



# **ELO CMIS**

Installation & operation



# Table of contents

<b>Operation</b>	<b>3</b>
Overview	3
Differences between the ELO standard and the CMIS interface	9
CMIS functions that are not supported	13
Search	14
Appendix: Initial settings with OpenCMIS Workbench	16

# Operation

## Overview

### Introduction to the technology

The interface is based on a JSR-compatible web application combined with the Apache Chemistry stack latest version (12.0-SNAPSHOT). In this context, the library package suggested by Chemistry is used. This includes the OpenCMIS libraries of the Chemistry project as well as the JAX-WS libraries for the SOAP binding.

### System requirements and specifications

You need to take the following system requirements and specifications into account:

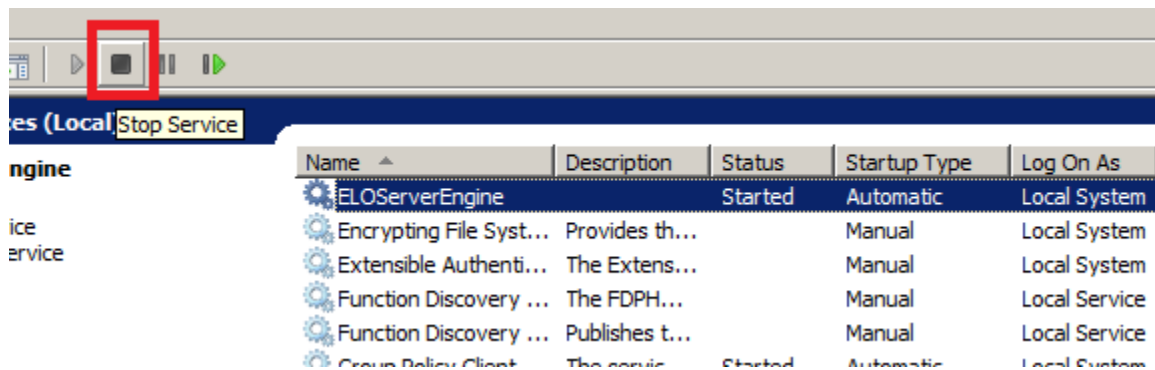
System	Requirements	Versions used
Java Runtime	Java 13 and later	13.0.1
Server engine	ELO CMIS 21.1 and previous versions require Servlet API 4.0 or other compatible servlet container (for instance Tomcat 9.0.31).	Tomcat 9.0.31
	ELO CMIS since version 21.3 requires Servlet API 5.0 or other compatible servlet container (for instance Tomcat 10.0.27).	Tomcat 10.0.27
ELO	Version 11 and higher	
	If ELO CMIS connects to an ELO Indexserver 20 or earlier (ELO 11, ELO 12, ELO 20), the interface provided by the module will be CMIS API 1.0.	
	If ELO CMIS connects to an ELO Indexserver 21 or later and if the <i>cmisVersion</i> parameter in the configuration is set to <i>1.1</i> , the interface provided by the module will be CMIS API 1.1.	
Memory & CPU	Depending on load. However, we recommend 250 MB for the web app and 5 MB for each user. This value must be increased due to the ELOix sessions, as the CMIS application is not used for managing session data. The recommended load per user is 30 MB. This value may need to be scaled up to take into account larger document uploads, as binding can cause binary data to be cached to memory.	
Operating system	See Java VM	Windows 2012 R2, Linux
HDD	The interface caches data to the file system when uploading documents. This needs to be taken into account when computing the load per user.	

System	Requirements	Versions used
CMIS Client	CMIS Client 1.0 or 1.1 with an AtomPub, SOAP	OpenCMIS Workbench 1.1 (AtomPub, SOAP), SoapUI

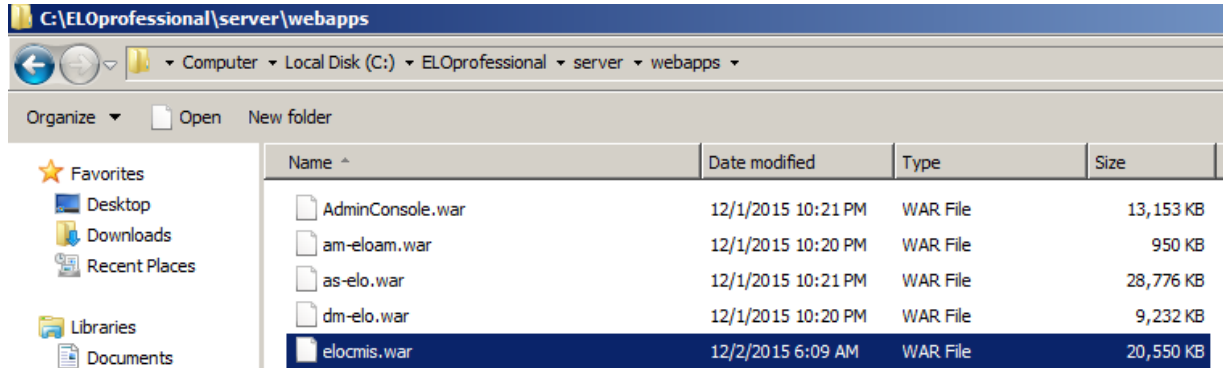
## Installation

The installation describes the procedure for a Tomcat-compatible servlet container.

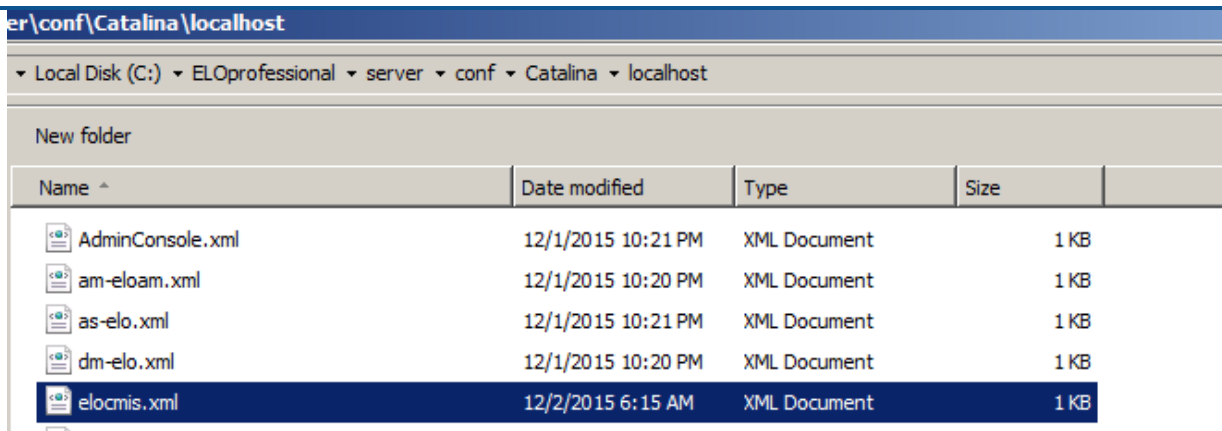
The installation behaves just like other ELO web applications.



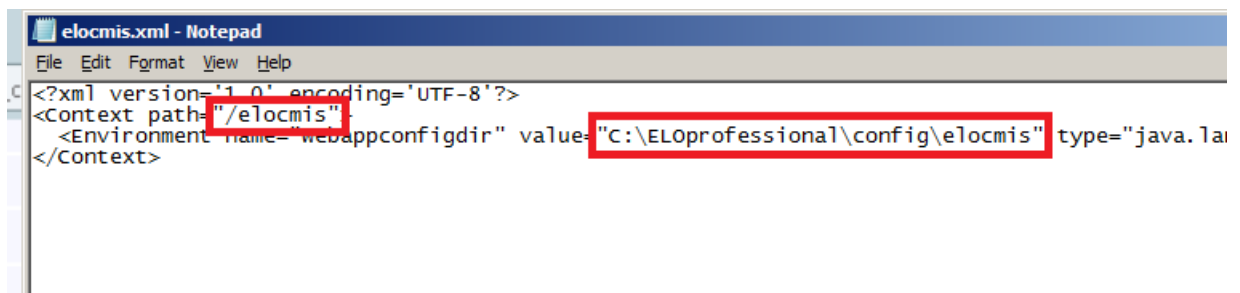
1. First, you have to stop the ELO Server Engine/Servlet Container.



2. Place the web application from the installation package (*cmis.war*) in the webapps folder. Change the name of the web application, if necessary.



3. Create a context descriptor for the web applications in the `conf//Catalina//localhost` directory. The name of the descriptor must be identical to the base name of the application.



4. Modify the parameters of the context descriptor.
5. Copy the `config.xml` and `logback.xml` files to the configuration directory and modify as required.
6. Import the provided script `RF_ELOcmis_getCheckedOutDocs.js` and load it in each ELO Indexserver, which has to be connected:
  - a) Copy this script at the location `Administration\IndexServer Scripting Base\_ALL`.
  - b) Reload the ELO Indexserver scripts.

## Configuration parameters

### Global parameters

Name	Description	Example
<code>repositories.enabled</code>	List of active repositories, separated with a comma	<code>reponame1,reponame2</code>

### Repository-specific parameters

Name	Description	Example
<code>&lt;reponame&gt;.id</code>	Repository ID (ID for the CMIS interface)	20

Name	Description	Example
<reponame>.name	Repository name	ELO20
<reponame>.indexserverurl	ELO Indexserver URL	http://localhost:9090/ix-ELO20/ix
<reponame>.username	ELO user name (administrative user). This entry is required for an alternative authentication mechanism described in the Authentication section.	EloService
<reponame>.password	ELO password. If this entry is in clear text, it will be automatically encrypted in the configuration file at ELO CMIS start. Required for an alternative authentication mechanism.	52-247-139-10-8-11-59-34
<reponame>.secret	A secret passphrase used to perform an alternative authentication mechanism. Can be MD5-encrypted.	
<reponame>.rootfolderid	GUID of the repository root folder. By default, this entry is set to 1, which means that the CMIS repository root corresponds to the ELO repository root. If there is an entry for the root folder in the configuration file and this 1 root folder cannot be found, all structure changes in the repository are forbidden. A <i>CmisNotFoundException</i> is thrown with an explicit error message.	
<reponame>.eloprefix	Prefix for all types and properties associated with ELO. By default, this prefix is set to <i>elo</i> .	sapbo
<reponame>.maxsearchtimeout	Maximum timeout for <i>getFolderTree/getDescendants</i> in seconds (default = 30 20)	
<reponame>.ixSearchDelimiter	Replaces the ELO Indexserver default search delimiter ('...').	###
<reponame>.cmisVersion	If you want the module to serve the CMIS interface in version 1.1, this parameter must be set to <i>1.1</i> . Any other value or no value at all will result in a CMIS 1.0 interface.	1.1

Name	Description	Example
	With this parameter, no connection to the Indexservers is established when calling the <i>getRepositories</i> method.	

### Information

Some configuration entries present in previous ELO CMIS versions were obsolete and have been removed in ELO CMIS version 21.00.010. Three of them were used to enable some functions that are now enabled by default: *supportfoldertree*, *supportdescendants*, and *supportdeletecontentstream*. Another entry was used to define sub-repositories in the configuration file: *subrepositories*.

## Authentication

### Normal authentication with ELO user

In standard cases, the CMIS authentication information sent with the requests corresponds to a given ELO user and their password. The rights and privileges of this user are considered for all actions they perform within the repository.

### Alternative authentication mechanism with 'runAs'

An alternative authentication mechanism exists for some special cases, like the execution of automatic tasks, which should be executed with different roles and users. For such a scenario, the use of user passwords is unpractical: Authentication is then based on a single ELO administrative account that performs a runAs for the users. The parameters *username* and *password* in the configuration file allow the system to authenticate the administrative user. The parameter *secret* in the configuration defines a passphrase, which has to be used in CMIS requests instead of the user's password.

Note that the normal authentication mechanism can also be performed simultaneously for users who need to perform interactive actions thanks to a peer login/password.

## Version information

### ELO CMIS versions

Previously, an ELO CMIS release (e.g. ELO CMIS 11, ELO CMIS 12, etc.) was built for each ELO release (e.g. ELO 11, ELO 12, ELO 20, etc.). From ELO CMIS 21, a single release will be maintained, which can serve all ELO versions. In practice, ELO CMIS 21 can connect with ELO 12, ELO 20, and ELO 21.

---

## **ELO versions and CMIS APIs**

ELO 20 and previous versions are compliant with *CMIS API 1.0*. *CMIS API 1.1* has been applied since ELO 21.

## **Supported bindings**

AtomPub and SOAP/WSDL bindings are supported. The JSON binding is not supported.



## Differences between the ELO standard and the CMIS interface

### Object types

CMIS requires object types to have unique identifiers (=TypeID). Every object type can inherit from the *document* or *folder* base class. This does not follow the ELO principle that allows a metadata form to be assigned to a document or a folder. Metadata form names are passed on. If there is a metadata form named *MetadataForm1* that is only assigned to folders, this name will always be used. If the metadata form is assigned to documents as well as folders, then two types are generated: *Metadata Form1* for documents and *Metadata Form1\_Folder* for folders.

Types can be created and deleted, but not updated. Note that all types have to inherit the base class *document* or *folder*. Inheritance between types isn't allowed: Each type has a *typeMutability.create* property set to *false*.

### Object IDs

CMIS folder IDs correspond to their object GUID in ELO. CMIS document IDs follow this format: *<objectGUID>\_<DocGUID>*. Each document version is therefore assigned a unique ID as per the CMIS specification. The ID of the private working copy follows the same format but has the prefix *PWC*.

#### Please note

In the previous ELO CMIS versions, the CMIS ID was always the corresponding ELO object GUID, even for the documents with content. This new CMIS ID format is an intern aspect, which doesn't impact the work with ELO CMIS, through CMIS clients. Nevertheless, this new format can lead to problems for third-party modules, which persist such information from CMIS servers in order to implement some functionalities, absent from the CMIS specification, like workflows for instance. In this case, the third-party module's database has to be updated in order to replace the old CMIS ID with the new one for all entries corresponding to a document with content. This update can easily be done since the old ID corresponds to the CMIS property *versionSeriesId*. Running the method *getObjectOfLatestVersion* with the old CMIS ID as parameter allows to retrieve the new CMIS ID.

```
<?xml version='1.0' encoding='UTF-8' standalone='no' ?>
<soapenv:Header><wsse:Security xmlns:wsse="http://docs.oasis-open.org/soapenv/2003/11/soap-envelope" />
</soapenv:Header>
<soapenv:Body>
  <ns:getObjectOfLatestVersion>
    <ns:repositoryId>vm21</ns:repositoryId> Old CMIS ID
    <ns:objectId>{EA5D8CB1-A23D-D061-B84C-F42458FEBF82}</ns:objectId>
  </ns:getObjectOfLatestVersion>
</soapenv:Body>
</soapenv:Envelope>
```

```

<S:Header>
  <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
  </S:Header>
<S:Body>
  <getObjectOfLatestVersionResponse xmlns="http://docs.oasis-open.org/ns/cmis/messaging/200908/" xmlns:ns2="http:
    <object>
      <ns2:properties>
        <ns2:propertyId propertyDefinitionId="cmis:objectId">
          <ns2:value>EASD8CB1-A23D-D061-B84C-F424587EBF82) (4F80B5EB-E37B-BE98-EF20-4480A941F043)</ns2:value>
        </ns2:propertyId>
      </ns2:properties>
    </object>
  </getObjectOfLatestVersionResponse>
</S:Body>

```

## Document name

CMIS expects the object in repository to have exactly the same name as the file that was checked in. ELO removes the extension and keeps only the main part to name the newly created object in the repository.

## Versions

Since ELO doesn't have private working copies, PWC corresponds to the last version, but it is assigned a unique ID thanks to its prefix *PWC*. Therefore, PWC is not the latest version, in accordance with the CMIS specification.

## Relationships

Relationships aren't supported.

## Renditions

Renditions aren't supported.

## Policies

Policies aren't supported.

## Change tokens

Change tokens aren't supported.

## Items

Items are available for ELO 21 and higher, in accordance with CMIS API 1.1.

## Secondary types

Secondary types are available for ELO 21 and higher, in accordance with *CMIS API 1.1*. Note that, like the primary types, the secondary types can also be searched.

However, ELO doesn't have any similar concept. The secondary type implementation therefore relies on dynamic modifications of the metadata forms. These modifications are transparent in

CMIS and CMIS types don't show any changes. Nevertheless, note that ELO metadata forms may be modified and extended with further properties using secondary types.

When creating a secondary type, its description won't be saved for technical reasons.

## Map fields

The additional information in ELO (map fields) is currently not supported.

## Special characters

As CMIS clients are to some extent also file system clients, special characters such as `<>:"/|?*` are not supported. However, these do occur in ELO. These objects are currently not transferred to CMIS clients.

## Historical metadata

Historical metadata is returned in compliance with the CMIS specification.

## Activities and workflows

There is currently no function on the CMIS ports that can map activities and workflows.

## Types of metadata fields

The following types defined in the CMIS specification aren't available in ELO and thus are converted to string: *BOOLEAN, ID, HTML, URI*.

ELO type	CMIS type
ISO date	DATETIME
Number, number 0 decimal places	NUMBER
Number with fixed width	DECIMAL (32-BIT)
Other types	STRING

## XDate

For each repository object, in addition to the CMIS and type-related properties, ELO CMIS also provides an additional ELO-specific property: *eloproperty:date* which corresponds to the ELO *xdate* property.

## Caching

To avoid unnecessary and numerous requests, ELO CMIS includes a system of caches:

- a cache containing the IDs of the objects in the repositories: this cache allows to check without expensive requests that no access is done outside the CMIS repository (defined thanks to the *rootfolderid* configuration parameter).
- A cache containing the CMIS types (and the associated ELO DocMasks and Aspects).

These caches are of course updated if changes are made to the structure or types using CMIS methods. If changes are made outside ELO CMIS (with an ELO client, ELO Java Client or ELO Administration Console for example), the caches must be refreshed using the '*Clear caches*' control. This control is available for each repository on the ELO CMIS status page.

## CMIS functions that are not supported

### Object Service

Function	Description
createRelationship	Creates a relationship object of the specified type.
createPolicy	Creates a policy object of the specified type.
getRenditions	Gets the list of associated renditions for the specified object. Only rendition properties are returned, not rendition streams.

### Policy Service

Function	Description
applyPolicy	Applies a specified policy to an object.
getAppliedPolicy	Gets the list of policies currently applied to the specified object.
removePolicy	Removes a specified policy from an object.

### Type Service

Function	Description
updateType	Updates a type definition.

### Discovery Service

Function	Description
getContentChanges	Gets a list of content changes. This service is intended to be used by search crawlers or other applications that need to efficiently understand what has changed in the repository.

## Search

DiscoveryService/query are used to support searches.

However, these are limited to searches within folders, basic properties such as short text, or the ELO metadata.

The *cmis:name* is always made up of the short text followed by the extension of the document version (for file system CMIS clients). However, this is not available for the search in this form. It is only possible to search the *cmis:name* without the extension.

### FROM

In FROM expressions, the following targets are supported:

<i>cmis:folder</i>	Search for folders only
<i>cmis:document</i>	Search for documents only
<i>elo:&lt;METADATA_FORM&gt;</i>	Search for items with metadata form <i>&lt;METADATA_FORM&gt;</i> . The metadata form name or the Mask ID are supported.

### WHERE

In WHERE expressions, the following criteria are supported:

<i>cmis:name</i>	Search according to the item name. Wildcards ( %) are supported if the LIKE operator is used.
<i>cmis:objectId</i>	Filter search result by object ID
<i>cmis:objectTypeId</i>	Search according to the metadata form name
<i>cmis:creationDate</i>	Search according to the creation date of the item. Here, you can search according to ISO Strings (YYYYMMDD) or with <i>TIMESTAMP '&lt;string&gt;'</i> (the exact syntax is documented in the CMIS specification). A special feature of ELO is the search within an interval. Beside ISO strings, searches within a range are also supported. For instance, a search for <i>20180101...20191231</i> will return all elements which were created between January 2018 and December 2019 (the short request <i>201801...201912</i> will actually return the same result).
<i>cmis:lastModificationDate</i>	Search according to the last date the item was modified. For the syntax, see <i>cmis:creationDate</i> .
<i>elo:&lt;METADATA_FIELD&gt;</i>	Search according to values from metadata fields. Search values can contain escaped single quotes. Supported escape characters are backslash or (a second) single quote.
<i>IN_FOLDER</i>	Search within a given folder (see <i>Example 1</i> ).

The AND and OR operators are supported if they follow the pattern:

WHERE *exp1* AND *exp2* AND (*exp3* OR *exp4* OR ...) AND *exp5* AND (*exp6* OR *exp7* OR ...) AND ...

The following has to be considered when writing a WHERE expression:

- If there is at least one AND operator, OR operators have to be encapsulated in blocks with parentheses.
- No block encapsulated in another block.
- As many AND and OR operators as needed: nevertheless consider performance issues; the ELO search API isn't designed for such SQL-Like WHERE expressions with OR operators used this way. To mimic this behavior, a dedicated search has to be made for each OR expression. The global result is computed and sorted afterwards. Considering this, a complex WHERE expression can lead to a significant latency.

The LIKE predicates are case-insensitive:

A predicate of the form ... where cmis:name LIKE 'a%' ... will match a *cmis:name* starting with a or A.

## ORDER BY

The ORDER BY expression supports the following criteria:

*cmis:XXX* All CMIS item properties (refer to the CMIS specification): e.g. cmis :name  
or cmis:creationDate

*elo:<METADATA\_FIELD>* Order according to values from metadata fields

## Examples

Example 1: Searching for documents within a folder

```
SELECT \* FROM cmis:document WHERE IN\_FOLDER(' (C9A6CC1D-C820-D957-5712-C2C629E973EF) ')
```

Example 2: Searching for documents using the short text

```
SELECT \* FROM cmis:document WHERE cmis:name LIKE 'Test%'
```

Example 3: Searching using a metadata form and a metadata field

```
SELECT \* FROM elo:Email WHERE elo:EL00UTL2 = '222'
```

Example 4: Searching for documents created after March 18, 2018

```
SELECT \* FROM cmis:document WHERE cmis:creationDate \> TIMESTAMP '2018-03-18T00:00:00.000'
```

```
SELECT \* FROM cmis:document WHERE cmis:creationDate = '20180318...'
```

Example 5: Searching for documents within a folder

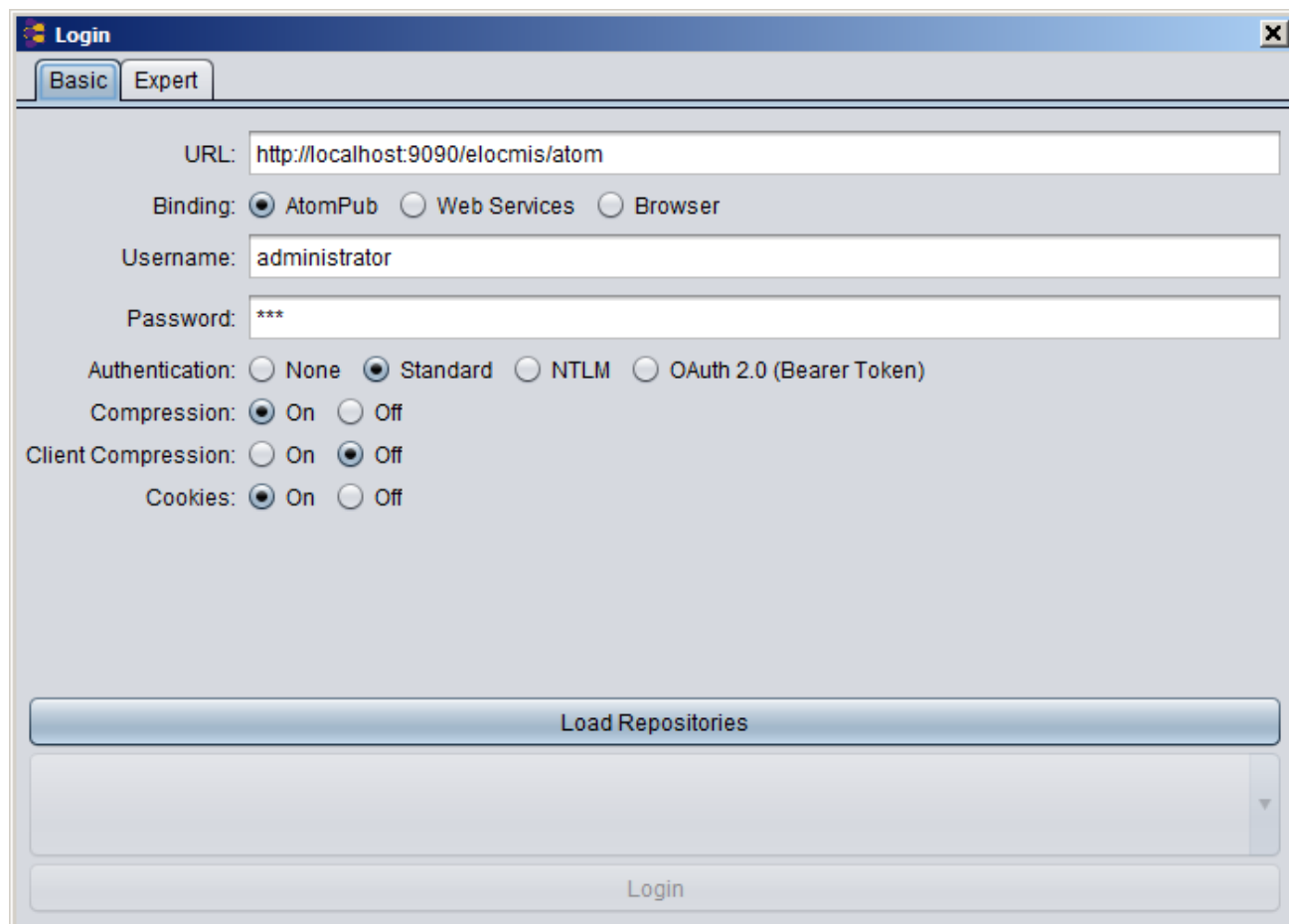
```
SELECT \* FROM cmis:document WHERE IN\_FOLDER(' (C9A6CC1D-C820-D957-5712-C2C629E973EF) ')
```

## Appendix: Initial settings with OpenCMIS Workbench

Sample settings to help you get started.

Current version for download: <https://chemistry.apache.org/java/download.html>

Java 8 version tested with SOAP: <http://archive.apache.org/dist/chemistry/opencmis/0.7.0/>



The screenshot shows the 'Login' dialog box of the OpenCMIS Workbench. It has a title bar with a close button. Below the title bar are two tabs: 'Basic' (selected) and 'Expert'. The 'Basic' tab contains the following fields and options:

- URL:
- Binding: ☒ AtomPub ☐ Web Services ☐ Browser
- Username:
- Password:
- Authentication: ☐ None ☒ Standard ☐ NTLM ☐ OAuth 2.0 (Bearer Token)
- Compression: ☒ On ☐ Off
- Client Compression: ☐ On ☒ Off
- Cookies: ☒ On ☐ Off

Below these settings is a large button labeled 'Load Repositories'. At the bottom of the dialog is a 'Login' button.



The screenshot shows a 'Login' dialog box with a title bar and a close button. It has two tabs: 'Basic' (selected) and 'Expert'. The 'Basic' tab contains the following fields and options:

- URL:
- Binding: ☐ AtomPub ☒ Web Services ☐ Browser
- Username:
- Password:
- Authentication: ☐ None ☒ Standard ☐ NTLM ☐ OAuth 2.0 (Bearer Token)
- Compression: ☒ On ☐ Off
- Client Compression: ☐ On ☒ Off
- Cookies: ☒ On ☐ Off

Below these fields is a 'Load Repositories' button. Underneath this button is a list box containing the text 'elo' and '1000'. At the bottom of the dialog is a 'Login' button.

Login

BasicExpert

URL:

Binding: ☐ AtomPub ☒ Web Services ☐ Browser

Username:

Password:

Authentication: ☐ None ☒ Standard ☐ NTLM ☐ OAuth 2.0 (Bearer Token)

Compression: ☒ On ☐ Off

Client Compression: ☐ On ☒ Off

Cookies: ☒ On ☐ Off

Load Repositories

elo

1000

Login