FACTSET ⟩ SEE THE ADVANTAGE

FactSet Tick History - API

Programmer’s Manual and Reference Version
1.2A

May 24th, 2023

Notice

This manual contains confidential information of FactSet Research Systems Inc. or its affiliates ("FactSet"). All proprietary rights, including intellectual property rights, in the Licensed Materials will remain property of FactSet or its Suppliers, as applicable. The information in this document is subject to change without notice and does not represent a commitment on the part of FactSet. FactSet assumes no responsibility for any errors that may appear in this document.

Revision History

Effective Date	Version Number	Description	Changes made
05/24/2021	1.0A	Initial release	
10/20/2022	1.1A	Enhancements	
03/22/2023	1.2A	Description change	Updated number of tickers supported.

Table of Contents

Document Organization and Audience

This user guide provides information regarding the Tick History RESTful API. The API delivers Tick History files to the user as per the user requirement(s) and set parameters

Tick History RESTful API leverages AWS to deliver tick history files. This ensures scalability and stability to deliver large volume of data. The user can request data from 2012-01-01 to current date.

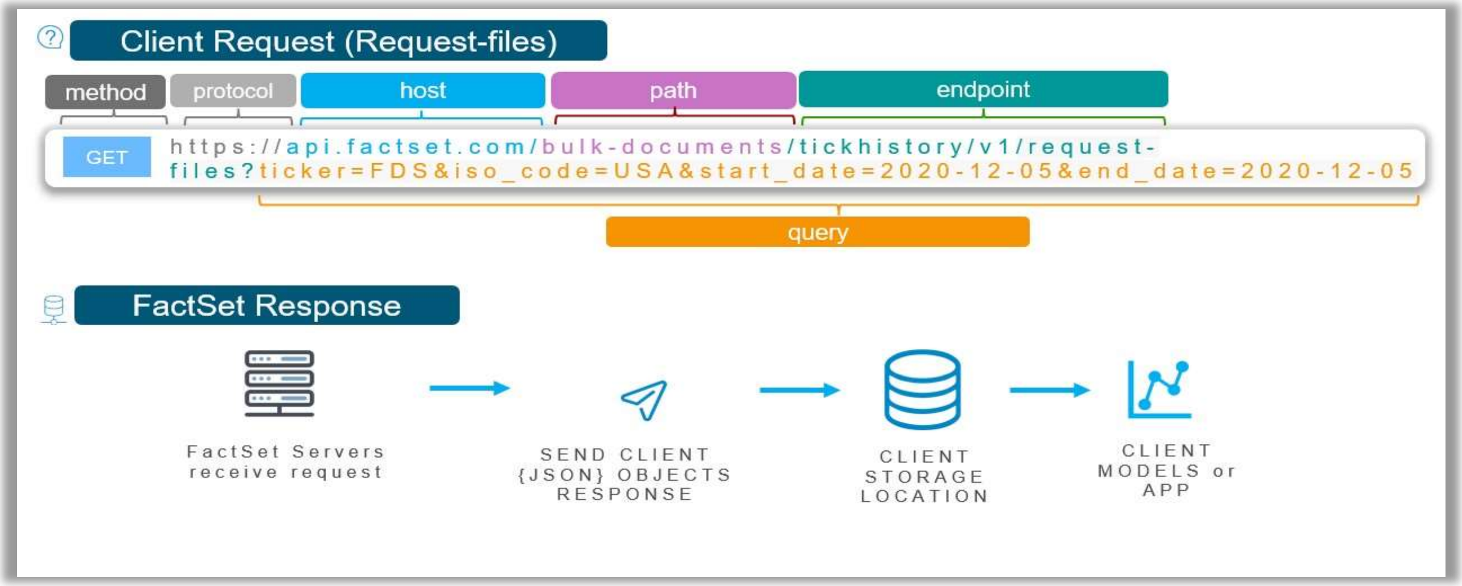
- Chapter 1 – Introduction
- Chapter 2 – FactSet Tick History RESTful API Functionality and Benefits
- Chapter 3 – Access and Security
- Chapter 4 – Requests
- Chapter 5 – Sample Requests
- Chapter 6 – Response Data Fields
- Chapter 7 - Error Response
- Chapter 8 - Options Identifiers
- Appendix

Chapter 1 Introduction

FactSet Tick History is an archive of historical tick data drawn from our real time content. Data coverage goes back as far as 2012, including OTC and exchange-traded instruments, from more than 200 Exchanges. Clients get access to Global historical Tick data across all asset classes.

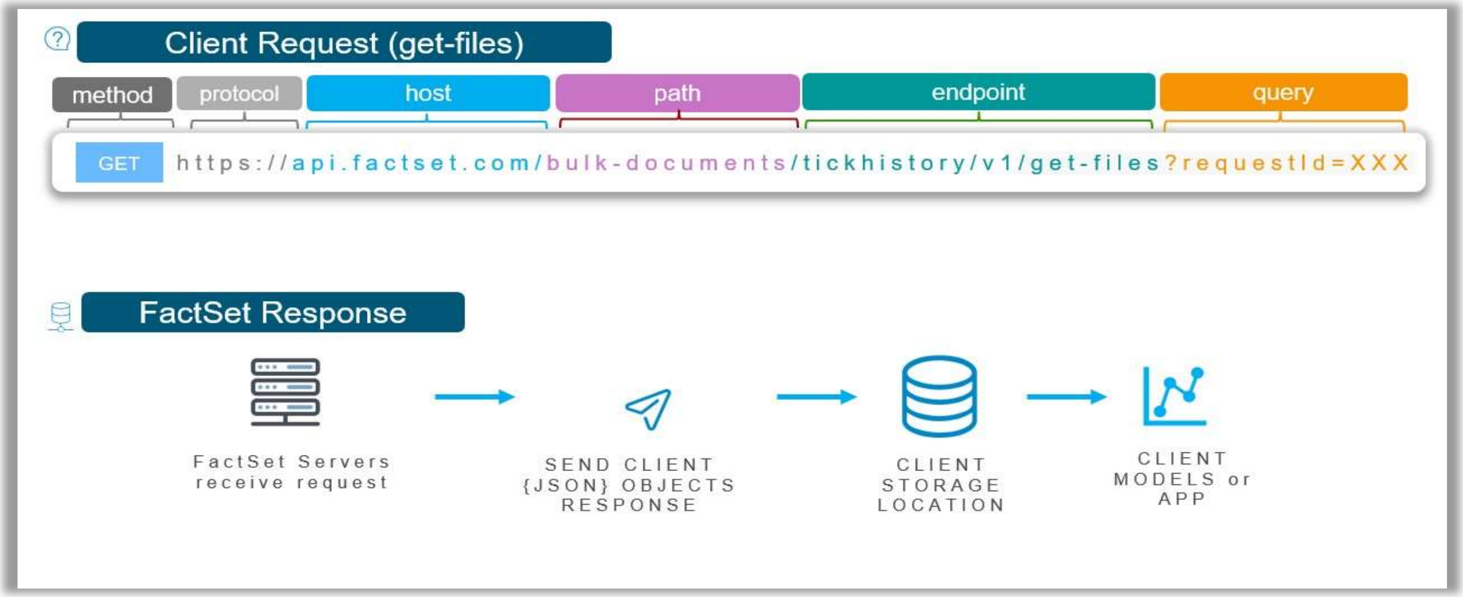
1.1 FactSet Tick History RESTful API

FactSet Tick History RESTful API provides an efficient programmatic access to the full content of the product via the standard HTTPS protocol. Requests made for a list of Tickers or by Exchange (Product) code, fields, and time periods. Data is delivered as links to the downloadable files produced based on user's request.



```
https://api.factset.com/bulk-documents/tickhistory/v1/request-files?ticker=FDS&iso_code=USA&start_date=2020-12-05&end_date=2020-12-05
```

Figure 1 FactSet Tick History RESTful API (Request-files)



```
https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=XXX
```

Figure 2 FactSet Tick History RESTful API (get-files)

1.2 Authentication

Tick History RESTful API provides easy access to historical tick data via the standard HTTPS protocol. All services are authenticated by a central source, so the same username and password works for all services. In addition, naming conventions have been standardized to simplify the consumption of multiple services.

To authenticate to the Tick History API FactSet leverages API Keys using Basic HTTP Authentication Scheme described in RFC 7617. Click here for more details on Authentication.

To generate an API key, go to https://developer.factset.com

- Logon with your FactSet .NET ID
- Go to https://developer.factset.com/manage-api-keys
- Select your **machine** account - this must be done to grant you access.
- Select “Generate new API key”
- Provide a description of your API Key (Tip: Describe IP range).
- Specify your required IP range and select ‘Add’. (The ‘From IP’ value will auto populate, while ‘To IP’ can be left blank.) You can edit this API key IP range in the future
- Select ‘Create’
- Copy your new ‘API Key’, this will not be retrievable again. Copy into a safe location for future reference, if needed.
- Please use USERNAME-MACHINE SERIAL along with the API key generated in the previous step for authentication.

1.2.1 Issuing a request with an API key

The specifics of an API request differ by the FactSet RESTful API being used, but the authentication remains the same. A valid Authorization header must be provided using Basic HTTP Authentication Scheme. This scheme requires a username and password. The FactSet credentials (username and serial separated by a single hyphen “-“ character) forms the username and the API key is the password. This username and password concatenated by a single colon “:“ character must be Base64 encoded and included with the Authorization header for all API requests.

Below is an example of GET request to the FactSet RESTful API using “usernameserial:XXXXXXXXXXXXXXXXXXXXXXXXXXX” as the credentials.

```
$ printf "username-serial:XXXXXXXXXXXXXXXXXXXXXXXXXXX" | base64
dXNlcm5hbWUtc2VyaWFsOlhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWA==
 $ curl -X GET "https://api.factset.com/bulk-documents/tickhistory"
-H "Accept: application/json"
-H "Authorization: Basic dXNlcm5hbWUtc2VyaWFsOlhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWA=="
```

If the credentials are not valid, the server will respond with a 401 HTTP status code (Unauthorized).

1.3 FactSet Tick History RESTful API Service

The Tick History RESTful API generates output files with data based on client specified input parameters. The input parameters include universe specification and date ranges for which the data will be generated. When the API request with specified input parameters has completed the output files will be made available back to the users through a secure URL to the location where the files are stored. For large universes and date ranges the output files will be chunked based on size (each file will be max 128MB, files are compressed after they are chunked so the compressed file size may differ) and multiple secure links will be returned by the API for the various files. Each secure link will be valid for up to 12 hours.

Chapter 2 FactSet Tick History RESTful API Functionality and Benefits

The FactSet Tick History RESTful provides the following features to applications:

- Synchronous access using a standard HTTP RESTful service.
- Stateless requests (No sticky cookies).
- Platform independence.
- Zero-based installation.
- Seamless failover.

2.1 Synchronous Access

Synchronous data access simplifies application programming and makes it straightforward to exploit the HTTP protocol.

2.2 Stateless Request/Response

The RESTful API is a stateless request/response service. The service does not make any use of HTTP cookies, and thus allows requests to be load-balanced over many servers. This in turn, improves both throughput and response time.

2.3 Platform Independence

The RESTful API can be used without any FactSet-supplied software. Applications can access data from any system that supports HTTPs requests.

2.4 Zero-Based Installation

Proprietary software is not needed to access this service. Any standard library or application that can issue HTTP queries can request data via the RESTful API.

2.5 Seamless Failover

If a FactSet Data Server or an entire data center goes down, requests will be routed to an available Data Server using state-of-the-art load balancers. You do not have to change your software, as this failover will be automatic.

Chapter 3 Access and Security

Host Name	Service	Example URL
api.factset.com	bulk-documents/tickhistory	https://api.factset.com/bulk-documents/tickhistory

3.1 Easy Access Through FactSet Tick History RESTful API

The RESTful API uses HTTP Basic Authentication which requires the following HTTP header be added to the request:

Authorization: Basic {Base-64 Encoding of "user:password"}

See the Authentication Section for details on how to generate the password.

3.2 Certificates and Certificate Chains

FactSet recommends clients should not hardcode reliance on any particular certificate or advertised certificate chain into their applications and expects clients to rely on Public Key Infrastructure verification and validity of the certificates.

- FactSet's Certificates will change over time as they are renewed, and the complexity of the algorithms employed increases (i.e. SHA-2 rather than SHA-1 signatures). This is the constant evolution of security as old security algorithms are retired and new security algorithms are included. Often these certificates get updated on a rolling multi-year basis. Validating a certificate dynamically during TLS connect must be incorporated by the client as a necessary practice.
- FactSet's Certificate chains, including intermediate Certificate Authorities, may change over time, and it is important that clients dynamically validate FactSet's certs against a modern CA trusted root certificate store. FactSet's current root certificate is Thawte Primary Root CA.

3.3 Security Protocols

Clients should not hardcode dependencies on any specific security protocol as FactSet is continuously reviewing security policies and reserves the right to disable support for older security protocols with short notice. The current supported protocols are TLSv1.1 and TLSv1.2 but at a future date, these may be replaced with future versions. Clients should make sure that their software can handle ever changing Security Protocols.

3.4 Firewall

Clients should have access to api.factset.com through their firewall.

Chapter 4 Requests

4.1 Description

The FactSet RESTful API supports HTTP GET requests.

4.2 Request Headers

Header name	Description
Authorization	Standard HTTP header. Value needs to use ‘Basic <base64 encoded value>’ format.
Accept-Encoding	FactSet will send compressed responses. Value needs to be deflated, gzip

4.3 Request Parameters

request-files end point

Parameters	Values	Description
ticker[1]	string	Supports one or multiple tickers separated by comma in the request-files URL. Note: The maximum number of tickers depends on the length of the URL. The request length should be less than 8192 bytes(8192 characters in length)
iso_code[1]	string	ISO CODE (Refer to FactSetTickHistoryDataFeed_DataModel_V1.0.pdf for details) Only one ISO CODE allowed
OR		
product[1]	integer	Product (Exchange) code. Refer to the Product code section in the FactSetTickHistoryDataFeed_DataModel_V1.0.pdf for list of available codes.
start_date[1]	date	Earliest date from which data is required. Should be used in conjunction with end_date. Note: Tick Data is available from 2012-01-01
end_date[1]	date	This is the latest date until which the data is to be fetched. Should be used in conjunction with start_date.
fields[2]	string	All fields are included by default. Refer to the Fields section in the FactSetTickHistoryDataFeed_DataModel_V1.0.pdf for list of available fields. Each field identifier should be separated by a comma.
order[2]	string	Order by date, year, month & ticker. To sort results in chronological order. Results are in reverse chronological order by default.
start_time[2]	time	Earliest time from which data is required. should be in HH:MM:SS format. Should be used in conjunction with end_time
end_time[2]	time	This is the latest time until which the data is to be fetched. should be in HH:MM:SS format. Should be used in conjunction with start_time Note: Using start_time and end_time will fetch the data on particular days in between the timestamps given
single[2]	boolean	(single=true) Returns data in a single file. Default value is false Note: This parameter works for requests with shorter data ranges having filesize <= 128MB

regionalISOCode[2]	string	Works in conjunction with iso_code parameter(Refer to FactSetTickHistoryDataFeed_DataModel_V1.0.pdf for details)

get-files end point

Parameters	Values	Description
requestId[1]	string	Request Id returned by the request-files endpoint to poll and collect results of the query
_paginationLimit[2]	integer	The number of results to display per page. "_paginationLimit" overrides the default (20) results per page.
_paginationOffset[2]	integer	For requests that generates more than 20 results. This parameter is used to identify the beginning of next set of results

[1] **Required parameter**

[2] **Optional parameter**

4.4 Request Return Codes

HTTP status code	Description
0	No Error
400	Invalid request or "Improper formatting of request parameters"
401	Missing or invalid authentication.
403	User is forbidden access with current credentials. Missing parameters or Invalid input value in the request. Refer to Error Response
405	Invalid HTTP method. Either the method is not GET or exceeds the maximum request length (currently set at 5000 bytes). The description field will indicate the exact reason.
408	The request timed out.
429	Too many requests.
500	There was an error while retrieving the data

Chapter 5 Sample Requests

5.1 URL Syntax

A URL is defined as following:

```
<protocol>://<base URL>/<service><endpoint>?<optional query string parameters>
```

5.2 Example:

Request Fileshttps://api.factset.com/bulk-documents/tickhistory/v1/request-files?ticker=XXX&iso_code=XXX&start_date=YYYY-MM-DD&end_date= YYYY-MM-DD

Get Files:
https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=XXXX

5.2.1 Request by Ticker/s and One Iso code

Request Files:

https://api.factset.com/bulk-documents/tickhistory/v1/request-files?ticker=FDS&iso_code=USA&start_date=2021-05-24&end_date=2021-05-24

```
{
   "requestId": "  52650005-4b95-456d-bb3e-
   "status":       4fbf8a06b3d3                  "Submitted"
}
```

Get Files: https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=52650005-4b95-456d-bb3e-4fbf8a06b3d3

The response status is updated at each step of the file creation process.

Example:

1. Submitted status

```
{

   "status": "Submitted",
   "requestTimestamp": "2021-05-26 13:33:02",
   "updateTimestamp": "2021-05-26 13:36:31",
   "not_authorized_tickers": []
}
```

2. InProgress status

```
{
   "status": "InProgress",
   "requestTimestamp": "2021-05-26 13:33:02",
   "updateTimestamp": "2021-05-26 13:36:31",
   "not_authorized_tickers": []
}
```

3. Completed status

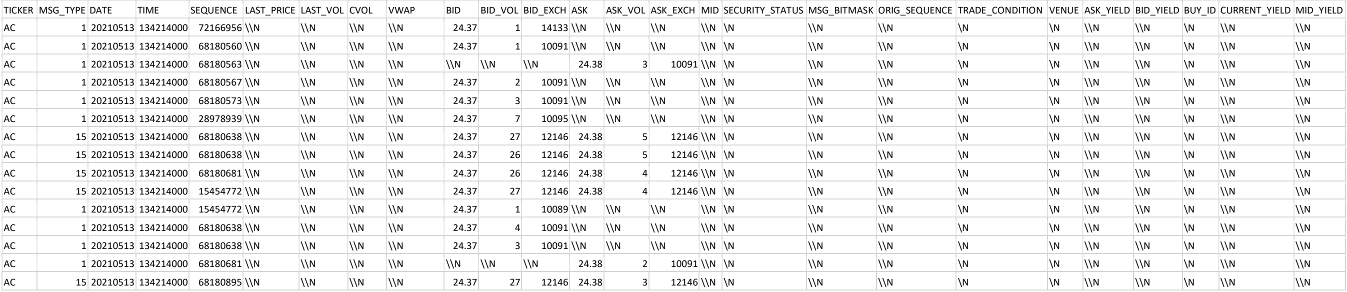
```
{
   "status": "Completed",
   "requestTimestamp": "2021-03-11 21:53:46",
   "updateTimestamp": "2021-03-11 22:05:14",
   "not_authorized_tickers": [],
   "data": [
      {
         "fileName": "data_0_0_0.csv.gz",
         "url": "https://fdss3-structured-data-delivery-tickhistory-
prod.s3.amazonaws.com/2e0b69498518f3e3f8154bc7b08b9d5d/2305960670/data_0_0_0.csv.gz?X-Amz-Algorithm=AWS4-
HMACSHA256&X-Amz-Credential=ASIA6B6XG7VEWT42TPFT%2F20210311%2Fus-east-1%2Fs3%2Faws4_request&X-
AmzDate=20210311T222049Z&X-Amz-Expires=86400&X-Amz-SignedHeaders=host&X-Amz-Security-
Token=IQoJb3JpZ2luX2VjEG8aCXVzLWVhc3QtMSJGMEQCIDnqMiSqlmM%2BgcbKxX8vPRYHq3E864PoezIKzjbgqrTKAiARus%2B
KBFnO7UL9Y%2BibPCkCRTGpjp%2BIFblJyI%2FBkRJz4CrhAQiX%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F8BEAAaDDk2NjI4Mj
ExNjQyNSIMiX9hwMJQZPx48IGYKrUBHPg9JZ3%2B3WIQu%2FB1HQiZ4f2RwUEQgzlLOy%2BrcpPUIBhhjUmfSu41JScpEPPlPpFK
RlkqNBbPjkD5Roy6AIamv3aC7EUG92aVg5hbkvk3P1nuFnXp3deKH5xEd8NeeeEmWFR2UBkjI%2BhmAmWf8CkVJW9jjpuRMmydd
YYPkIqj1scJEoqisLMIkJzjmTuMU7zruC2P6aMSxDYJGNil12473ttbnQ9%2BHcMGVTMfXYQLIj%2FqHJmvNDC%2Fr6qCBjrhAYKS%
2BWXv7hFDsA6p7%2BV3BHKn2cw6GNfag7DSiahCZ11q8orehHLyjOh9bdleMED%2BwBuPb6hdectAp8cd3y%2BBa5TXT0UfeadUL
FMrXW1bx095wHvn1amhvhpty%2BFU3XoEuCSCZgKln4Pl4a7RGHd4tqyWlmcSHvN80aCmYhCB1D8pNnEd9af8WKYdntnWBa3xsv
0WPd9WeD48Zwa5Hwy2aZyAMPa4Z0ceVTOHcojaL791bQZI6lK35FLVkjxb71lvvbjRWx%2F5G2Be3w8OEjg8Aqhm%2FBOWVyPml
oaIuq2QOkM9iw%3D%3D&X-Amz-Signature=79233c651f4254618bd83542061071f38f0edd2e01c0fb21cf28c802471075df"
      },
      { … }    ],
   "meta": {

      "pagination": {
         "total": 128,
         "isEstimatedTotal": false,
         "limit": 20,
         "offset": 20
      },
      "partial": {
         "isPartial": true
      }   }   }
```

Note: The secure links are examples.

- Example output.

NULL fields are populated with “\\N” values



TICKER	MSG_TYPE	DATE	TIME	SEQUENCE	LAST_PRICE	LAST_VOL	CVOL	VWAP	BID	BID_VOL	BID_EXCH	ASK	ASK_VOL	ASK_EXCH	MID	SECURITY_STATUS	MSG_BITMASK	ORIG_SEQUENCE	TRADE_CONDITION	VENUE	ASK_YIELD	BID_YIELD	BUY_ID	CURRENT_YIELD	MID_YIELD
AC	1	20210513	134214000	72166956	\\N	\\N	\\N	\\N	24.37	1	14133	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180560	\\N	\\N	\\N	\\N	24.37	1	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180563	\\N	\\N	\\N	\\N	\\N	\\N	\\N	24.38	3	10091	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180567	\\N	\\N	\\N	\\N	24.37	2	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180573	\\N	\\N	\\N	\\N	24.37	3	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	28978939	\\N	\\N	\\N	\\N	24.37	7	10095	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	68180638	\\N	\\N	\\N	\\N	24.37	27	12146	24.38	5	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	68180638	\\N	\\N	\\N	\\N	24.37	26	12146	24.38	5	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	68180681	\\N	\\N	\\N	\\N	24.37	26	12146	24.38	4	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	15454772	\\N	\\N	\\N	\\N	24.37	27	12146	24.38	4	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	15454772	\\N	\\N	\\N	\\N	24.37	1	10089	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180638	\\N	\\N	\\N	\\N	24.37	4	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180638	\\N	\\N	\\N	\\N	24.37	3	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180681	\\N	\\N	\\N	\\N	\\N	\\N	\\N	24.38	2	10091	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	68180895	\\N	\\N	\\N	\\N	24.37	27	12146	24.38	3	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N

Secure links in the get-files response is active for up to 24 hours. Output files are archived for 7 days and users can rerun get-files requests using the “requestId” created by their request-files query if they do not download files with the 24hour period.

5.2.2 Request by Product code

Request Files:

https://api.factset.com/bulk-documents/tickhistory/v1/request-files?product=9001&start_date=2020-12-29&end_date=2020-12-30

Note: Only one product code should be included in a request

```
"requestId":  a1b2959b-dd95-49b2-87aa-                 ",
"             14b2f8470dba
"status": "Submitted"
```

Get Files:

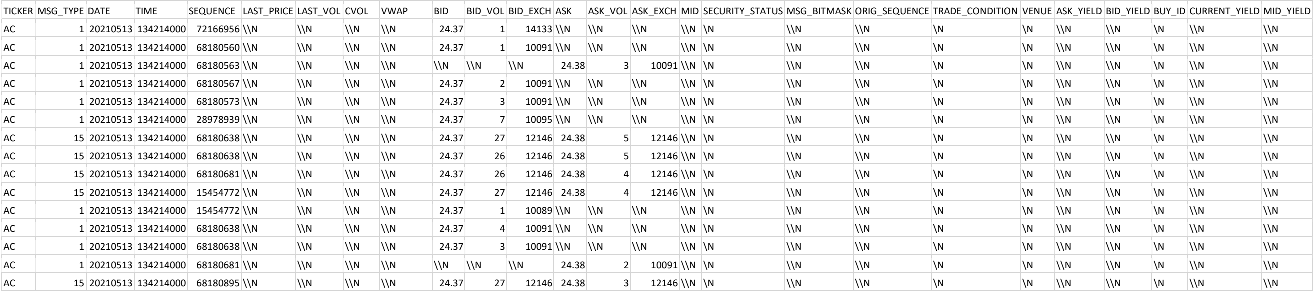
https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=a1b2959b-dd95-49b2-87aa-14b2f8470dba

```
{
   "status": "Completed",
   "requestTimestamp": "2021-01-15 19:44:10",
   "updateTimestamp": "2021-01-15 20:20:16",
   "data": [
      {
         "fileName": "data_0_0_0.csv.gz",
         "url": "https://fdss3-structured-data-delivery-tickhistory-
prod.s3.amazonaws.com/343b930be20467ce7c79a0196f694282/1754993474/data_0_0_0.csv.gz?X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Credential=ASIA6B6XG7VEWC5DY2MP%2F20210115%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-
Date=20210115T212211Z&X-Amz-Expires=86400&X-Amz-SignedHeaders=host&X-Amz-Security-
Token=IQoJb3JpZ2luX2VjEEYaCXVzLWVhc3QtMSJHMEUCIHRviimFSJdYZPg%2F09f3Q2F8GieCV94uIYlhnBl2ytY%2FAiEAuYRd7
nXdc0zy8%2BlBjmhJoha%2FA%2FNIj29spi16M6Fpza4q2AEIHhAAGgw5NjYyODIxMTY0MjUiDD5Z0LMLf5aY7FmCxSq1AWvQUCG
nBb5fZtArIVHjf%2F0Qo%2F21RRT%2BcmZlro3QK8uLYhG%2B5%2FScTssNHrp0xxKBto10MFQCUWviqH4b67dDQy2mtnDMmq9Ul
EBoWR25PMJCYKiv6n80%2F7wBBD%2BHDadfznb%2FFCqOtAtTtSVcvl4piHHZ%2BgPF0Zddb50j24kIbktOddGMT36KeHpFA1E51t
WPKnOZL%2BtfZtlp1RBBI%2FWixvQCs36k0NgPjP99Upktg1QgBwhqYOow1I6IgAY64AFGznAYCINqpmNPhFmxUJ3%2BThrCdEbN
MrqTPBIs3QPf4hs9MNY%2BU7bVtVZbUa3Lx0mObkAzVEdNfxp2Vc1KKlPBHYnJsJIsJaN7%2B6AZ%2BOjyae43hYgP%2BMAC2%2
Fa%2FeCHYuYHuVHtqy56ZlPzNQOn9BDBqsPX2hmebXBSfhmYa33x4JRwel8LrSKFIQ%2FWn4lZnH2C9cRLXAuFngZJfxr2Ep9Psp
zalaR9W5YUEUgUzXjN4gI8GMDIRSJ5QhpDP6tFz8GBI9ooi28qJ5V5XXhKTJkixadcCdSqQLrScAgLnY%2BZfzA%3D%3D&X-
AmzSignature=ebe1530417539b94bd2d690f7c24816e0403e2c6f1bba517265bae7fe2eb925f"


        },
        {. . . },     ],
    "meta": {
        "pagination": {            "total": 256,
            "isEstimatedTotal": false,
            "limit": 20,
            "offset": 20
        },
        "partial": {


            "isPartial": true
        }  }  }
```

Example output



TICKER	MSG_TYPE	DATE	TIME	SEQUENCE	LAST_PRICE	LAST_VOL	CVOL	VWAP	BID	BID_VOL	BID_EXCH	ASK	ASK_VOL	ASK_EXCH	MID	SECURITY_STATUS	MSG_BITMASK	ORIG_SEQUENCE	TRADE_CONDITION	VENUE	ASK_YIELD	BID_YIELD	BUY_ID	CURRENT_YIELD	MID_YIELD
AC	1	20210513	134214000	72166956	\\N	\\N	\\N	\\N	24.37	1	14133	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180560	\\N	\\N	\\N	\\N	24.37	1	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180563	\\N	\\N	\\N	\\N	\\N	\\N	\\N	24.38	3	10091	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180567	\\N	\\N	\\N	\\N	24.37	2	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180573	\\N	\\N	\\N	\\N	24.37	3	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	28978939	\\N	\\N	\\N	\\N	24.37	7	10095	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	68180638	\\N	\\N	\\N	\\N	24.37	27	12146	24.38	5	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	68180638	\\N	\\N	\\N	\\N	24.37	26	12146	24.38	5	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	68180681	\\N	\\N	\\N	\\N	24.37	26	12146	24.38	4	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	15454772	\\N	\\N	\\N	\\N	24.37	27	12146	24.38	4	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	15454772	\\N	\\N	\\N	\\N	24.37	1	10089	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180638	\\N	\\N	\\N	\\N	24.37	4	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180638	\\N	\\N	\\N	\\N	24.37	3	10091	\\N	\\N	\\N	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	1	20210513	134214000	68180681	\\N	\\N	\\N	\\N	\\N	\\N	\\N	24.38	2	10091	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N
AC	15	20210513	134214000	68180895	\\N	\\N	\\N	\\N	24.37	27	12146	24.38	3	12146	\\N	\N	\\N	\\N	\N	\N	\\N	\\N	\N	\\N	\\N

5.2.3 Request by multiple tickers/one iso code and specified list of fields

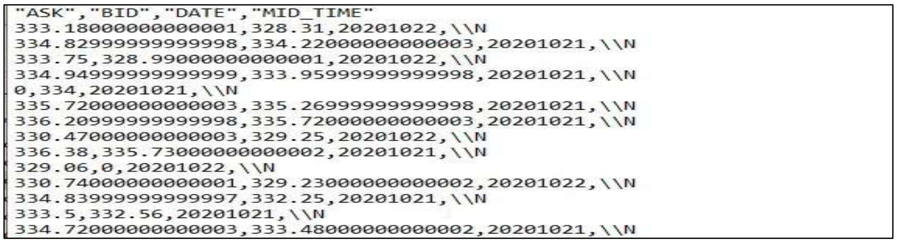
Request Files:

https://api.factset.com/bulk-documents/tickhistory/v1/request-files?ticker=FDS,IBM,AAPL&iso_code=USA&start_date=2021-05-21&end_date=2021-05-24&fields=DATE,MID_TIME,BID,ASK

```
“requestId”: “018c1e49-e3c4-4bf7-9b4b-f98f6778e75f“,
“status”: “Submitted”
```

Get Files: https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=018c1e49-e3c4-4bf7-9b4b-f98f6778e75f

Example output



```
"ASK","BID","DATE","MID_TIME"
333.18000000000001,328.31,20201022,\\N
334.82999999999998,334.22000000000003,20201021,\\N
333.75,328.99000000000001,20201022,\\N
334.94999999999999,333.95999999999998,20201021,\\N
0,334,20201021,\\N
335.72000000000003,335.26999999999998,20201021,\\N
336.20999999999998,335.72000000000003,20201021,\\N
330.47000000000003,329.25,20201022,\\N
336.38,335.73000000000002,20201021,\\N
329.06,0,20201022,\\N
330.74000000000001,329.23000000000002,20201022,\\N
334.83999999999997,332.25,20201021,\\N
333.5,332.56,20201021,\\N
334.72000000000003,333.48000000000002,20201021,\\N
```

5.2.4 Request using Order filter

Request Files: https://api.factset.com/bulk-documents/tickhistory/v1/request-files?ticker=FDS&iso_code=USA&start_date=2020-10-21&end_date=2020-10-22&fields=DATE,MID_TIME,BID,ASK&order=MID_TIME,ASK,DATE,BID

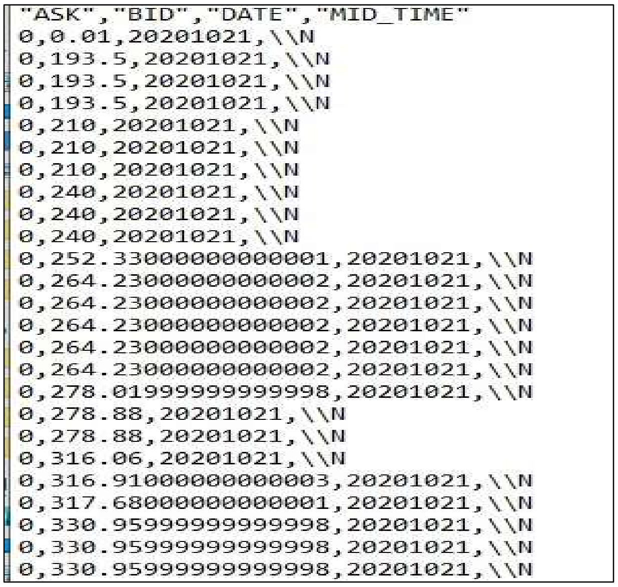
Data is sorted based on the order specified in the request.

```
    "requestId":  8745c4be-e3ad-4ab3-8b48-dc02277eb3a7",
"
    "status": "Submitted"
```

Get Files:

https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=8745c4be-e3ad-4ab3-8b48-dc02277eb3a7

Example output



```
"ASK","BID","DATE","MID_TIME"
0,0.01,20201021,\\N
0,193.5,20201021,\\N
0,193.5,20201021,\\N
0,193.5,20201021,\\N
0,210,20201021,\\N
0,210,20201021,\\N
0,210,20201021,\\N
0,240,20201021,\\N
0,240,20201021,\\N
0,240,20201021,\\N
0,252.33000000000001,20201021,\\N
0,264.23000000000002,20201021,\\N
0,264.23000000000002,20201021,\\N
0,264.23000000000002,20201021,\\N
0,264.23000000000002,20201021,\\N
0,264.23000000000002,20201021,\\N
0,278.01999999999998,20201021,\\N
0,278.88,20201021,\\N
0,278.88,20201021,\\N
0,316.06,20201021,\\N
0,316.91000000000003,20201021,\\N
0,317.68000000000001,20201021,\\N
0,330.95999999999998,20201021,\\N
0,330.95999999999998,20201021,\\N
0,330.95999999999998,20201021,\\N
```

Chapter 6 Response Data Fields

6.1 Response fields

Response fields	Values and Datatype	Description
fileName	e.g. data_0_0_0.csv.gz	Tick data compressed filenames

6.2 Meta fields

```
"meta": {
  "pagination": {
    "total": 256,
    "isEstimatedTotal": false,
    "limit": 20,
    "offset": 20
  },
  "partial": {
    "isPartial": true
  }  }
```

6.2.1 Pagination

An endpoint where the primary data in the response is an array or an object containing an array **MAY** support pagination of the respective array to limit the number of resources returned in a response, using a pagination meta-attribute.

Meta Member name	Format	Description
pagination	object	Object defining the pagination attributes for an endpoint response.
pagination.total	int32	Total number of entries in the result set.
pagination.isEstimatedTotal	boolean	Flag indicating that the value of total is estimated.

Example

```
"meta": {
        "pagination": {
            "total": 256,

            "isEstimatedTotal": false,

            "limit": 20,
            "offset": 20
        },
```

6.2.2 Total

Specifies the number of records returned in the response. In this example the total number of records are 256

```
"meta": {
        "pagination": {
            "  total": 256
            "isEstimatedTotal": false,
            "limit": 20,
            "offset": 20
        },
        "partial": {
            "isPartial": true
```

6.2.3 _paginationLimit

The default results per page is 20 (offset). “_paginationLimit” parameter allows users to override the default.

Get Files:
https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=ec17e512-d1fb-45ba-bde2e043d67321b1&_paginationLimit=100

shows 100 results per page instead of default 20

```
Default Limit value
"meta": {
        "pagination":           {
"total": 256,
            "isEstimatedTotal": false,
            "  limit": 20
            "offset": 20
        },
        "partial": {
            "isPartial": true


with “_paginationLimit=100”
    "meta": {
        "pagination": {
            "total": 256,
            "isEstimatedTotal": false,
            "  limit": 100
            "offset": 100
        },
        "partial": {
            "isPartial": true
```

6.2.4 _paginationOffset

The “_paginationOffset” option allows users to navigate to the next set of records. By default the Pagination Limit = Pagination Offset.

Get Files:

https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=ec17e512-d1fb-45ba-bde2-e043d67321b1

Default Offset value

```
"meta": {
       "pagination": {
          "total": 256,
          "isEstimatedTotal": false,
          "limit": 20,
          "offset": 20
       },
       "partial": {
          "isPartial": true
```

"_paginationOffset=100" returns next set of 100 records (101-200)

Get Files:
https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=ec17e512-d1fb-45ba-bde2e043d67321b1&_paginationLimit=100&_paginationOffset=100

with "_paginationOffset=100"

```
   "meta": {
       "pagination":          {
"total": 256,
          "isEstimatedTotal": false,
          "limit": 20,
          "offset": 100
       },
       "partial": {
          "isPartial": true

       }
   }
}
```

Get Files: https://api.factset.com/bulk-documents/tickhistory/v1/get-files?requestId=ec17e512-d1fb-45ba-bde2e043d67321b1&_paginationLimit=100&_paginationOffset=200

"_paginationOffset=200" provides the next set of 100 records (201-300), in this example it provides the last 56 records (201-256)

```
  "meta": {
      "pagination":          {
"total": 256,
          "isEstimatedTotal": false,
          "limit": 100,
          "offset": 300
      },
      "partial": {
          "isPartial": false
      }
    }
}
```

6.2.5 isPartial

6.2.5.1 isPartial: true

Denotes there are additional results/pages for the request

```
    "meta": {
      "pagination": {
          "total": 256,
          "isEstimatedTotal": false,
          "limit": 20,
          "offset": 100
      },
      "partial": {
          "isPartial": true
      }   }}
```

6.2.5.2 isPartial: false

Denotes there are no further results/pages for the request.

```
 "meta": {
      "pagination":          {
"total": 256,
          "isEstimatedTotal": false,
          "limit": 100,
          "offset": 300


      },
      "partial": {
          "isPartial": false
      }   } }
```

Chapter 7 Error Response

7.1 Permission Error

Error message if the user does not have a subscription to the product

```
"id": "4f8f5586-eba7-4fca-a189-2204110193b7",
"error_message": "You are not subscribed to this product: Tick History"
```

7.2 Param Error

Example of missing parameter and not entitled error messages:

1. Missing mandatory parameters (Ticker, iso_code, product, or start_date/end_date) or incorrect values included in the request-file query

```
"id": "41cf4e5a-da82-4fbc-a841-0320b7cc5fd2",
"error_message": "parameterError:Missing required parameter start_date"
```

```
"id": "3c1dd330-89d9-4e68-829f-4d08ec8d3979",

"error_message": "parameterError:Missing required parameter iso_code"
```

2. User does not have entitlements for the tickers included in their request

```
  "status": "Completed",
  "requestTimestamp": "2021-01-14 14:38:36",
  "updateTimestamp": "2021-01-14 15:30:12",
   not_authorized_tickers": [BNGO-USA,MSFT-USA,RIOT-USA],
"
```

Note: There is a polling interval of 30 seconds to fetch the latest status of the query

Chapter 8: Options Identifiers

Options Clearing Corporation

Symbology convention:

Root +. + Region + # + P or C (for Put or Call) + randomly generated four-character alphanumeric code + - + Exchange.

Example: RACE.IT#P7Y8J-DMI is a Ferrari NV put option traded on Italian Derivatives Options (DMI) with a strike price of 188 that expires on 21-June-2024.

Note: Make sure converting the standard '#' into '% 23' to the API /request-files endpoint.

FactSet's Options of Futures symbology

FactSet Option on Futures symbology is based on the Globex Exchange symbol

Globex option root from the exchange + 1-letter future month code (see section 5.4) + 1-digit year code + # + 6-digit option expiration date (YYMMDD) + P/C + 11-digit Strike Price (6 decimals) +

Exchange ISO

Example: LNEK31#310425C00002700000-NYME, is the May 2031 Natural Gas future call Option expiring on 25th of April 2031 with a $2.70 strike price.

Example: LOZ30#301115P00093000000-NYME which is the Dec 2030 Crude Oil future Put Options expiring on 15th Nov 2030 with a $93.00 strike price.

Appendix D: FactSet Symbology and Data Policy

Refer to FactSetTickHistoryDataFeed_DataModel_V1.0C.pdf document for **FactSet Symbology and Data Policy.**

Global Client Support

If you have any questions, submit a request through https://issuetracker.factset.com under the “Tick History DataFeed” category.
If you do not have login credentials for Issue Tracker, Email to datafeed_support@factset.com.
For general assistance, contact your local FactSet Consultant or Salesperson or Email support@factset.com.

Trademarks

- FactSet is a registered trademark of FactSet Research Systems, Inc.
- Microsoft is a registered trademark, and Windows is a trademark of Microsoft Corporation.
- All other brand or product names may be trademarks of their respective companies.

Acknowledgements

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org).