

# CMIS Open Source

## Introduction

CMIS Repo is an implementation of the CMIS open standard for document management. It can be used as the backend server for any document or content management application, and provides the basic features needed, such as authentication, ACL management, filing, etc. It currently implements versions 1.0 and 1.1 of the standard, and is fully functional. It will be adapted to the current version of the standard in the future.

This solution was developed under ISA<sup>2</sup> action **2016.38. Legislation Interoperability Tools – LEGIT**.

## A. Download and Installation

### Prerequisites

1. Install Apache Tomcat version 8.x or above.
2. Oracle database with an up and running instance.

### Download

1. [Latest CMIS Repo](#)
2. [Apache CMIS Workbench](#) (If not installed already)

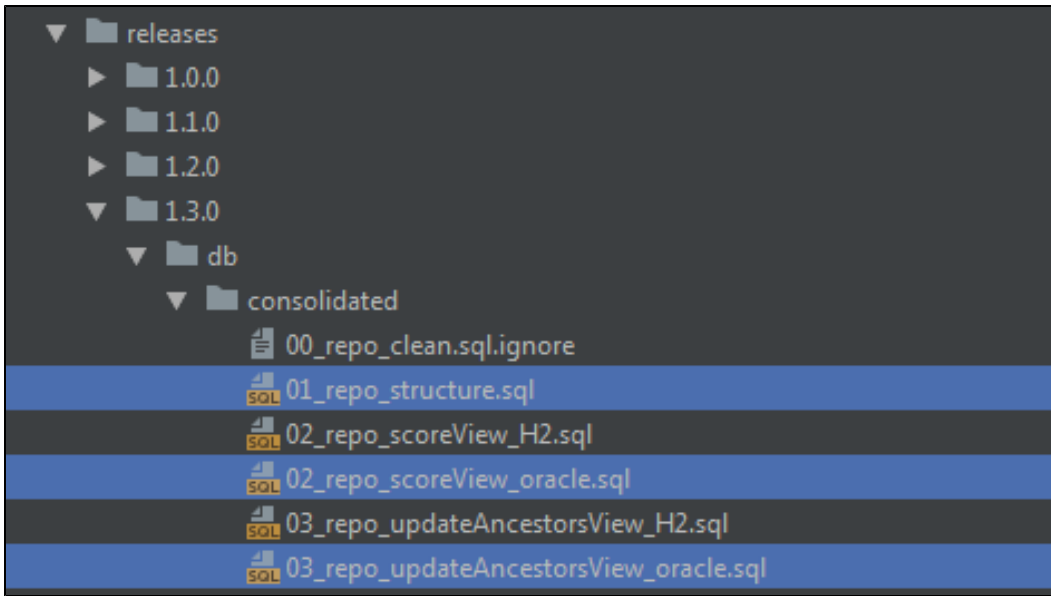
### Build and Install CMIS Application

1. Unzip the CMIS release, open command prompt and run below maven command to generate the war
2. Package CMIS war

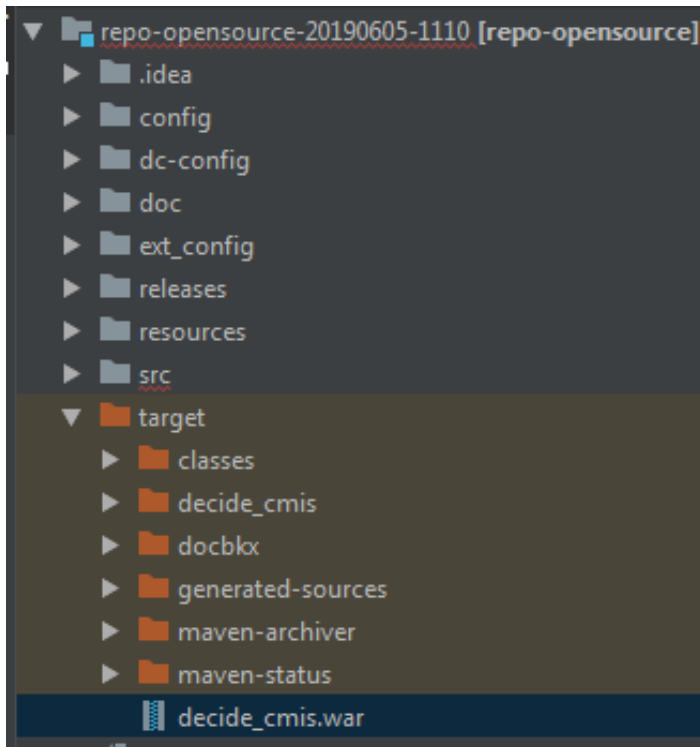
```
mvn clean package -Dmaven.test.skip=true
```

### Schema creation and Deployment in Tomcat

1. Execute the three SQL-scripts against the database (**01\_repo\_structure.sql**, **02\_repo\_scoreView\_oracle.sql** and **03\_repo\_updateAncestorsView\_oracle.sql**) to create the schema. these SQLs are present in **release** folder, see below:



2. Go to **target** folder, which is created as a result of previously run maven command and Unpack the war-file in [CATALINA\_HOME] /webapps/[APP\_BASE] - where APP\_BASE is the name you have chosen for the application.



3. Modify properties file, in extracted release package go to folder named **ext-config**:

1. environment.properties

Property Name	Value
REPO_DB_URL	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=...
REPO_DB_USERNAME	LEOS_CMIS
REPO_DB_PASSWORD	*****

REPO_INDEX_DB_FILE	/ec/prod/app/webroot/home/repo/repo.index
--------------------	---

## 2. log4j\_repo.properties

Property Name	Value
log4j.appender.logfile.File	/ec/prod/app/webroot/home/repo/logs/repo.log
log4j.appender.repoupdate.File	/ec/prod/app/webroot/home/repo/logs/repoupdate.log
log4j.appender.timelogappender.File	/ec/prod/app/webroot/home/repo/logs/repo-timer.log

## 3. repo\_adminConfig.xml, bydefault username is **username** and password is **\*\*\*\*\*** (6 stars)

Property Name	Value
<username>	username
<password>	*****

## 4. repo\_securityConfig.xml, admin\_dev is the default password

Property Name	Value
<password>	admin_dev

4. Add the path of configuration files (**ext-config**) in server classpath.
  1. Go to apache tomcat **conf** folder and open catalina.properties, add the path in common.loader property
5. Add your Oracle JDBC driver to either [CATALINA\_HOME]/lib or [CATALINA\_HOME]/webapps/[APP\_BASE]/WEB-INF/lib
6. Start tomcat.
7. In a standard install the web app can be accessed on [http://localhost:9090/\[APP\\_BASE\]/](http://localhost:9090/[APP_BASE]/)

# REPO

## Trade Document Repository

CMIS 1.0 bindings	CMIS 1.1 bindings	CMIS 1.0 client configurations	CMIS 1.1 client configurations	Other
<b>Web Services binding (combined)</b> <a href="http://localhost:9090/decide_cmis/services/cmis?wsdl">http://localhost:9090/decide_cmis/services/cmis?wsdl</a>				
<b>Web Services binding (divided)</b> <a href="http://localhost:9090/decide_cmis/services/RepositoryService?wsdl">http://localhost:9090/decide_cmis/services/RepositoryService?wsdl</a> <a href="http://localhost:9090/decide_cmis/services/NavigationService?wsdl">http://localhost:9090/decide_cmis/services/NavigationService?wsdl</a> <a href="http://localhost:9090/decide_cmis/services/ObjectService?wsdl">http://localhost:9090/decide_cmis/services/ObjectService?wsdl</a> <a href="http://localhost:9090/decide_cmis/services/VersioningService?wsdl">http://localhost:9090/decide_cmis/services/VersioningService?wsdl</a> <a href="http://localhost:9090/decide_cmis/services/RelationshipService?wsdl">http://localhost:9090/decide_cmis/services/RelationshipService?wsdl</a> <a href="http://localhost:9090/decide_cmis/services/DiscoveryService?wsdl">http://localhost:9090/decide_cmis/services/DiscoveryService?wsdl</a> <a href="http://localhost:9090/decide_cmis/services/MultiFilingService?wsdl">http://localhost:9090/decide_cmis/services/MultiFilingService?wsdl</a> <a href="http://localhost:9090/decide_cmis/services/ACLService?wsdl">http://localhost:9090/decide_cmis/services/ACLService?wsdl</a> <a href="http://localhost:9090/decide_cmis/services/PolicyService?wsdl">http://localhost:9090/decide_cmis/services/PolicyService?wsdl</a>				
<b>AtomPub binding</b> <a href="http://localhost:9090/decide_cmis/atom">http://localhost:9090/decide_cmis/atom</a>				

**Note:** change the default port(8080) in tomcat server.xml , this port may be already used.

## Creating a Repository in CMIS Application

1. Open the application in browser [http://localhost:9090/decide\\_cmis](http://localhost:9090/decide_cmis) (use the APP\_BASE name if different than decide\_cmis)
2. Go to the tab '**Other**', and click on link give below 'Administrative Panel'

## REPO

### Trade Document Repository

[CMIS 1.0 bindings](#)
[CMIS 1.1 bindings](#)
[CMIS 1.0 client configurations](#)
[CMIS 1.1 client configurations](#)
[Other](#)

#### Administration panel

[http://localhost:9090/decide\\_cmis/admin](http://localhost:9090/decide_cmis/admin)

#### Web interface

[http://localhost:9090/decide\\_cmis/web](http://localhost:9090/decide_cmis/web)

3. Entering the username and password as configured in the file "repo\_adminConfig.xml"  
e.g.  
User: **username**  
Password: **\*\*\*\*\*** (6 stars)

**Login with Username and Password**

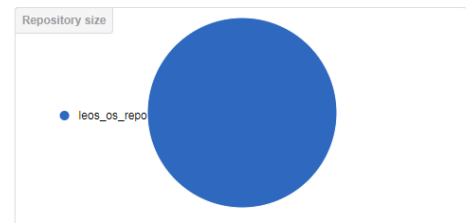
User:   
Password:

### Administration page

Home

[Create new repository](#)  
[Sessions](#)  
[Configuration](#)  
REPOSITORIES  
[Open Source Repository](#)

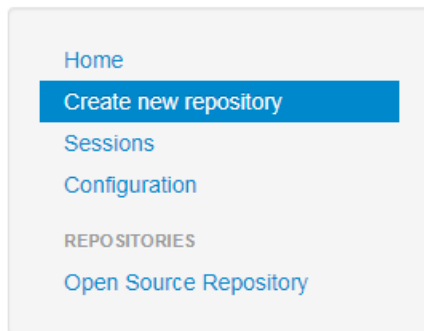
Repository Id	Repository Name	Repository Description
leos_os_repo	Open Source Repository	Open Source Repository



**Note:** in your case the repository will be empty first time you use it

4. To create an empty repository click on 'Create new repository' and set repository id, name and description and click on button 'Create Repository'

# Create new repository

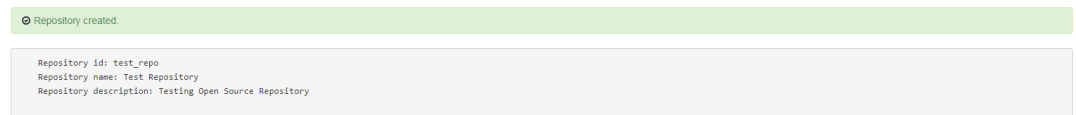
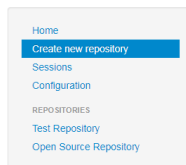


Repository ID:

Repository name:

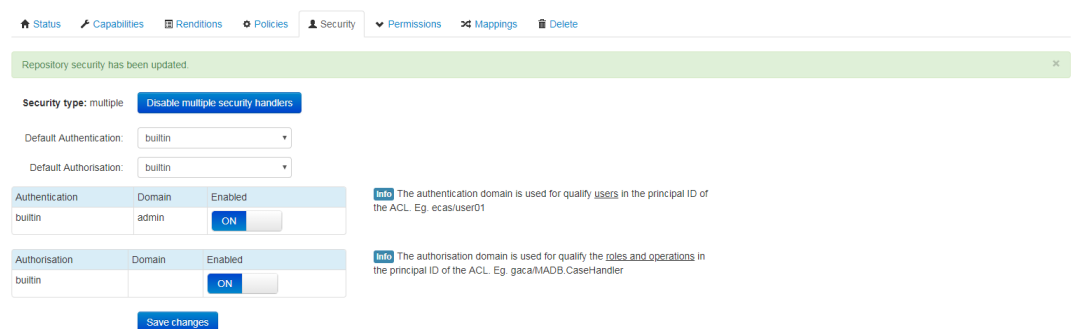
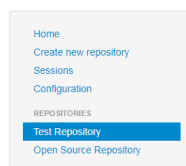
Repository description:

## Create new repository



5. After repository is created, click in 'Security option' and then click on Enable multiple security handlers . Check that the configuration is the same as in the following image. Authentication and Authorization handlers will be builtin.

## test\_repo / Test Repository / Testing Open Source Repository



Click on 'Save changes'.

6. Click on Capabilities tab in the menu and check that all properties are configured in the same way as in the following image.

test\_repo / Test Repository / Testing Open Source Repository

Home  
Create new repository  
Sessions  
Configuration  
REPOSITORIES  
**Test Repository**  
Open Source Repository

Status Capabilities Renditions Policies Security Permissions Mappings Delete

Get descendants: ☒ ON  
Get folder tree: ☒ ON  
Content Stream updatability: anytime  
Changes: objectidsonly  
Renditions: read  
Multifiling: ☒ ON  
Unfiling: ☒ ON  
Version specific filing: ☐ OFF  
PWC updatable: ☒ ON  
PWC searchable: ☒ ON  
All versions searchable: ☒ ON  
Query: bothcombined  
Join: innerandouter  
ACL: manage  
ACL Propagation: propagate  
Save changes

**Note:** Check "Version specific filing" capability. By default it is configured as ON and this causes that different versions of the same file appear as individual files in the repository. Set it to OFF and 'Save changes'.

## Login in Repository using Apache Cmis Workbench

1. Open previously downloaded Apache Chemistry Workbench (if not downloaded , use the this link <http://chemistry.apache.org/java/download.html>)
2. Unzip the file.
3. Execute it (you don't need to install it)
  - Windows: double click workbench.bat
  - Linux/Mac OS : start a terminal session, go to the folder where the downloaded file was unzipped, start from the terminal the workbench.sh script
4. In parameter screen, put the connection details:
  - url: [http://localhost:9090/\[APP\\_BASE\]/services11/cmis?wsdl](http://localhost:9090/[APP_BASE]/services11/cmis?wsdl) (adapt it to your host)
  - binding: web services
  - username: builtin/admin
  - password: xxxxx (the password defined on the property **<password>admin\_xxx</password>** on the file repo\_securityConfig.xml)
5. Click on load repositories.
6. From drop down, select a repository (repository created in the previous section).
7. Click login.

Login

Basic Expert Discover

URL:

Binding: ☐ AtomPub ☒ Web Services ☐ Browser

Username:

Password:

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

Compression: ☒ On ☐ Off

Client Compression: ☐ On ☒ Off

Cookies: ☒ On ☐ Off

CSRF Header:

Language:

Connect timeout (secs):

Read timeout (secs):

Load Repositories

Test Repository

test\_repo

Testing Open Source Repository

Login

CMIS Workbench - (builtin/admin) - Test Repository

Connection Repository Info Types Query Change Log Console TCK Create Object Log Info

up / go

Name	Type	Content Type	Size	Created
------	------	--------------	------	---------

**Name:** rootFolder

Object ID: [e35106f557e98bc319ab8e6c84cc20257257fbc](#)

Latest State ID:

Type: cmis:folder

Base Type: cmis:folder

Secondary Types:

Paths: /

MIME Type:

Size:

Version Label:

PWC:

Content URL: (not available)

Allowable Actions:

- CAN\_UPDATE\_PROPERTIES
- CAN\_GET\_PROPERTIES
- CAN\_GET\_OBJECT\_RELATIONSHIPS
- CAN\_GET\_OBJECT\_PARENTS
- CAN\_GET\_DESCENDANTS
- CAN\_APPLY\_POLICY
- CAN\_GET\_APPLIED\_POLICIES
- CAN\_REMOVE\_POLICY
- CAN\_GET\_CHILDREN
- CAN\_CREATE\_DOCUMENT
- CAN\_CREATE\_FOLDER
- CAN\_CREATE\_RELATIONSHIP
- CAN\_DELETE\_TREE
- CAN\_GET\_ACL
- CAN\_APPLY\_ACL

ACL: is not exact

Refresh Check specification compliance Open console

Now you have logged in to the empty repository.

## B. Integration with Leos Joinup

1. Update the Properties file in LEOS

1. Below Properties needs to be updated in **local.properties**, please check the comments between ##

```
leos.cmis.repository.id=test_repo ##Id of the repository given
at the time of repository creation##
leos.cmis.repository.username=builtin/admin ##As given in
repo_securityConfig.xml##
leos.cmis.repository.password=admin_dev ##As given in
repo_securityConfig.xml##
leos.cmis.repository.url=http://localhost:9090/[APP_BASE]
```

2. Create Type

1. Open Apache CMIS Workbench using AtomPub Binding, as shown below, can see atom url on Home Page

# REPO

## Trade Document Repository

CMIS 1.0 bindings

CMIS 1.1 bindings

CMIS 1.0 client configurations

CMIS 1.1 client configurations

Other

**Web Services binding (combined)**  
[http://localhost:9090/decide\\_cmis/services/cmis?wsdl](http://localhost:9090/decide_cmis/services/cmis?wsdl)

**Web Services binding (divided)**  
[http://localhost:9090/decide\\_cmis/services/RepositoryService?wsdl](http://localhost:9090/decide_cmis/services/RepositoryService?wsdl)  
[http://localhost:9090/decide\\_cmis/services/NavigationService?wsdl](http://localhost:9090/decide_cmis/services/NavigationService?wsdl)  
[http://localhost:9090/decide\\_cmis/services/ObjectService?wsdl](http://localhost:9090/decide_cmis/services/ObjectService?wsdl)  
[http://localhost:9090/decide\\_cmis/services/VersioningService?wsdl](http://localhost:9090/decide_cmis/services/VersioningService?wsdl)  
[http://localhost:9090/decide\\_cmis/services/RelationshipService?wsdl](http://localhost:9090/decide_cmis/services/RelationshipService?wsdl)  
[http://localhost:9090/decide\\_cmis/services/DiscoveryService?wsdl](http://localhost:9090/decide_cmis/services/DiscoveryService?wsdl)  
[http://localhost:9090/decide\\_cmis/services/MultiFilingService?wsdl](http://localhost:9090/decide_cmis/services/MultiFilingService?wsdl)  
[http://localhost:9090/decide\\_cmis/services/ACLService?wsdl](http://localhost:9090/decide_cmis/services/ACLService?wsdl)  
[http://localhost:9090/decide\\_cmis/services/PolicyService?wsdl](http://localhost:9090/decide_cmis/services/PolicyService?wsdl)

**AtomPub binding**  
[http://localhost:9090/decide\\_cmis/atom](http://localhost:9090/decide_cmis/atom)



**Login**

Basic Expert Discover

URL:

Binding: ☒ AtomPub ☐ Web Services ☐ Browser

Username:

Password:

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

Compression: ☒ On ☐ Off

Client Compression: ☐ On ☒ Off

Cookies: ☒ On ☐ Off

CSRF Header:

Language:


Connect timeout (secs):

Read timeout (secs):

Load Repositories

Test Repo  
test\_repo  
Test Repo

Login

2. Click **Types** option in Workbench ,  , here we will create LEOS specific types

**CMIS Types - Test Repo**

Reload Check Compliance Save Type Definition Update Type Delete Type Create Type

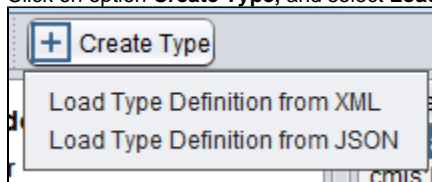
cmis:folder (cmis:folder)  
cmis:relationship (cmis:relationship)  
cmis:policy (cmis:policy)  
cmis:document (cmis:document)

**Name:** cmis:folder  
Description: cmis:folder  
Id: cmis:folder  
Local Namespace: http://ec.europa.eu/trade/r...  
Local Name: cmis:folder  
Query Name: cmis:folder  
Base Type: cmis:folder  
Creatable: ☒ Yes  
Fileable: ☒ Yes  
Queryable: ☒ Yes  
In Super Type Queries: ☒ Yes  
Full Text Indexed: ☒ Yes  
ACL Controlable: ☒ Yes  
Policy Controlable: ☒ Yes  
Type Mutability:  
Versionable:  
Content Stream Allowed:  
Allowed Source Types:  
Allowed Target Types:  
Extensions:

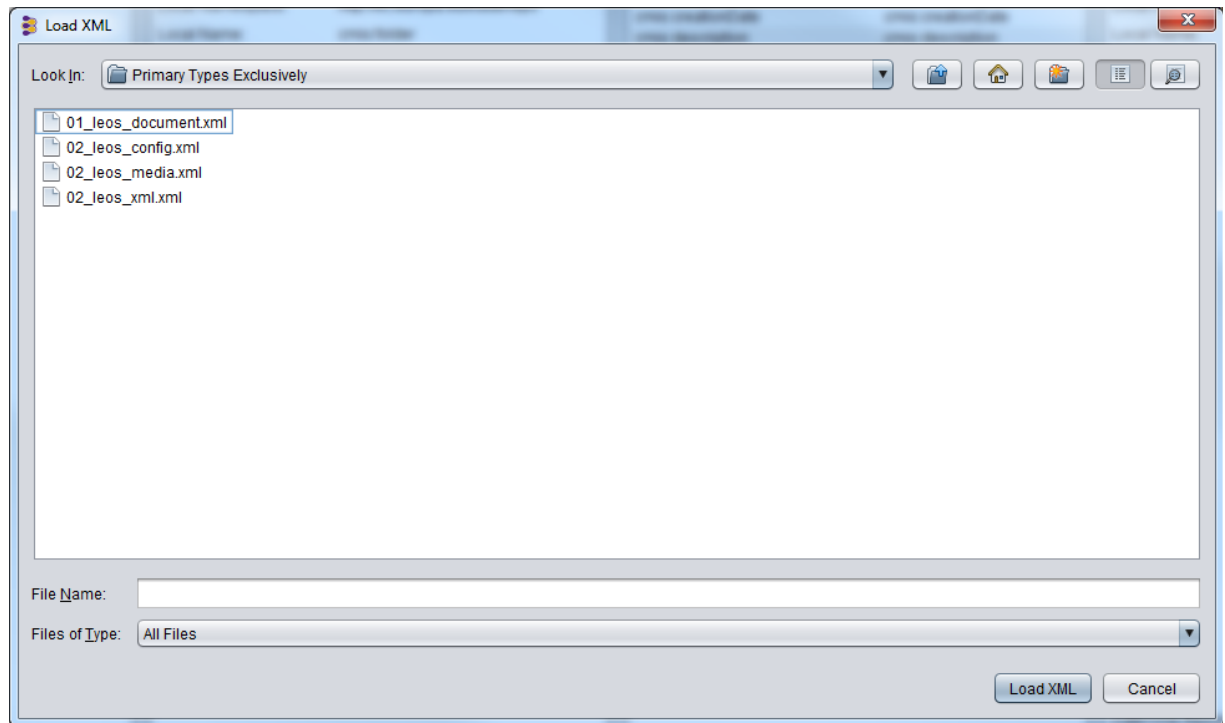
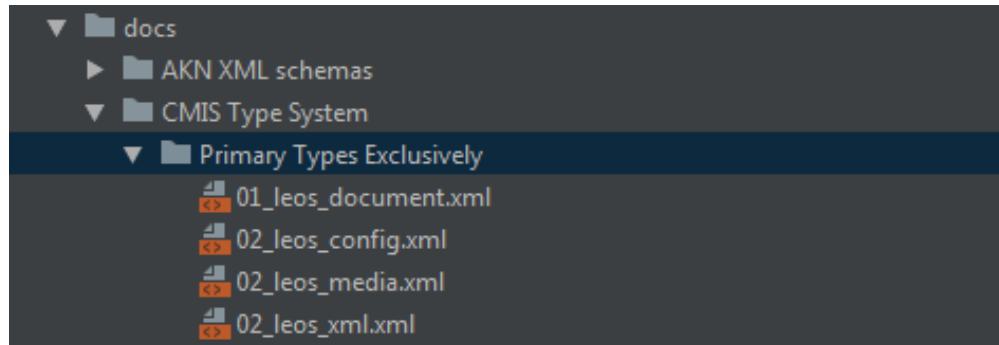
Name	Id
cmis:allowedChildObjectTypes	cmis:allowedChildObjectTypes
cmis:baseTypeId	cmis:baseTypeId
cmis:changeToken	cmis:changeToken
cmis:createdBy	cmis:createdBy
cmis:creationDate	cmis:creationDate
cmis:description	cmis:description
cmis:lastModificationDate	cmis:lastModificationDate
cmis:lastModifiedBy	cmis:lastModifiedBy
cmis:name	cmis:name
cmis:objectId	cmis:objectId
cmis:objectTypeId	cmis:objectTypeId
cmis:parentId	cmis:parentId
cmis:path	cmis:path
cmis:secondaryObjectTypes	cmis:secondaryObjectTypes

**Name:** cmis:allowedChildObjectTypes  
Description: Id's of the set of Object-type  
Id: cmis:allowedChildObjectTypes  
Local Namespace: http://ec.europa.eu/trade/r...  
Local Name: cmis:allowedChildObjectTypes  
Query Name: cmis:allowedChildObjectTypes  
Property Type: id  
Cardinality: multi  
Updatability: readonly  
Queryable: ☒ No  
Orderable: ☒ No  
Required: ☒ No  
Inherited: ☒ No  
Default Value:  
Open Choice: ☒ No  
Choices:   
Max Length:  
Min:  
Max:

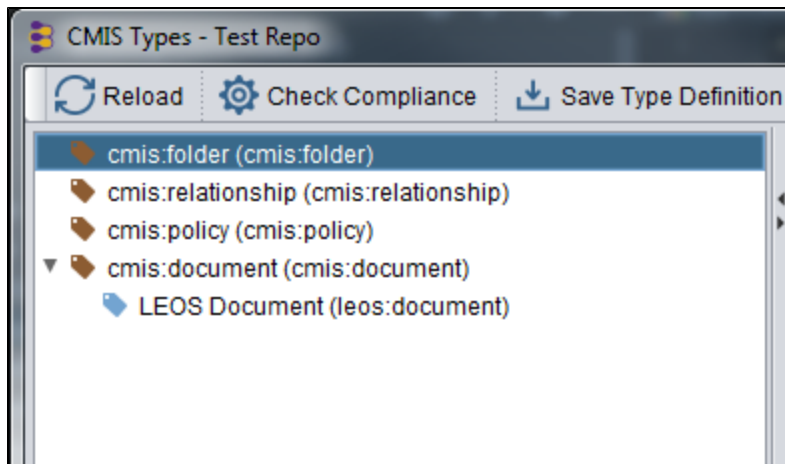
3. Click on option **Create Type**, and select **Load Type Definition fromXML**



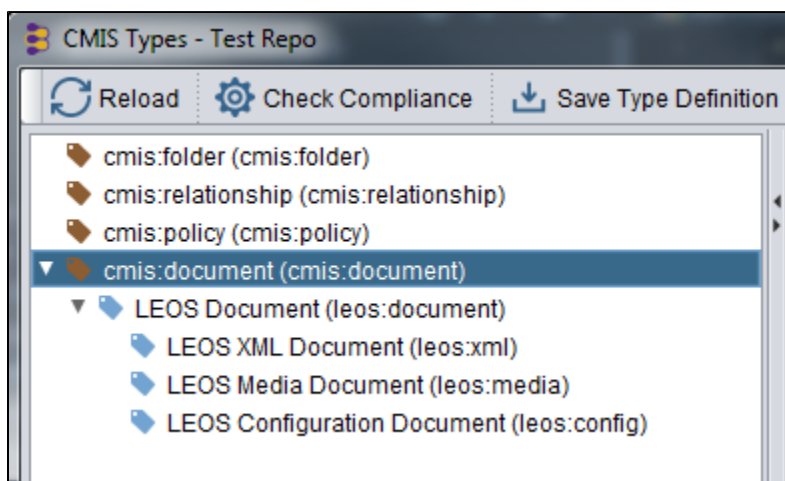
4. XML definition of LEOS types are present in **docs\CMIS Type System\Primary Types Exclusively**, in LEOS Joinup version(if not downloaded before, can be downloaded from [LEOS-Pilot 2.0.0](#))



select **01\_leos\_document.xml** and click on **Load XML**. Now this type definition can be found as a child of cmis:document, as shown below:



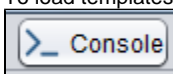
5. Load other types one by one ( 02\_leos\_xml.xml, 02\_leos\_media.xml, 02\_leos\_config.xml )



These types will be added as child of leos:document, verify and close the CMIS Types window.

6. Next step would be to load document templates and update the metadata using repository.properties, CMIS Types window can be closed now.

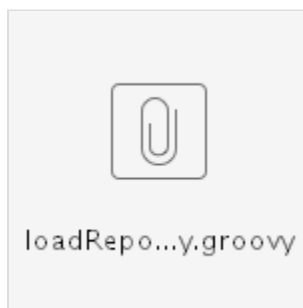
3. To load templates and metadata from repository properties, we need to run the given below Groovy script, to run the script Click on



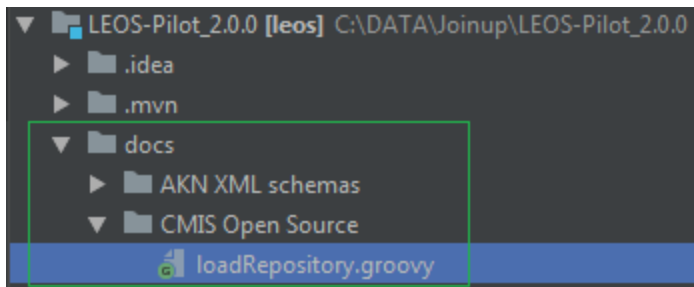
and select



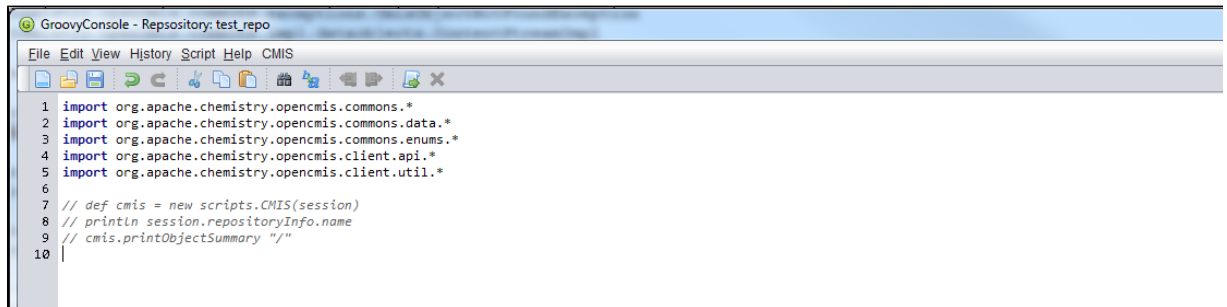
, it will open GroovyConsole



this script is present in Leos Joinup package in doc folder at below path



1. In Groovy Console, remove everything which is already there and Copy the above script completely

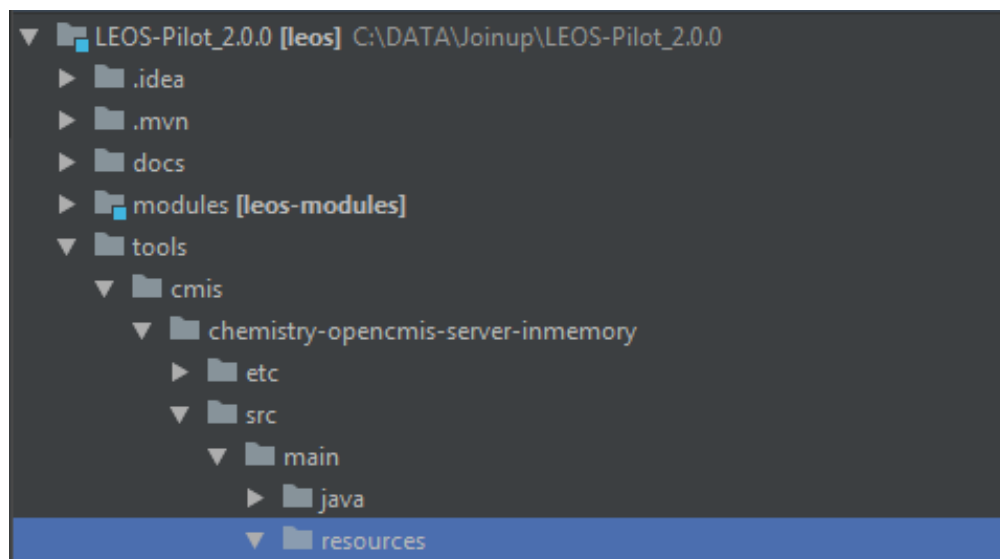


After putting the script in console we need to modify below <TODO>

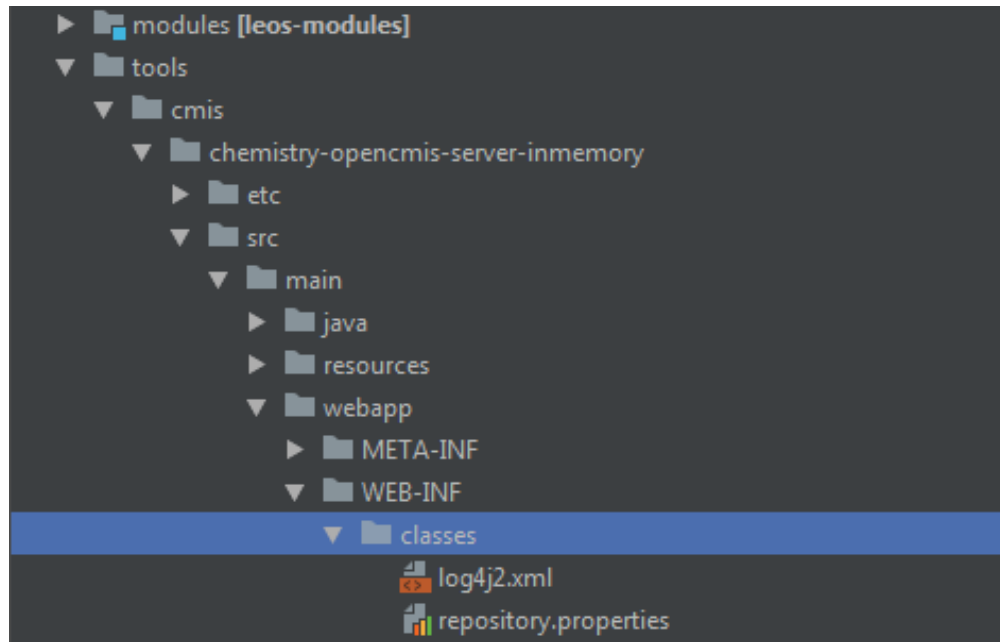
```
// ex. C:/devel/sources/pilot/releases/1.0.0-alpha7-council/CMIS/resources
def resourcesLocalPath = '<TODO>'

// ex. C:/devel/sources/pilot/releases/1.0.0-alpha7-council/CMIS/scripts/repository
def repoPropertiesLocalPath = '<TODO>'
```

1. for `resourcesLocalPath` we need to provide the path of resources, these templates are present inside 'cmis' folder in LEOS Joinup version at below location



2. for `repoPropertiesLocalPath` we need to provide the path of repository.properties, this file is present inside 'cmis' folder in LEOS Joinup version at below location

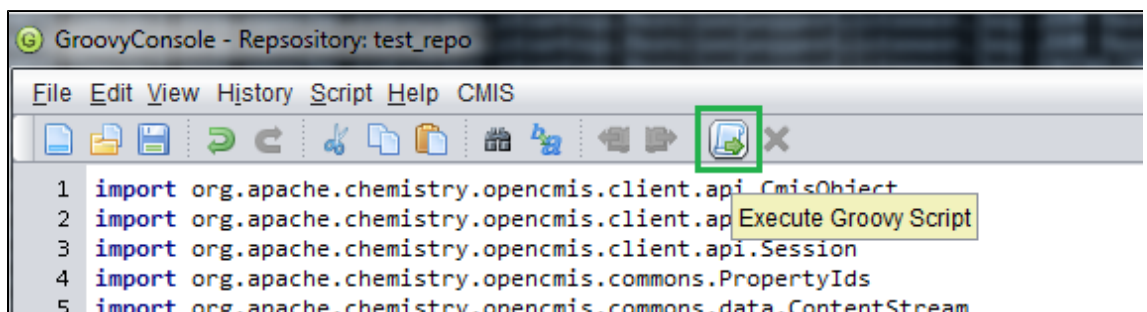


3. After updating, your script should look similar to this

```

GroovyConsole - Repository: test_repo
File Edit View History Script Help CMIS
1 import org.apache.chemistry.opencmis.client.api.CmisObject
2 import org.apache.chemistry.opencmis.client.api.ObjectId
3 import org.apache.chemistry.opencmis.client.api.Session
4 import org.apache.chemistry.opencmis.commons.PropertyIds
5 import org.apache.chemistry.opencmis.commons.data.ContentStream
6 import org.apache.chemistry.opencmis.commons.enums.BaseTypeId
7 import org.apache.chemistry.opencmis.commons.enums.VersioningState
8 import org.apache.chemistry.opencmis.commons.exceptions.CmisObjectNotFoundException
9 import org.apache.chemistry.opencmis.commons.impl.dataobjects.ContentStreamImpl
10
11 // NOTE set variables with full path on local file system
12
13 // ex. C:/devel/sources/pilot/releases/1.0.0-alpha7-council/CMIS/resources
14 def resourcesLocalPath = 'C:/DATA/Joinup/LEOS-Pilot_2.0.0/tools/cmis/chemistry-opencmis-server-inmemory/src/main/resources'
15
16 // ex. C:/devel/sources/pilot/releases/1.0.0-alpha7-council/CMIS/scripts/repository
17 def repoPropertiesLocalPath = 'C:/DATA/Joinup/LEOS-Pilot_2.0.0/tools/cmis/chemistry-opencmis-server-inmemory/src/main/webapp/WEB-INF/classes'
18
19 def loader = new LeosLoader(resourcesLocalPath, repoPropertiesLocalPath, session)
20 loader.loadRepository()
  
```

4. Execute the script by clicking on below option given in console









Result:



click on  to navigate back

open workspaces and it contains a sample proposal

up	/leos/workspaces/proposal_1				▼	go
	Name	Type	Content Type	Size		
	annex_1	leos.xml	application/akn+xml	36020		
	media	cmis:folder				
	proposal_1	leos.xml	application/akn+xml	35641		
	memorandum_1	leos.xml	application/akn+xml	43794		
	bill_1	leos.xml	application/akn+xml	41024		
	annex_2	leos.xml	application/akn+xml	36020		

All the proposals created from LEOS application will be created in workspaces folder.

## C. Running LEOS with CMIS Open source version

Now we have CMIS Open source version up and running.

Since we are do not want to connect to **chemistry-opencmis-server-inmemory** , all we need to do it to run below scripts individually present LEOS Joinup release.

Execute below scripts

**run-annotate.bat** (optional)

**run-leos.bat** (required)

**run-user-repository.bat** (required)

Login to LEOS application <http://localhost:8080/leos-pilot/ui>



The screenshot shows the LEOS application interface. At the top, there is a header with the European Commission logo and the LEOS logo. Below the header, the main content area is titled "Repository Browser". On the left side, there is a "FILTERS" panel with checkboxes for "PROCEDURE" (Ordinary legislative procedure), "ACT" (Proposal for a regulation), and "LANGUAGES" (English). The main content area displays a list of proposals. The first proposal is titled "Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on XYZ" and has a language of "English". There is a "Create proposal" button in the top right corner and an "Open" button next to the first proposal.