SailPoint IdentityIQ

Version: 8.0

File Access Manager SCIM API Reference Guide

Revision Table

1.0	Initial release	
2.0	Added endpoints: • DataClassificationCategories • DataClassificationResults • Capabilities •	30 Jan 2019
3.0	• New filter "parentResourceId" for BusinessResources Endpoint • New endpoint: KPIs, which returns the count, and score of the requested KPI.	21 Apr 2019

Table of Contents

1 Introduction

Welcome to the SailPoint IdentityIQ File Access Manager API. This API provides access to the IdentityIQ File Access Manager platform. The API is standards-based, built upon the RESTful SCIM 2.0 specification. You can use this API to access IdentityIQ File Access Manager API endpoints, which allows you to programmatically interact with objects within the File Access Manager.

2 Getting Started

1. For more information about the SCIM 2.0 specification, see the next section: SCIM Protocol
2. Ensure you have IdentityIQ File Access Manager version 8.0 or higher installed.
3. Read the IdentityIQ File Access Manager documentation.
4. Participate in the forums. Ask questions, read about requested and upcoming functionality, and assist others.
5. Send us feedback. We want to hear from you.

3 SCIM Protocol

SCIM (System for Cross-Domain Identity Management), is an HTTP-based protocol that makes managing identities in multi-domain scenarios easier to support through a standardized RESTful API service. It provides a platform neutral schema and extension model for representing users, groups and other resource types in JSON format.

We implement SCIM with the following restrictions:

- Filter - Currently we only support the "and" logical operator between filter expressions ("or" is not supported)
- Filter - If filter expression values include the reserved URL characters " '$-_.+!*'(),' ", they need to be changed to their encoded value
- Sorting - Currently we do not support SCIM Sorting capabilities (SortBy and SortOrder). Each method implements its own sorting by default.

4 Authentication

4.1 Basic Authentication

Basic Authentication is used to allow access to the API. It is a simple technique for enforcing access controls to API resources because it doesn't require session IDs, cookies, or login pages but instead uses standard fields in the HTTP header. For more information on Basic authentication, please see https://tools.ietf.org/html/rfc1945#section-11 and https://www.ietf.org/rfc/rfc2617.txt. Support for Basic Authentication will continue to exist in future releases.

Basic Authentication can be used by IdentityIQ File Access Managerinternal users that have the "API User" role. You can create internal users and grant them the role using the administrative client.

4.2 OAuth 2.0

The Client ID and Client Secret will be automatically generated during installation (or upgrade) of version 6.1.

For upgrades from version 6.1 or above, the client ID and client secret will remain the same.

You can find the client parameters in the "API Authentication" screen in the IdentityIQ File Access Manager web client.

4.3 "API Authentication" screen

4.3.1 Navigation

The screen can be found under Settings -> General -> API Authentication

4.3.2 General

On this screen you can:

- Check your Client ID and Client Secret
- Generate a new Client Secret

4.3.3 Get Token - Sample Request

```
"curl -X POST http://localhost/identityiqfamapi/token -H 'content-type: applicatio
n/x-www-form-urlencoded' -d 'grant_type=client_credentials&client_id=6779ef20e7581
7b79602&client_secret=mY5zM5nh7MR8gpj5yG9iIQ%3D%3D'"
```

Get Token - Sample response

```
{

"access_token": "gCV2VxetE7vgRxG77pqztGSs-3lWLTJhLG5K3dL7YbtyV6Ys1z0CnTcmv__NwTuOd
IcUq4_bM9q2xRPa8I4ab7JW31T6XVZ70eMLdAnOy3tgZpaz3UWTJwfLKEi8pqN6ZcF57kYmSKWrBYOabmY
9JrvWtqSLsTBaX9ALWgK2JADHMvpXsbqjkI2MV9xh3nIYKyTX0mW8EOZx9JhtqC3XIQ",
```

```
"token_type": "bearer",

"expires_in": 1199,

".issued": "Thu, 09 Aug 2018 08:00:21 GMT",

".expires": "Thu, 09 Aug 2018 08:20:21 GMT"

}
```


Using the access_token value you can then make requests to any SCIM endpoint using “Authorization: Bearer” in the header

4.3.4 Sample SCIM endpoint request header parameter


```
{"Authorization":"Bearer gCV2VxetE7vgRxG77pqztGSs-3lWLTJhLG5K3dL7YbtyV6Ys1z0CnTcmv
__NwTuOdIcUq4_bM9q2xRPa8I4ab7JW31T6XVZ70eMLdAnOy3tgZpaz3UWTJwfLKEi8pqN6ZcF57kYmSKW
rBYOabmY9JrvWtqSLsTBaX9ALWgK2JADHMvpXsbqjkI2MV9xh3nIYKyTX0mW8EOZx9JhtqC3XIQ"}
```

5 Supported Protocols

- HTTP
- HTTPS

6 Endpoints

6.1 Applications

Application is the name of the File Access Manager component that represents the monitored system (such as, Microsoft Outlook, Active Directory, and file servers). File Access Manager monitors and analyzes permissions of built-in applications.

The IdentityIQ File Access Manager Server Installation Guide contains a complete list of supported built-in applications.

Endpoint Description: The API provides information about applications that are configured in IdentityIQ File Access Manager. It allows you to retrieve a list of all defined applications (Which are configured in File Access Manager or a specific application.

6.2 BusinessResources

Endpoint Description: The API provides information about business resources of the organization (folders, shares etc.). It enables searching for business resources by folder name (full or partial) across all defined applications (servers) or in a specific application. Note: You can query Business Resource owners using this Endpoint. This endpoint can be used to build a resource tree, using the parentResourceId filter.

6.2.1 Business Resource type mapping

One of the returned business resource parameters is **type (number)**. The table below describes the types according to the returned type ID:

Note: The content of the table may vary according to the application types installed.

Business Resource Type ID	Business Resource Type	Business Resource Type ID	Business Resource Type
0	Folder	1	Active Directory Computer
2	Active Directory Container	3	Active Directory Domain
4	Active Directory Group	5	Active Directory OU
6	Active Directory User	7	SharePoint Document
8	SharePoint List	9	SharePoint List Item
10	SharePoint Site	11	Unknown
12	Folder	13	SharePoint Web
14	Exchange Folder	15	Exchange Mailbox
16	Exchange Public Folder	18	UserSAMAccountName
24	Active Directory GPO	25	Active Directory GPO Container
801	Windows Cluster Server Name	908	Google Folder
909	Google User	910	Dropbox Folder
911	Dropbox User	912	Box Folder

Business Resource Type ID	Business Resource Type	Business Resource Type ID	Business Resource Type
913	Box User	914	Box File
950	SharePoint File	951	SharePoint Hidden List
952	SharePoint Hidden Folder	953	SharePoint Hidden File
1000	Active Directory Builtin Domain	1100	Dfs Namespace
1101	Dfs Link		

6.3 Capabilities

Capabilities are objects defining access rights within the IdentityIQ File Access Manager module.

A Capability includes

- Capability name and description
- Rights that each capability has
- Users and groups associated with each capability

Endpoint Description: The API retrieves a list of capabilities, including the capability description, the rights each capability includes, and associated users and groups. Optional filters include capability, right, and user names.

6.4 DataClassificationCategories

Data Classification categories describe the different types of sensitive data which the File Access Manager can identify, according to the data content and context.

Endpoint Description: The API retrieves a list of all File Access Manager Data Classification categories. An optional filter of category enables calling a single category record.

6.5 DataClassificationResults

The Data Classification mechanism provides the ability to discover and classify resources and files containing sensitive information, according to configurable rules and policies.

Endpoint Description: For each resource requested, this endpoint returns an object including the file name, policy, rule, and categories that triggered the classification for this file, as well as the number of times a category match was found. This endpoint supports DFS addresses, if the DFS applicationId is requested.

6.6 IdentityUsers

Identities are collected from different identity repositories, such as Active Directory, Azure, and NIS. This information is used in Permissions Collection, as well as to analyze users, the relation between users, groups, users' membership in groups, the structure of groups, and other information.

Endpoint Description: The API provides information about the Identity Users collected by IdentityIQ File Access Manager's Identity Collectors. It allows querying them and changing their business resources' ownership.

6.7 KPIs

Endpoint Description: The API returns the count and score of KPIs calculated in IdentityIQ File Access Manager. This is a read only endpoint.

6.8 Permissions

Endpoint Description: The API provides information about a user or group's direct permissions on each business resource.

Unlike other objects, the Permission object does not stand on its own and its id cannot be used as a filter This mean that getting a permission object by id is not supported (/Permissions/[identifier]).

The reason there's no ID for a permission lies in the underlying data model of how permissions are stored. Since most application types support an inheritance model, permissions in File Access Manager are stored only for business resources which are uniquely managed.

Uniquely managed business resources are either business resources which do not inherit their permissions, or business resources which inherit permissions but add more on top of them. A business resource which fully inherits its permissions without adding to them, only holds a reference to the parent business resource it inherits the permissions from.

A single permission is uniquely identified by the following attributes:

- identity id (either user or group)
- identity type - user or group
- business resource id
- permission type id
- inherited - a single user/group can have the same permission on a business resource. Once as an inherited permission and another as a non-inherited explicit permission
- allow/deny - a single user/group can have the same permission on a business resource. Once as an allow permission and another as a deny permission

In some application types, the first 4 attributes will be enough to uniquely identify a permission. Those are application types that do not support an inheritance model and allow/deny permissions, or partially support an inheritance model without allow/deny, such as SharePoint, where a business resource can either inherit its permissions or be uniquely managed, it cannot inherit and add on top of it.

7 Endpoint Details and Usage

7.1 Applications

GET/v2/applications/{id}

Retrieves the Application by ID

Filter
Filter is not supported

Attributes
Returns all attribute values by default

Paging
Paging is not supported. Returns a specific application.

Sample Requests
```
./identityiqfamapi/scim/v2/Applications/2
```

7.2 BusinessResources

GET/v2/businessresources

Retrieves a list of Business Resources according to a given query. The results are sorted by name.

Filter
All attributes to filter by are optional. If no filter is specified, the first 1000 records are returned.

Supported filter attributes:
- **name** – Can be used to filter by the business resource name. If it is called without a parentApplicationId, it will return the first 1000 records.
  Operators supported: contains, starts with and equals
  Constraints: cannot be sent with the fullPath filter attributes.
- **fullPath** – Can be used to filter by the business resource full path. Cannot be sent with the name filter attribute.
  Operators supported: equals
  Constraints:
    - Must be sent with the parentApplicationId attribute filter.
    - Cannot be sent with the name filter attribute.
- **parentApplicationId** - Can be used to filter by the business resource application id. If it is called without other filter attributes, it will return the top-level resources in the hierarchy.
  Operators supported: equals
- **isDfs** – Use this filter attribute to get business resources from DFS applications.
  Operators supported: equal
  Valid values: “false” (default), “true” or "both"
  Constraints: Must be sent with name or fullPath filter attributes.

- **owners** – Use this filter attribute to get business resources that have data owners assigned to them.
  Operators supported: **present** operator (pr) only
- **parentResourceId** - If sent, the response will contain only the direct children of the parent resource. If parentApplicationId is sent without parentResourceId, the result will contain the direct children of the application, meaning the top-level resources in the hierarchy.
  For DFS resources, use the parentResourceId and parentApplicationId.

  Operators supported: equals
  Constraints: Cannot be sent with other filters besides parentApplicationId

Attributes

Returns all attributes values by default except for the owners attribute.
Owners attribute value will be returned if it was specifically requested in the attributes parameter.
Owners attribute can only be used when the owners filter is present in the query.

Paging

- startIndex - The 1-based index of the first result in the current set of list results (starts from 1)
- count - The number of objects returned in a list response per page. Max page size = 200.
- In case no filter was specified, or a filter was sent with the name attribute without the parentApplicationId attribute, the first 1000 records are returned. Paging parameters are irrelevant in these 2 cases.

Sample Requests

```
/identityiqfamapi/scim/v2/BusinessResources?filter=name co "MyFolderName"
/identityiqfamapi/scim/v2/BusinessResources?filter=fullPath
eq "\\server\share\folder1" and parentApplicationId eq
"2"&count=200&startIndex=1
/identityiqfamapi/scim/v2/BusinessResources?filter=owners
pr&attributes=owners
• /identityiqfamapi/scim/v2/BusinessResources?filter=name sw "DFS
folder" and isDfs eq "both"
```

Parameters

Try it out

Name	Description
Filter **string** *(query)*	To filter results, use the following syntax: attributeName operator “value”.
Attributes **string** *(query)*	To retrieve specific attributes values, add the attributeName to the attributes query part.
startIndex **int($int32)** *(query)*	An integer indicating the 1-based index of the first query result.
Count **int($int32)** *(query)*	An integer indicating the desired maximum number of query results per page.

7.3 Capabilities

GET /v2/Capabilities

Retrieves a list of capabilities, the rights for each capability, and associated users and groups, according to the given query. The results are sorted by capability name.

Filter

The attributes to filter by are optional. If no filter is specified, the list will include all the capabilities.

Supported logical operators: None

Supported grouping operators: None

Supported filter attributes:

- capabilityName – Returns the capability selected.
  Operators supported: contains, starts with and equals.
- rightName – returns all capabilities that contain this right.
  Operators supported: contains, starts with and equals.
-
- userUniqueIdentifier – returns capabilities that this user belongs to. either directly, as part of a group, or a nested group, depending on the value of the filter ‘searchNested’ (see below).
  Operators supported: equals
  Format: The filter must be entered in the form 'domain\user'
- searchNested’ – Determines how to search for users within the groups
- Default value: False
  False: Return only capabilities that contain this user as a direct member
  True: Return capabilities that contain this user as a direct member, or a member through nested groups (ex, capability A contains Group B -> Group C -> User D)
  Constraints: Must be sent with the filter 'userUniqueIdentifier'

Attributes

All attributes are of type "always' and must be returned.
All attributes are of type "readOnly".

Paging

Paging is not supported.

7.4 DataClassificationCategories

GET /v2/DataClassificationCategories

Returns a list of categories containing the categories in the File Access Manager database, according to the requesting filter. For each category it returns the id, name and description.

Filter

The attributes to filter by are optional. If no filter is specified, all the data classifications are returned.
Supported logical operators: None
Supported grouping operators: None

Supported filter attributes:

categoryName – Return the data classification category requested
Operators supported: contains, starts with and equal

Attributes
All attributes are of type "always' and must be returned
All attributes are of type "readOnly"

Paging
Paging is not supported

7.5 DataClassificationResults

GET /v2/DataClassificationResults

Returns the data classification results for the requested application and path. For each file analyzed, it lists the policy, rule and categories that triggered the classification.

Filter
The attributes to filter by are optional. If no filter is specified, all the data classification results are returned.

If no filter is applied, only the physical resources will be returned. For the DFS resources, use the DFS applicationId and logical resource full path in the filter.
Supported logical operators: and
Supported grouping operators: None

Supported filter attributes:
- applicationId – Return business resources from this application
  Operators supported: equals
  Format: Integers
  Constraints: Must be sent with the filter 'fullPath'
- fullPath – Can be used to filter by the business resource full path. Supports the equals operator only. Must be sent with the ApplicationId attribute filter. Cannot be sent with the name filter attribute
  Operators supported: equals
  Constraints: Must be sent with the filter 'applicationId'

Attributes
All attributes are of type "always' and must be returned.
All attributes are of type "readOnly".

Paging
Paging is not supported.

7.6 Groups

GET/v2/groups

Parameters
Try it out

Name	Description
queryOptions.filter **string** *(query)*	To filter results, use the following syntax: attributeName operator "value".

Name	Description
queryOptions.attributes **string** *(query)*	To retrieve specific attributes values, add the attributeName to the attributes query part.
queryOptions.startIndex **int($int32)** *(query)*	An integer indicating the 1-based index of the first query result.
queryOptions.count **int($int32)** *(query)*	An integer indicating the desired maximum number of query results per page.

7.7 IdentityUsers

GET/v2/identityusers/{id}

Retrieves a specific IdentityUser, where ID in the request is the ID of the identity

Filter
Filter is not supported

Attributes
Returns all attribute values by default

Paging
Paging is not supported. Returns a specific IdentityUser.

Sample Requests
- `/identityiqfamapi/scim/v2/IdentityUsers/135`

GET/v2/identityusers

Retrieves a list of IdentityUsers according to a given query

Filter
Supported filter attributes:
- uniqueIdentifier – the domain\username representation of the IdentityUser. Supports only the equals operator.
- ownedResources– returns only users that are owners of business resources and supports only present operator. It cannot be used with the attribute uniqueIdentifier.

Attributes
Returns all attribute values by default

Paging
- startIndex - The 1-based index of the first result in the current set of list results (starts from 1)
- count - The number of objects returned in a list response per page. Max page size = 200.

Sample Requests

• /identityiqfamapi/scim/v2/IdentityUsers?filter=uniqueIdentifier eq "domain\username"&count=200&startIndex=1
• /identityiqfamapi/scim/v2/IdentityUsers?filter=ownedResources pr&count=50&startIndex=2

Parameters

Try it out

Name	Description
filter **string** *(query)*	To filter results, use the following syntax: attributeName operator “value”.
attributes **string** *(query)*	To retrieve specific attributes values, add the attributeName to the attributes query part.
startIndex **int($int32)** *(query)*	An integer indicating the 1-based index of the first query result.
count **int($int32)** *(query)*	An integer indicating the desired maximum number of query results per page.

PATCH /v2/identityusers/{id}

Update specific IdentityUser's owned resources. Should pass the IdentityUser Id in the URL. Returns the updated IdentityUser object.

Request

This is a SCIM Patch request that is based on JSON Patch.
The body of each request MUST contain the “schemas” attribute with the URI value of “urn:ietf:params:scim:api:messages:2.0:PatchOp” and the Operations object.
The Operations object has 3 parts: “op” for operation, “path” for the attribute and “value” for the new resources.

Operation - “op”

• Add - adds the new resource to the owned resources list. If the resource already exists, it does not add the resource, but the action is successful.
• Remove - removes all resources from the owned resources list. Does not currently support removing specific resources, any value is ignored.
• Replace - replacing all owned resources\specific resource, with given resources as value. The specific resource to be removed can be passed in the filter under "path". If the value is empty, it will remove the specific resource, if given. If not, it removes all resources.

Path - “path”

Supports “OwnedResources” attribute only, the only writable attribute of the User object. Any other attribute will return an error of unsupported.

Value - “value”

Must contain the FullPath and ParentApplicationID of the BusinessResource, see example below.

Sample Requests

• URL - /identityiqfamapi/scim/v2/IdentityUsers/135

• **Add body:**


```
{
"schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
"Operations": [{
"op": "add",
"path": "ownedResources",
"value": [{ "fullPath": "\\server\share\folder1", "parentApplicationId": "1" },
{ "fullPath": "\\server\share\folder2", "parentApplicationId": "1" }]
}]}
```


• **Remove body**:


```
{
"schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
"Operations": [{
"op": "remove",
"path": "ownedResources"
}]}
```


• **Replace body**:


```
{
"schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
"Operations": [{
"op": "replace",
"path": "ownedResources",
"value": [{ "fullPath": "\\server\share\folder2", "parentApplicationId": "1" },
{ "fullPath": "\\server\share\folder3", "parentApplicationId": "1" }]
}]}
```


• **Replace body (With filter)**:


```
{
"schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
"Operations": [{
"op": "replace",
"path": "ownedResources[fullPath eq "\\server\share\folder1" and
parentApplicationId eq "1"]",
"value": [{ "fullPath": "\\server\share\folder2", "parentApplicationId": "1" },
{ "fullPath": "\\server\share\folder3", "parentApplicationId": "1" }]
}]}
```


Parameters

Try it out

Name

Description

id *

string

(path)

patchRequest *

(body)

7.8 KPIs

GET/v2/KPIs/

Returns the values of the KPI requested. KPI name must be from the valid list below

Filter
The name filter is required. If no filter is specified, or if the name is not in the list of valid KPIs, the API will not return results.

Supported logical operators: None
Supported grouping operators: None

Supported filter attributes:
- name – The name of the KPI to return
  Operators supported: equals
  Format: String
  Valid values:
    - 'Sensitive Resources Missing Owners'
    - 'Overexposed Sensitive Resources'

Attributes
- Name – name of the KPI
- Count – the KPI value (for example: The number of sensitive resources without data owners)
- Score

All attributes are of type "always' and must be returned.
All attributes are of type "readOnly".

Paging
Paging is not supported.

Sample Requests
```
• /identityiqfamapi/scim/v2/kpis?filter=name eq `"Overexposed Sensitive Resources"`
```

7.9 Permissions

GET/v2/permissions

Retrieves a list of Permissions according to a given query.

Filter
All attributes to filter by are optional, but at least one should be selected.
Supported filter attributes:

- userUniqueIdentifier - supports the equal operator only. Must be in the form of 'domain\user'. If the domain is empty must be in the form of 'user' only. Description: the parameter can be used to specify the user. This is the domain\user representation in each Identity Collector type:

- Active Directory - domain is the Netbios name of the domain, user is the samAccountName
- Azure Active Directory - domain is the fqdn of the Azure AD domain, user is the user upn
- NIS - domain is empty, user is the user name in the NIS server
- Google Drive - domain is empty, user is the user email
- Box - domain is the Box domain, user is the user email
- Dropbox - domain is the Dropbox Team name, user is the user email

- groupUniqueIdentifier – the domain\groupname representation of the identity group. Supports the equal operator only.

  Constraint: The filter cannot contain both the filters userUniqueIdentifier and groupUniqueIdentifier.

- classificationCategory - Use this filter attribute to get permissions that have classification categories assigned to their business resource. Supports the present operator only.
- fullPath – Can be used to filter by the permission's business resource full path. Supports the equal operator only. Must be sent with the applicationId attribute filter.
- applicationId - Can be used to filter by the permission's business resource application id. Supports the equal operator only. To query permissions in DFS applications, you must use this attribute with the DFS application id.
- permissionTypeName – Use this filter attribute to get permissions with a specific permission type (Read, Write etc.). Supports the equals operator only.
- inherited - Use this filter attribute to get permissions by their inheritance value. Supports the equals operator only and the values "false" (default), "true" or "both".

Attributes

Returns all attribute values by default except for the classificationCategories attribute of business resource.
classificationCategories attribute value is returned if it was specifically requested in the attributes parameter.

Paging

• startIndex - The 1-based index of the first result in the current set of list results (starts from 1)
• count - The number of objects returned in a list response per page. Max page size = 200.
• Only the first 100,000 results are returned in pages. If the requested page exceed 100,000 results, an error of tooMany will be returned.
• Results are ordered by the Id of Groups' Permissions and then the by the Id of Users' Permissions.

Sample Requests

```
• /identityiqfamapi/scim/v2/Permissions?filter=applicationId eq "1"
• /identityiqfamapi/scim/v2/Permissions?filter=classificationCategory pr
• /identityiqfamapi/scim/v2/Permissions?filter=fullPath
eq "\\server\share\folder1" and applicationId eq "2"&count=200&startIndex=1
• /identityiqfamapi/scim/v2/Permissions?filter=permissionTypeName eq "Full
Control"&attributes=classificationCategories
• /identityiqfamapi/scim/v2/Permissions?filter=inherited eq "both"
```

Parameters

Name	Description
filter **string**	

(query)

To filter results, use the following syntax: attributeName operator “value”.

attributes
string

(query)

To retrieve specific attributes values, add the attributeName to the attributes query part.

startIndex
int($int32)

(query)

An integer indicating the 1-based index of the first query result.

count
int($int32)

(query)

An integer indicating the desired maximum number of query results per page.

7.10 Users

DELETE /v2/users/{userId}

GET /v2/users

Parameters

Try it out

Name	Description
queryOptions.filter **string** *(query)*	To filter results, use the following syntax: attributeName operator “value”.
queryOptions.attributes **string** *(query)*	To retrieve specific attributes values, add the attributeName to the attributes query part.
queryOptions.startIndex **int($int32)** *(query)*	An integer indicating the 1-based index of the first query result.
queryOptions.count **int($int32)** *(query)*	An integer indicating the desired maximum number of query results per page.