FINRA

fileX User Guide

Effective Date: October 7, 2025
Current Document Version 1.5.0

Revision History

Document Version	Published on	Status	Comments
0.1.0	08/01/2018	Approved	First draft of fileX service user guide
0.1.1	08/15/2018	Approved	Changes to SFTP Hostnames for Production and CT Environment
1.0.0	9/18/2018	Approved	Enhancement to FAQs & Troubleshooting section along with additional minor changes
1.0.3	10/18/2018	Approved	Changed the ACATS and eFOCUS upload/download paths and added FAQs
1.1	12/19/2018	Approved	Added HTTPS REST API Details. Formatting Updates.
1.2.0	02/06/2019	Approved	Added S3 Direct details and HTTPS REST API details for File Tracking
1.2.1	03/25/2019	Approved	Add File Archive feature details
1.2.2	04/30/2020	Approved	Added SSH config details for SFTP Added directory information for newly onboarded apps
1.2.3	8/10/2020	Approved	Added HTTPS certificate information for REST API usage, section 5.1. Added (in FAQs) SFTP connection options to prefer password instead of keys. Added (in FAQs) Guidance around Packet handling. Added directory information for newly onboarded apps
1.2.4	1/29/2021	Approved	Added HTTPS TLS v1.2 policy, section 5.1. Added (in FAQs) TLS v1.2 Cipher information.
1.3.0	9/01/2021	Approved	Added details about FINRA Identity platform (FIP) to support OAUTH Updated HTTPS and S3 direct method instructions Added FIP HTTP Response codes Deprecated filex-{Attribute} header(fileX metadata) as the feature was not utilized
1.4.0	11/05/2022	Approved	Added application and application sub-space details to the existing tracking APIs Updated OAuth2 details
1.5.0	10/07/2025	Approved	Added UI file transfer method in section 2 Added FINRA Gateway Link in Section 3 Updated Supported Ciphers and Algorithms for SFTP in section 3.1 Added details for On Behalf of Feature (OBO) in section 4.1 Moved Trouble Shooting and FAQs to section 8 Created a new section for FINRA Gateway Upload App

Contents

1 Introduction 3
2 Access Methods 4
3 Environment and Connectivity 5
3.1 SFTP Transfer Method 5
3.1.1 Using OBO Feature via SFTP 6
3.2 HTTPS REST API Endpoints 7
3.3 AWS S3 DIRECT TRANSFER 7
4 Entitlement & Access Control 8
4.1 On Behalf Of (OBO) 8
5 HTTPS REST APIs 9
5.1 Security 9
5.1.1 OAUTH2.0 9
5.1.2 TLS Certificates 9
5.2 Response Codes 10
5.3 Request Headers 11
5.4 REST API Catalog 12
5.4.1 List Available Applications 12
5.4.2 List Available Application Sub-Spaces 13
5.4.3 List Available Files in Application Sub-Space 15
5.4.4 Download File from Application Sub-Space 17
5.4.5 Upload File to Application Sub-Space 18
5.4.6 Track File uploaded to Application Sub-Space 21
6 S3 Direct Transfer 23
6.1 Generate Token for AWS S3 23
7 Upload App in FINRA Gateway 25
8 Troubleshooting & FAQs 27
9 Contact Information 32
10 Supported Applications and Relevant Parameters 33

1 Introduction

fileX is a centralized, secure file transfer service from FINRA, where customers (member firms and industry participants) can send or receive batch file(s) to FINRA Applications like CRD, ACATS etc., (referred within this document as 'application').

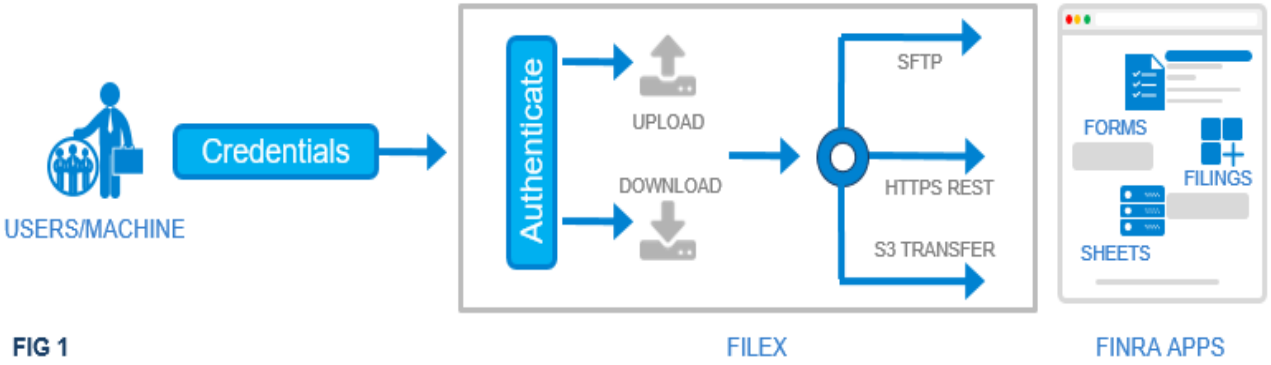


FIG 1

The purpose of this document is to provide details for using fileX services to transfer file(s) with FINRA applications.

fileX supports multiple access methods to send/receive files, and a customer may choose to use any of the supported access methods to transfer file(s). FINRA Entitlement process controls the account provisioning and entitlements which control access to various application sub-spaces. Sub-spaces are the locations within each application in fileX where customers can upload or download files.

Please refer to section 10 for more details on sub-spaces.

2 Access Methods

fileX supports four access methods detailed below.

	FINRA customers can upload or download files through Secure File Transfer Protocol (SFTP), a standard file transfer mechanism to securely transmit files between systems/machines. fileX supports the full security and authentication functionalities provided by SFTP.
	FINRA customers can upload or download files using REST APIs over HTTPS protocol. Customers can make standard REST API calls to the endpoint URL with valid credentials for authentication. REST API calls are encrypted through HTTPS.
	FINRA customers can upload or download files natively through AWS S3 APIs. Customers who are already using Amazon AWS S3 can take advantage of this method to send/receive files directly to/from their S3 bucket.
	FINRA customers can upload and download files, one at a time, through the FINRA Gateway Upload app. Customers can also use Upload app to manage their file history and view file feedback.

3 Environment and Connectivity

FINRA recommends customers to first test their setup in lower environments before cutting it over to 'Production'.

	PRODUCTION	CUSTOMER TEST	LOWER (QA)
Environment description	Live/production environment	Production-like customer test environment	Non-Production environment for test purposes
Credentials	Production FINRA Enterprise Web Security (EWS) credentials	Production FINRA Enterprise Web Security (EWS) credentials	Contact the respective FINRA application to get credentials for this environment
Hostname /URL	filex.finra.org	filex.ct.finra.org	filex-int.qa.finra.org
Static IP Addresses for SFTP	52.207.197.35 35.171.199.181	18.209.156.254 34.225.135.103	52.201.46.30 52.70.2.197
Port for SFTP	22	22	22
Port for REST methods	443	443	443
FIP OAUTH 2 URL	https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token	https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token	https://ews-qaint.fip.qa.finra.org/fip/rest/ews/oauth2/access_token
FINRA Gateway	https://gateway.finra.org	N/A	N/A

3.1 SFTP Transfer Method

The following steps are required to use the fileX SFTP service

- Open firewall to allow outbound traffic on port 22 to FINRA SFTP host IP addresses
- Request FINRA to allow inbound traffic on port 22 for your outbound server's internet routable IP address. Please call FINRA support at 800-321-6273 and provide a list of external IPs to be whitelisted to open the network/firewall from FINRA.
- Request FINRA Credentials (if not already available) with entitlements for specific FINRA application file transfer (see section 4)
- Install and configure a SFTP client/library to connect and transfer files
- Please ensure the following SFTP SSH configurations are set appropriately on your SFTP client

		Supported Ciphers and Algorithms
1	Supported SSH Ciphers	1. aes128-ctr 2. aes128-gcm@openssh.com 3. aes192-ctr 4. aes256-ctr 5. aes256-gcm@openssh.com
2	Supported SSH Key Exchange Algorithms	1. mlkem768x25519-sha256 2. mlkem768nistp256-sha256 3. mlkem1024nistp384-sha384

		Supported Ciphers and Algorithms
		4. diffie-hellman-group14-sha256 5. diffie-hellman-group16-sha512 6. diffie-hellman-group18-sha512 7. ecdh-sha2-nistp384 8. ecdh-sha2-nistp521 9. ecdh-sha2-nistp256 10. diffie-hellman-group-exchange-sha256 11. curve25519-sha256@libssh.org 12. curve25519-sha256
3	Supported SSH MAC Algorithms	1. hmac-sha2-256-etm@openssh.com 2. hmac-sha2-512-etm@openssh.com

3.1.1 Using On Behalf Of (OBO) Feature via SFTP

Once you have obtained a Prime account by following the steps mentioned in section 4.1, follow these steps to upload and download files On Behalf Of (OBO) a third party:

Note: Ensure you have your machine set up to do fileX through SFTP.

1. Open your SFTP client.
2. Log in with your Prime account username and password, or you can use your SSH key.
3. Once you are connected to the host, you can see the profile org IDs that you are entitled to submit through OBO, and you can upload and download files with that profile org.

Note: You can view an org ID to see the entitlement for the profile. For example, if an org ID shows 606 NMS, the profile is entitled for SEC Rule 606 NMS Reporting.

Here is an example of submitting a file OBO using CURL commands:

```
[DG4Q0CQK6H:~ k31850$ sftp ytaoprime2@sftp-filex.qa.finra.org
[ytaoprime2@sftp-filex.qa.finra.org's password:
Connected to sftp-filex.qa.finra.org.
sftp> ls
143546  143547  9349
sftp> cd 9349
sftp> ls
606nms
sftp> cd 606nms
sftp> ls
in        in_arcv   out
sftp> cd in
sftp> ls
sftp> put testfile.txt
Uploading testfile.txt to /9349/606nms/in/testfile.txt
testfile.txt                                  100%   19     0.8KB/s   00:00
sftp>
```

3.2 HTTPS REST API Endpoints

The following steps are required to use the fileX HTTPS REST API service

- Open firewall to allow outbound traffic on port 443 from your outbound server IP address to fileX HTTPS REST APIs and to FINRA Identity Platform (FIP) API endpoints
- Request FINRA Credentials (if not already available) with entitlements for specific FINRA application file transfer (see section 4)
- Open firewalls to allow traffic to/from AWS S3. Upload and Download services currently use native AWS S3 endpoints.
    - For AWS S3 IP ranges refer to: https://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html
- Invoke FIP OAUTH API to obtain OAUTH token
- Invoke/Call fileX HTTP REST APIs with FIP OAUTH token as detailed in section 5 below

3.3 AWS S3 DIRECT TRANSFER

The following steps are required to use the fileX S3 Direct service

- Open firewall to allow outbound traffic on port 443 from your outbound server IP address to fileX HTTPS REST APIs and to FINRA Identity Platform (FIP) API endpoints
- Request FINRA Credentials (if not already available) with entitlements for specific FINRA application file transfer (see section 4)
- Open firewalls to allow traffic to/from AWS S3.
    - For AWS S3 IP ranges refer to: https://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html
- Invoke FIP OAUTH API to obtain OAUTH token
- Obtain S3 STS access token from the fileX REST API for AWS S3 as detailed in section 6.
- Use the S3 STS access token to upload/download using AWS S3 APIs or AWS CLI or AWS SDK wrapper libraries.

4 Entitlement & Access Control

FINRA Entitlement Service controls access and privileges granted to customer accounts to access various services provided by FINRA, including fileX service. Customers will have the option of creating multiple file transfer accounts with different access privileges. Various supported access levels for users includes, but not limited to,

- read/download only
- submit only
- submit and download

This allows customers to support separation of duties within the firm across different departments. Administrator (typically SAA) of the firm needs to contact FINRA Entitlement Service to create file transfer accounts and request respective FINRA Application File Transfer entitlements. Currently, entitling file transfer or machine-to-machine account is a paper-based process handled by FINRA entitlement service. If you have any question about FINRA Entitlement program, please check https://www.finra.org/industry/entitlement-program.

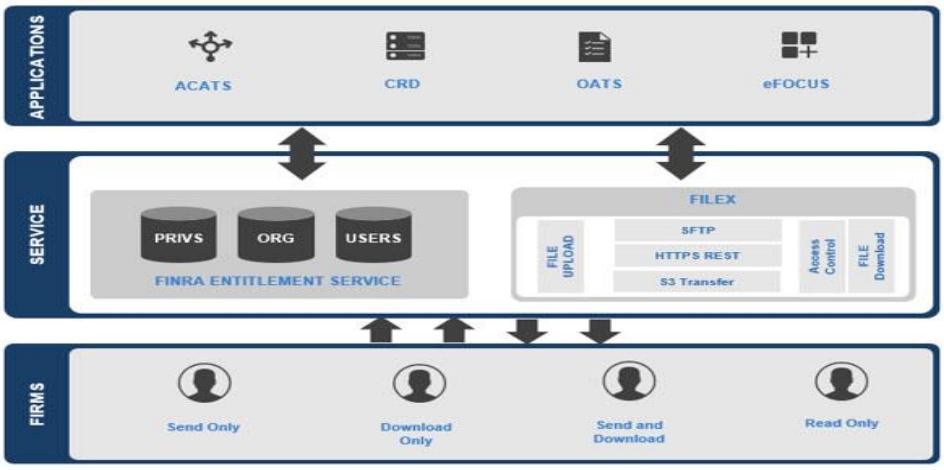


FIG 2

fileX leverages FINRA provisioned entitlements to control access to specific upload/download directories or the respective HTTPS REST endpoints. Once an application is onboard the fileX platform, access to the upload and download capabilities will be controlled by specific FINRA entitlements.

4.1 On Behalf Of (OBO)

Third parties can submit or download secure file transfers through fileX on behalf of (OBO) firms or organizations that have granted them permission. The OBO feature is currently only available through **UI or SFTP transfer methods.**

Pre-requisite

To submit or retrieve a file on behalf of another party, you must have a Prime account. See here for more information on how to create **On Behalf Of (OBO) Relationship** Requests.

5 HTTPS REST APIs

fileX service provides HTTPS REST API Endpoints to perform functions such as list of applications/files and download/upload files from/to FINRA applications. The APIs support both JSON and XML responses.

Details and sample code (for reference only) are provided for each API in the following sections. In the sample codes and other examples within this document curly braces, like {ORGID}, indicate input variables.

5.1 Security

5.1.1 OAUTH2.0

fileX authentication and authorization scheme is based on OAuth 2.0. For OAuth 2.0, there are two steps. The first call is made to FINRA Identity Platform (FIP) using credentials provided by FINRA Entitlement Service to get an OAUTH2 token. The second call is made to the fileX API endpoint with the "Authorization= Bearer" header and with the token generated in step 1.

To invoke FIP OAUTH2 API:

1. Use any HTTP client, make a HTTPS POST call with FINRA Entitlement service provided credentials as Basic Auth header.
2. Use the environment specific domain with the OAUTH URI: fip/rest/ews/oauth2/access_token
3. Use any standard library that supports OAUTH2 client credentials grant.
    a. The library should be configured with grant_type=client_credentials
    b. Client id, Client secret will be the existing user account id and password provided by FINRA Entitlement services

FIP OAUTH2 API response will be in JSON format. Example of the response:

```
{
   "access_token" :
"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

5.2 Response Codes

The FIP API can return a range of standard HTTP response codes based on the request made to the service. Table below lists the response codes and possible reasons (This is not an exhaustive list)

Response Code	Possible Reasons
200 OK	Indicates a successful request
400 Bad Request: error_message: “Invalid Credentials”	Either the username or password provided were incorrect or there is an issue with the user
400 Bad Request: error_message: “Invalid credentials format”	The Basic Authorization header’s string is malformed
400 Bad Request: error_message: “Grant type is required”	The “grant_type” parameter key was either malformed or not provided
400 Bad Request: error_message: “Only client_credentials grant type is supported”	The value for the “grant_type” parameter was something other than “client_credentials”
403 Forbidden	Indicates request to an invalid resource
500 Internal Server Error	Indicates a possible problem with FIP

fileX APIs return a range of standard HTTP response codes based on the request made to the service. Table below lists the response codes and possible reasons

Response Code	Possible Reasons
200 OK	Indicates a successful request
400 Bad Request	Indicates the request was malformed
401 Unauthorized	Indicates that either the account lacks sufficient entitlements to access requested resource or provided credentials are invalid
403 Forbidden	Indicates request to an invalid resource
404 Not Found	Indicates the file/directory is not found/available
500 Internal Server Error	Indicates a possible problem with fileX service

If you encounter issues or response code other than 200 (success), please check your request for errors/invalid content and retry.

If you still encounter issues after fixing your request, please contact FINRA Support with the response code and the request details for further assistance. See section 9 for contact information.

5.3 Request Headers

The following header parameters should be passed (as applicable) when making all the fileX REST API calls:

Header	Required (Y/N)	Default	Example
Authorization	Y	N/A	Authorization: Bearer (OAUTH Token from FIP)
Accept	N	application/json	application/json, application/xml
Content-Type	N	application/json	application/json, application/xml

5.4 REST API Catalog

5.4.1 List Available Applications

HTTP Method: GET

GET data URI: files/{ORG_ID}

Parameters	Required	Accepted Values	Case Sensitive	Description
ORG_ID	Y	Numerical values 0-9+	N	Number representing the account's orgId provided by FINRA (e.g. 0001 or 888888)

Response (JSON):

The request returns a JSON response containing the following:

Element/Attribute Key	Description
name	Name of the present working space/directory
type	Type of this object DIR = directory
metadata -> writable	denotes if this is a writable space true/false
children	Child spaces/directories accessible Name, type, and metadata values are similar to self
_links	API links to self and children


```
{
    "name": "0001",
    "type": "DIR",
    "metadata": {
        "writable": false
    },
    "children": [
        {
            "name": "app1",
            "type": "DIR",
            "metadata": {
                "writable": false
            }
        },
        {
            "name": "app2",
            "type": "DIR",
            "metadata": {
                "writable": false
            }
        }
    ],
    "_links": {
        "self": [
            {
                "href": "https://filex.finra.org/files/0001"
            }
        ],
        "parent": [
            {
                "href": "https://filex.finra.org/files"
            }
        ],
        "children": [
            {
                "href": "https://filex.finra.org/files/0001/app1"
            },
            {
                "href": "https://filex.finra.org/files/0001/app2"
            }
        ]
    }
}
```


Sample Code:

cURL:

```
Step1, get OAUTH2 access_token from FIP OAUTH2 API (HTTP Method: POST)

curl --location --request POST
'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials' --header
'Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
```

```
Step2, call fileX API (HTTP Method: GET)

curl https://filex.finra.org/files/0001 -H "Accept: application/json" -H "Authorization: Bearer
<access_token>"
```

Python:

```python
import base64, requests, sys, json

username = '1234ftp12'
password = '******'
# Base64 encode username:password
userpass = username + ":" + password
authorization = base64.b64encode(userpass.encode()).decode()
# Prepare headers for POST call to FIP OAUTH2
fipRequestHeaders = {
    'Authorization': 'Basic ' + authorization,
}
# Make the request to FIP
fipUrl = 'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials'
fipResponse = requests.post(fipUrl, headers=fipRequestHeaders)
fipResponseJson = json.loads(fipResponse.text)
# Retrieve access_token from FIP response
access_token = fipResponseJson["access_token"]
# Prepare headers for GET call to fileX API
fileXRequestHeaders = {
    'Authorization': 'Bearer ' + access_token,
    'Accept': 'application/json'
}
fileXUrl = "https://filex.finra.org/files/0001"
# Make the request
fileXResponse = requests.get(fileXUrl, headers=fileXRequestHeaders)
```

5.4.2 List Available Application Sub-Spaces

HTTP Method: GET

GET data URI: files/{ORG_ID}/{APPLICATION}

Parameters	Required	Accepted Values	Case Sensitive	Description
ORG_ID	Y	Numerical values 0-9+	N/A	Number representing the account's orgId provided by FINRA (e.g. 0001 or 888888)
APPLICATION	Y	Alphanumeric [a-zA-Z0-9]	Y	Name of the FINRA application (e.g. focus, crd, appX)

Response (JSON):

The request returns a JSON response containing the following:

Element/Attribute Key	Description
name	Name of the present working space/directory
type	Type of this object DIR = directory
metadata -> writable	denotes if this is a writable space true/false
children	Child spaces/directories accessible Name, type, and metadata values are similar to self
_links	API links to parent, self and children

```
{
    "name": "app1",
    "type": "DIR",
    "metadata": {
        "writable": false
    },
    "children": [
        {
            "name": "in_arcv",
            "type": "DIR",
            "metadata": {
                "writable": false
            }
        },
        {
            "name": "out",
            "type": "DIR",
            "metadata": {
                "writable": false
            }
        },
        {
            "name": "in",
            "type": "DIR",
            "metadata": {
                "writable": true
            }
        }
    ],
    "_links": {
        "self": [
            {
                "href": "https://filex.finra.org/files/0001/app1"
            }   ],
        "parent": [
            {
                "href": "https://filex.finra.org/files/0001"
            }     ],
        "children": [
            {
                "href": "https://filex.finra.org/files/0001/app1/out"
            },
            {
                "href": "https://filex.finra.org/files/0001/app1/in"
            },
            {
                "href": "https://filex.finra.org/files/0001/app1/in_arcv"
            }
        ]
    }
}
```


Sample Code:

cURL:

Step 1, get OAUTH2 access_token from FIP OAUTH2 API (HTTP Method: POST)

```
curl --location --request POST
'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials' --header
'Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
```

Step2, call fileX API (HTTP Method: GET)

```
curl https://filex.finra.org/files/0001/app1 -H "Accept: application/json" -H "Authorization: Bearer
<access_token>"
```

Python:

```python
import base64, requests, sys, json

username = '1234ftp12'
password = '******'
# Base64 encode username:password
userpass = username + ":" + password
authorization = base64.b64encode(userpass.encode()).decode()
# Prepare headers for POST call to FIP OAUTH2
fipRequestHeaders = {
    'Authorization': 'Basic ' + authorization,
}
# Make the request to FIP
fipUrl = 'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials'
fipResponse = requests.post(fipUrl, headers=fipRequestHeaders)
fipResponseJson = json.loads(fipResponse.text)
# Retrieve access_token from FIP response
access_token = fipResponseJson["access_token"]
# Prepare headers for GET call to fileX API
fileXRequestHeaders = {
    'Authorization': 'Bearer ' + access_token,
    'Accept': 'application/json'
}
fileXUrl = "https://filex.finra.org/files/0001/app1"
# Make the request
fileXResponse = requests.get(fileXUrl, headers=fileXRequestHeaders)
```

5.4.3 List Available Files in Application Sub-Space

HTTP Method: GET

GET data URI: files/{ORG_ID}/{APPLICATION}/{SUB-SPACE}

Parameters	Required	Accepted Values	Case Sensitive	Description
ORG_ID	N	Numerical values 0-9+	N/A	Number representing the account's orgId provided by FINRA (e.g. 0001 or 888888)
APPLICATION	Y	Alphanumeric [a-zA-Z0-9]	Y	Name of the FINRA application
SUB-SPACE	Y	Alphanumeric [a-zA-Z0-9]	Y	Name of the FINRA application subspace

Response (JSON):

The request returns a JSON response containing the following:

Element/Attribute Key	Description
name	Name of the present working space/directory
type	Type of this object DIR = directory
metadata -> writable	denotes if this is a writable space true/false
children -> name	Name of the file
children -> type	Type of this object FILE = file
children -> metadata -> writable	denotes if this file can be modified true/false
children -> metadata -> contentLength	Size of the file in bytes
children -> metadata -> lastModified	Time file was last modified with UTC time zone
_links	API links to parent, self and children

```
{
    "name": "out",
    "type": "DIR",
    "metadata": {
        "writable": false
    },
    "children": [
        {
            "name": "file1.zip",
            "type": "FILE",
            "metadata": {
                "writable": false,
                "contentLength": 7473,
                "lastModified": "2018-12-07T14:18:34.000+0000"
            }
        },
        {
            "name": "file2.pdf",
            "type": "FILE",
            "metadata": {
                "writable": false,
                "contentLength": 2408,
                "lastModified": "2018-11-29T18:59:07.000+0000"
            }
        }
    ],
    "_links": {
        "self": [
            {
                "href": "https://filex.finra.org/files/0001/app1/out"
            }
        ],
        "parent": [
            {
                "href": "https://filex.finra.org/files/0001/app1"
            }
        ],
        "children": [
            {
                "href": "https://filex.finra.org/files/0001/app1/out/file1.zip"
            },
            {
                "href": "https://filex.finra.org/files/0001/app1/out/file2.pdf"
            }
        ]
    }
}
```


Sample Code:

cURL:

Step 1, get OAUTH2 access_token from FIP OAUTH2 API (HTTP Method: POST)

```
curl --location --request POST
'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials' --header
'Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
```

Step2, call fileX API (HTTP Method: GET)

```
curl https://filex.finra.org/files/0001/app1/out -H "Accept: application/json" -H "Authorization: Bearer
<access_token>"
```

Python:

```python
import base64, requests, sys, json

username = '1234ftp12'
password = '******'
# Base64 encode username:password
userpass = username + ":" + password
authorization = base64.b64encode(userpass.encode()).decode()
# Prepare headers for POST call to FIP OAUTH2
fipRequestHeaders = {
    'Authorization': 'Basic ' + authorization,
}
# Make the request to FIP
fipUrl = 'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials'
fipResponse = requests.post(fipUrl, headers=fipRequestHeaders)
fipResponseJson = json.loads(fipResponse.text)
# Retrieve access_token from FIP response
access_token = fipResponseJson["access_token"]
# Prepare headers for GET call to fileX API
fileXRequestHeaders = {
    'Authorization': 'Bearer ' + access_token,
    'Accept': 'application/json'
}
fileXUrl = "https://filex.finra.org/files/0001/app1/out"
# Make the request
fileXResponse = requests.get(fileXUrl, headers=fileXRequestHeaders)
```

5.4.4 Download File from Application Sub-Space

Downloading a file is a two-step operation (after obtaining the FIP token). Details are below.

HTTP Method: GET (Step 1)

GET pre-signed URL URI: files/{ORG_ID}/{APPLICATION}/{APPLICATION_SPACE}/{FILE_NAME}

Parameters	Required	Accepted Values	Case Sensitive	Description
ORG_ID	N	Numerical values 0-9+	N/A	Number representing the account's orgId provided by FINRA (e.g. 0001 or 888888)
APPLICATION	Y	Alphanumeric [a-zA-Z0-9]	Y	Name of the FINRA application
SUBSPACE	Y	Alphanumeric [a-zA-Z0-9]	Y	Name of the FINRA application subspace
FILE_NAME	Y	Alphanumeric [a-zA-Z0-9]!@,&-='.	Y	Name of the file

Response (JSON):

The request returns a JSON response containing the following:

Element/Attribute Key	Description
url	URL used to execute the file download via GET call
expirationTime	Time URL expires. Cannot initiate download after expiration time.


```
{
    "url": "https://7234-0120-8410-filex.s3.amazonaws.com/0001/app1/in/file1.zip?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEJb%2F%2F%2F%2F%2F%
2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJGMEQCIA3Lkm%2BTB6g3XIJTUNVZGS%2BuwAbtGdczYecgdZBPjRVKAiAL8oDo0HE4dj9sSY02jGEfPKEPHEXotLRoGrm%2FJUmvvC
q0AwhOEAEaDDcyMzQwMTIwODQxMCIMiN%2BhyyAzWjdewqBrKpED%2F5VvYmk0FGaC%2Fvl9wzDQljKN1hiRtXjqaioBTrUM8sNlNLDL%2B6n%2FgUGyHSNMS3gQmsmWIjTKE
9a0wSqVA0wOqoHQCDE7gacbsntjGnJwHfkEqj0YjmPXF47KezJyrjoUZPcpZsya5kze45U%2FD5l%2BERmiKazWR7TCieSLuahX%2BqVmi4F%2BAG49uqIOLZEE7HicL7RGfQ
pdxetlBiZnX5l4wsCXY0jfdMfQlo1%2F%2FfXp0av8IQo5yWIMUo0KIWeOYkxdW3K%2FzbPiztL%2BBUf6Uago3wHY0sol1Qblld6RH386QZYwenkJIiz3NkJzIhNQE1wov%
2FQFnNb53%2BNdZX6KCLwzhPWoK%2F05cI%2FDi1E%2FgXKupVzxUjK%2BlG2GMRLqsh2j%2FkqLlkVxJhro%2BsN8%2Fq5zGNVc3SwBg1V2oXhahrTTImLbJJHwFaUU8bRF0
hy7fzCaSOHt1oPVpD%2F2Wh1RcrFTHemD2%2BEUDm2mokEMLDYwKw7FL5CapjQPKssOqD0MkB6uVmZsREAfE9PqPlaOjF35zhNKj4wh9KR8gU67AHmvmgXSeoH2KmCtXnIF5u
xFZ6ObN6CpVLZwzGDSrT0YVIPjeppuaCZsQVrtiK6Hb3t5a3oigLN64Q%2Bgp7Ci%2Fha86uNMD1ROxnBBTAvL41nCXpoTSy975bNC%2FwrTiG1GSoyLVeXDXjTGkm3pEjwfl
t6K4%2Bgb7RdzOdrATgWRug24zk5Y3Y%2Biu8OAIPXgM354tpMkXUlMeNB75qNXFotprZLIRpWkTJnL5%2Bs1AizWW10uniPQBZf4LEBaDZCZFmwZLV06hcM31TjKgDYeB%2F
w0SsgRJYAzpOPx0kDe9h5t1hTw2XGqf9WYbypSBaBOQ%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20200212T213525Z&X-Amz-SignedHeaders=ho
st&X-Amz-Expires=60&X-Amz-Credential=ASIA2Q3QSOJNNZHHAE6M%2F20200212%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=023661120b94f4a8
3c10400336bb745b2b8fdf76cc3d1d3ecfd49e2d663f36a1",
    "expirationTime": "2020-02-12T16:36:25.273-05:00"
}
```


HTTP Method: GET (Step 2)

GET file URI: <pre-signed URL received in Step 1>

Response:

You will get 200 OK HTTP response code on successful download of the file.

Sample Code:

```
Step 1, get OAUTH2 access_token from FIP OAUTH2 API (HTTP Method: POST)

curl --location --request POST
'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials' --header
'Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXX'

Step2, get the pre-signed URL from fileX API (HTTP Method: GET)

curl GET https://filex.finra.org/files/0001/app1/out/file1.zip -H "Accept: application/json" -H
"Authorization: Bearer <access_token>"

Step3, cURL command to download the file using pre-signed URL:

curl -X GET -o file1.txt "<pre-signed URL received in Step 2>"
```

Python code sample to get the pre-signed URL:

```
import base64, requests, sys, json

username = '1234ftp12'
password = '******'
# Base64 encode username:password
userpass = username + ":" + password
authorization = base64.b64encode(userpass.encode()).decode()
# Prepare headers for POST call to FIP OAUTH2
fipRequestHeaders = {
    'Authorization': 'Basic ' + authorization,
}
# Make the request to FIP
fipUrl = 'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials'
fipResponse = requests.post(fipUrl, headers=fipRequestHeaders)
fipResponseJson = json.loads(fipResponse.text)
# Retrieve access_token from FIP response
access_token = fipResponseJson["access_token"]
# Prepare headers for GET call to fileX API
fileXRequestHeaders = {
    'Authorization': 'Bearer ' + access_token,
    'Accept': 'application/json'
}
fileXUrl = "https://filex.finra.org/files/0001/app1/out/file1.zip"
# Make the request
fileXResponse = requests.get(fileXUrl, headers=fileXRequestHeaders)
```

Python code sample to download the file using pre-signed URL:

```
import requests, sys

# Make the request
fileXResponse = requests.get('<pre-signed URL received in Step 2>')
```

5.4.5 Upload File to Application Sub-Space

Uploading a file is a two-step operation (after obtaining the FIP token). Details are below.

HTTP Method: PUT (Step 1)

PUT pre-signed URL URI: files/{ORG_ID}/{APPLICATION}/{APPLICATION_SPACE}/{FILE_NAME}

Parameters	Required	Accepted Values	Case Sensitive	Description
ORG_ID	N	Numerical values 0-9+	N/A	Number representing the account's orgId provided by FINRA (e.g. 0001 or 888888)
APPLICATION	Y	Alphanumeric [a-zA-Z0-9]	Y	Name of the FINRA application
SUBSPACE	Y	Alphanumeric [a-zA-Z0-9]	Y	Name of the FINRA application subspace
FILE_NAME	Y	Alphanumeric [a-zA-Z0-9]!@,&-='.	Y	Name of the file
filex-{attribute}	N	Alphanumeric [a-zA-Z0-9]!@,&-='.	N	Optional metadata that can be attached with the file e.g. filex-InternalReference

Response (JSON):

The request returns a JSON response containing the following:

Element/Attribute Key	Description
url	URL used to execute the file upload via PUT call
expirationTime	Time URL expires. Cannot initiate download after expiration time.
trackingId	Unique Identifier to track the upload file request


```
{
    "url": "https://7234-0120-8410-filex.s3.amazonaws.com/0001/app1/in/file1.zip?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEJb%2F%2F%2F%2F%2F%
    2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJGMEQCIA3Lkm%2BTB6g3XIJTUNVZGS%2BuwAbtGdczYecgdZBPjRVKAiAL8oDo0HE4dj9sSY02jGEfPKEPHEXotLRoGrm%2FJUmvvC
    q0AwhOEAEaDDcyMzQwMTIwODQxMCIMiN%2BhyyAzWjdewqBrKpED%2F5VvYmk0FGaC%2Fvl9wzDQljKN1hiRtXjqaioBTrUM8sNlNLDL%2B6n%2FgUGyHSNMS3gQmsmWIjTKE
    9a0wSqVA0wOqoHQCDE7gacbsntjGnJwHfkEqj0YjmPXF47KezJyrjoUZPcpZsya5kze45U%2FD5l%2BERmiKazWR7TCieSLuahX%2BqVmi4F%2BAG49uqIOLZEE7HicL7RGfQ
    pdxetlBiZnX5l4wsCXY0jfdMfQlo1%2F%2FfXp0av8IQo5yWIMUo0KIWeOYkxdW3K%2FzbPiztL%2BBUf6Uago3wHY0sol1Qblld6RH386QZYwenkJIiz3NkJzIhNQE1wov%
    2FQFnNb53%2BNdZX6KCLwzhPWoK%2F05cI%2FDi1E%2FgXKupVzxUjK%2BlG2GMRLqsh2j%2FkqLlkVxJhro%2BsN8%2Fq5zGNVc3SwBg1V2oXhahrTTImLbJJHwFaUU8bRF0
    hy7fzCaSOHt1oPVpD%2F2Wh1RcrFTHemD2%2BEUDm2mokEMLDYwKw7FL5CapjQPKssOqD0MkB6uVmZsREAfE9PqPlaOjF35zhNKj4wh9KR8gU67AHmvmgXSeoH2KmCtXnIF5u
    xFZ6ObN6CpVLZwzGDSrT0YVIPjeppuaCZsQVrtiK6Hb3t5a3oigLN64Q%2Bgp7Ci%2Fha86uNMD1ROxnBBTAvL41nCXpoTSy975bNC%2FwrTiG1GSoyLVeXDXjTGkm3pEjwfl
    t6K4%2Bgb7RdzOdrATgWRug24zk5Y3Y%2Biu8OAIPXgM354tpMkXUlMeNB75qNXFotprZLIRpWkTJnL5%2Bs1AizWW10uniPQBZf4LEBaDZCZFmwZLV06hcM31TjKgDYeB%2F
    w0SsgRJYAzpOPx0kDe9h5t1hTw2XGqf9WYbypSBaBOQ%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20200212T213525Z&X-Amz-SignedHeaders=ho
    st&X-Amz-Expires=60&X-Amz-Credential=ASIA2Q3QSOJNNZHHAE6M%2F20200212%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=023661120b94f4a8
    3c10400336bb745b2b8fdf76cc3d1d3ecfd49e2d663f36a1",
    "expirationTime": "2020-02-12T16:36:25.273-05:00"
}
```


HTTP Method: PUT (Step 2)

PUT pre-signed URL URI: <pre-signed URL received in Step 1>

Header:

The following headers are required for this PUT operation:

Element/Attribute Key	Description
Content-Length	Size of the file to be uploaded in bytes (0 or +ve integer)
Content-Type	application/octet-stream

Response:

You will get 200 OK HTTP response code on successful upload of the file.

Sample Code:

```
Step 1, get OAUTH2 access_token from FIP OAUTH2 API (HTTP Method: POST)

curl --location --request POST
'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials' --header
'Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXX'

Step2, cURL to get the pre-signed URL (HTTP Method: PUT)

curl -X PUT https://filex.finra.org/files/0001/app1/in/newFile.zip -H "Accept: application/json" -H
"Authorization: Bearer <access_token>"

Step3, cURL to upload the file using pre-signed URL in Step 2 (HTTP Method: PUT):

curl -X PUT -T ./newFile.zip <pre-signed URL received in Step 2>
```

Python (Step 1,2):

```python
import base64, requests, sys, json

username = '1234ftp12'
password = '******'
# Base64 encode username:password
userpass = username + ":" + password
authorization = base64.b64encode(userpass.encode()).decode()
# Prepare headers for POST call to FIP OAUTH2
fipRequestHeaders = {
    'Authorization': 'Basic ' + authorization,
}
# Make the request to FIP
fipUrl = 'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials'
fipResponse = requests.post(fipUrl, headers=fipRequestHeaders)
fipResponseJson = json.loads(fipResponse.text)
# Retrieve access_token from FIP response
access_token = fipResponseJson["access_token"]
# Prepare headers for GET call to fileX API
fileXRequestHeaders = {
    'Authorization': 'Bearer ' + access_token,
    'Accept': 'application/json'
}
fileXUrl = "https://filex.finra.org/files/0001/app1/in/newFile.zip"
# Make the request
fileXResponse = requests.put(fileXUrl, headers=fileXRequestHeaders)
```

Python code to upload the file using pre-signed URL in Step 2 (Step 3):

```python
import requests, sys
# Prepare for put header
requestHeaders = {
    'Content-Type': 'application/octet-stream'
}
# Make the request
with open('newFile.zip', 'rb') as f:
    fileXResponse = requests.put('<pre-signed URL received in Step 2>', headers=requestHeaders, files=('newFile.zip': f))
```

5.4.6 Track File uploaded to Application Sub-Space

HTTP Method: GET

GET data URI: /tracking?id={TRACKING_ID}

Parameters	Required	Accepted Values	Case Sensitive	Description
TRACKING_ID	N	UUID in alphanumeric with '-'	Y	Unique identifier returned when file was uploaded to an application sub-space

Response (JSON):

The request returns a JSON response containing the following:

Element/Attribute Key	Description
trackingId	Unique identifier returned when file was uploaded to an application sub-space
submissionTime	Time at which file was submitted
userName	Username of the account which was used to upload the file
fileName	Name of the file submitted
status	Two possible values: Request Received (pre-signed URL sent by fileX for the upload request) Received by FINRA (file received by fileX through the pre-signed URL)
downstreamStatus	Four possible values: RECEIVED - (file received by fileX) ACCEPTED - (FINRA has received and successfully processed the file) REJECTED - (FINRA has received and found processing error(s) with the file. The entire file needs to be resubmitted) PARTIALLY_REJECTED - (FINRA has received the file and found processing error(s) with the file. A new file with only corrected data needs to be submitted again.)
statusUpdatedTime	Time when status was last updated. Will be used in future by applications to update file processing status
fileSize	Size of the file submitted
uploadLocation	Location at which file was submitted
application	Application the file was uploaded to
applicationSpace	Specific application space the file was uploaded to
arcvFileName	Name of the submitted file in in_arcv location
downloadable	If the arcv file is available for download (If the submission time is within 30 days)

Note: /tracking without a specific identifier {TRACKING_ID} would return status of all the submissions made in last 90 days.

```
[
    {
        "trackingId": "50e1b1c5-6150-4423-9da6-3c51ea34fc30",
        "submissionTime": "2019-03-01T13:55:50.795-05:00",
        "userName": "filexhttpuser11,
        "fileName": "file1.txt",
        "status": "Request Received",
        "statusUpdatedTime": "2019-03-01T13:56:01.838-05:00",
        "fileSize": 0,
        "uploadLocation": "0001/app1/in",
        "_links": [
            {
                "rel": "self",
                "href": "https://filex.finra.org/tracking?id=50e1b1c5-6150-4423-9da6-3c51ea34fc30"
            },
            {
                "rel": "parent",
                "href": "https://filex.finra.org/tracking"
            }
        ]
    }
]
```


Sample Code:

Step 1, get OAUTH2 access_token from FIP OAUTH2 API (HTTP Method: POST)

```
curl --location --request POST
'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials' --header
'Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
```

Step 2, invoke fileX API (HTTP Method: GET)

```
Curl -X GET https://filex.finra.org/tracking?id=50e1b1c5-6150-4423-9da6-3c51ea34fc30 -H "Accept:
application/json" -H "Authorization: Bearer <access_token>"
```

Python:

```python
import base64, requests, sys, json

username = '1234ftp12'
password = '******'
# Base64 encode username:password
userpass = username + ":" + password
authorization = base64.b64encode(userpass.encode()).decode()
# Prepare headers for POST call to FIP OAUTH2
fipRequestHeaders = {
    'Authorization': 'Basic ' + authorization,
}
# Make the request to FIP
fipUrl = 'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials'
fipResponse = requests.post(fipUrl, headers=fipRequestHeaders)
fipResponseJson = json.loads(fipResponse.text)
# Retrieve access_token from FIP response
access_token = fipResponseJson["access_token"]
# Prepare headers for GET call to fileX API
fileXRequestHeaders = {
    'Authorization': 'Bearer ' + access_token,
    'Accept': 'application/json'
}
fileXUrl = "https://filex.finra.org/tracking?id=50e1b1c5-6150-4423-9da6-3c51ea34fc30"
# Make the request
fileXResponse = requests.get(fileXUrl, headers=fileXRequestHeaders)
```

6 S3 Direct Transfer

The fileX service supports S3 Direct Transfers, allowing customers to perform upload/download actions directly to the fileX S3 bucket. Customers can perform S3 actions by first obtaining a S3 STS access token from the fileX REST API for AWS S3. Customers should use the S3 STS access token to upload/download using AWS S3 APIs or AWS CLI or AWS SDK wrapper libraries.

6.1 Generate Token for AWS S3

Customers can call the fileX GenerateToken REST API by using the OAUTH token mentioned in section 5.1.1. The REST API returns a S3 STS access token containing a set of access keys, as described in the response json below, which should be used to send/receive files. The generated access token is valid for 1 hour and allows file transfer for the requested application only. If you want to transfer files to a different application, make another API call to request a new access token for that application.

HTTP Method: GET

GET token URI: /S3TransferTokens/{ORG_ID}/{APPLICATION}

Parameters	Required	Accepted Values	Case Sensitive	Description
ORG_ID	Y	Numerical values 0-9+	N/A	Number representing the account's orgId provided by FINRA (e.g. 0001 or 888888)
APPLICATION	Y	Alphanumeric [a-zA-Z0-9]	Y	Name of the FINRA application (e.g. focus, crd, appX)

Response (JSON):

The request returns a JSON response containing the following:

Element/Attribute Key	Description
region	AWS region where S3 bucket is located
sessionName	Unique identifier generated by fileX for the request to generate the token
readPaths	List of application spaces available to the users to download file(s)
writePaths	List of application spaces available to the users to upload file(s)
accessKeyId	20 character alphanumeric string that identifies the temporary access token
secretAccessKey	40 character alphanumeric string used to sign the request
sessionToken	Token that users must pass to the service API to use the access token
expiration	Time at which current access token expire

Note: Token is valid for 1 hour only. Users will have to generate and use new tokens to upload/download additional file(s) to AWS S3 after the previous token is expired.


```
{
    "region": "us-east-1",
    "sessionName": "username@373a8cb3-8c38-40cd-a1ed-eaff73319d3a",
    "readPaths": [
        "bucket-filex/0001/app1/out/",
        "bucket-filex/0001/app1/in_arcv/"
    ],
    "writePaths": [
        "bucket-filex/0001/app1/in/"
    ],
    "credentials": {
        "accessKeyId": "ASIASCHVEHT4NOKPGZO6",
        "secretAccessKey": "+YkWP7X8Q8b4yL7ZCjsTGglKERdPtglG1sVG5BY0",
        "sessionToken": "FQoGZXIvYXdzEP//////////wEaDPW1H/sNtSay9UJRGyLPBDgFy6ydYzlTha/
        XpmEeZO5fwhx2aNXNjLCrFPj/5xHHX8DbGAqxVAHt4qIgXw7N7cRlyOPZ7JM1R/Y0uQNqNZeyHvIE9c6EYoJ/oykFRmIBC8aTGvualrF7AHU8S5O9c5Pv5udfL3oQatK
        TWljLY9MC08YPyFIZ/10lX2kjJYsHh448AaF+zVYfcl0MqUA4pP2SlzF6b/9RtXvEi2RmDB892hdXDhsFrgJPMuI4S4QXBJp3bko31ar2OJZaR81my0MIRcqPUrB/
        RM4aweF62AuTARCPJnccp35Mm0jIAjxsUU9ustsdjeHchNHnUQMLXDIMPvMx/
        FVg1I34OnlRIimIBWdWYEBz4lAAT7ikU4PIvRNZAJHLzH9Zh+ort2lPrnwCrEolJG715we06xnQllHRgSaUwbcKmBGzk2QSPNgIw3AAs4nS5tEyh/
        Ns+/1kC3Brgfrevgm2AzgX1ImYTq6X/flcXdmbTZY4BKOsBziWefey+TXJN75MQ0O5DgTtwNPWQlLkDa+5KLziv04tx4NgQ+oYGqZi1Sn1a/QDsfOboZnq3f3eOmKTVX
        vzs7N5vnIrWf2d7In24VyI7s7KFHoAfq2d32x2omz+1JMXShV2E01gnpp68cG5uEu7Id8VxpTBxvylBxTs0HuBZLBvZBOwslNc4Xn9RaFG6qJ+MSc2t9aWVPDN60VSkX
        NARSS1bylLZYYp4LgD3fEFEB5wLyrRliFKvDTLgNKtHwny9npFsa4Vt+PjIVci35ylMc3Q6GJ9euqyziDY+Gy6ndWfoiiKoNXkBQ==",
        "expiration": "2019-03-22T22:13:46.000Z"
    }
}
```

Sample Code:

```
Step 1, get OAUTH2 access_token from FIP OAUTH2 API (HTTP Method: POST)

curl --location --request POST
'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials' --header
'Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXX'

Step 2, invoke fileX API (HTTP Method: GET)

curl -X GET https://filex.finra.org/S3TransferTokens/0001/app1 -H  "Accept: application/json" -H
"Authorization: Bearer <access_token>"
```

Python:

```python
import base64, requests, sys, json

username = '1234ftp12'
password = '******'
# Base64 encode username:password
userpass = username + ":" + password
authorization = base64.b64encode(userpass.encode()).decode()
# Prepare headers for POST call to FIP OAUTH2
fipRequestHeaders = {
    'Authorization': 'Basic ' + authorization,
}
# Make the request to FIP
fipUrl = 'https://ews.fip.finra.org/fip/rest/ews/oauth2/access_token?grant_type=client_credentials'
fipResponse = requests.post(fipUrl, headers=fipRequestHeaders)
fipResponseJson = json.loads(fipResponse.text)
# Retrieve access_token from FIP response
access_token = fipResponseJson["access_token"]
# Prepare headers for GET call to fileX API
fileXRequestHeaders = {
    'Authorization': 'Bearer ' + access_token,
    'Accept': 'application/json'
}
fileXUrl = "https://filex.finra.org/S3TransferTokens/0001/app1"
# Make the request
fileXResponse = requests.get(fileXUrl, headers=fileXRequestHeaders)
```

After getting the access token using fileX S3Token API, customers can use any of the AWS S3 APIs, AWS CLI or AWS SDK wrapper libraries to upload/download files. Here is a sample AWS CLI Code to upload a file to an application sub-space

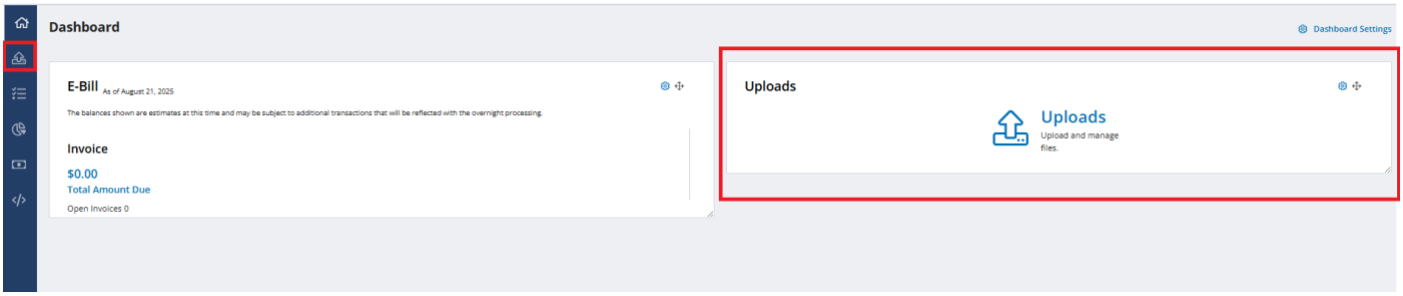
```
export AWS_ACCESS_KEY_ID=ASIA2Q3QSOJNCHBALYNN
export AWS_SECRET_ACCESS_KEY=phshuUXUANHMXx9awFsZx7YjSXi074t4dsM2nG2p
export AWS_SESSION_TOKEN=FQoGZXIvYXdzEGoaDMxSLP2oOaRUA5QzyiKoBE/BrAiV9FRPb/i1ip/rXfBnRgMT0DCo5eMa2J48lK5//gYcRCVOIcCzfjW9v+HFkQOBrorcBso9ROiZBHVBIAjJWNNjyb
0taxSvU0QzSzTioWA+UiPDnhpgUy+fC1PnfgXAAbugy61xu1rolyW0L9TWhCfnURzsbH0UD8+HMZtSqGSsEQgljrkb/9Gxad+NXFgQ568cheffk9yn25NcaER95R1lSi60/dEIE2WZkA10h/
  Xt12dpQeesA+vKEWwAq2bN3DwB0YeoG8IA60x1lckn8Mu6LJQ8dPcWbxwzgZpTG5cDPdqR6/C6XvuFa9l8yelxMFBBNf7PoRaOx1KTeT/m/d+Sd/LWpa/JlaTeCFkFoBPkeG/
  gpJPLrvrw1KuFxQkVVcRfFwdTkezauvQFov4D9JQZkMaHFs6nWMitxrrTZQUVtgzZd5BayxDLmyoK9q2gE2fQfg9RrSFarSDs3xynR3dl94nbiH5PjzonGI1saHe4Rm0cT+Ta8f07iDB/
  T+nU5Kb8c4t7alyh2rXc+KXBBiRrtCWVCocYbcMZFUzBKMoO2xNa5msimSM21CC/
  e3jPF14erS0EsAyu+qXBX2NjJoV0kqQ0UaRa1YvgGTLdR82WzfblxFfUMznjoWIt7uZmNh8/fqmpV6NsmAXHUVck+SnIrGAXLmHgW6CFZDS/R/
  cLByLXV54zkDqATLezheYIGhC2XsGzFYukAM3BD39akr2UWg0iwSjjo/zjBQ==

aws s3 cp test.txt s3://bucket-filex/0001/app1/in/test.txt
```

Note: Latest documentation on upload/download using AWS S3 APIs or AWS CLI or AWS SDK wrapper libraries can be found here - https://docs.aws.amazon.com/AmazonS3/latest/dev/MakingRequests.html

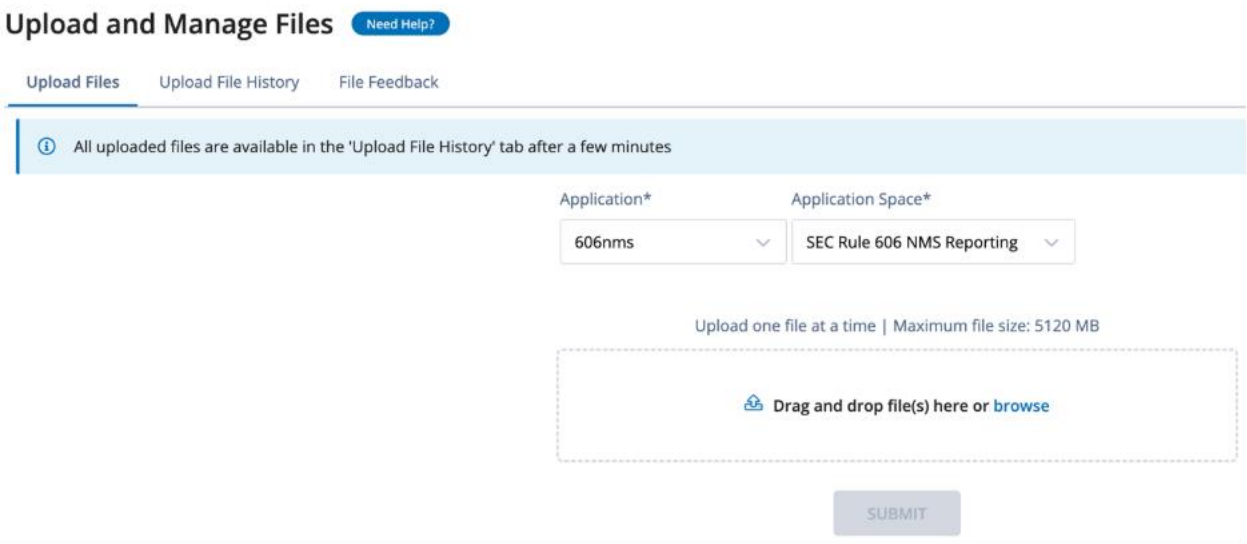
7 Upload App in FINRA Gateway

Complete the following steps to upload your file with FINRA Gateway:

1. Log in to FINRA Gateway: https://gateway.finra.org/. In case of using the On Behalf Of feature, login using your prime account.
2. Continue to multi-factor authentication and complete the prompt in your MFA client. For example, Duo is an MFA client. In case of using On Behalf Of feature, you will see your Prime account listed first, followed by profiles from organizations that have given you permission to submit files for them. **Profile accounts always start with [obo+the Org ID]**. Select the profile you want to submit on behalf of.
3. Select Upload in the dashboard page or from the left menu bar

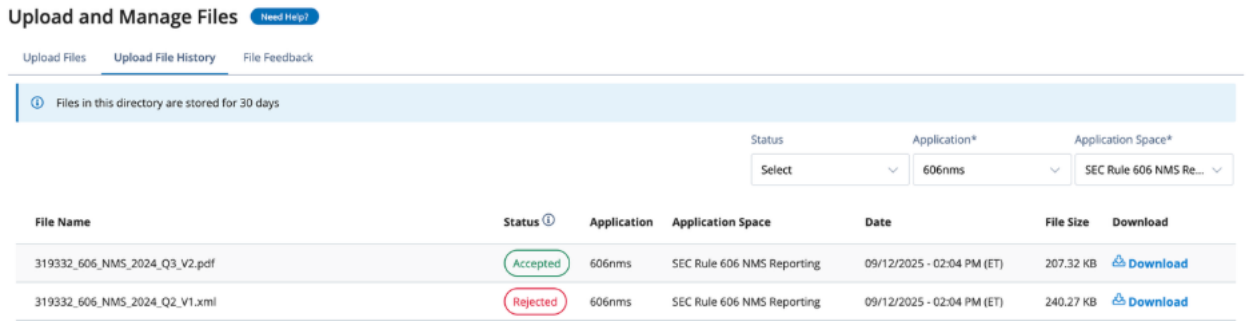


4. Make your selections for Application and Application Space, drag and drop the files and select **submit**



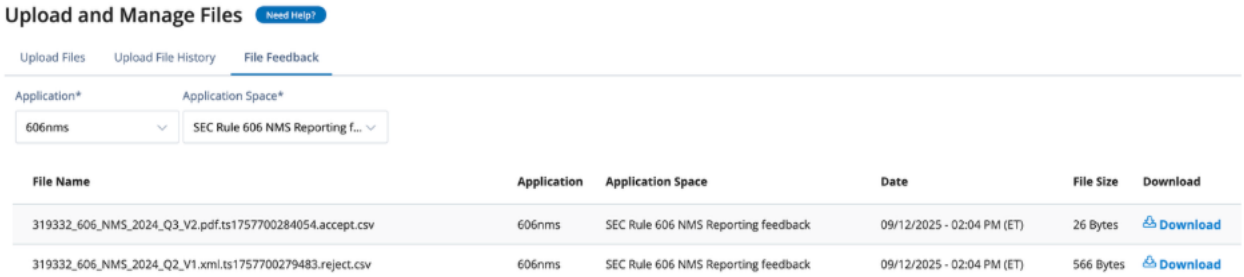
5. Upload File History

You can check the file's status in **Upload File History**. If your file is rejected, File Feedback will show the reason why it was rejected.



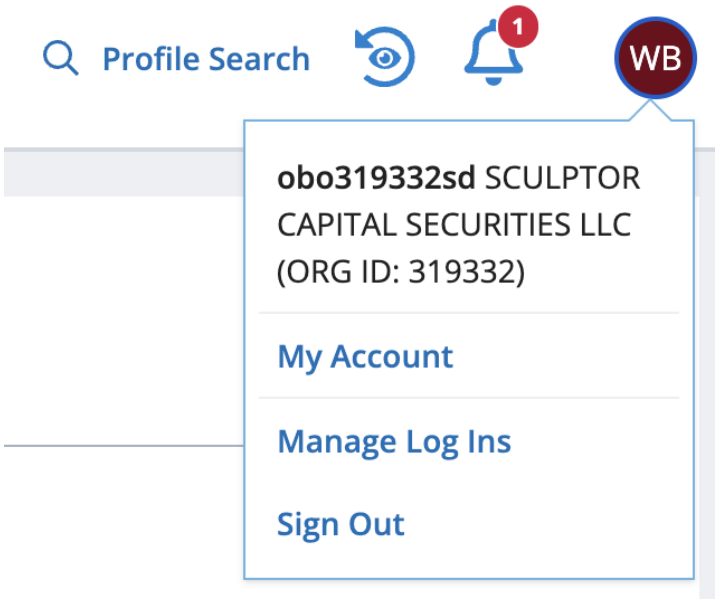
6. File Feedback

You can check for information about your submission on the File Feedback tab.

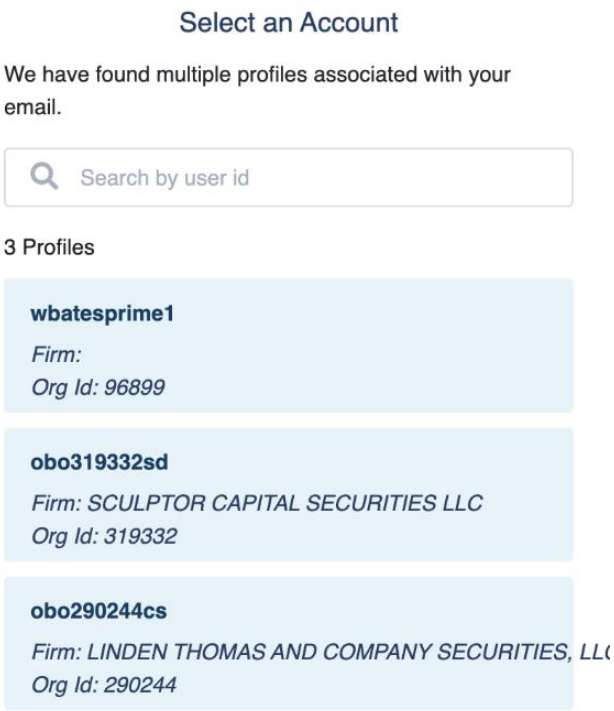


7. Switching Profiles while using On Behalf Of (OBO) feature

To switch profiles, go to your account and select **Manage Log Ins**.



From the login screen, you can change accounts, complete MFA, and then you can upload files for another profile.



8 Troubleshooting & FAQs

1. What are the supported and unsupported SFTP actions?

SFTP commands supported by fileX: put, get, cd, ls

SFTP commands not supported by fileX: mkdir, rmdir, ln, chgrp, chown, chmod, move, rename

fileX SFTP does not support put append or put resume.

2. What happens when a file is re-uploaded (uploaded twice)?

FINRA handles every uploaded file as a separate submission and delivers to the application that processes that submitted data. Please contact the FINRA support team for guidance in handling re-uploads in specific scenarios.

3. Can I do recursive calls to upload multiple files through fileX SFTP service?

fileX does not allow recursive PUT operations.

4. How do I view the files that I submitted?

fileX service keeps a copy of uploaded files in an archive sub-space, corresponding to the incoming application sub-space where file was originally uploaded. All files are retained in the archive sub-space for 30 calendar days.
How to track a file's progress?

File Tracking API allows users to track files uploaded through REST API. Please refer to section 5.4.6 for more details. For SFTP and S3 Direct, please contact the FINRA support to check the status of a submitted file.

5. Are there any restrictions on the file name?

fileX service allows all letters (a-z)(A-Z), numbers (0-9) and special characters (: ! @ , & - = ' _). However, file name should not contain '#' as anything after this character in the filename would be ignored.

Each FINRA application may enforce a certain naming convention, please check with that application contacts for additional restrictions.

6. Are there any restrictions on the file size?

fileX service does not put any limit on the file size for SFTP. However, it is recommended to keep the file size under 60GB to have better control over the file transfer process.

If you have a need to transfer very large files, please contact FINRA support to facilitate such transmissions.

7. Can I change the file immediately after it is uploaded?

fileX does not allow any changes/operations to the file like changing the owner (CHMOD) or renaming the file. If customers perform these tasks after uploading the file, it will become unusable, and customers will have to re-upload the file.

Some SFTP clients attempt these actions by default, so they should be disabled.

8. What could be the problem if the file is still in the upload directory and has not moved?

This may happen due to variety of reasons ranging from slow network connection to large file size. It is recommended to wait for few minutes for fileX service to pick up the file(s) before reaching out to FINRA support team.

9. I am not able to see the folder to upload/download file/s. What could be problem?

This could be due to lack of proper/required FINRA entitlements to the specific application you are trying to upload/download. Please contact FINRA support team.

10. I am not able to upload any file. What could be problem?

fileX allows upload operation only to specific directories for the respective FINRA application. Make sure you are trying to upload to the correct directory. See section 9 for FINRA Application specific directory information. In addition, you need to make sure the account performing the upload has the proper entitlements for that application. Please contact FINRA support team for questions related to FINRA Entitlements.

11. Am I allowed to rename or delete file/s in download location?

fileX does not allow rename or delete operations on files at the download locations.

12. How many files can I transfer together using fileX service?

SFTP and AWS S3 Direct transfer methods do not have any limit on the number of files that could be transferred in a single operation, but HTTPS REST method allows only ONE file per API call.

13. What happens when two sessions from the same firm log in with the same entitlements?

Both sessions will see the same directories and files. fileX service moves the file(s) immediately after the upload is complete, so a file that gets uploaded in session 'A' may "disappear" from view of session 'B'.

14. Is there a SFTP Client UI to exchange file(s) through fileX?

fileX service with FINRA is tested with some SFTP clients like FileZilla, SecureFX and WinSCP.

15. I am a new member/customer of FINRA. How do I request access to fileX services?

As mentioned in this document, fileX service access is controlled by credentials provided and permissioned by FINRA Entitlement services. Please contact FINRA support to request access. You may also check https://www.finra.org/industry/entitlement-program for information related to Entitlement process.

16. I am not able to connect. What should I do now?

Your account may not be active or locked out. Also check with FINRA support team to check the status of the account and also ensure it has correct privileges to use fileX services.

17. My account got locked out, how do I reset my account?

The account will be locked out if it connects with invalid credentials, please contact FINRA support for further help with the account.

18. Does fileX SFTP support SSH Keys?

Yes, we support key based auth for SFTP.

19. My connection got disconnected while I was uploading the file, what should I do now?

You may still see the file in the folder but it is highly recommended to upload the file again. fileX must receive a successful upload complete status from client applications before it starts processing file(s). When re-uploading the same file, name sure the file name is not changed.

20. I am getting network connection errors while trying to connect to fileX and perform file transfer operations using SFTP. What could be the issue?

Network connection errors may occur because IP Address or DNS names are not whitelisted by either your network/firewall department or FINRA's network/firewall department. Please ensure you have whitelisted fileX SFTP IP addresses as required and verify outflow of traffic at your end. Inform FINRA of your outgoing IP address (internet routable IP addresses) so it could be whitelisted on FINRA's side (see also question 22) by contacting FINRA support team.
Network connection errors could also happen due to encryption key related issues. If the problem still persists after whitelisting IP addresses, please contact FINRA support team to troubleshoot.

21. I am currently transferring files through other FINRA SFTP systems and have provided IP's to whitelist in the past. Do I need to request to whitelist lP Addresses again to use fileX?

No, FINRA will retain the previously whitelisted IP address for fileX service also.

22. How can I check the health of fileX service status?

You can check the status of the fileX service through the below health check URL: https://filex.finra.org/health.

23. Why am I getting 'Too Many Connections Error' with SFTP method?

You may have too many concurrent connections opened to fileX. Make sure your connections are terminated gracefully after successful upload/download operations.
As a good practice avoid frequent connections to check for a file, only connect at the expected file availability time. Contact the respective FINRA Application to know the expected file availability time/s.

24. My SFTP session is timing out during file transfer. What could be the issue?

This could happen due to network related issues at your end, please check with your network support to ensure proper outbound data traffic flow. If you are still getting this error, please contact FINRA support team to troubleshoot.

25. Can I make a POST request to download the file?

No, fileX only supports GET method to download the file

26. Which coding/scripting languages are supported when calling HTTPS REST API endpoints?

Any client language can be used to call fileX API endpoints as long as the requests are valid REST calls.

27. Do I need to provide authentication headers again when making call to pre-signed URL to download the file?

No. You do not have to provide the authentication details again to download the file. URL is pre-signed and valid for 1 minute following the GET request. Please make sure to initiate the download request within the 1 minute time limit. If you try to download after the expiration time, the download operation will fail with error message similar to below

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
    <Code>AccessDenied</Code>
    <Message>Request has expired</Message>
    <X-Amz-Expires>59</X-Amz-Expires>
    <Expires>2018-12-11T19:58:13Z</Expires>
    <ServerTime>2018-12-11T23:38:49Z</ServerTime>
    <RequestId>CD51795F7323938B</RequestId>
<HostId>GZmAO5Wk70of2S+xkd47hM2oT2UYCs7z9IfZy++C4qmR/cU5nM1paCih0LH7JmfrxTyRHdeqkNI=</Host
Id>
</Error>
```

28. Can some users within the same firm use SFTP and other APIs?

Yes. fileX service experience will be consistent across all the supported methods. Same files will be available in all the supported methods.

29. Why I am getting Not Found 404 Error Code when I make tracking API request?

It is a standard HTTP Response Code you get when the tracking ID provided in the API request is either invalid or older than 90 days

30. Is tracking API supported for files uploaded through all the access methods?

Tracking API from fileX is only supported for upload requests made through HTTP REST APIs.

31. How can I renew the AWS S3 access token after it expires?

You have to call the S3TransferToken API from fileX again to get the new access token.

32. Can I use S3 multipart upload?

Yes. You can find the latest AWS documentation here - https://docs.aws.amazon.com/AmazonS3/latest/dev/uploadobjusingmpu.html

33. Can I download files from archive sub-space?

Yes, a user with right entitlement can download file available in the archive sub-space

34. Is file archive supported for all the access methods?

Yes, files uploaded through any of the supported access methods, SFTP, REST APIs and S3 Direct, gets archived for 30 days.

35. Is it recommended to connect to the hostname or IP addresses directly for SFTP connectivity?

It is recommended to connect to the hostname referenced in Section 3: Environment and Connectivity

36. File download is interrupted and ends up in a partial file. What could be the reason?

Some of the most common reasons are

- Insufficient space at the location where the file is being downloaded to.
- Some SFTP client libraries are unable to handle packets out of order and the file download stops after a certain size. As a workaround, you can set the QueueDepth/queue_size to 1 which will force the packets to arrive in order. This may slowdown the overall file download operation when downloading very large files.

37. What are the Ciphers supported in TLS v1.2?

The following Ciphers are supported in TLS v1.2

- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- AES128-GCM-SHA256
- AES128-SHA256
- AES256-GCM-SHA384
- AES256-SHA256

38. How long do the FIP tokens last?

The token lasts for 12 hours.

9 Contact Information

Please call FINRA support at 800-321-6273 or visit the web page at https://tools.finra.org/cc_support/ to open a support request. Please provide the following information when contacting support to help us troubleshoot issues faster:

1. Org ID
2. User account
3. Access method (SFTP, HTTPS REST or AWS S3 Direct)
4. Application space (where you are trying to transfer files)
5. File Name
6. Brief description of the error (include information on tool/language used)

10 Supported Applications and Relevant Parameters

This section lists the application sub-spaces and retention policies for the respective applications in fileX. This section will be updated as FINRA applications migrate to fileX service but if any application space you are trying to use is not listed below, please contact FINRA support to get details.

Key:

- *Immediately Moved* - these files are moved out of the indicated sub-space as soon as they are processed by the downstream applications. There should be no expectation that the files in these application sub-spaces will persist beyond the transfer itself.
- *N days* - Number of days for which files will remain in the indicated application sub-space. Files will be removed from the application sub-space after N days have passed.

Application Name	Application Sub-Spaces	Allowed Actions	File Retention duration
ACATS	{orgId}/acats/in	Upload	Immediately Moved
ADDS	{orgId}/tradejrnls/{various}	Download	30 calendar days
Advertising	{orgId}/adv/in	Upload	Immediately Moved
	{orgId}/adv/out	Download	30 calendar days
Bluesheets	{orgId}/blshts/finra_in	Upload	Immediately Moved
	{orgId}/blshts/sec_in	Upload	Immediately Moved
CCCS	{orgId}/cccs/in	Upload	Immediately Moved
	{orgId}/cccs/out	Download	30 calendar days
cboe (DNC)	{orgId}/cboednc/in	Upload	Immediately Moved
CRD	{orgId}/crd/incoming_designations	Upload (Designating Authorities Only)	Immediately Moved
	{orgId}/crd/incoming_filings	Upload (Firms Only)	Immediately Moved
	{orgId}/crd/results	Download	30 calendar days
	{orgId}/crd/reports	Download	30 calendar days
eFOCUS	{orgId}/focus/in	Upload	Immediately Moved
	{orgId}/focus/out	Download	30 calendar days
INSITE	{orgId}/ifdf/in	Upload	Immediately Moved
	{orgId}/ifdf/out	Download	30 calendar days
phlx (DNC)	{orgId}/phlxdnc/in	Upload	Immediately Moved
Stock Record and Allocations	{orgId}/stockrecord/in	Upload	Immediately Moved
	{orgId}/stockrecord/out	Download	30 calendar days
4530	{orgId}/r4530/in	Upload	Immediately Moved
	{orgId}/r4530/out	Download	30 calendar days
Short Interest Reporting	{orgId}/shorts/in	Upload	Immediately Moved
	{orgId}/shorts/out	Download	30 calendar days
Regulatory Extensions (REX)	{orgId}/rex/in	Upload	Immediately Moved
	{orgId}/rex/out	Download	30 calendar days
Regulation M Filings	{orgId}/regm/in	Upload	Immediately Moved
	{orgId}/regm/out	Download	30 calendar days

Note: By default, every incoming application sub-space will have a corresponding <application sub-space>_arcv space. This sub-space will store files received within the last 30 days. The filenames will be prefixed with a timestamp yyyymmddhh24missSSS_<filename>. Timestamp is in US Eastern.

An example would be /0001/app1/in_arcv/20190322190521330_file1.zip