



Contents

- 1. Version History..... 3**
- 2. Product Description..... 4**
- 3. Webservice Introduction..... 6**
 - 3.1. REST End Points..... 6
 - 3.2. SOAP End Points..... 6
- 4. Authorization..... 7**
 - 4.1. REST Authorization..... 7
 - 4.2. SOAP Authorization..... 7
- 5. Selecting a repository..... 9**
- 6. Selecting the output format..... 10**
- 7. myInsight Objects..... 11**
- 8. References..... 13**
 - 8.1. Info Service References..... 13
 - 8.2. Web Service References..... 13
- 9. Errors and Exceptions..... 18**

1. Version History

Date	Changes	Version number
2-June-2016	Release of myInsight 5.1, which includes the new webservises.	1.0
18-June-2016	Small corrections	1.1
19-October-2018	Update myInsight 7.0	1.2
26-March-2021	Update myInsight 7.4	1.3
7-April-2023	Update for myInsight 8.0.	1.4
22-September-2023	Update for myInsight 8.1.	1.5

2. Product Description

With myInsight for Documentum, end-users can request reports from predefined report definitions. They can see the reports displayed on their computer screen or receive reports automatically in their e-mail or at a specified location inside or outside the Documentum repository. The preferred format can be chosen by the end-user, without the end-user needing any knowledge about DQL, HTML or XSL.

myInsight for Documentum categorizes functionalities according to the user's role in the report generator. There are 3 predefined roles:

- Users in the *Report User* role, which supplies predefined reports from the system, require no knowledge of either DQL or style sheets.
- Users in the *Report Administrator* role can schedule reports so that they are generated automatically at a predefined time and location.
- Users in the *Report Builder* role can define new report specifications by configuring the DQL statements, and they can identify and compose the desired style sheets.

myInsight for Documentum can be accessed by anyone who has been given one of these default roles.

myInsight for Documentum can produce output in any format that can be generated using XSL style sheets. For example: reports can be presented in PDF, HTML, text file, Microsoft Excel spread sheet or Microsoft Word format. myInsight for Documentum can also e-mail the output file automatically. In this case the recipient does not need to be a Documentum user and can even be someone from outside the organisation.

myInsight for Documentum is integrated into Documentum Webtop, Documentum Administrator, Documentum D2 and Documentum xCP, in line with the corporate philosophy of Documentum. This enables end users to work in an environment that they are already familiar with. Due to its full integration within the Documentum environment, no additional components need to be installed on the end user's local machine.

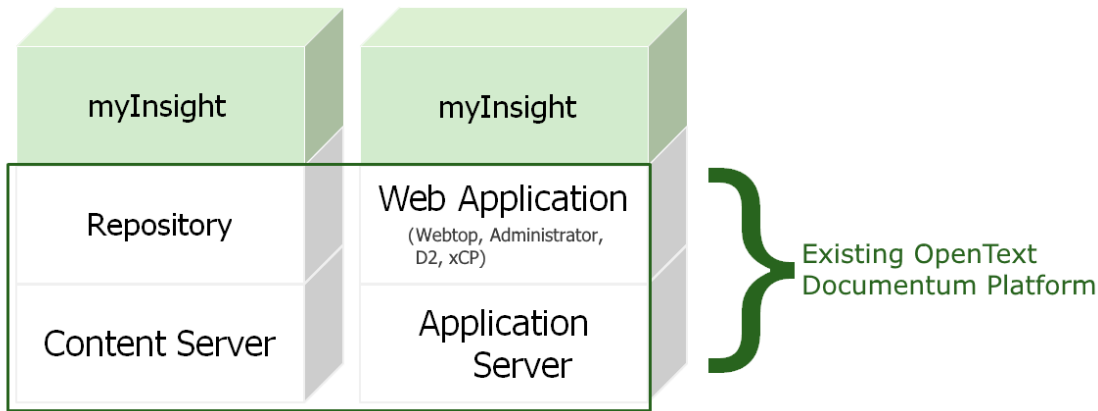


Figure 1: myInsight integration into the OpenText Documentum platform

As shown in the previous figure, myInsight for Documentum components are located on both Repository level as Web Application Level. Both use standard OpenText | Documentum subcomponents for its functionality.

As of version 7.0, myInsight for Documentum can be installed separate from the Content Server.

In December 2015 eurosript has been renamed to Amplexor. The product eurosript Documentum Report Generator has been renamed to myInsight for Documentum. In 2020, Amplexor joined the Acolad Group under the name Acolad Digital. In October 2023, Acolad Digital became AmeXio.

For the reader's convenience, the abbreviation 'myInsight' will be used in this document, instead of the full product title 'myInsight for Documentum'.

3. Webservice Introduction

With myInsight 5.1, two web services have been introduced for myInsight:

- **MyInsightInfoService:** This service is used to check for available webservices and available repositories. To view the methods for this service view [Info Service References](#) on page 13.
- **MyInsightWebService:** This service is used to get myInsight objects and to run reports. To view the methods for this service view [Web Service References](#) on page 13.

These two services provide you and your organization the possibility to integrate myInsight functionality and data in your own services.



Important

To use myInsight web services, a valid license must be obtained. Please contact myInsightSupport@group.amexio.com for more information.

The myInsight services can be reached by both REST and SOAP request formats.

3.1. REST End Points

These are the end points for REST requests:

- **MyInsightInfoService:** `http://[host]:[port]/myInsightService/services/rest/info/{method}`
- **MyInsightWebService:** `http://[host]:[port]/myInsightService/services/7.0/{repository}/objects/{method}`

3.2. SOAP End Points

These are the end points for SOAP requests:

- **MyInsightInfoService:** `http://[host]:[port]/myInsightService/services/soap/myInsightInfoService`
- **MyInsightWebService:** `http://[host]:[port]/myInsightService/services/soap/7.0/myInsightWebService`

4. Authorization

To make use of the myInsight web services, authorization is required. Each request needs to be authorized with a valid username and password.

4.1. REST Authorization

Each REST request should be authorized with a username and password. The web service accepts HTTP basic authorization, with base64 encoding. An example of a basic authorization parameter would look like the following:

```
http://[host]:[port]/myInsightService/services/rest/info/
repositories
Accept: application/json
Authorization: Basic VXNlcm5hbWU6UGFzc3dvcmQgdXNlcjE6bXlTZWNyZXQ=
```

Decoded the authorization parameter would look like the following, where 'user1' should be your username and 'mySecret' your password:

```
Authorization: Basic Username:Password user1:mySecret
```

4.2. SOAP Authorization

Each SOAP request needs to include a username and password. Authorization is passed using the WS-Security (Web Service Security) SOAP extension. The following is an example of passing credentials in a SOAP header:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:ser="http://schemas.myinsight.amplexor.com/7.0/
services" xmlns:com="http://schemas.myInsight.amplexor.com/5.1/
model/common">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken
        wsu:Id="UsernameToken-7B5E9DFBE9E9C48DAB14525996581921">
        <wsse:Username>user1</wsse:Username>
        <wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">mySecret</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
```



```
<ser:Repository>myRepository</ser:Repository>
<i18n:international xmlns:i18n="http://www.w3.org/2005/09/
ws-i18n">
  <i18n:locale>en_US</i18n:locale>
</i18n:international>
</soapenv:Header>
<soapenv:Body>
</soapenv:Body>
</soapenv:Envelope>
```


5. Selecting a repository

When accessing the myInsight web services a repository is accessed. The way the repository selection is made differs between RESTful and SOAP requests.

To access a repository using REST.

Include your repository name in the request in the following way:

```
http://[hostname]:[port]/myInsightService/services/rest/7.0/
{repository}/objects/{method}
Accept: application/json
Authorization: Basic VXNlcm5hbWU6UGFzc3dvcmQgdXNlcjE6bXlTZWNyZXQ=
```

To access a repository using SOAP.

To select a repository with a SOAP request, include the repository name in the SOAP header as demonstrated below:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:ser="http://schemas.myinsight.amplexor.com/7.0/
services" xmlns:com="http://schemas.myInsight.amplexor.com/5.1/
model/common">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken
        wsu:Id="UsernameToken-7B5E9DFBE9E9C48DAB14525996581921">
        <wsse:Username>user1</wsse:Username>
        <wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">mySecret</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
    <ser:Repository>myRepository</ser:Repository>
    <i18n:international xmlns:i18n="http://www.w3.org/2005/09/
ws-i18n">
      <i18n:locale>en_US</i18n:locale>
    </i18n:international>
  </soapenv:Header>
  <soapenv:Body>
  </soapenv:Body>
</soapenv:Envelope>
```

6. Selecting the output format

Your requests can specify which format response data should be returned: XML or JSON.



Note

If no output format is specified the default format JSON will be used.

To specify which output to receive, add the 'accept' parameter to the request header and specify 'application/json' or 'application/xml' as the value.

7. myInsight Objects

The following myInsight Objects can be returned via the API.

Object Name	Description	Related Methods	
Repository	Place where the objects are stored.	getRepositories	
ReportCategory	Parent of report definitions or other nested ReportCategories	getRootReportCategories getReportCategory getReportCategories	
ReportDefinition	XML properties used to generate a report.	getReportDefinitions getReportDefinition	
Dialog	Configuration for running the reportdefinition.	getRunDialog	
AllowedValues	Contains possible values of dropdown lists in a Dialog.	getDynamicAllowedValues	
Document	Documentum Document in the repository.	getRepositoryDocuments	
Folder	Documentum Folder in the repository.	getRepositoryFolders	
ReportSchedule	Schedule to generate a report at one or more points in time.	getReportSchedules getReportSchedule	

Object Name	Description	Related Methods	
Presentation	Definition of the transformation of data, this object is returned in the reportDefinition object.	-	
PresentationImage	Image data and type.	getPresentationImage	
Report	Result of generateReport. Possible returned values are defined in ReportDefinition.	generateReport	
ReportTile	Pre-defined reports	getDefaultTiles	

8. References

8.1. Info Service References

Request type	GET
REST path	"/versions"
SOAP method	getWebServiceVersions()
Description	Returns the list of available versions of the webservises
Parameters	-
Returns	Returns a list of strings with available versions.

Request type	GET
REST path	"/repositories"
SOAP method	getRepositories()
Description	Returns the list of available repositories.
Parameters	-
Returns	List of Repository

8.2. Web Service References

Request Type	GET
REST path	"/reportcategories"
SOAP method	getRootReportCategories
Description	Return all the root report categories.
Parameters	-
Returns	List of ReportCategory

Request Type	GET
REST path	"/reportcategory/{id}"
SOAP method	getReportCategory

Description	Returns the specified report category.
Parameters	id (required)
Returns	ReportCategory

Request Type	GET
REST path	"/reportcategory/{id}/reportcategories"
SOAP method	getReportCategories
Description	Returns the specified report categories under the given category.
Parameters	id (required)
Returns	List of ReportCategory

Request Type	GET
REST path	"/reportcategory/{id}/reportdefinitions"
SOAP method	getReportDefinitions
Description	Returns the report definitions under the specified report category
Parameters	id (required)
Returns	List of ReportDefinition

Request Type	GET
REST path	"/reportdefinition/{id}"
SOAP method	getReportDefinition
Description	Returns the report definition with the specified id
Parameters	id (required)
Returns	ReportDefinition

Request Type	GET
REST path	"/reportdefinition/{id}/{index}/rundialog"
SOAP method	getRunDialog
Description	Returns the run dialog configuration for the specified report definition and presentation.

Parameters	id (required, id of the reportDefinition.) index (required, index of the list of presentations linked to the reportDefinition.)
Returns	Dialog

Request Type	POST
REST path	"/reportdefinition/{id}/dynamicallowedvalues/{variable}"
SOAP method	getDynamicAllowedValues
Description	Returns dynamic allowed values for the specified report definition and variable
Parameters	id (required, id of the reportDefinition) variable (required, string which represents the variable) submittedValues (required, needs to be included in the body of the POST request)
Returns	AllowedValues

Request Type	POST
REST path	"/reportdefinition/{id}/generatereport"
SOAP method	generateReport
Description	Generate the report
Parameters	id (required) submittedValues (required, needs to be included in the body of the POST request)
Returns	Report

Request Type	GET
REST path	"/repository/folder{id}/folders"
SOAP method	getRepositoryFolders
Description	Returns the documents from the repository
Parameters	id
Returns	List of Folder

Request Type	GET
--------------	-----

REST path	"/repository/folder/{id}/documents"
SOAP method	getRepositoryDocuments
Description	Returns the documents from the repository
Parameters	id (required)
Returns	List of Document

Request Type	GET
REST path	"/reportschedules"
SOAP method	getReportSchedules
Description	Return all the report schedules
Parameters	-
Returns	List of ReportSchedule

Request Type	GET
REST path	"/reportschedules/{id}"
SOAP method	getReportSchedule
Description	Return the selected report schedule
Parameters	-
Returns	ReportSchedule

Request Type	GET
REST path	"/presentationimage/{id}/{index}"
SOAP method	getPresentationImage
Description	Returns the presentation image data and type
Parameters	id (required, id of the reportDefinition) index (required, index of the presentation in the reportDefinition)
Returns	PresentationImage

Request Type	GET
REST path	"/defaulttiles"
SOAP method	getDefaultTiles
Description	Returns the list of pre-defined reports

Parameters	-
Returns	ReportTile

9. Errors and Exceptions

Errors and exceptions thrown from REST requests will return three elements:

- Status Code
- errorMessage
- detailMessage

Possible Status Codes	Description
400	Bad Request
401	Unauthorized
500	Internal Server Error
503	Service Unavailable

Code	Exception	errorMessage
400	NotSupportedException	Request is not supported
400	ErrorMyInsightException	MyInsight returned an error
401	ErrorAuthenticationFailed	User authentication failed
500	RunReportException	There was an error running the report
500	ErrorInvalidDQLQuery	The DQL query returned an error
503	ErrorCannotReachServer	Failed to reach server

detailMessage: The detailMessage part of the thrown exception can be from a variety of sources. The contents should provide the developer with detailed information about the source of the problem.