

Getting started with the Data Discovery Gateway API

Data Discovery

16th April 2020

Contents

Contents	2
Before you get started	3
Introduction	3
Swagger Documentation on the API	4
Testing calls on your tenant	4
Authentication	6
Example – Listing Workspaces	7
Example – Getting a single user	7
Example – Listing Repositories.....	8
Example – Creating a workspace	8
Example – Creating a data subject for a workspace	9
Example – Creating a data source for a workspace	10
Micro Focus Trademark Information	12
Company Details	12

Before you get started

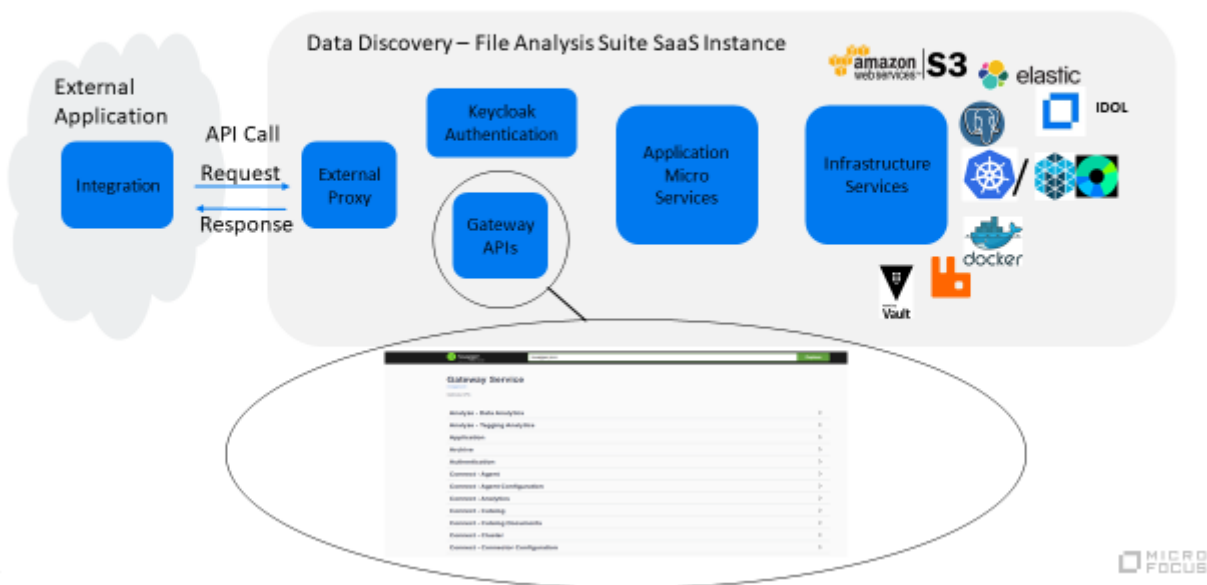
As with all product APIs it's important to state there is always a risk associated with the use of API and it's the responsibility of those writing the integrations to ensure that there is no adverse impact on the production tenant. Therefore it is suggested only use API with the help from a Data Discovery Subject Matter Expert (SME) e.g. Micro Focus or enabled partner. If incorrect method calls are made it is possible to cause problems on your tenant. Suggest all integrations are reviewed by the SME. You could use Micro Focus professional services or your service provider partner to test out the integration on a demo instance before applying in production. Micro Focus support will be able to provide general advice on the rest based APIs but will not be able to help on specific custom code.

Introduction

Micro Focus® File Analysis Suite—Data Discovery SaaS platform provides file analysis, tagging, analytics and personal data discovery across mission critical unstructured repositories including NT file shares, Exchange, Office365, SharePoint/SharePoint Online and Micro Focus Content Manager. The integration with Content Manager provides data management actions for disposition, records management and long-term preservation capabilities. Data Discovery addresses key challenges around data privacy and associated risk assessment use cases. It represents a key solution for mitigating risk associated with data privacy challenges, including GDPR (EU—General Data Protection Regulations), KVKK—Protection of Personal Data Law (Turkey), POPI— Protection of Personal Information Act (South Africa) and the CCPA (California Consumer Privacy Act), as customers look for technology to assist in identifying, managing and securing personal identifiable information and other sensitive data.

Data Discovery has a Rest based APIs (Gateway Service) which represents many of the features that are shown in the standard web user interface. Each call needs to first gain an authentication token from Keycloak which can then be added as a parameter to the feature calls which in the main are named similar to the labelled features in the user interface. The tokens provide secure communications over the API but do expire after a period of time, similar to the login timeout on the web interface. New tokens can be retrieved or existing tokens refreshed for continuous operation.

Data Discovery APIs



Details of the methods are self-documented using swagger but this document aims to provide an introduction to explain how the calls are made and some examples of how to use it.

Swagger Documentation on the API

All Gateway APIs are documented through the following URL. You can use this resource to research possible calls. Most of the methods are self-explanatory and relate to the UI interactions. However some of the names relate to the original functions e.g. for repositories use the “Job” method.

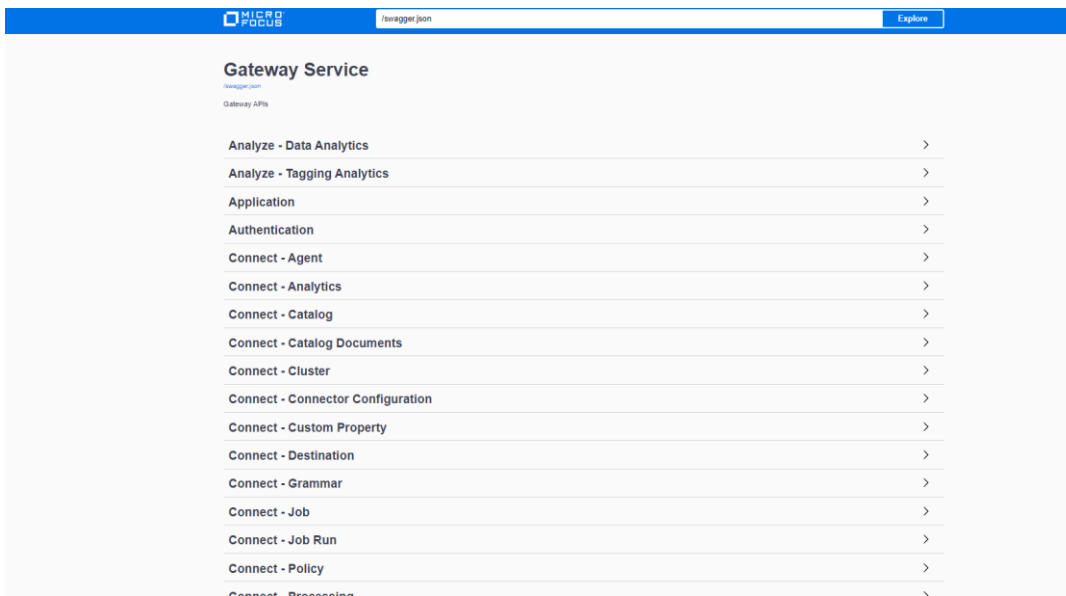
<https://www.demo.microfocusfileanalysis.com:9310/swagger/?TENANT=boswell>

Testing calls on your tenant

Use tenants swagger page

Each tenant has its own Gateway Service interface which allows for testing calls to the tenant. Please note there is no test system or rollback. Note that all most calls require a security token. This token is generated by the login page in the UI but can be called from using the Authentication Object

<https://www.demo.microfocusfileanalysis.com:9310/swagger/?TENANT=<tenantID>>



Example – Listing Workspaces

This examples displays the details of a specific case

```
$params = @{
    Uri          = 'https://www.demomicrofocusfileanalysis.com:9310/ca/v1/case/95'
    Headers      = @{ 'Authorization' = "Bearer $token" }
    Method       = 'GET'
    Body         = $jsonSample
    ContentType  = 'application/json'
}
```

`Invoke-RestMethod @params`

```
PS C:\Users\boswellj.CORPDOM> C:\Users\boswellj.CORPDOM\OneDrive - Micro
Focus\Software\Data Discovery - SaaS\Rest Calls\3.2 RestAPI to Data Discovery - list
workspaces.ps1

id                : 95
name              : DSAR for Jean Guiral
createdBy         : 479516b0-02f7-4fa0-8e90-6d3d8df2b094
updatedBy        :
createDate       : 1575902132071
updateDate       : 1587029533051
description       :
caseId           :
status           : Open
templateType     : Data Subject Access Request
allowApprovalWithoutReview : False
items            : 25
diskSpace        : 1531278.0
caseOwner        : 479516b0-02f7-4fa0-8e90-6d3d8df2b094
startDate        : 1586905200000
endDate          : 1587041186930
dataSubjects     : {}
workbooks        : {}
datasources      : {}
categories       : {}
security         : @{{51b35a41-290f-4b1e-b1cc-49f1b58dab75=System.Object[]}}
validPermissions : {EcaCaseEditSecurity, EcaCaseContentSearch, EcaCaseDownload,
EcaCaseDeactivateDataSource...}

PS C:\Users\boswellj.CORPDOM>
```

Example – Getting a single user

This call gets a user based on a query result. It's a useful call to get a user's ID as this is required when creating workspaces

```
#Using query parameter to search for user
$params = @{
    Uri          = 'https://victor-ext01.swinfra.net:9310/v1/users?query=boswell'
    Headers      = @{ 'Authorization' = "Bearer $token" }
    Method       = 'GET'
    Body         = $jsonSample
    ContentType  = 'application/json'
}

#Invoke-RestMethod @params
$response = Invoke-RestMethod @params

write-host "Found Total Users = " $response.items.Count

$i=0
Do{
```

```

write-host $response.items[$i].username $response.items[$i].email
$response.items[$i].id
}while (++$i -lt $response.items.Count)

```

```

PS C:\Users\boswellj.CORPDOM> C:\Users\boswellj.CORPDOM\OneDrive - Micro
Focus\Software\Data Discovery - SaaS\Rest Calls\3.2 RestAPI to Data Discovery - get
single user.ps1
Found Total Users = 1
boswell@microfocus.com boswell@microfocus.com 8f66c7ea-e703-4ccf-b78a-f183678c6d51
PS C:\Users\boswellj.CORPDOM>

```

Example – Listing Repositories

The call retrieves all the details for the repositories. The ids for the repository will be required when creating the data source.

```

#Generate params for get requests for all users
$params = @{
    Uri          = 'https://victor-ext01.swinfra.net:9310/ajp/v1/job'
    Headers      = @{ 'Authorization' = "Bearer $token" }
    Method       = 'GET'
    Body         = $jsonSample
    ContentType  = 'application/json'
}

```

```

#Execute Rest call
Invoke-RestMethod @params

```

```

PS C:\Users\boswellj.CORPDOM> C:\Users\boswellj.CORPDOM\OneDrive - Micro
Focus\Software\Data Discovery - SaaS\Rest Calls\3.2 RestAPI to Data Discovery - list
repositories.ps1

```

```

id                : 24149
name              : Vika Repo1 with msg files
description       :
createDate        : 1583870345808
createdBy         : 1acd091c-2d17-4c6e-98bb-85d0dd7535b8
type              : FILESYSTEM
repositoryId      : 24149
clusterId         : 24142
clusterName       : VikaAgent3
policy            : @{id=-2; name=Index metadata and content; archive=False;
processContent=True; processMetadata=True; processContentAnalysis=False}
captureRule       : @{paths=System.Object[]; filter=.*; ruleId=}
properties        : @{dateRange=; maxFileSize=0; includeExtensions=True}
totalItems        : 87
totalItemsSize    : 0
active            : True
scheduleType      : MANUAL
useCustomGrammars : True
updateDate        : 1585679808680
tenantId          : vika26
repositoryType    : FILESYSTEM
credentialId      : 24150

```

Example – Creating a workspace

This call creates a workspace which is useful for remotely instantiating a process e.g. Data Subject Access Request. This call is required before the data subject and data source are added.

```

#Body for workspace creation post
$json = @'
{
    "allowApprovalWithoutReview":true,
    "caseOwner":"1acd091c-2d17-4c6e-98bb-85d0dd7535b8",

```



```

    "createdBy": "1acd091c-2d17-4c6e-98bb-85d0dd7535b8",
    "description": "",
    "name": "JB test 3",
    "security": {
      "ebb34b2d-9baa-4a52-8bdd-d539389c408c": [
        {
          "id": "1acd091c-2d17-4c6e-98bb-85d0dd7535b8",
          "userType": "User"
        }
      ]
    },
    "startDate": 1585281600000,
    "status": "Open",
    "templateType": "Data Subject Access Request"
  }
}
"@

```

#Header for workspace creation post

```

$header = @{
  "Accept"="application/json"
  "Authorization"="Bearer $token"
  "Content-Type"="application/json"
}

```

#Call to create workspace

```

Invoke-RestMethod -Uri "https://victor-ext01.swinfra.net:9310/ca/v1/case" -Method 'Post'
-Body $JSON -Headers $header

```

```

PS C:\Users\boswellj.CORPDOM> C:\Users\boswellj.CORPDOM\OneDrive - Micro
Focus\Software\Data Discovery - SaaS\Rest Calls\3.2 RestAPI to Data Discovery - create
workspace.ps1

```

```

id                : 26300
name              : JB test 3
createdBy        : 1acd091c-2d17-4c6e-98bb-85d0dd7535b8
description      :
status           : Open
templateType     : Data Subject Access Request
allowApprovalwithoutReview : True
items            : 0
diskSpace        : 0.0
caseOwner        : 1acd091c-2d17-4c6e-98bb-85d0dd7535b8
startDate        : 1585281600000
dataSubjects     : {}
workbooks        : {}
datasources      : {}
categories       : {}
security         : @{"ebb34b2d-9baa-4a52-8bdd-d539389c408c=System.Object[]"}

```

Example – Creating a data subject for a workspace

This call adds a data subject to the previously created workspace.

#Body for workspace data subject post

```

$JSON = @'
{
  "name": "Boswell Declan declan.boswell@microsoft.com",
  "createdBy": "1acd091c-2d17-4c6e-98bb-85d0dd7535b8",
  "updatedBy": "",
  "criteria": "(case==25788) AND ((Boswell NEAR5 Declan) OR
(accounts.identifier_aggregate == \"declan.boswell@microsoft.com\")) ",
  "queryStructure": "{ \"anyFields\": { \"required_keyword-
0\": { \"component\": \"required_keyword\", \"value\": \"Boswell NEAR5
Declan\" }, \"accounts.identifier-
0\": { \"component\": \"accounts.identifier_aggregate\", \"value\": [ { \"name\": \"declan.boswel
l@microsoft.com\", \"account\": \"declan.boswell@microsoft.com\" } ] } } }",
  "primaryEmail": "declan.boswell@microsoft.com",
  "lastName": "Boswell",
  "firstName": "Declan",

```

```

    "secondaryEmails": [],
    "active": true,
    "type": "External"
  }
  '@

```

```

#Header for workspace creation post
$header = @{
  "Accept"="application/json"
  "Authorization"="Bearer $token"
  "Content-Type"="application/json"
}

```

```

#Call to create workspace

```

```

Invoke-RestMethod -Uri "https://victor-ext01.swinfra.net:9310/ca/v1/case/26300/data-
subject" -Method 'Post' -Body $JSON -Headers $header

```

```

PS C:\Users\boswellj.CORPDOM> C:\Users\boswellj.CORPDOM\OneDrive - Micro
Focus\Software\Data Discovery - SaaS\Rest Calls\3.2 RestAPI to Data Discovery - create
data subject.ps1

```

```

id                : 26306
name              : Boswell Declan declan.boswell@microfocus.com
createdBy         : 1acd091c-2d17-4c6e-98bb-85d0dd7535b8
updatedBy        :
createDate       : 1585688430491
updateDate       : 1585688432181
queryStructure   : {"anyFields":{"required_keyword-
0":{"component":"required_keyword","value":"Boswell NEAR5
Declan"},"accounts.identifier-
0":{"component":"accounts.identifier_aggregate","value":[{"name":"declan.boswell@microfoc
us.com","account":"declan.boswell@microfocus.com"}]}}}
primaryEmail     : declan.boswell@microfocus.com
lastName         : Boswell
firstName        : Declan
secondaryEmails  : {}
active           : True
type             : External

```

Example – Creating a data source for a workspace

This call adds a data source to the previously created workspace

```

#Body for workspace datasource creation post

```

```

$JSON = @'
{
  "name" : "test2",
  "createdBy" : "1acd091c-2d17-4c6e-98bb-85d0dd7535b8",
  "description": "created by rest api",
  "dataSubjectDataOnly": true,
  "exchangeDetails": {"repositoryIds": [24235]},
  "fileSystemDetails": {"repositoryIds": [24149, 24245, 24179, 24185, 24529, 24766,
24775, 24395, 25011], "pathNames": []},
  "sharePointDetails": {},
  "contentManagerDetails": {"repositoryIds": [24240]},
  "customRepoDetailsList": [],
  "active" : true
}
'@

```

```

#Header for workspace creation post
$header = @{
  "Accept"="application/json"
  "Authorization"="Bearer $token"
  "Content-Type"="application/json"
}

```

```

#Call to create workspace

```

```

Invoke-RestMethod -Uri "https://victor-
ext01.swinfra.net:9310/ca/v1/case/26300/datasource" -Method 'Post' -Body $JSON -Headers
$header

```

```
PS C:\Users\boswellj.CORPDOM> C:\Users\boswellj.CORPDOM\OneDrive - Micro Focus\Software\Data Discovery - SaaS\Rest Calls\3.2 RestAPI to Data Discovery - create datasource.ps1
```

```
id : 26309
name : test2
createdBy : 1acd091c-2d17-4c6e-98bb-85d0dd7535b8
updatedBy :
createDate : 1585688572839
updateDate : 1585688572839
description : created by rest api
dataSubjectDataOnly : True
exchangeDetails : @{repositoryIds=System.Object[]}
fileSystemDetails : @{repositoryIds=System.Object[]; pathNames=System.Object[]}
sharePointDetails :
contentManagerDetails : @{repositoryIds=System.Object[]}
customRepoDetailsList : {}
lastTagged : 1585688572765
active : True
```

Micro Focus Trademark Information

MICRO FOCUS and the Micro Focus logo, among others, are trademarks or registered trademarks of Micro Focus (IP) Limited or its subsidiaries in the United Kingdom, United States and other countries. All other marks are the property of their respective owners.

Company Details

Company name: Micro Focus International plc

Place of registration: England and Wales

Registered number: 5134647

Registered address: The Lawn, 22-30 Old Bath Road, Berkshire, RG14 1Q