



# API User Guide



## Contents

- Use Case & Limits.....2-3
- Overview & Getting Authenticated..... 4-6
- Browsing APIs..... 5-6
- Historical Bulk Data ..... 7-9
- Historical Data ..... 10-11
  - Metadata ..... 11
  - Tag List ..... 12
  - Backtest Only ..... 13
  - Generic ..... 13
  - Tag Browser .....14
  - Equity Identifier.....15-16
  - User Defined Tags ..... 17-18
- Streaming API ..... 19
  - Subscription Control ..... 20
  - Binary Format ..... 21
- Flat File Download Web Service..... 22
  - Overview..... 22
  - Terminology ..... 22
  - Tag ..... 22
  - Choosing the Correct Service ..... 23
- Workflow..... 23
- Request Format ..... 24
  - Frequency ..... 24
- Response Format..... 25
  - Link ..... 25
  - Limit ..... 25
  - Status ..... 25
  - Message ..... 25
  - Output File Format ..... 25-26
  - Get Data Set and Date Range Info ..... 26-27
- Your Questions Answered..... 28

API access is required and completed by reaching out to the Citi Velocity Sales team.  
For any other questions or concerns please email: [citivelocitysales@citi.com](mailto:citivelocitysales@citi.com) | [citivelocityapi@citi.com](mailto:citivelocityapi@citi.com)

*\* This User Guide has been consolidated and now includes API User Guides across the entire platform. All documents have been combined for user convenience.*

Use Case & Limits

Use Case		Velocity Charting Service
I want to export large amounts of data at one time.	Historical Bulk	Best for hourly data for some series going back 6 months.
I want to query for modest amounts of intraday data. (EOD/daily data)	Historical	
I want to receive Intraday streaming updates.	Streaming	
I want to export large amounts of historical intraday time series data for all data-series (tags) in a dataset.	Flat File	

Historical Bulk Data Use Case and Limits:

Counting Limits:

Type	Job Submission	API Status	Job Status	Download Output File
Max Calls	1,000	2,000	2,000	1,000
Max Tags	10,000	-	-	-

Frequency and Min History

Frequency	Description	Min History
Hourly	Hourly	1 Year
MI10	Ten minutely	2 Months
MI01	Minutely	1 Week

Request Size

Type	Citi Id-From	Citi Id-To	Filter
Max Calls	10,000	10,000	1,000
Max Items	100,000	100,000	1,000,000
Max Calls per Items	5	-	-

Bespoke tags created via Vol & Swap Pricer are available via Historical API

Max Items per Call	10
--------------------	----

Rate Limit

Type	Limit	Description
Concurrency	1 thread	Only one request is processed at a time
Rate	1 per second	At Most one request is processed per second

Streaming API Data Limits:

Lifespan of Subscriptions

The set of active subscriptions is bound to the connection. All of the subscriptions are canceled (unsubscribed) on the server automatically after the connection is closed.

Limits

The following limits are in place to restrict the amount of resources consumed by a single login:

Description		Limit
Simultaneous connections per user		1
Simultaneous connections per company (Limit count base on gp number)		10
Max number of SUB messages per user (max number of tags allowed to be subscribed in one day)		100,000
Max number of UNSUB messages per user		100,000
Max number of connects		100

Frequency	Max # of active subscriptions
MI01 (Minutely)	20,000

Max number of active subscriptions depends on the frequency. MI01 is the default frequency if user does not pass any frequency.

These limits reset at 2am EST.

Clients using all 20,000 subscriptions will be limited to at most 5 connects in practice due to the limit on the number of SUB requests.

Available API Data Options

There are various API's that the client can access on Citi Velocity.

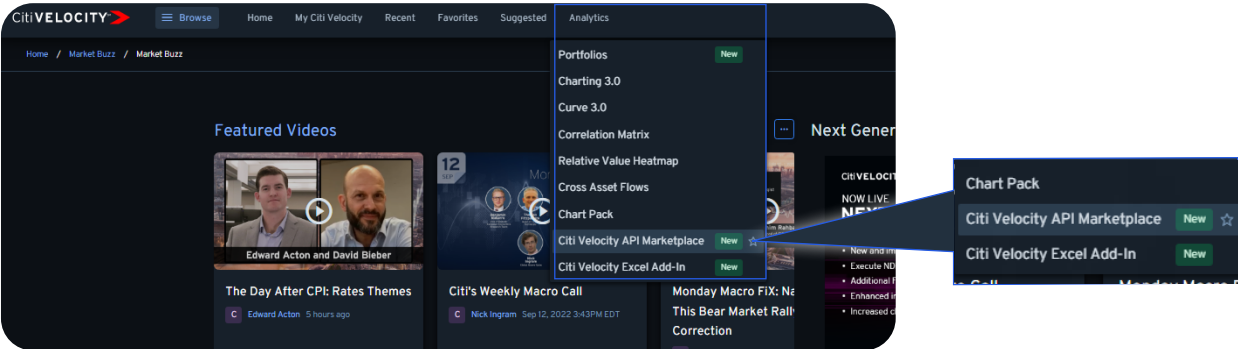
Asset Class	Meta Data	Tag List	Tag Browser	Bulk Data	Flat Files	Historical Data	Streaming Data	Eq Identifier
FX	✓	✓	✓	✓	✓	✓	✓	
Credit	✓	✓	✓	✓	✓	✓		
Equities	✓	✓	✓	✓	✓	✓	✓	✓
Municipals	✓	✓	✓	✓	✓	✓		
Rates	✓	✓	✓	✓	✓	✓	✓	
Commodities	✓	✓	✓	✓	✓	✓		
Economics	✓	✓	✓	✓	✓	✓		
Securitized Product	✓	✓	✓	✓	✓	✓	✓	

Overview

The purpose of this guide is to help clients of Citi Velocity to get started with API Marketplace so that they can access Citi Markets’ APIs and data offerings.

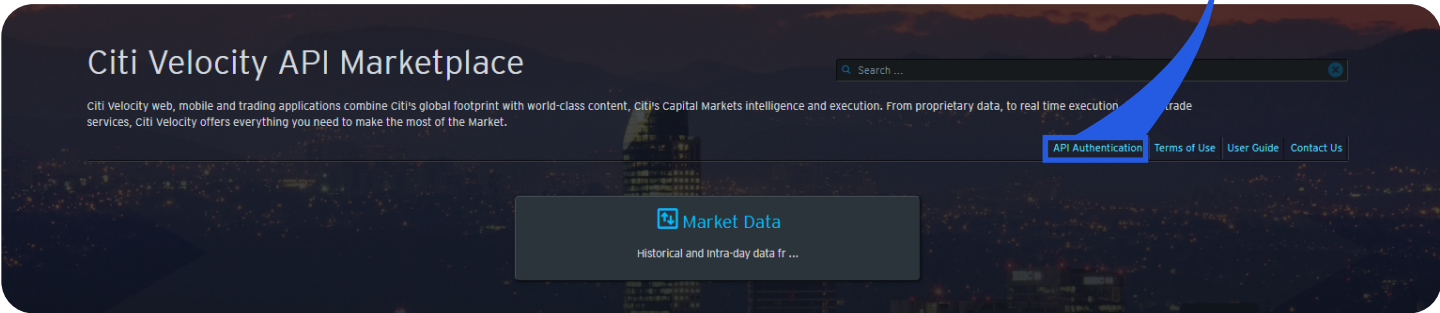
Getting Access to Citi Velocity API Marketplace

- 1. Once access is granted, the user can access the Citi Velocity API Marketplace by going to the menu on Citi Velocity: **Analytics > Citi Velocity API Marketplace**



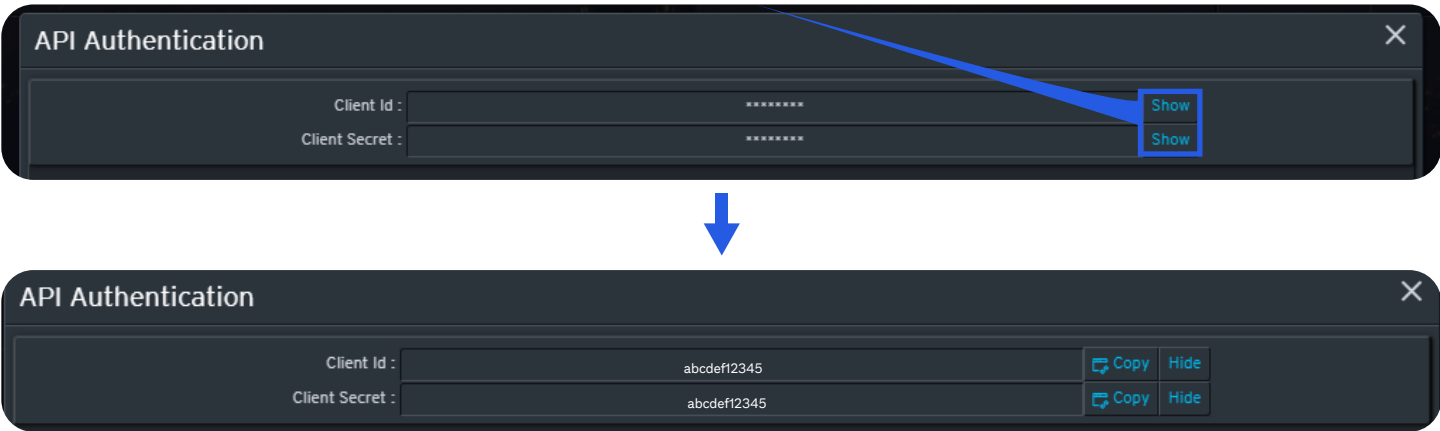
Getting Authenticated

- 2. Each API call requires authentication. User can get authenticated by clicking on the **API Authenticator** button on the top right.



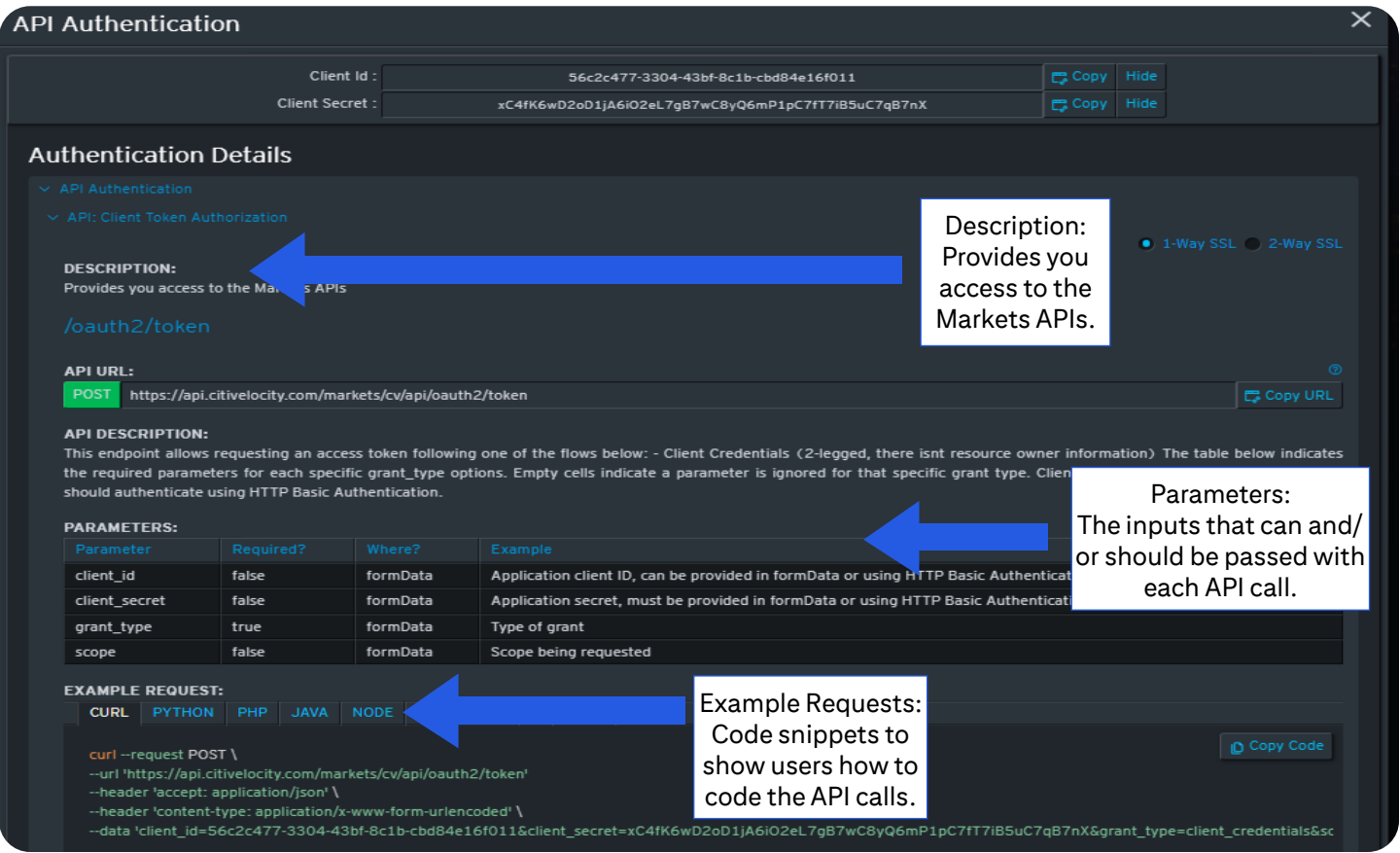
- 3. Each authentication API requires two inputs:
  - Client ID
  - Client Secret

This can be found by clicking the **Show** check boxes to the right of each field.

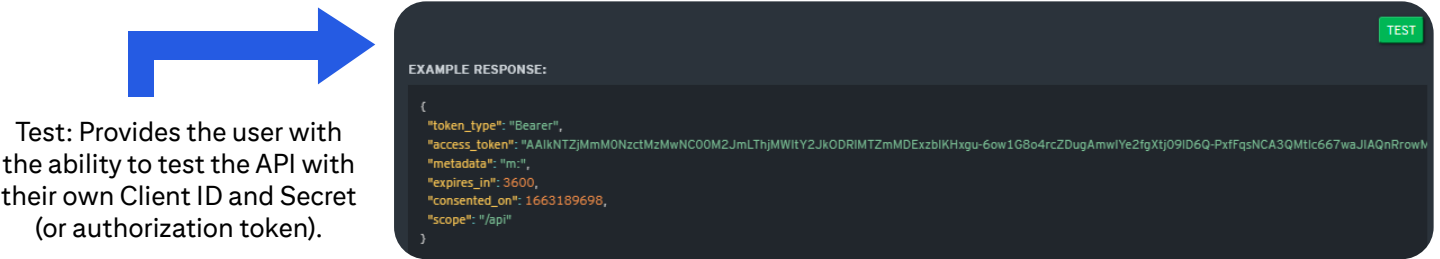


The **copy** button will copy the value of each field to the clipboard.

- 4. Details on how to make an authentication API call can be found below.



These details are available for each API and provide users with key information about the API and how to call it.

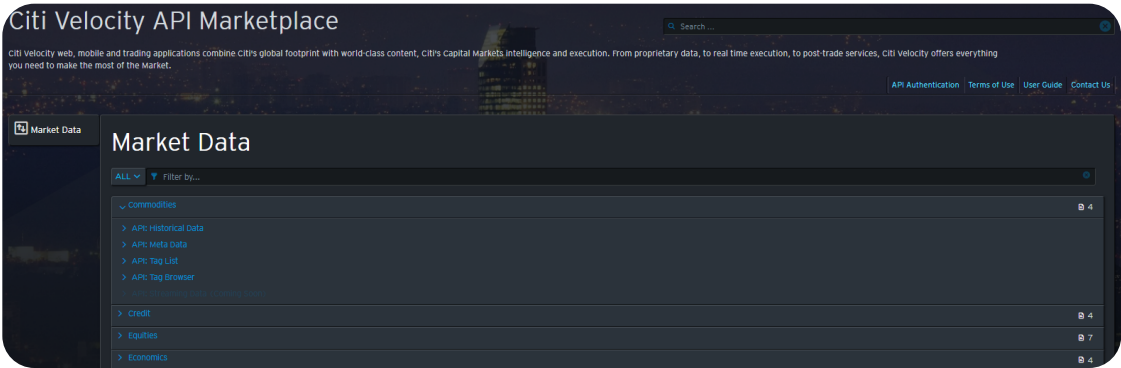


Test: Provides the user with the ability to test the API with their own Client ID and Secret (or authorization token).

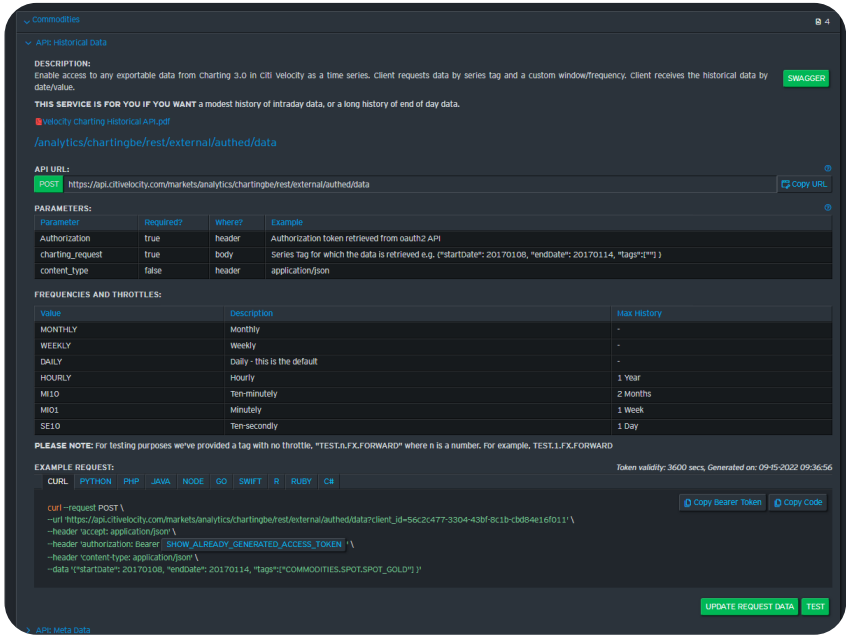
- 5. Once the user has received their Client ID and Client Secret, they need to request an authorization token. This token will be valid for 30 minutes, at which point the above step will need to be repeated.

Browsing APIs

- 6. Once the authorization token has been received, the user can start calling APIs. These can be found on the main API Marketplace screen.



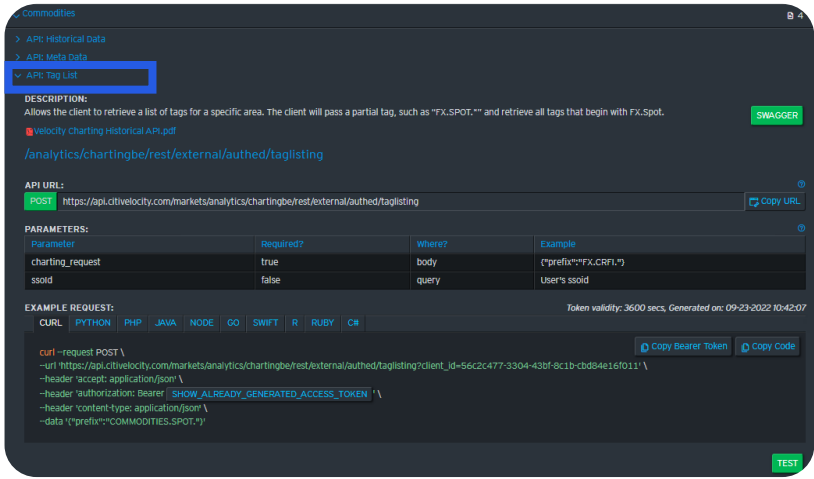
7. The user can click through to view the available APIs for a particular category (ex. Market Data APIs) and asset class (ex. Commodities, shown below).



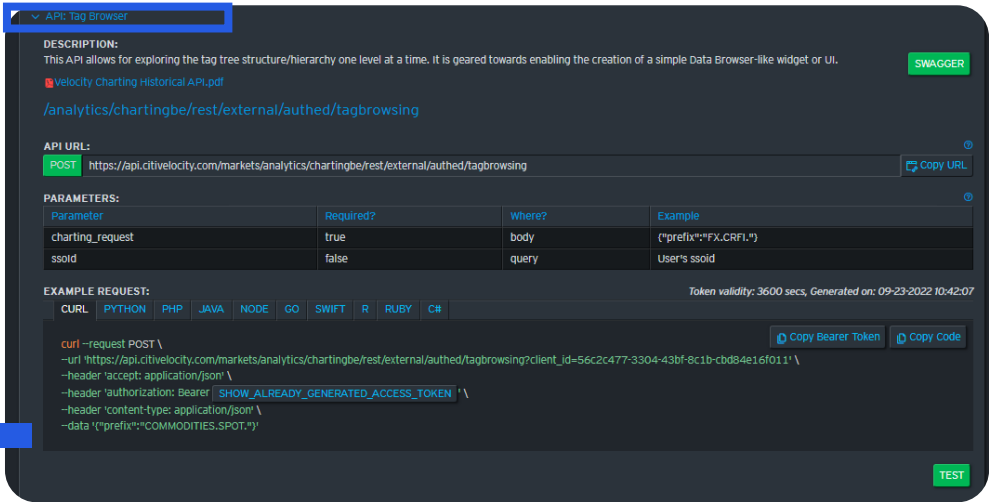
Using the information in the API details screen, the user can now code for the API calls in order to receive the data required.

The user can click through to view the details of each API similar to the authentication API detailed above.

8. The Tag List feature allows the client to retrieve a specific list of information relating to key words inputted, such as “FX.SPOT”, which will include all information with this tag.



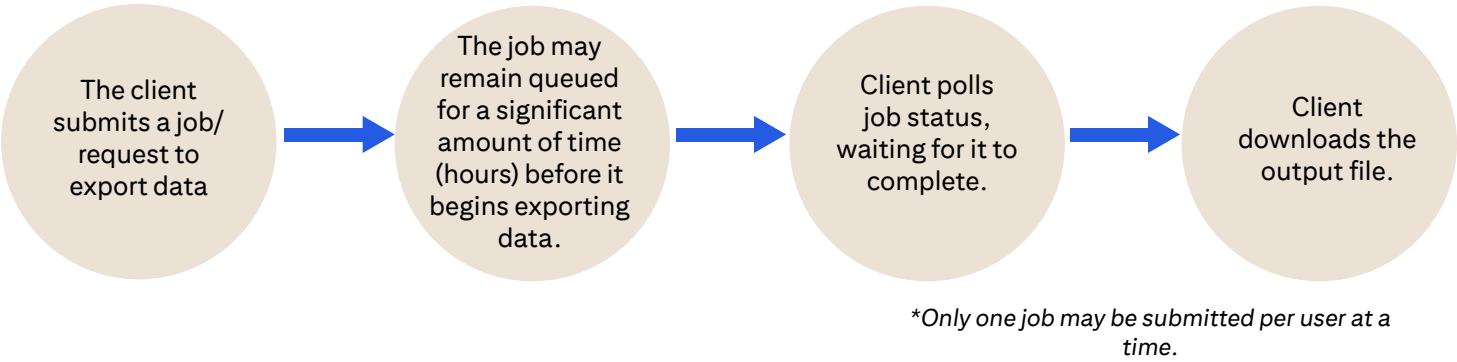
The Tag Browser allows the client to focus on specific asset classes giving historical and intraday data.



## Historical Bulk Data

The purpose of this web service is to enable clients of Citi Velocity to export large amounts of historical intraday time series data. This export service enforces the limits on the amount of history; you must pull at least one year of hourly data, at least one week of minutely data, etc.

Workflow process:



## API Status Check

<https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/authed/ibe/apistatus>

Example 1:

```
Response
{
  "api": {
    "state": "READY",
    "quotaUsed": 85370999,
    "quotaLimit": 10737418240
  },
  "status": "OK"
}
```

This is the request/response status, not to be confused with the API state. The two possible values are “OK” and “ERROR”. If there is an error then there will be a corresponding “message” field detailing the error.

## Job Submission

<https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/authed/ibe/submitjob>

Example 2:

```
Request
{
  "tags": ["FX.SPOT.USD.JPY.CITI", "FX.SPOT.USD.CAD.CITI"],
  "frequency": "HOURLY",
  "startDate": 20030101,
  "pricePoints": "C"
}
```

```
Response
{
  "jobId": "e848e585-573b-4f80-b020-eb04eee940b3",
  "status": "OK"
}
```

\*An array of between 1 and 100 distinct tags.



Job Status

https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/auth  
ed/ibe/jobstatus?jobId=jobid\_like\_above

State	Description
QUEUED	The job is waiting patiently for jobs from other users to complete
EXPORTING	The job is running - exporting the intraday data
READY_FOR_DOWNLOAD	The output file is ready for download
FINISHED	The job has completed, either successfully or due to an error

Each response will have the standard “status” field as well as the “state” field.

HTTP method for requests is GET.

Queued

```
{
  "job": {
    "position": 2,
    "state": "QUEUED"
  },
  "status": "OK"
}
```

Exporting

```
{
  "job": {
    "errors": 0,
    "successes": 1,
    "remaining": 2,
    "state": "EXPORTING"
  },
  "status": "OK"
}
```

Ready\_For\_Download

```
{
  "job": {
    "ttl": 14268,
    "state": "READY_FOR_DOWNLOAD"
  },
  "status": "OK"
}
```

Finished

```
{
  "job": {
    "succeeded": true,
    "totalTime": 17,
    "quotaUsed": 449985,
    "state": "FINISHED"
  },
  "status": "OK"
}
```

Definitions

**position** – The job’s position in the queue. The job at position 1 will be the next to run

**errors** – the number of tags which have failed due to a tag-specific error  
**successes** – the number of tags which have been exported successfully  
**remaining** – the number of tags which are still outstanding

**ttl** – number of seconds until the output file is deleted. Currently starts at 24 hours (86400)

**succeeded** – false if the job failed, in which case there will be an additional “message” field with the error.  
**totalTime** – time in seconds between when the job entered the QUEUED state and the FINISHED state.  
**quotaUsed** – size of the output file in bytes

Download Output File

The file container format is the standard “zip” format. The file name in the Content-Disposition header is

```
<login> _ <hash> .zip
|
sha256 digest of the file contents
```

The zip file contains two types of plain text files  
1. summary file–summary.txt  
2. tag-specific file: <tag> .txt

**Limits** – Limits are in place to restrict the frequency APIs can be called.  
Example 1:

```
Request/Response {
  "api": {
    "state": "READY",
    "quotaUsed": 85370999,
    "quotaLimit": 10737418240
  },
  "status": "OK"
}
```

Miscellaneous

- 1
- New Fields in Response JSON: Existing fields will never be removed or renamed, new fields can be added at any time without notice to the response JSON.
- 2
- New Files in the Output File: Existing file formats will not change, new file types can be added to the zip file at any time.
- 3
- HTTP Error Codes: Most errors are caught and handled at the JSON response level or in the summary.txt file (status “ERROR”). So the HTTP response code will usually be 200. However, it is possible that some unexpected errors (or service downtime) will result in HTTP error codes ≥ 400, and client code should handle that possibility.
- 4
- Downtime and Retry Logic: This web service can be down for up to 10 minutes at any time during the day (for maintenance). Client code must have retry logic built in to handle short periods of downtime. These short periods of downtime will not interfere with active jobs. Maintenance requiring more than 10 minutes of downtime will be scheduled for the 48-hour window between Friday 6pm and Sunday 6pm NY time. Weekend maintenance can disrupt active jobs, so it’s best to avoid this Intraday Bulk export service over the weekend, especially for large jobs.
- 5
- Testing: The amount of time required to export the intraday data is proportional to the number of points in the output file, so we recommend testing with an HOURLY frequency even if you are ultimately interested in minutely data. Just keep in mind the difference in format of the Time field in the output files.


Historical Data

This service returns historical information for the queried series tags, intended to be used to pull full daily values.

startTime & endTime format (GMT):  
hhmm (hour and minute) two digit (24) hour x  
100 + two digit minute

Value	Description	Max History
Monthly	Monthly	-
Weekly	Weekly	-
Daily	Daily	-
Hourly	Hourly	1 Year
MI01	Minutely	1 Week

Category	Sub Category
Equities	Stocks, ETFs, Indices
FX	Spot, Forward, Deposit Rate, Vol (Implied)
Rates	Sovereign, Par Swap

Intraday frequencies have a limit on the amount of history which can be pulled in a single request and there is a separate service in which bulk intraday data can be accessed. Intraday data is marked with double arrows to indicate the availability. 

*\*Hourly and finer are intraday frequencies/Daily and coarser (daily, weekly & monthly) are EOD frequencies*

Value	Sub Category
C	Closes only-this is the default
OHLC	Open, High, Low and Close

◀ One data point  
‘Y’ Values

◀ Four data points

startDate and endDate are still required in the request

Frequency	Format
Monthly	yyyyMM
Weekly	yyyyww
Daily	yyyyMMdd
Hourly	yyyyMMddHH
Ten-minutely	yyyyMMddHHm
Minutely	yyyyMMddHHmm

The ‘x’ values contains the dates and date-times

Date Field	Description
yyyy	4 digit: year (1980, 2017, etc.)
MM	2:digit: month, 0-12
yyyy	4 digit: ISO week based year
ww	2 digit: ISO week of week based year
dd	2 digit: date, 1-31
HH	2 digit: hour, 0-23
m	1 digit: ten minute value- 0-5
mm	2 digit: minute, 0-59

\* The ‘x’ & ‘c’ arrays can be empty if the tag is valid but has no data associated with it-this is not treated as an error.

Example 3: Below shows how OHLC data can be pulled.

Request

```
{
  "startDate": 20220320,
  "endDate": 20220324,
  "tags": [
    "FX.SPOT.EUR.USD.CITI"
  ],
  "pricePoints": "OHLC"
}
```

Response

```
{
  "frequency": "DAILY",
  "body": {
    "FX.SPOT.EUR.USD.CITI": {
      "x": [20220320, 20220321, 20220322,
        20220323, 20220324],
      "o": [1.0742, 1.0737, 1.0796, 1.0789, 1.0783],
      "h": [1.0778, 1.0812, 1.0825, 1.0806, 1.0818],
      "l": [1.0719, 1.0736, 1.0776, 1.0768, 1.0761],
      "c": [1.0737, 1.0796, 1.0789, 1.0783, 1.0799],
      "type": "SERIES"
    }
  },
  "status": "OK"
}
```

A series response will have the ‘x’ and ‘c’ fields

Metadata

This provides the user with timing and frequency of updates, returning the last 5 time stamps of the queried series tags, including three types of metadata per tag: tag description, first date / last date and history of series modification times.

<https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/authed/metadata>

Request

```
{
  "tags": [
    "RATES.VOL.USD.ATM_RFR.NORMAL.ANNUAL.1M.3M"
  ],
  "frequency": "EOD"
}
```

Response

```
{
  "body": {
    "RATES.VOL.USD.ATM_RFR.NORMAL.ANNUAL.1M.3M": {
      "description": "1M x 3M USD Normal Annual RFR Vol (BPS/ANNUM)",
      "modifiedTimes": [
        "2024-08-01T00:07:36Z[GMT]",
        "2024-07-31T00:07:38Z[GMT]",
        "2024-07-30T00:07:16Z[GMT]",
        "2024-07-27T00:07:12Z[GMT]",
        "2024-07-26T00:07:01Z[GMT]"
      ],
      "startDate": 20220103,
      "endDate": 20240731
    }
  },
  "status": "OK"
}
```

There are 1 to 1000 distinct tags.  
Two values are available, Intraday or EOD, where EOD is the default.

Tag List

https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/auth  
ed/taglisting

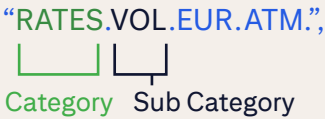
Request

```
{
  "prefix": "RATES.VOL.EUR.ATM.",
  "regex": ".*(1M).*"
}
```

Response

```
{
  "tags": [
    "RATES.VOL.EUR.ATM.BLACK.1M.10Y",
    "RATES.VOL.EUR.ATM.BLACK.1M.15Y",
    "RATES.VOL.EUR.ATM.BLACK.1M.1Y",
    "RATES.VOL.EUR.ATM.BLACK.1M.20Y",
    "RATES.VOL.EUR.ATM.BLACK.1M.2Y",
    "RATES.VOL.EUR.ATM.BLACK.1M.30Y",
    "RATES.VOL.EUR.ATM.BLACK.1M.3M",
    "RATES.VOL.EUR.ATM.BLACK.1M.3Y",
    "RATES.VOL.EUR.ATM.BLACK.1M.5Y",
    "RATES.VOL.EUR.ATM.BLACK.1M.7Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.10Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.15Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.1Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.20Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.2Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.30Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.3M",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.3Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.5Y",
    "RATES.VOL.EUR.ATM.FWDPREMIUM.1M.7Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.10Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.15Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.1Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.20Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.2Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.30Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.3M",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.3Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.5Y",
    "RATES.VOL.EUR.ATM.NORMAL.ANNUAL.1M.7Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.10Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.15Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.1Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.20Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.2Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.30Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.3M",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.3Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.5Y",
    "RATES.VOL.EUR.ATM.NORMAL.DAILY.1M.7Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.10Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.15Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.1Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.20Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.2Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.30Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.3M",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.3Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.5Y",
    "RATES.VOL.EUR.ATM.PREMIUM.1M.7Y"
  ],
  "status": "OK"
}
```

\*A tag prefix must include a category and sub category



\*A generic tag will have placeholder fields surrounded by <>  
This separation is needed when there are many securities with the same measure (ex. thousands of bonds have prices)

Backtest Only Tag

Request

```
{
  "prefix": "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E110.FRA_E110_1.O1495233639.ACTUAL",
  "tagType": "BT"
}
```

Response

```
{
  "tags": [
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E110.FRA_E110_1.O1495233639.ACTUAL",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E110.FRA_E110_1.O664065462.ACTUAL",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E28.FRA_E28_1.O69822702.ACTUAL",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E28.FRA_E28_1.O69822702.EXPECTED",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E28.FRA_E28_2.O69822702.ACTUAL",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E28.FRA_E28_2.O69822702.EXPECTED",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E28.FRA_E28_2.O69822702.PREVIOUS",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E28.FRA_E28_3.O69822702.ACTUAL",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E44.FRA_E44_1.O69822702.PREVIOUS",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E81.FRA_E81_1.O664065462.ACTUAL",
    "ECONOMICS.DJ_ECO.DJ_EUR.DJ_FRA.FRA_E82.FRA_E82_1.O664065462.EXPECTED"
  ],
  "status": "OK"
}
```

Generic Tag

Request

```
{
  "prefix": "EQUITY.STOCK",
  "tagType": "Security"
}
```

Response

```
{
  "tags": [],
  "downloadlink": "https://www.citivelocity.com/marketdata/aknetdownload/securitydetails/EQUITY.STOCK.xlsx?hash=I8k9EpVad1ImEUV1SsIlP-oDst-Thzqgo2ocYW1BRsUs=",
  "msg": "success",
  "status": "OK"
}
```

tagType

parameter tagType with value "Security", to get the ID list for given data set.

Support Prefix

```
{EQUITY.STOCK},{EQUITY.ETF},{EQUITY.EQUITY_INDEX},{EQUITY.EQIVOL.STOCK},{EQUITY.EQIVOL.ETF},{EQUITY.EQIVOL.EQUITY_INDEX},{CREDIT.BOND},{RATES.BOND}
```

Tag Browser

Allows for exploring the tag tree structure one level at a time to create a simple data browser widget or UI.

https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/auth  
ed/tagbrowsing

Request

```
{“prefix”:””}  
{“prefix”:”COMMODITIES.SPOT”}  
{“prefix”:”COMMODITIES.SPOT.SPOT_GOLD”}
```

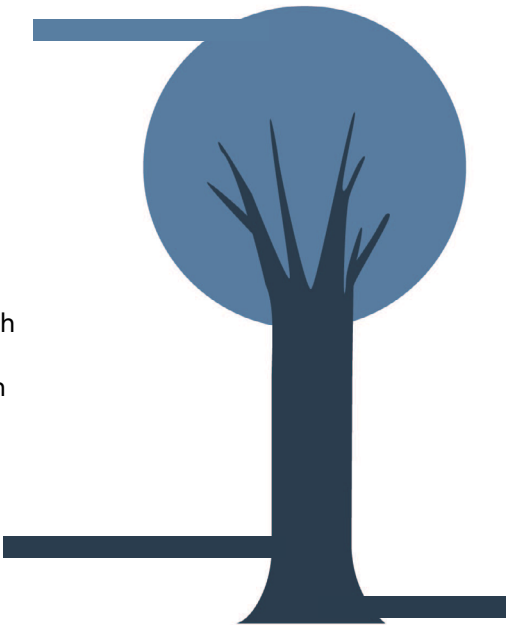
Response

Leaf Node (“COMMODITIES.SPOT.SPOT\_GOLD”)

```
{  
  “fields”: {},  
  “description”: “Spot Gold Index Data”,  
  “leaves”: [],  
  “status”: “OK”  
}
```

Interior Node

```
{  
  “header”: “Product”,  
  “fields”: {  
    “OIL_PRICE_BRENT”: “Brent Spot Month  
(ICE)”,  
    “OIL_PRICE_NYMEX”: “WTI Spot Month  
(NYMEX)”,  
    “SPOT_GOLD”: “Gold (Spot Price)”,  
    “SPOT_SILVER”: “Silver (Spot Price)”  
  },  
  “leaves”: [  
    “OIL_PRICE_NYMEX”,  
    “SPOT_GOLD”,  
    “SPOT_SILVER”,  
    “OIL_PRICE_BRENT”  
  ],  
  “status”: “OK”  
}
```



Root Node

```
{  
  “header”: “Category”,  
  “fields”: {  
    “COMMODITIES”: “Commodities”,  
    “CREDIT”: “Credit”,  
    “ECONOMICS”: “Economics”,  
    “EM”: “Emerging Markets”,  
    “EQUITY”: “Equities”,  
    “FX”: “FX”,  
    “MUNI”: “Municipals”,  
    “RATES”: “Rates”,  
    “SECURITIZED”: “Securitized”  
  },  
  “leaves”: [],  
  “status”: “OK”  
}
```

UI Implementation

The tree structure should be cached on the client side with an expiration of 12 hours. Different users will have access to different tags, therefore cache is per user.

**Equity (eq) Identifier**- With this API you can convert between Citi Equity Identifiers and other “street” identifiers.

https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/auth  
ed/citiids/from  
https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/auth  
ed/citiids/to

/from - Sample Request

```
{“ids”: [123, 306888, 56280, 92141]}
```

There are up to 10,000 integers of equity identifiers available.

/from - Sample Response

```
{  
  “body”: [ null, {  
    “id”: 306888,  
    “ric”: “IBM.N”,  
    “mic”: “XNYS”,  
    “ets”: “IBM”,  
    “bbt”: “IBM”,  
    “isin”: “US4592001014”  
  }, {  
    “id”: 56280,  
    “ric”: “SN.L”,  
    “mic”: “XLON”,  
    “ets”: “SN.”,  
    “bbt”: “SN/”,  
    “isin”: “GB0009223206”  
  }, {  
    “id”: 92141,  
    “ric”: “.SPX”,  
    “ets”: “SPX”,  
    “bbt”: “SPX”,  
    “isin”: “US78378X1072”  
  } ],  
  “status”: “OK”  
}
```

Each index in the response body corresponds to the same index in the request. If the request Citi Id is not identified then the value will be null.

Field	Description
id	Citi Equity Identifier
ric	Reuters Instrument Code
mic	Market Identifier
ets	Exchange Traded Symbol
bbt	Bloomberg Ticker
isin	International Securities Identification Number

Always guaranteed to be present in the response

Will often be missing for Indices



```
/to - Sample Request

{
  "queries": [
    {
      "productType": "STOCK",
      "identifier": "IBM.N",
      "identifierType": "RIC"
    },
    {
      "productType": "STOCK",
      "identifier": "IBM",
      "identifierType": "BBT",
      "mic": "XNYS"
    },
    {
      "productType": "STOCK",
      "identifier": "IBM",
      "identifierType": "BBT",
      "primaryOnly": true
    }
  ]
}
```

```
/to - Sample Response

{
  "ids": [ 306888, 306888, 306888 ],
  "status": "OK"
}
```

Each index in the response corresponds to the same index in the request. The value at a given index in the response will be null if the corresponding query did not result in exactly one Citi Id.

A list of up to 1,000 query objects. Each query should identify exactly one instrument and exchange combination. This can be achieved three ways:

- 1. Use a Ric Identifier with a built in exchange.
- 2. Specify Stocks or ETFs using the mic field.
- 3. Use ‘primaryonly’ field to specify the primary exchange.

Field	Optional	Description
productType	N	STOCK, ETF, or INDEX, STOCKVOL, INDEXVOL, ETFVOL
identifier	N	
identifierType	N	RIC, BBT, ETS OR ISIN
mic	Y	
primaryOnly	Y	filter for primary exchanges when true

*\*Bloomberg Ticker is not consistent in all regions, use caution when inputting a ticker.*

Miscellaneous

Using one of the following exchanges for stocks or ETFs:

ARCX, XAMS, XASE, XASX, XBRU, XCSE, XETR, XFRA, XHEL, XHKG, XJPX, XLIS, XLON, XNAS, XNCM, XNGS, XNMS, XNYS, XPAR, XSTO, XTKS

User Defined Tags

We are offering user defined tags for easier usage of equity and credit products. Tags will be in format UDT.ADMIN.<Tag> where the Citi ID of equity/credit instruments replaced by RIC/BBT.

Support Data Set
EQUITY.STOCK
EQUITY.EQUITY_INDEX
EQUITY.EQIVOL.ETF
EQUITY.EQIVOL.STOCK
EQUITY.EQIVOL.EQUITY_INDEX
CREDIT.CDS

Sample Charting Tag	User Definer Tag
Equity.EQIVOL.EQUITY_INDEX.92141.EQIVOL_DELTA.STRIKE_C25.2W.CITI	UDT.ADMIN.EQUITY.EQIVOL.EQUITY_INDEX..SPX.EQIVOL_DELTA.STIKE_C25.2W.CITI
EQUITY.EQUITY_INDEX.92141.LEVEL.REUTERS	UDT.ADMIN.EQUITY.EQUITY_INDEX..SPX.LEVEL.REUTERS
EQUITY.STOCK.306888.VOLUME.REUTERS	UDT.ADMIN.EQUITY.STOCK.IBM.N.VOLUME.REUTERS
CREDIT.CDS.MKS-M+SPLC.SNRFOR.EUR.MM14.5Y.PAR.BLENDED	UDT.ADMIN.CREDIT.CDS.MARSPE.SNRFOR.USD.XR14.5Y.PAR.BLENDED
	(Special character “.” will be treated as underscore while processing)

Example:

```
Request

{
  "startDate": 20170323,
  "endDate": 20170323,
  "tags": [
    "UDT.ADMIN.EQUITY.STOCK.IBM UN.VOLUME.REUTERS"
  ],
  "startTime": 0,
  "endTime": 2359,
  "frequency": "HOURLY"
}
```

Miscellaneous

Downtime and Retry Logic

This web service can be down for up to 10 minutes at any time during the day. Client code must have retry logic built in to handle short periods of downtime. Maintenance requiring more than 10 minutes of downtime will be scheduled for the 48 hour window between Friday 6pm and Sunday 6pm NY time.

gzip

Please verify that your client accepts the gzip Content-Encoding. This will greatly reduce network traffic.

Corporate Proxy

To check if you’re behind a corporate proxy on Windows, execute:

Start >> cmd >> ping www.citivelocity.com  
“Request timed out.” indicates the presence of a proxy

Sample Python

This sample code makes use of the “requests” HTTP library for Python. Installation instructions [here](#). Help with a corporate proxy [here](#).

```
>>> import requests
>>> url =
'https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/aut
hed/data?client_id=your_client_id'
>>> payload = {'startDate': 20170108, 'endDate': 20170114, 'tags':
['COMMODITIES.SPOT.SPOT_GOLD']}
>>> headers = {'authorization': 'Bearer your_access_token'}
>>> r = requests.post(url, json=payload, headers=headers)
>>> r.text
```

Sample Curl

The curl command line utility can be used to quickly test connectivity and the request JSON format. Below is a sample curl command corresponding to the first example above. See [here](#) for help with a corporate proxy.

```
curl -H 'Content-Type: application/json' -H 'authorization: Bearer
your_access_token' -X POST -d '{"startDate": 20170108, "endDate": 20170114,
"tags":["COMMODITIES.SPOT.SPOT_GOLD"]}'
'https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/aut
hed/data?client_id=your_client_id'
```

Sample R

```
library(httr)
library(jsonlite)

client_id <- "your_client_id"
access_token <- "your_access_token"
endpoint <-
"https://api.citivelocity.com/markets/analytics/chartingbe/rest/external/aut
hed/data"

url <- paste(endpoint,"?client_id=",client_id,sep="")

req <- '{"startDate": 20170108,"endDate":
20170114,"tags":["COMMODITIES.SPOT.SPOT_GOLD"]}'

resp <- POST(url, body=req, add_headers("Content-Type" = "application/json",
"authorization" = paste("Bearer ", access_token, sep="")))

resp_content <- content(resp, "text")

resp_content
```

Streaming API

Refers to the action of pushing out updates for the active subscriptions. This Service would most accurately be called the “streaming API”.

Sample Client

A sample client written in Java 8 can be downloaded [here](#). You may find it helpful to reference the sample code as you read the “Websocket API” section below. The sample client follows Maven conventions, and more specifically is an Eclipse Maven project. The tgz file can be imported directly into Eclipse by:

1. Right click on  
2. Import  
3. Select archive file:
- ➡➡
- Package Explorer  
Existing Projects into Workspace

Sample code:

cva.logging.level	One of the java.util.logging.level levels
cva.frequency	One of the MIO1 (minutely), SE01 (secondly). Defaults to MIO1
java.util.logging.SimpleFormatter.format	ex. . "%1\$tF %1\$tT %4\$s %2\$s %5\$s%6\$s%n"
cva.eventBus.threadCount	# threads used to deliver updates, defaults to 1
proxy.host	Hostname of corporate proxy
proxy.port	Port for Corporate proxy, default to 8080

Text vs Binary Messages

The Websocket protocol makes a distinction between text and binary messages. This API uses text messages for control purposes in both directions: server ↔ client. All text messages are serialized using JSON.

Binary messages are used exclusively to deliver the actual updates from the server to the client: Server →<sup>binary</sup> client. No binary messages should be sent from the client to the server – any such messages are treated as a fatal error and the connection will be closed.

Connection Control Messages

There are three connection messages: CONN\_READY, CONN\_ERROR and KEEP\_ALIVE

```
CONN_READY      {"type":"CONN_READY", "id":"<UUID>"}
Will be sent shortly after a new connection is established

CONN_ERROR      {"type":"CONN_ERROR", "message":"<some error message>"}
Will be generated during the connection initialization, due to incorrect login, pass-code, duplicate login or IP address

KEEP_ALIVE      {"type":"KEEP_ALIVE"}
Messages will be sent periodically to prevent the connection from being closed due to inactivity
```

Subscription Control Messages

There are four messages used to control the set of active subscriptions: SUB, SUBACK, UNSUB, UNSUBACK

Control Message	Code
SUB	{“type”:”SUB”, “id”:<msg id>, “tag”:”<tag>”, “pricePoint”:”<price point>”}
SUBACK	{“type”:”SUBACK”, “id”:<msg id>, “status”:”OK”, “subid”:<subscription id>} {“type”:”SUBACK”, “id”:<msg id>, “status”:”ERROR”, “message”:”<error msg>”}
UNSUB	{“type”:”UNSUB”, “id”:<msg id>, “subid”:”<subscription id>”}
UNSUBACK	{“type”:”UNSUBACK”, “id”:<msg id>, “status”:”OK”} {“type”:”UNSUBACK”, “id”:<msg id>, “status”:”ERROR”, “message”:”<err msg>”}

**SUB Messages**– Are sent by the client to request a subscription to a new time series identified by the specified tag and price point.

Categories	
id	Mandatory signed int32 value which acts as a correlation id between a SUB message and the corresponding SUBACK messages (see below). The client is responsible for choosing the ids. The server does not use this message id for anything beyond echoing it back in the SUBACK message.
tag	Mandatory. E.g. FX.SPOT.USD.JPY.CITI
pricePoint	Optional. Will be defaulted to CLOSE for series which support OHLC (see below for details). Conversely, must be omitted or null or CLOSE for series which do not support OHLC.

**SUBACK Messages**– Message sent after the subscription has been established. The SUBACK message will always contain the message id from the corresponding SUB message, along with a status field which will have value OK or ERROR. If the status is ERROR then there will be an error message field.

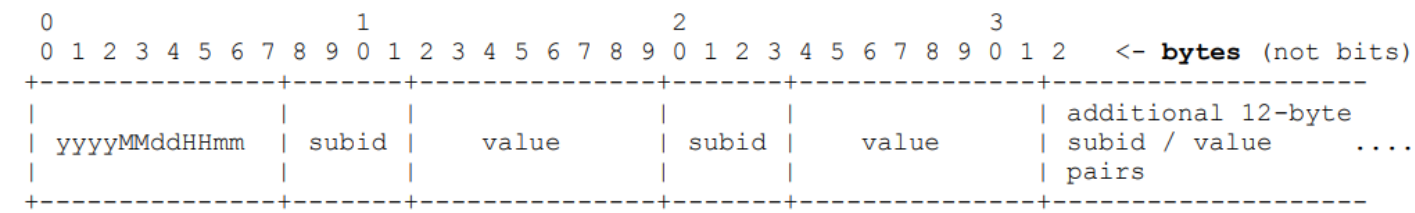
subid	A signed int32 value chosen by the server to identify the subscription. This subscription id can be used to subsequently unsubscribe (see below). The subscription id and message id are independent concepts.
-------	--

**UNSUB Messages**– Are sent by the client to unsubscribe from a time series identified by the specified subscription id.

id	Same logic as the message id in a SUB message (see above). Will be echoed back in the corresponding UNSUBACK message (see below).
----	---

Binary Format

The updates are sent in batches in a simple binary format:



Each batch starts with an 8-byte timestamp followed by a sequence of 12-byte pairs. Each pair is a 4-byte subscription id and then an 8-byte IEEE double value.

The number of subid/value pairs can be calculated as follows: the Websocket protocol provides the total length of the binary message, so #pairs = (length – 8) / 12

Delivery

In this section we will refer to minutes in the real world, i.e. the actual clock time as “real-clock minutes”. We will refer to the yyyyMMddHHmm formatted minutes at the beginning of each batch as “HHmm minutes”.

Batches will typically start being delivered within a couple of seconds of rolling into a new real-clock minute. As the word “batch” implies, there can be more than one batch per real-clock minute for the same HHmm minute.

Example:

This client has subscribed to OPEN, HIGH, LOW, and CLOSE for one tag, say FX.SPOT.USD.JPY.CITI, with subscription ids 51, 52, 53, and 54 respectively. For example: realclock minute has recently rolled to 13:05. In this situation, the client will typically receive three batches shortly after rolling to 13:05.

USD. JPY, OPEN (51)	13:03	13:04	13:05
USD. JPY, HIGH (52)		51,110.76	51,110.90
USD. JPY, LOW (53)	52,110.76	52,110.85	
USD. JPY, CLOSE (54)	53,110.69	53,110.74	
	54,110.75	54,110.85	
	44 bytes	56 bytes	20 bytes

All three of the batches in this example are arriving at almost the same real-clock time shortly after we’ve rolled into real-clock minute 13:05, but each of the three is for a different HHmm minute.

After one minute at 13:06:

USD. JPY, OPEN (51)	13:04	13:05	13:06
USD. JPY, HIGH (52)		51,110.90	51,110.89
USD. JPY, LOW (53)	52,110.85	52,110.90	
USD. JPY, CLOSE (54)	53,110.74	53,110.81	
	54,110.85	54,110.84	
	44 bytes	56 bytes	20 bytes

The restated values are in italics.

Miscellaneous

Streaming Data Availability

See column F in [Intraday Tags](#) for a breakdown of which tags having streaming data available via this service.

Mock Data for Testing

Testing with real tags like FX.SPOT.USD.JPY.CITI can be tricky (e.g. data is rarely published 24x7), so we have provided mock tags MOCK.n where n is an integer between 1 and 9999. These mock tags support OHCL, and will always publish a value each minute. The value for MOCK.n will be:

Price Point	Value
OPEN	$n$
HIGH	$n + 0.5$
LOW	$n - 0.5$
CLOSE	$n$

Testing: The following is excluded from the “max calls per item” and can be used for testing

TEST . n . FX . FORWARD,  $n \in \{0 \dots 999\}$

Flat Files Download Web Service

Overview


The purpose of this web service is to enable Citi Velocity clients to export large amounts of historical intraday time series data for all data-series (tags) in a dataset. Data is made available in zip files containing up-to 1 calendar month of historical intraday data on a go forward basis.

Terminology

Tag

A hierarchial namespace (or path) which identifies a time series in Citi Velocity Charting, e.g.

FX.SPOT.USD.JPY.CITI

The tag structure can be explored using the Data Browser in Charting. Note however that only a relatively small number of series have intraday data. Series with intraday data are marked with an  icon in the Data Browser. A detailed listing of series with intraday data can be found in the Daily and Intraday OHLC Dataset List document under help menu of Charting 3.0. We are working to onboard additional intraday data.

Choosing The Correct Service

Citi Velocity offers four different services which provide intraday time series data. Please select the service which best fits your needs. The term “amount” means “length of history” (e.g. 6 months) below:

Use Case	Citi Velocity Charting Service
I want to recieve live minutely update going forward.	Streaming API
I want to query for modest amounts* of intraday data (or any amount of EOD/daily data).	Historical API
I want to bulk export large amounts of intraday data for a selected set of tags	Bulk Data API
I want to bulk export large amounts of intrday data for all tags in a dataset	Flat Files API

\*The historical API has the following limitaions on how much history can be pulled at a given frequency.

Frequency	Description	Max History
Hourly	Hourly	1 Year
MI01	Minutely	1 week

For example, if you want to pull the hourly data for some series going back 6 months, then you should use the Historical service. If you want to download greater than 1 month of Minutely data for a limited set of tags then you should use the Bulk Data API. If you want to download greater than 1 month of Minutely data for all tags in a dataset (e.g. Minutely data for all available FX Spot data currency pairs) then you should use the Flat Files API.

Workflow

- The high-level workflow for this Intraday Bulk Flat Files Download series is:
1. Use Flat Files API to get a list of download links for selected dataset and time range.
  2. Download zip file from each download link.

Download Limit : Each file can only be downloaded at most 5 times.

Request Format

The request is serialized using JSON. HTTP method for requests is POST. Here is an example of the simplest possible request:

Request

```
{
  "dataset": "FX.SPOT",
  "endYYYYMM": 202312,
  "frequency": "MIO1",
  "startYYYYMM": 202307
}
```

Datasets Available for Download Using Flat Files API

Dataset Name	Description	Start Date	Supported Frequencies	OHLC Support
FX.SPOT	FX Spot	2000 01	MIO1/ HR01	Y
FX.VOL	FX Implied Volatility	2000 01	MIO1/ HR01	N
FX.FORWARD	FX Forward	2000 01	MIO1/ HR01	Y
RATES.VOL	Rates Volatility	2016 06	MIO1/ HR01	N
RATES.TSY	Rates Treasury OTR	2016 06	MIO1/ HR01	N
RATES.TIPS	Rates Tips	2020 11	MIO1/ HR01	N
RATES.SPREAD_OPTIONS	Rates Single LookSpread Option	2022 06	MIO1/ HR01	N
RATES.SOV	Rates Sovereign	2016 06	MIO1/ HR01	N
RATES.SOV_CMT	Rates Sovereign CMT	2020 05	MIO1/ HR01	N
RATES.XCCY_OIS_SWAP	Rates Cross Currency OIS Swaps	2019 03	MIO1/ HR01	N
RATES.SWAP_LIBOR	Rates Swap (Libor)	2016 06	MIO1/ HR01	N
RATES.OIS	Rates OIS Swaps (RFR)	2018 02	MIO1/ HR01	N
RATES.FRA_OIS	Rates FRA/OIS	2020 11	MIO1/ HR01	N
RATES.FRA	Rates FRA	2022 03	MIO1/ HR01	N
RATES.FORWARD	Rates Forward	2021 06	MIO1/ HR01	N
RATES.MIDCURVES	Rates MidCurves	2022 06	MIO1/ HR01	N
RATES.OIS_MEETING	Rates OIS Meeting	2020 11	MIO1/ HR01	N
RATES.SSA	Rates SSA EUR	2022 04	MIO1/ HR01	N

Frequency

Support MIO1 as known as Minutely data and HR01 as known as Hourly data.

Response Format

The response is serialized using JSON. Here is an example of a possible response:

Response

```
{
  "body": {
    "dataset": "FX.SPOT",
    "description": "FX.SPOT",
    "frequency": "MIO1",
    "details": [
      {
        "link": "https://uat.citivelocity.com/marketdata/aknetdownload/dataset/FX.SPOT/MIO1/FX.SPOT.072023.zip?hash=wzQGOJ40sSI8sh1L1eEtffPkhHVes-fHUxQFTper3cM=",
        "filename": "FX.SPOT.072023.zip",
        "limit": 5
      },
      {
        "link": "https://uat.citivelocity.com/marketdata/aknetdownload/dataset/FX.SPOT/MIO1/FX.SPOT.082023.zip?hash=PXOgKUe5mqj5nlMH8g1avPkhHVes-fHUxQFTper3cM=",
        "filename": "FX.SPOT.082023.zip",
        "limit": 5
      }
    ],
    "status": 200,
    "message": null
  }
}
```

Link

The link in the response contains a hash that uniquely identifies the request and is regenerated for every request. The link is set to expire after a 24-hour period of the original request to obtain the links.

Limit

“Limit” parameter indicates the remaining number of downloads available for the file. We allow the same file to be downloaded a max of 5 times. Please be careful not to interrupt the download process, as the limit usage is recorded when download process starts.

Status

200	Success
400	Incorrect parameters
401	Not entitled to the specified date set via API
408	Link has expired
429	Exceeded max limit for download

Message

Message will be given in the response body if request date range exceeds supported date range. Information of available date range for given date set will be provided as reference.

Output File Format

The file container format is the standard “zip” format. The zip file contains multiple plain text files which are tag-specific files (.txt).



Example

Time	OPEN	HIGH	LOW	CLOSE		
202307010000	119.720000			119.720000	119.720000	119.720000
202307010001	119.720000			119.720000	119.720000	119.720000
202307010002	119.720000			119.720000	119.720000	119.720000
202307010003	119.720000			119.720000	119.720000	119.720000

Time format is yyyyMMddHHmm, and all times are in GMT.

Get Data Set and Date Range Info

https://api.citivelocity.com/markets/cva-chartingbe/bulk/getUrl?getDatasetInfo=true

getDatasetInfo

With this parameter getDatasetInfo with value “true” given in url, will return user’s entitled data set list with supported date range info.

Request Format

HTTP method for requests is POST. Request body could be null when “getDatasetInfo=true” given in url.

Response Format

```
{
  "body": {
    "RATES.VOL": [
      {
        "freq": "MIO1",
        "maxDate": "202308",
        "minDate": "201606"
      }
    ],
    "FX.SPOT": [
      {
        "freq": "MIO1",
        "maxDate": "202308",
        "minDate": "200001"
      },
      {
        "freq": "HR01",
        "maxDate": "202311",
        "minDate": "200001"
      }
    ],
    "status": 200,
    "message": null
  }
}
```

Miscellaneous  
Downtime and Retry Logic

This web service can be down for up to 10 minutes at any time during the day (for maintenance). Client code must have retry logic built in to handle short periods of downtime. These short periods of downtime will not interfere with active jobs. Maintenance requiring more than 10 minutes of downtime will be scheduled for the 48-hour window between Friday 6pm and Sunday 6pm NY time. Weekend maintenance can disrupt active jobs, so it’s best to avoid this Intraday Bulk Flat Files Download service over the weekend, especially for large jobs.

The curl command line utility can be used to quickly test connectivity and the request JSON format. Below is a sample curl command to submit a job. See [here](#) for help with a corporate proxy.

Sample curl to get download link

```
curl --location --request POST 'https://api.citivelocity.com/markets/cva-chartingbe/bulk/getUrl?client_id=your_client_id' \
--header 'Authorization: Bearer your_access_token' \
--header 'Content-Type: application/json' \
--data-raw '{"dataset": "FX.VOL","endYYYYMM": 202302,"frequency": "MIO1","startYYYYMM": 202301}'
```

Sample curl to get dataset info

```
curl --location --request POST 'https://api.citivelocity.com/markets/cva-chartingbe/bulk/getUrl?client_id=your_client_id&getDatasetInfo=true' \
--header 'Authorization: Bearer your_access_token' \
--header 'Content-Type: application/json' \
```

# Your Questions Answered

## How do I retrieve data using API?

Each successful data extraction requires two steps:

**Step 1** – Generation of authentication Token. Using the Client Id and Client Secret, generate an authentication token as explained in Step-5.

### Step 2 – Make a Data Call.

While you make a data call, pass the token, generated in Step-1 above. Refer various programming languages offered on the portal. Please note you need to insert the token after the word ‘Bearer’. Do not omit the word ‘Bearer’ from the code. Do not forget to leave a space between the word ‘Bearer’ and token value.

## Why is the authentication token only valid only for one hour?

For your security the authentication token is only valid for one hour to prevent third parties from leveraging your session while it is inactive.

## I do not see the data I need in these APIs. How can I request additional API offerings?

Requesting additional API offerings can be done by reaching out to your Citi Velocity Sales Representative or the regional Citi Velocity Sales groups.

## Historical Bulk Questions:

### Why can’t I use Historical Data export service to export short histories?

The Historical service is simpler, more flexible to use for shorter histories.

### Can I get a listing of all tags?

We provide a tag listing API found on the Historical API section.

### Why is the file deleted after 24 hours?

Output files tend to be very large and in order to save space they need to be deleted after use.

## Historical Questions:

### Why do I need to provide prefix separately from the regex?

For accuracy and speed, we need to know the appropriate prefix since there are millions of possible tags.

### Why not use the standard symbols for equities?

We require that the identifier specifies both the instrument and the exchange, since there is no universal standard.

### Given the concurrency limit, is there any benefit to multi-threading the requests against the same API?

Two request threads might give a minor boost. Using more than two threads will not yield any improvements and may lead to requests timing out.

## Streaming Questions:

### Why are there 8 values in a batch for a client who has subscribed to OPEN, HIGH, LOW and CLOSE?

Values from the previous minute are restated for series with OHLC data.

### Why so some values in the batch get restated?

The series with OHLC data are the ones Citi generates or receives a significant number of ticks per minute. Those ticks are coming from an external system with some delay. This delay means that we will receive some of the ticks meant for HHmm minute T during real clock minute T+1. If we were going to use a fixed delay then it would have to be something conservative, for example 10 seconds, which introduces a 10 second delay by default. A more accurate approach would be to not wait, but send out a provisional value as soon as we roll into T+1 and then send the restated value a minute later.

