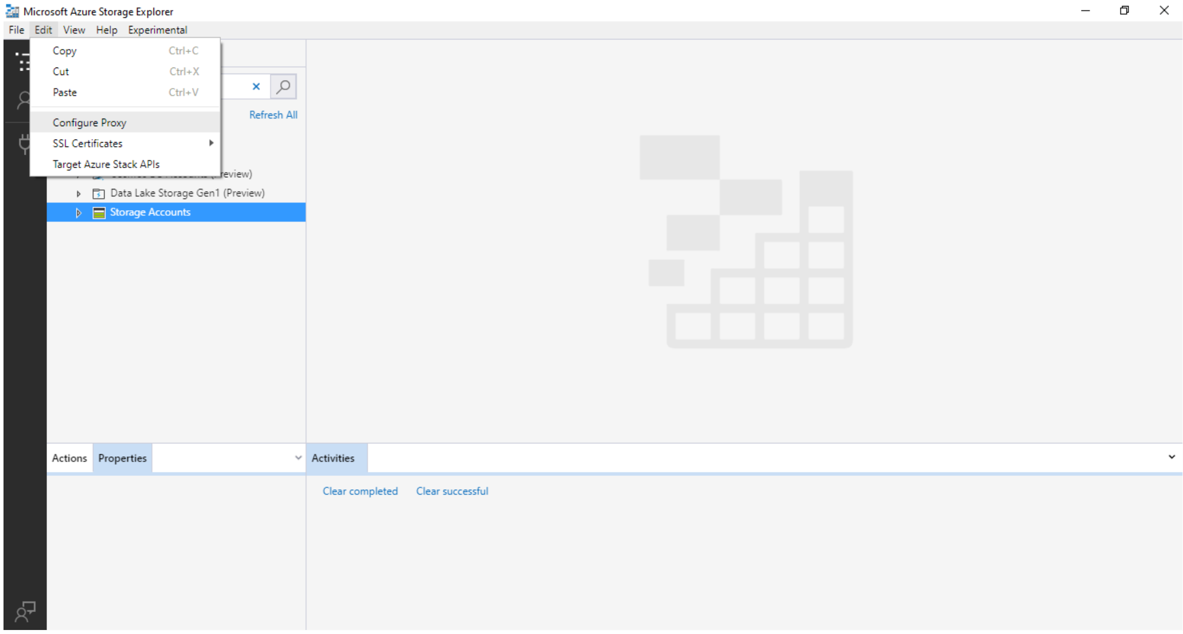
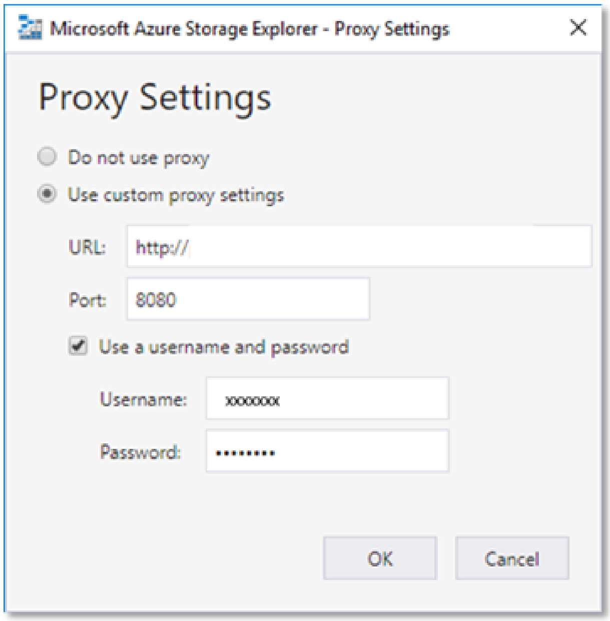


Figure 10- View of edit options in Microsoft Azure

8. Check if the proxy information is correct



○ Do not use proxy

● Use custom proxy settings

URL: http://

Port: 8080

☑ Use a username and password

Username: xxxxxxx

Password: ••••••••

Figure 12- Proxy Settings view in Microsoft Azure

5. Automation via Microsoft AzCopy

In this section, we give some examples of how to get UP2DATA files from a command line using Microsoft Azcopy.

First, remember some things about the SAS key:

The SAS (Shared Access Signature) is made up of three parts:

- **Azure Account Name** – This is the host subdomain part of the URI, so we have in UP2DATA: **up2datacloudxxx** (channel name).

- **Container name** – Represents the root directory if the entire UP2DATA repository. You will find it right after the web address //up2datacloudxxxx.blob.core.windows.net/. It's the first word after the "/" windows.net. Our case: **up2data.**

- **SAS Search suffix:** All text after the "?"

Structurally we have: https://**{a}**.blob.core.windows.net/**{b}**?**{c}**

API REST from Blob Service

Inside Azure's storage service, there are many different API. The "Blob Service" is one of them. We focus on "list", "filter" and "download data", as we understand that they will be the most used to consume UP2DATA files.

- ***List***

Start with Listing Blobs

To use the List API, it is necessary to add the string operation of Search key. In this case, ***restype=container&comp=list***. The operation needs to be placed right after the beginning of the Search string. It would look like: https://**{a}**.blob.core.windows.net/**{b}**?**{operation}**&**{c}**

In the end, the key will look like: https://up2datacloudeod.blob.core.windows.net/up2data?**restype=container&comp=list...** (SAS KEY Continuation)

- ***Filter***

As a result of the Search, you will see all files (represented by blobs in this API) that are available in UP2DATA. If you want to access files from a specific date, you need to put that information in the filter.

The listing API provides a parameter called prefix, which is simply the beginning off the full filename, including directories and subdirectories. So in the filter, you can add the prefix as a parameter. (exemplo: **&prefix={filter}**).

An example: to find a trade information file for the Janurary 14 indices, the prefix parameter will be: **20200114/Index/TradeInformation**.

The complete key will look like: https://up2datacloudeod.blob.core.windows.net/up2data?**restype=container&comp=list&sv=2017-07-29&prefix=20200114%2FIndex%2FTradeInformation...** (SAS Key continuation)

Downloading the data:

Still using the example above, it returned na XML with 9 entries, but we would like to refine the Search to arrive at the download file format. In this example, we choose the CSV format.

To download, you need to enter the full path of the file/blob in the original SAS Key. The download is done by the Get Blob API.

The chosen file was: **20200114/Index/TradeInformation/Index_TradeInformationIndexFile_20200114_2.csv**

Structurally we have: https://**{a}**.blob.core.windows.net/**{b}**/**{file-path}**?**{c}**

Soon, the key will look like: https://up2datacloudeod.blob.core.windows.net/up2data/**20200114/Index/TradeInformation/Index_TradeInformationIndexFile_20200114_2.csv?**... (continuation of SAS Keys)

The contents of the file will appear as part of the HTTP Response.

Another way to access the UP2DATA Cloud data is using AzCopy, a command line that can be accessed by the most popular operating systems (Windows, Linux and MacOS). How to Use:

- Download the AZCopy executable on your computer, accessing the website:https://docs.microsoft.com/pt-br/azure/storage/common/storage-use-azcopy-v10?toc=%2fazure%2fstorage%2fblobs%2ftoc.json. Choose one of the available options and follow the instructions.
- Create a folder in the C directory ti store the UP2DATA data.
- Next, open a terminal (When Linux or MacOS) oro ne of the Windows command prompts (CM or Powershell) and run one of the commands below:

Download all UP2DATA documents from the last 30 days

```
azcopy copy “https://up2datacloudadcert.blob.core.windows.net/up2data?st=2020-12-31T13XXXXXXXX&se=YYYYYYY&sp=rl&sv” "c:\up2data" --recursive=true
```

Download all files of the day

```
azcopy copy "https://up2datacloudadcert.blob.core.windows.net/up2data? st=2020-12-31T13XXXXXXXX&se=YYYYYYY&sp=rl&sv&prefix=20201111" "c:\up2data" --recursive=true
```

Download a specific day and channel – more suitable for the “Old”.

```
azcopy list  st=2020-12-31T13XXXXXXXX&se=YYYYYYY&sp=rl&sv &prefix=20201110/Index/TradeInformation "c:\up2data" --recursive=true
```

Download only one type of file except for files from the “Old” folder

```
azcopy copy "https://up2datacloudadcert.blob.core.windows.net/up2data? st=2020-12-31T13XXXXXXXX&se=YYYYYYY&sp=rl&sv&prefix=20201110/Index/TradeInformation" "c:\up2data" --include-pattern "*.csv" --recursive
```