File*Catalyst* Server

General REST API Use Cases

Contents

Copyright © 2020 Unlimi-Tech Software, Inc.

Copyright © 2020 Unlimi-Tech Software, Inc.

Introduction

This document is designed to serve as a language-agnostic guide on how to carry out certain workflows and procedures via the FileCatalyst Server REST services. The majority of these operations all have the same initial requirements:

1. Have an instance of FileCatalyst Server installed and licensed.

2. Configure the FileCatalyst Server administration credentials for REST authentication. This can be done through the “Administration” panels located on the FileCatalyst Server administration GUIs.

Copyright © 2020 Unlimi-Tech Software, Inc.

1 Authenticating REST Services

To perform REST services against the FileCatalyst Server, you must authenticate the REST calls that you are be submitting. The authentication process requires that you construct an **encoded authentication token** containing the credentials for the FileCatalyst Server administration account that you are using. To construct this token you need to Base64 encode the username and password delimited by a ":". For example:

- Sample admin account username: admin
- Sample admin account password: system
- Sample non-encoded token value: admin:system
- Sample encoded token value: YWRtaW46c3lzdGVt

With the encoded authentication token now defined you can now use it to authenticate your REST calls with the FileCatalyst Server. There are two methods to perform this authentication:

1.1 Authentication via the "RESTAuthorization" header

To authenticate the REST service, construct the REST call that you wish to perform, and then add the "**RESTAuthorization**" header to the request.

REST Example
RESTAuthorization: YWRtaW46c3lzdGVt
HTTP Method: GET
REST URL: http://localhost:12480/rs/system

Note: If you use this method to authenticate your REST services you have to supply the RESTAuthorization header with every REST request made.

1.2 Authentication via the "rs/AuthorizationSessionRev1" REST resource (More secure)

This version is more secure and is recommended for using the REST services for FileCatalyst Server. To use this authentication method you have to execute a REST call to the **"rs/AuthorizationSessionRev1"** resource. Doing this causes the Server to authenticate the encoded token and establish your individual session as a trusted session with the Server. As long as the credentials remain the same all calls made with the session will be fully authenticated and will not require any additional data to be provided to the system.

REST Example:
HTTP Method: POST
REST URL Example: http://localhost:12480/rs/AuthorizationSessionRev1
API Documentation for this resource: Documentation Link

Copyright © 2020 Unlimi-Tech Software, Inc.

2 FileCatalyst User Use Cases

2.1 Adding a FileCatalyst User to the Server

To add a new User account to the Server, you must first generate a UserModel object that contains all of the individual configuration values that you want that User to have. Once the object has been constructed you create a REST payload containing the data and then send that request as POST to the "rs/users" REST entry point. Upon submission, the Server will validate the resource and then add it to the system for future use.

REST Example:
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/users
API Documentation for this resource: Documentation Link

Note 1: If you have a predetermined list of groups and/or virtual folders that you want the new User to be connected to then you can add those connections to the UserModel data "group" and "folder" properties. Simply append the requested connections into the two arrays and the Server will ensure that the connections are established once the user is created.

2.2 Retrieving information for a User on the Server

To request the current data for all users defined on the Server make a GET request to the "rs/users" resource. Submitting the request provides you with an array containing all of the Users and their currently defined configurations. If you are only interested in a particular User on the Server then you can make a GET request to "rs/users/{username}". This will return only the individual UserModel object for the user that you requested information for.

REST Example (All users):
HTTP Method: GET
Example REST URL: http://localhost:12480/rs/users
API Documentation for this resource: Documentation Link

REST Example (Single User):
HTTP Method: GET
Example REST URL: http://localhost:12480/rs/users/exampleUser
API Documentation for this resource: Documentation Link

2.3 Editing a User on the Server

To edit the data for a User, construct a UserModel data object and send that data as part of the payload to the "rs/users/{username}" REST resource.

REST Example:
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/users/exampleUser
API Documentation for this resource: Documentation Link

Copyright © 2020 Unlimi-Tech Software, Inc.

2.4 Deleting a User from the Server

To delete a User from the Server you can evoke one of two available REST services. The first is the DELETE method that is available to the "rs/users" REST resource. This method allows you to supply a GenericIDModel to bulk delete a series of users. If you want to delete only a single user then you can use the DELETE call located at the "rs/users/{username}" REST entry point.

REST Example (Bulk Delete):
HTTP Method: DELETE
Example REST URL: http://localhost:12480/rs/users
API Documentation for this resource: Documentation Link

REST Example (Single Deletion):
HTTP Method: DELETE
Example REST URL: http://localhost:12480/rs/users/exampleUser
API Documentation for this resource: Documentation Link

2.5 Linking a User to groups and virtual folders

To link a user to a collection of groups and virtual folders there are a couple of options that you can use:

2.5.1 Managing individual links

To add a single link between a user and a group or virtual folder you can evoke the "rs/users/{user}/link" resource and provide a LinkModel containing the information for the link that you want to establish. If you wish to link the user to a group, then you must set the "linkType" property to have the value of "GROUP". If you want to link the user to a virtual folder/file then you must set the "linkType" property to have the value of "FOLDER".

REST Example:
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/users/exampleUser/link
API Documentation for this resource: Documentation Link

To remove a linkage from a user you can construct a LinkElementModel object and then provide it as the payload for the POST method available with the "rs/users/{username}/unlink" REST service.

REST Example:
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/users/exampleUser/unlink
API Documentation for this resource: Documentation Link

2.5.2 Linking multiple resources at the same time

If you have multiple resources that you want to link at one time then you can provide them by editing the user and supplying the links through the "group" and "folder" properties of the object.

Note: Applying linkages to resources in this fashion will **remove all existing linkages** with the user and then replace them with the values that are provided.

Copyright © 2020 Unlimi-Tech Software, Inc.

3 UserGroup Use Cases

3.1.1 Adding a Group

To add a group to the Server you will have to supply a GroupModel to the "rs/groups" resource. For example:

REST Example:
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/groups
API Documentation for this resource: Documentation Link

Note 1: If you have a predetermined list of users and/or virtual folders that you want the new Group to be connected to then you can add those connections into the GroupModel data "user" and "folder" properties.

3.1.2 Retrieving the current data for a Group

To retrieve the current data for the Groups on the Server you need to invoke a GET REST call against the "rs/groups" resource. For example:

REST Example:
HTTP Method: GET
Example REST URL: http://localhost:12480/rs/groups
API Documentation for this resource: Documentation Link

3.1.3 Modifying a Group

To make changes to an existing group provide a GroupModel containing all of the group data that you want to apply to the POST method found at "rs/groups/{groupname}". For example:

REST Example:
HTTP Method: GET
Example REST URL: http://localhost:12480/rs/groups
API Documentation for this resource: Documentation Link

Note: When making changes to the group it is possible to provide a list of Users and virtual items that you want the group to be linked to. If you use this option, please keep in mind that any existing connections will be removed from the group you are making changes to.

3.1.4 Deleting a Group

Similar to the Users of the FileCatalyst Server, the Groups objects allow you to perform deletions in two different ways. The first method is a bulk deletion that can be found at the "rs/groups" DELETE REST resource. The second deletion method you can use is the DELETE call located at the "rs/groups/{groupname}" REST entry point.

REST Example (Bulk Delete):
HTTP Method: DELETE
Example REST URL: http://localhost:12480/rs/groups
API Documentation for this resource: Documentation Link

Copyright © 2020 Unlimi-Tech Software, Inc.

REST Example (Single Deletion):
HTTP Method: DELETE
Example REST URL: http://localhost:12480/rs/groups/exampleGroup
API Documentation for this resource: Documentation Link

3.1.5 Linking multiple resources at the same time

Linking groups works in a similar manner to how the User linking resource works. To link a group to a singular user or folder you can call the POST method located at ”rs/groups/{groupname}/link” to perform the operation. To unlink a virtual item from a singular group or folder, you can call the POST method located at “rs/groups/{groupname}/unlink”.

REST Example (Link resource):
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/groups/exampleGroup/link
API Documentation for this resource: Documentation Link

REST Example (Remove a link between resources):
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/groups/exampleGroup/unlink
API Documentation for this resource: Documentation Link

4 Virtual Folder/File Use Cases

4.1.1 Adding a virtual folder/file

To add a virtual folder/file to the Server you must submit a FolderModel to the POST method available on the “rs/folders”. For example:

REST Example:
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/folders
API Documentation for this resource: Documentation Link

Note 1: When a virtual folder/file is added to the Server, the REST will return with a FolderModel containing an assigned unique “**foldername**” that is based on the “label” value that was provided. If there was an existing virtual item containing the label you provided then the “foldername” will be the value of the “label” with a number appended to the end of the label. If you are going to be performing any further actions with the virtual item that you created you should make sure to keep the “foldername” value, since it will be **necessary to access the virtual item at a later time**.

Note 2: If you have a predetermined list of groups and/or users that you wish to connect the new virtual item to, then you can add those connections into the FolderModel data “user” and “group” properties.

4.1.2 Retrieving the data for a virtual folder/file

To retrieve all of the currently established virtual items and their respective configurations, construct a GET REST request against the “rs/folders” resource. For example:

Copyright © 2020 Unlimi-Tech Software, Inc.

REST Example:
HTTP Method: GET
Example REST URL: http://localhost:12480/rs/folders
API Documentation for this resource: Documentation Link

4.1.3 Modifying virtual folders/files

To make changes to an existing virtual folder/file, provide a FolderModel containing all of the configuration data that you want to apply to the POST method found at "rs/folders/{foldername}". For example:

REST Example:
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/folders
API Documentation for this resource: Documentation Link

Note: When making changes to the virtual item it is possible to provide a list of Users and groups that you want the group to be linked to. If you use this option, please keep in mind that any existing connections will be removed from the virtual item that you are making changes to.

4.1.4 Deleting virtual folders/files

Similar to both the Users and Group data objects, the Virtual folders/files have two methods for deleting resources. The first is the standard bulk deletion operation that is found at the DELETE method for the "rs/folders" resource. The second method is for individual virtual item deletions and it can be found at "rs/folders/{foldername}. For example:

REST Example (Bulk Delete):
HTTP Method: DELETE
Example REST URL: http://localhost:12480/rs/folders
API Documentation for this resource: Documentation Link

REST Example (Single Deletion):
HTTP Method: DELETE
Example REST URL: http://localhost:12480/rs/folders/exampleFolderName
API Documentation for this resource: Documentation Link

4.1.5 Linking virtual folders/files with Users and Groups

To link a virtual item to a singular group or folder, you can call the POST method located at "rs/folders/{foldername}/link" to perform the operation. This method will require a LinkPermissionsModel to be set within the LinkModel that is provided within the REST payload. This additional model defines the degrees of access that the linkage has with the virtual item and fundamentally impacts what can be done when performing operations.

REST Example (Link resource):
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/folders/exampleFolderName/link
API Documentation for this resource: Documentation Link

Copyright © 2020 Unlimi-Tech Software, Inc.

To unlink a virtual item from a singular group or folder you can call the POST method located at "rs/folders/foldername/unlink".

REST Example (Remove link between resources):
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/folders/exampleFolderName/unlink
API Documentation for this resource: Documentation Link

5 External File System (EFS) Use Cases

To perform operations against the external file systems collections of the FileCatalyst Server you must use the REST resource located at the "rs/filesystems" entry point. This resource provides you with access methods for the retrieval, addition, modification, and deletion of external file systems defined within the Server's configurations.

5.1.1 Adding a new EFS

To add a new EFS to the Server you must define a valid FileSystemModel object to be used for the payload for a POST REST call to "rs/filesystems". Once that data is ready to be submitted, execute the call and the Server's back-end resources will create the new EFS if it is able to properly secure a connection with the provided credentials.

REST Example:
HTTP Method: POST
Example REST URL: http://localhost:12480/rs/filesystems
API Documentation for this resource: Documentation Link

5.1.2 Retrieving data for a current EFS

To retrieve the data for all of the currently defined file systems on the Server, you will have to make a GET REST call to the root of the "rs/filesystems" REST entry point. Doing this will provide you with a JSON response that contains a mapping of each file system type (Example types: smb, s3, b2, gcs, etc.) to a corresponding FileSystemModel. This model object will contain basic information about the FileSystem as well as a small nested CustomFileSystemLabelsModel, that contains visual label aliases for what each file system refers to their respective username, passwords, and hostname values as.

REST Example:
RESTAuthorization: YWRtaW46c3lzdGVt
HTTP Method: GET
Example REST URL: http://localhost:12480/rs/filesystems
API Documentation for this resource: Documentation Link

5.1.3 Deleting an EFS

To delete an EFS from the Server, you must evoke the DELETE REST resource located "rs/filesystems/.
For example:

REST Example:
HTTP Method: DELETE
Example REST URL: http://localhost:12480/rs/filesystems

Copyright © 2020 Unlimi-Tech Software, Inc.

API Documentation for this resource: Documentation Link

5.1.4 Browsing the current folder roots/buckets of the EFS

Performing a GET request at "rs/filesystems/{filesystemLabel}/filelist", will return a collection of FileObjectModels that contain information about each individual file/folder/bucket that resides on the roots of the mounted filesystem.

REST Example:
HTTP Method: GET
Example REST URL: http://localhost:12480/rs/filesystems/smb/filelist/
API Documentation for this resource: Documentation Link

5.1.5 Browsing specific folders and files within the EFS

To browse an individual location contained within the file system you will need to provide the path that you want to browse as part of the call. When navigating through EFS directory trees it is recommended that you use the "path" parameter within the FileObjectModels that returned through the last browse request. By appending the value from this "path" parameter to the file list REST URL, you should ensure seamless navigation across the file system and its contents.

REST Example:
HTTP Method: GET
Example REST URL:
https://172.20.5.222:12480/rs/filesystems/swift3/filelist/swift3%3A//cloud.swiftstack.com/filecatalyst-test
API Documentation for this resource: Documentation Link

6 Retrieving local file listings from the Server

To retrieve a local file listing from the Server you need to invoke the REST resource located at "/rs/local". Submitting a request at the root level of this resource will provide you with all of the file system roots that the Server has access to. Submitting requests with a file path parameter after the "rs/local" will give you a directory listing for the specified directory.

Note 1: The "rs/files/local" resource is set up to support listings of files that follow the standard local file system syntax (Ex: C:\Users\admin, or /home/Users/admin), and traditional URL file path definitions.

6.1.1 Local file root listing

HTTP Method: GET
REST URL Example: http://localhost:12480/rs/local
API Documentation for this resource: Documentation Link

6.1.2 Specific directory local file listing

HTTP Method: GET
REST URL Example: http://localhost:12480/rs/local/C:/temp/SomeDirectory/
API Documentation for this resource: Documentation Link

Copyright © 2020 Unlimi-Tech Software, Inc.

7 Retrieving statistic information for ongoing client connections

To receive information regarding ongoing sessions of file transfer clients connected to the FileCatalyst Server you must perform a REST call against the REST resource found at **"rs/sessionRev1".** Invoking this resource will return a collection of FCSessionsModels (one for each connection), that contains all of the Session information for the current connections. Each one of these individual session objects contains helpful information such as the unique sessionID, the current status of the connection, the IP address of the client, and which FileCatalyst User the connection is going to.

7.1.1 Full session listing

HTTP Method: GET
REST URL Example: http://localhost:12480/rs/sessionsRev1
REST API Documentation Link: Documentation Link

7.1.2 Individual session listing

If you want to view the information for a particular session on the Server instead, you can perform a request to **"rs/sessionRev1/{sessionID}**". This call will limit the scope of your session lookup and only return the data relating to that session and the individual clients contained within. For example:

HTTP Method: GET
REST URL Example: http://localhost:12480/rs/sessionsRev1/M4846028585801553-1-NULL
REST API Documentation Link: Documentation Link

8 Requesting diagnostic information

Sometimes, during the Server's operations, you may encounter an issue that can't be resolved without the help of the FileCatalyst Support team. When this occurs it will be beneficial for you to capture a diagnostics report that contains a collection of useful information that the support engineers can use to assist you. To capture this information you execute a GET REST request to the **"rs/applicationDiagnostics"** REST entry point. Doing this starts the diagnostic data collection process and, once the process is complete, a zip archive will be created within the install directory under the "diagnostics" folder.

REST Example:
HTTP Method: GET
REST URL Example: http://localhost:12480/rs/applicationDiagnostics

9 Performing Server REST Calls through the Central REST Proxy

To perform Server REST calls through Central's REST proxy there are a couple of steps that you must perform:

9.1.1 Connecting the Server application to Central

To perform REST calls against a Server through Central, you must first connect the Server to a FileCatalyst Central instance. This can be accomplished by specifying the connection information of your FileCatalyst Central instance through the Server administration UI.

Copyright © 2020 Unlimi-Tech Software, Inc.

9.1.2 Determining the RESTAuthorization Header value for Central

To determine the **RESTAuthorization header** that will be needed to authenticate your REST calls against Central, you must first gather the credentials for a FileCatalyst Central user account that has access to your Server node. Once the user credentials are collected, you can construct the RESTAuthorization header in the same fashion as the one generated for the authentication with the Server.

9.1.3 Finding the Server node ID

With the RESTAuthorization now known, perform a GET request to Central's /rs/nodes REST resource. This will return the currently connected nodes within Central.

REST Example:
HTTP Method: GET
Example Central REST URL: http://127.0.0.1:8080/rs/nodes
Central REST API Documentation Link: Link

9.1.4 Listing the available REST resources for proxying

Once your Server node has been found in the current node list, you will be able to see the list of available REST services available to that node. All of these REST services use a different URL prefix than the normal Server REST services, but they function identically to the methods outlined in the Server's system. If you need help with any of the REST services listed, open the Server's REST documentation and view the service you want to access.

10 Support

10.1 Online Support Portal

Support is available online at: http://support.filecatalyst.com

Available services include:

- Submitting a trouble ticket
- Searching the Knowledgebase for solutions to common questions
- Live Chat

10.2 Email

File*Catalyst* and Unlimi-Tech Software, Inc. may be reached at: support@filecatalyst.com

10.3 Telephone

Reach us by phone at:

main: +1.613.667.2439
toll-free: +1.877.327.9387
fax: +1.613.667.5950

Copyright © 2020 Unlimi-Tech Software, Inc.