

# CMIS AIRS Dossier

Digitech

Version 1.0, 04/07/2023 14:31:40

# Sommaire

1. Introduction .....	5
2. Concepts CMIS .....	6
2.1. CMIS repository .....	6
2.2. CMIS query .....	6
2.3. CMIS services .....	6
2.4. CMIS object model .....	8
2.4.1. CMIS folder object .....	8
2.4.2. CMIS document object .....	9
2.4.3. CMIS policy object .....	10
2.4.4. CMIS versioning .....	10
2.4.5. CMIS bindings/protocoles .....	10
3. Browser Binding .....	12
3.1. Paramètres additionnels communs .....	12
3.2. Interrogation d'un repository .....	12
3.3. Récupération d'éléments .....	13
3.4. Opération Création/Mise à jour/Suppression .....	15
3.4.1. Création : .....	15
3.4.2. Suppression : .....	30
3.4.3. Mise à jour des documents : .....	31
3.5. Recherche .....	35
3.6. Récupération du contenu d'un document .....	36
3.7. Bonnes pratiques .....	37
4. Dossier CMIS serveur .....	39
4.1. Arborescence / object model .....	39
4.2. Authentification .....	40
4.3. Limitations .....	40
4.4. Préconisations .....	40
5. Clients CMIS .....	41
5.1. Apache Chemistry / opencmis .....	41
5.2. Dossier CMIS client .....	41
5.3. Dossier CMIS : Cas d'utilisation .....	42
5.3.1. Récupération d'un objet .....	43
5.3.2. Récupération des enfants d'un objet .....	44
5.3.3. Récupération du contenu binaire d'un document .....	44
5.3.4. queries .....	45
5.3.5. Suppression d'un document .....	45
5.3.6. Mise à jour d'un document .....	45
5.3.7. Création d'un commentaire .....	46

5.3.8. Création d'un document AIRS .....	46
5.3.9. Création d'une pièce jointe .....	46
6. Annexes.....	48
6.1. Interface client Dossier .....	48
6.2. librairie <b>dossier-cmis-client</b> .....	57

**Digitech SA**

*ZAC Saumaty Séon - 21 av. Fernand Sardou - CS 40173  
13322 Marseille Cedex 16*

Copyright (c) 2020 Digitech SA, tous droits réservés.

*Table 1. Notes de suivi*

Date	Version	Etat	Objet	Rédacteur	Valideur
01/05/2020	1.0	Validé	Création du document	Nicolas Félix	

# 1. Introduction

**CMIS** (*Content Management Interoperability Services*) est une norme/un standard géré par le consortium **OASIS**, dont le but premier est de permettre une interopérabilité entre des systèmes de gestion de contenu (type *geide* par exemple).

Cette norme permet à des **clients** et des **serveurs** (ou à des **consommateurs** et des **fournisseurs**) de *communiquer, échanger, dialoguer* ensemble via **HTTP** (**REST json** ou **XML AtomPub**).

La dernière version publiée de la norme est **CMIS 1.1**.



Les **webservices** de type **SOAP** ne sont pas disponibles, car obsolètes et moins performants (ils ont été simplement désactivés).

**Dossier** est compatible avec la norme **CMIS** dans sa version **1.1**, permettant à des applications tierces d'accéder au contenu de l'application **Dossier**.

Ce document décrit brièvement :

- le format des APIs **CMIS** REST (leurs appels)
- une description du format des réponses



**Dossier** fournit une implémentation restreinte de la norme **CMIS**, dans le sens où l'intégralité de la norme n'est pas implémentée.

Cependant, celle-ci évolue au fil des demandes et des versions de **Dossier**. Par exemple :

- la version **7.0** de **Dossier** a intégré pour la première fois une implémentation de la norme **CMIS**, partie navigation.
- la version **7.1** de **Dossier** a ajouté les **queries** et la possibilité d'ajouter, modifier ou supprimer des **documents**.

Les chapitres suivants décrivent

- (*succinctement*) la norme **CMIS** et les concepts introduits par celle-ci (2, *Concepts CMIS*)
- de manière un peu plus détaillée le protocole 3, *Browser Binding*
- l'utilisation de clients pour interroger **Dossier** via le protocole **CMIS** (5, *Clients CMIS*)

## 2. Concepts CMIS

La norme CMIS introduit un certain nombre de concepts. Ce chapitre décrit succinctement les principaux.

### 2.1. CMIS repository

Le point d'entrée de la norme CMIS est la notion de repository (qui pourrait être traduit par *référentiel*). Celui-ci représente une instance d'un *système de gestion de contenu*, avec ses *métadonnées*, ses *index*, son *contenu*, ses *fichiers*...

Le repository est l'entité racine vers laquelle toutes les requêtes sont adressées, notamment en appels REST où le repository sert de chemin racine.

Le repository, par simple appel, renvoie sa description et ses capacités.

### 2.2. CMIS query

CMIS query est une syntaxe de requêtage basée sur la norme SQL-92.



Seules les queries de recherche/sélection sont gérées, c'est-à-dire qu'aucune mise à jour ou manipulation de données n'est permise.

Une syntaxe de requête consiste:

- **SELECT**, suivi d'une liste des propriétés recherchées.
- **FROM**, précisant la liste des objets interrogés/recherchés.
- **JOIN**, dans le but d'effectuer une jointure entre différents types d'objets
- **WHERE**, précisant les critères de sélection
- **IN** et **ANY**, afin de requêter des propriétés multi-évaluées
- **CONTAINS**, applicable sur des champs full-text (et donc le contenu des documents)
- **IN\_FOLDER** et **IN\_TREE**, afin de rechercher dans une hiérarchie de documents.
- **ORDERBY**, dans le but de trier les résultats.

Une query, et ses résultats, peut être paginée afin de faciliter la présentation (côté client) et optimiser les performances.

### 2.3. CMIS services

CMIS fournit un ensemble de service accessibles via **AtomPub** ou **Browser**, selon votre choix.

Ces services sont les suivants:

#### Repository service

Permet de découvrir la liste des repositories disponibles, leurs capacités, leur structure et quels types sont disponibles au sein de ceux-ci.

## Navigation service

Permet de naviguer dans un **repository** et son arborescence, en parcourant sa hiérarchie de dossiers parents/enfants.

Généralement, vous pouvez utiliser ces services pour obtenir les enfants ou parents d'un objet.

## Object service

Fournit les opérations basiques de type CRUD (Create, Read, Update, Delete) et les services de contrôle de tout objet (**document**, **folder**, **policy** et **relationship**).

Pour les objets **document**, cela inclut la récupération et la mise à jour des *propriétés*, des *policies* et des fichiers associés (*content stream*).

Ces services retrouvent un (ou des) objet(s) à partir :

- d'un **chemin**
- ou d'un **ID**

Les clients **CMIS** peuvent alors également découvrir les actions disponibles, et autorisées à l'utilisateur connecté.

## Multi-filing service

Ces services vous permettent de constituer une hiérarchie entre **documents** et **folders** en ajoutant ou en supprimant un objet dans ou à partir d'un **folder**.

## Discovery service

Fournit les services de requêtage (**query**) et de modifications.

## Change service

Service de découverte de modification de contenu depuis la dernière vérification.

## Versioning service

Fournit un service d'accès à l'historisation des versions d'un objet, si celui-ci est versionné.

## Relationship service

Permet de créer, de gérer et d'accéder à des relations ou des associations entre des objets.

## Policy services

Permet d'appliquer des **policies** à un objet de type **document**.

Les **policies** sont des entités notamment utilisées pour gérer l'accès sécurité et accès aux **documents**.

## ACL service

Permet de gérer l'accès et les contrôles (**Access Control Lists**) permettant alors d'effectuer certaines opérations sur un objet/une entité.

## 2.4. CMIS object model

**Dossier** fournit un modèle d'objet **CMIS** permettant d'abstraire le contenu de **Dossier**.

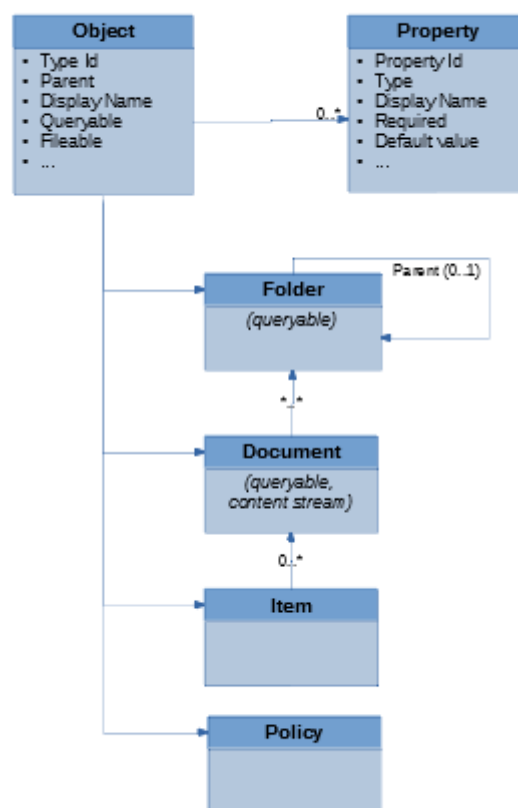
Il implémente la navigation entre **folders** et **documents**.

**CMIS** permet la définition de *types* d'objets, chacun ayant ses propriétés, par exemple, chaque objet possède un **type**, un **ID**, ...

Les *types* d'objet supportent l'héritage (hiérarchie entre objets).

Les *types* **document** permettent d'accéder à du contenu binaire (généralement un fichier joint) si disponible, nommé **content stream** dans le jargon **CMIS**.

1.



CMIS data model

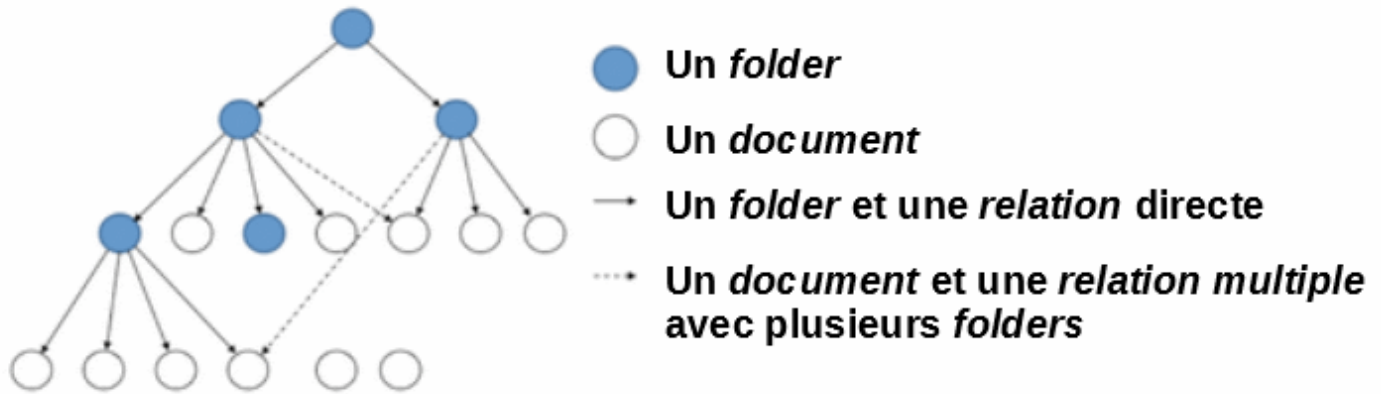
### 2.4.1. CMIS folder object

Tout **document** est rattaché à un **folder** (ou plus exactement à une hiérarchie de **folders**).

Un **document** peut cependant appartenir à plusieurs **folders**.

CMIS folder/document relationships





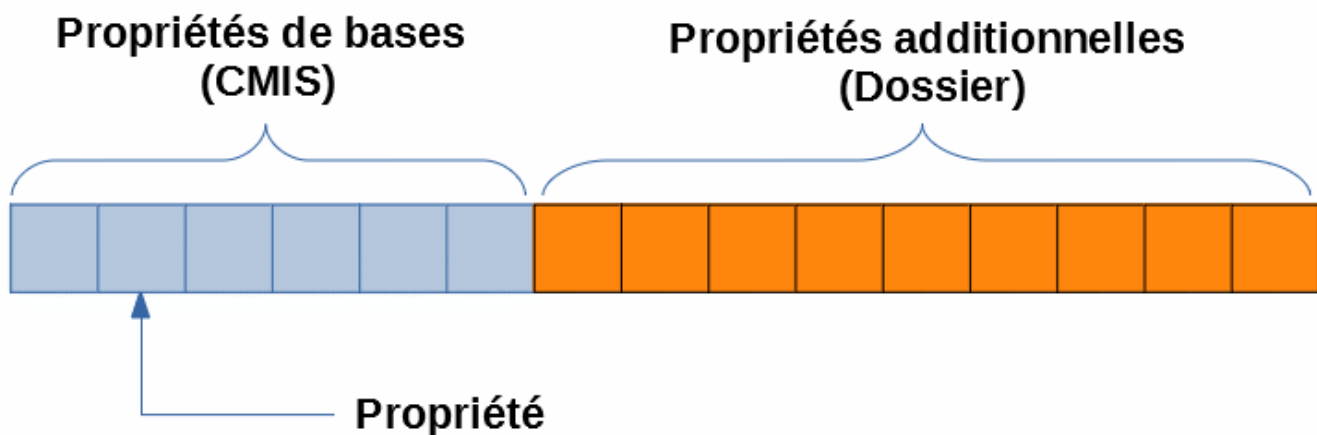
### 2.4.2. CMIS document object

Les **documents** ont des propriétés - qui peuvent être multi-évaluées - et (optionnellement) un contenu binaire (**content stream**) - qui peut être versionné.

Chaque propriété :

- est **typée** : *Object ID, string, integer, decimal, boolean, float, datetime.*
- et possède des **attributs** : *nom, required, queryable, type ...*

CMIS document object



Un **document** peut également posséder des **renditions**, représentant une version alternative du **content stream**.

Les exemples les plus courants sont :

- *thumbnail*: version *image* (aperçu) d'un **content stream** du **document**, utilisée par les applications mobiles.
- *application/pdf*: version *PDF* d'un **content stream** du **document**, utilisée par tous clients ne souhaitant pas effectuer de conversion de fichiers.



**Dossier** ne fournit pas de **rendition** alternative.

### 2.4.3. CMIS policy object



Ce type d'objet n'est actuellement pas géré par Dossier

Un objet de type **policy** permet de représenter la stratégie/la politique d'accès à un document.

Le type **ACL** (Access Control List) est un sous-type de **policy**.

### 2.4.4. CMIS versioning



Dossier ne gère le versionning que sur les fichiers binaires associés à un document.  
Ceci n'est pas géré au travers de la norme CMIS

### 2.4.5. CMIS bindings/protocoles

Tout client peut communiquer avec un **repository CMIS** en utilisant l'un des protocoles disponibles:

- **Browser** (**recommandé**) :

CMIS fournit un accès de type **REST** basé sur des échanges au format **JSON**.

Ce *binding* a été ajouté en version **1.1** de la norme **CMIS** afin de simplifier les échanges avec les applications *web*, *mobiles*, ce type d'échanges étant populaires et très répandus depuis plusieurs années.

Cette implémentation utilise les seuls verbes **GET** et **POST**.

Les URLs d'accès et d'interrogation sont simples, par exemple, dans le but d'obtenir la liste des **repositories**.

```
http(s)://<hostname>:<port>/<context>/cmis/browser
```

où :

- **hostname**: serveur (IP ou Nom d'hôte) hébergeant l'application Dossier
- **port**: port du service d'accès à l'application Dossier
- **context**: context applicatif

Vous trouverez de plus amples informations dans le chapitre dédié 3, *Browser Binding*

- **AtomPub**

Ce protocole **RESTful** est basé sur **Atom Publishing**.

Les clients qui souhaitent exploiter ce protocole utiliseront l'url du **Service Document**

```
http(s)://<hostname>:<port>/<context>/cmis/atom
```

Pour une description complète de ces protocoles, veuillez vous reporter aux [spécifications officielles CMIS](#).



Les **Web Services** de type **SOAP** ne sont pas disponibles car obsolètes et moins performants.

## 3. Browser Binding



Les informations indiquées dans ce chapitre sont succinctes, veuillez vous référer à la [documentation officielle](#) pour plus d'informations+ Il est préconisé d'employer un client, disponible par exemple [ici](#) pour abstraire et masquer la complexité des échanges.



En utilisant un simple appel **HTTP** de type **GET** et quelques paramètres additionnels, le contenu d'un [repository](#) peut être récupéré/interrogé.

### 3.1. Paramètres additionnels communs

Les requêtes d'interrogation peuvent avoir des paramètres additionnels (optionnels)

Table 2. Liste des paramètres http optionnels

Nom	Valeur par défaut	Description
filter	*	Liste des propriétés attendues en réponse ( <b>queryname</b> au sens <a href="#">CMIS</a> ), séparées par une virgule
includeAllowableActions	false	Si <b>true</b> , la réponse doit indiquer également la liste des actions disponibles
includeRelationships	IncludeRelationships.NONE	La réponse doit indiquer également les relations avec ce noeud. valeurs possibles: <ul style="list-style-type: none"><li>• NONE</li><li>• SOURCE</li><li>• TARGET</li><li>• BOTH</li></ul>
renditionFilter	cmis:none	Liste des <i>renditions</i> attendues en réponse
includePolicyIds	false	Si <b>true</b> , la réponse doit indiquer également la liste des <i>polices</i> disponibles
includeAcl	false	Si <b>true</b> , la réponse doit indiquer également les <b>ACL</b> ( <b>A</b> ccess <b>C</b> ontrol <b>L</b> ists) pour ce noeud

### 3.2. Interrogation d'un [repository](#)

Afin d'obtenir des informations sur le [repository](#), le paramètre **callback** devra être précisé.

```
http(s)://<hostname>:<port>/<context>/cmis/browser?callback=showRepositoryInfo
```

dont la réponse sera similaire à ce qui suit :

```
showRepositoryInfo(
{ "Dossier-2":
  {"repositoryId":"Dossier-2",
   "repositoryName":"Dossier-2AC",
   "repositoryDescription":"2AC",
   "vendorName":"Digitech",
   "productName":"Dossier",
  ...
}
```



l'attribut **repositoryId** est important car permet d'accéder aux données du référentiel ciblé.

### 3.3. Récupération d'éléments



Dans les exemples ci-après, **Dossier-2** correspond au **repositoryId** ciblé.

La récupération d'un objet s'effectue simplement en spécifiant l' **ID** de l'objet souhaité.

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2/root?objectId=ZmxvdzpDUg%3D%3D
```

*Réponse (mode succinct)*

```
{
  "succinctProperties": {
    "cmis:objectId": "ZmxvdzpDUg==",
    "cmis:objectTypeId": "dossier:flow",
    "cmis:baseTypeId": "cmis:folder",
    "cmis:name": "Compte Rendu",
    "cmis:description": "Flow created in simplified mode",
    "cmis:createdBy": "dossier",
    "cmis:creationDate": null,
    "cmis:lastModifiedBy": "dossier",
    "cmis:lastModificationDate": null,
    "cmis:changeToken": null,
    "cmis:parentId": "QGRvc3NpZXJfcm9vdEA=",
    "cmis:path": "\\flow:CR",
    "cmis:allowedChildObjectIds": [
      "cm:basket"
    ],
    "dossier:flow:code": "CR",
    "dossier:flow:contentType": "CR"
  },
  "allowableActions": {
    "canDeleteObject": false,
    "canUpdateProperties": false,
    "canGetFolderTree": true,
    "canGetProperties": true,
  }
}
```

```

"canGetObjectRelationships": false,
"canGetObjectParents": true,
"canGetFolderParent": true,
"canGetDescendants": true,
"canMoveObject": false,
"canDeleteContentStream": false,
"canCheckOut": false,
"canCancelCheckOut": false,
"canCheckIn": false,
"canSetContentStream": false,
"canGetAllVersions": false,
"canAddObjectToFolder": false,
"canRemoveObjectFromFolder": false,
"canGetContentStream": false,
"canApplyPolicy": false,
"canGetAppliedPolicies": false,
"canRemovePolicy": false,
"canGetChildren": true,
"canCreateDocument": false,
"canCreateFolder": false,
"canCreateRelationship": false,
"canCreateItem": false,
"canDeleteTree": false,
"canGetRenditions": false,
"canGetACL": true,
"canApplyACL": false
},
"acl": {
  "aces": []
},
"exactACL": true
}

```

Dans le but de récupérer les **enfants** d'une entité, il faudra préciser ici le paramètre suivant :

```
cmisselector=children
```

Ce qui donne, pour obtenir les **enfants** de l'objet d'ID **ZmxvdzpDUg==** dans le **repository Dossier-2**

```

http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-
2/root?objectId=ZmxvdzpDUg%3D%3D&cmisselector=children&succinct=true

```

La réponse obtenue sera alors un *json*

```

{
  "objects": [
    {

```

```

"object": {
  "succinctProperties": {
    "cmis:objectId": "YmFza2V0kNSI0Rlcm5pZXJzIGNvbXB0ZSBYZW5kdXM=",
    "cmis:objectTypeId": "cm:basket",
    "cmis:baseTypeId": "cmis:folder",
    "cmis:name": "Derniers compte rendus",
    "cmis:createdBy": "dossier",
    "cmis:creationDate": null,
    "cmis:lastModifiedBy": "dossier",
    "cmis:lastModificationDate": null
  }
}
],
"hasMoreItems": false
}

```

D'autres paramètres (que ceux indiqués dans le chapitre [Section 3.1, "Paramètres additionnels communs"](#)) sont disponibles

Table 3. paramètres additionnels

Nom	Valeur par défaut	Description
orderBy	*	Liste des <b>queryname</b> (suivis de <b>ASC</b> ou <b>DESC</b> ), séparées par une virgule, définissant le tri souhaité.
includePathSegments	false	Si <b>true</b> , la réponse contiendra pour chaque enfant le <i>path</i> relatif.
maxItems		Nombre maximum d'éléments retournés
SkipCount		Nombre d'éléments <i>passés</i> (et donc non renvoyés en réponse). A utiliser en combinaison avec <b>maxItems</b>

## 3.4. Opération Création/Mise à jour/Suppression

Les opérations de création, de mise à jour ou de suppression sont effectuées via une commande **HTTP** de type **POST**.

Le paramètre additionnel précisant le type d'opération est **cmisaction**, par exemple, pour une création

```
cmisaction=createDocument
```

### 3.4.1. Création :

L'application fournit 3 opérations de création en fonction du type d'objet que l'on veut créer : **createItem**, **createFolder** et **createDocument**.

Il existe 3 types d'objets qui sont créables via CMIS dans AIRS Dossier :

- Les documents AIRS (**createFolder**)
- Les pièces jointes (**createDocument**)
- Les commentaires (**createItem**)

### Création d'un commentaire (**createItem**)

La création d'un commentaire nécessite les paramètres suivants :

```
cmisaction=createItem
objectId=<IDENTIFIANT DE L'OBJET PARENT>
propertyId[0]=cmis:name
propertyValue[0]=<TEXTE DU COMMENTAIRE>
propertyId[1]=cmis:objectTypeId
propertyValue[1]=cm:comment
propertyId[1]=cmis:baseTypeId
propertyValue[1]=cmis:item
...
```

Voici un exemple :

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-
5/root?cmisaction=createItem&propertyId[0]=cmis:name&propertyValue[0]=TEST
COMMENT&propertyId[1]=cmis:objectTypeId&propertyValue[1]=cm:comment&propertyId[2]=cmis:baseTy
peId&propertyValue[2]=cmis:item&objectId=ZG9jdW1lbnQ6Q1I6RGVybml1cnMgY29tcHRlIHJlbnR1czo3NDc2
```

La réponse obtenue sera alors un *json* reprenant l'ensemble des propriétés du document mis à jour.

```
{
  "properties": {
    "cmis:objectId": {
      "id": "cmis:objectId",
      "localName": "cmis:objectId",
      "queryName": "cmis:objectId",
      "type": "id",
      "cardinality": "single",
      "value": "ZG9jdW1lbnQ6Q1I6RGVybml1cnMgY29tcHRlIHJlbnR1czo3NDc2"
    },
    "cmis:objectTypeId": {
      "id": "cmis:objectTypeId",
      "localName": "cmis:objectTypeId",
      "queryName": "cmis:objectTypeId",
      "type": "id",
      "cardinality": "single",
      "value": "cm:document:cr"
    },
    "cmis:baseTypeId": {
      "id": "cmis:baseTypeId",
      "localName": "cmis:baseTypeId",

```



```

    "queryName": "cmis:baseTypeId",
    "type": "id",
    "cardinality": "single",
    "value": "cmis:folder"
  },
  "cmis:name": {
    "id": "cmis:name",
    "localName": "cmis:name",
    "queryName": "cmis:name",
    "type": "string",
    "cardinality": "single",
    "value": "CHANGEMENT_DE_VALEUR"
  },
  "cmis:description": {
    "id": "cmis:description",
    "localName": "cmis:description",
    "queryName": "cmis:description",
    "type": "string",
    "cardinality": "single",
    "value": "CHANGEMENT_DE_VALEUR"
  },
  "cmis:createdBy": {
    "id": "cmis:createdBy",
    "localName": "cmis:createdBy",
    "queryName": "cmis:createdBy",
    "type": "string",
    "cardinality": "single",
    "value": "dossier"
  },
  "cmis:creationDate": {
    "id": "cmis:creationDate",
    "localName": "cmis:creationDate",
    "queryName": "cmis:creationDate",
    "type": "datetime",
    "cardinality": "single",
    "value": 1670248188000
  },
  "cmis:lastModifiedBy": {
    "id": "cmis:lastModifiedBy",
    "localName": "cmis:lastModifiedBy",
    "queryName": "cmis:lastModifiedBy",
    "type": "string",
    "cardinality": "single",
    "value": "dossier"
  },
  "cmis:lastModificationDate": {
    "id": "cmis:lastModificationDate",
    "localName": "cmis:lastModificationDate",
    "queryName": "cmis:lastModificationDate",
    "type": "datetime",
    "cardinality": "single",

```

```

    "value": 1674660862000
  },
  "cmis:changeToken": {
    "id": "cmis:changeToken",
    "localName": "cmis:changeToken",
    "queryName": "cmis:changeToken",
    "type": "string",
    "cardinality": "single",
    "value": "MjUvMDEvMjAyMyAxNjozNDoyMg=="
  },
  "cmis:parentId": {
    "id": "cmis:parentId",
    "localName": "cmis:parentId",
    "queryName": "cmis:parentId",
    "type": "id",
    "cardinality": "single",
    "value": "YmFza2V0OksNS0kRlcm5pZXJzIGNvbXB0ZSB5ZW5kdXM="
  },
  "cmis:path": {
    "id": "cmis:path",
    "localName": "cmis:path",
    "queryName": "cmis:path",
    "type": "string",
    "cardinality": "single",
    "value": "/orga:5/flow:CR/basket:CR:Derniers compte rendus/document:CR:Derniers compte rendus:7476"
  },
  "cmis:allowedChildObjectTypeIds": {
    "id": "cmis:allowedChildObjectTypeIds",
    "localName": "cmis:allowedChildObjectTypeIds",
    "queryName": "cmis:allowedChildObjectTypeIds",
    "type": "id",
    "cardinality": "multi",
    "value": [
      "cm:attachment",
      "cm:comment",
      "cm:event"
    ]
  },
  "cm:document:cr:crTheme": {
    "id": "cm:document:cr:crTheme",
    "localName": "cm:document:cr:crTheme",
    "displayName": "Thématique",
    "queryName": "cm:document:cr:crTheme",
    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cm:document:cr:dModif": {
    "id": "cm:document:cr:dModif",
    "localName": "cm:document:cr:dModif",

```

```

    "displayName": "Date de modification",
    "queryName": "cm:document:cr:dModif",
    "type": "datetime",
    "cardinality": "single",
    "value": 1674660862000
  },
  "cm:document:cr:dCreat": {
    "id": "cm:document:cr:dCreat",
    "localName": "cm:document:cr:dCreat",
    "displayName": "Date de création",
    "queryName": "cm:document:cr:dCreat",
    "type": "datetime",
    "cardinality": "single",
    "value": 1670248188000
  },
  "cm:document:cr:crRedacteur": {
    "id": "cm:document:cr:crRedacteur",
    "localName": "cm:document:cr:crRedacteur",
    "displayName": "Rédacteur",
    "queryName": "cm:document:cr:crRedacteur",
    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cm:document:cr:crDes": {
    "id": "cm:document:cr:crDes",
    "localName": "cm:document:cr:crDes",
    "displayName": "Désignation",
    "queryName": "cm:document:cr:crDes",
    "type": "string",
    "cardinality": "single",
    "value": "CHANGEMENT_DE_VALEUR"
  },
  "cm:document:cr:crResume": {
    "id": "cm:document:cr:crResume",
    "localName": "cm:document:cr:crResume",
    "displayName": "Résumé",
    "queryName": "cm:document:cr:crResume",
    "type": "string",
    "cardinality": "single",
    "value": "test creat"
  },
  "cm:document:cr:crDate": {
    "id": "cm:document:cr:crDate",
    "localName": "cm:document:cr:crDate",
    "displayName": "Date événement",
    "queryName": "cm:document:cr:crDate",
    "type": "datetime",
    "cardinality": "single",
    "value": null
  },

```

```

"cm:document:cr:tPrior": {
  "id": "cm:document:cr:tPrior",
  "localName": "cm:document:cr:tPrior",
  "displayName": "xml_priority_field_label",
  "queryName": "cm:document:cr:tPrior",
  "type": "string",
  "cardinality": "single",
  "value": null
},
"cm:document:cr:multi": {
  "id": "cm:document:cr:multi",
  "localName": "cm:document:cr:multi",
  "displayName": "test champs multi",
  "queryName": "cm:document:cr:multi",
  "type": "string",
  "cardinality": "single",
  "value": ""
},
"cm:document:cr:corres": {
  "id": "cm:document:cr:corres",
  "localName": "cm:document:cr:corres",
  "displayName": "correspondent",
  "queryName": "cm:document:cr:corres",
  "type": "string",
  "cardinality": "single",
  "value": null
},
"cm:document:cr:masqueNum": {
  "id": "cm:document:cr:masqueNum",
  "localName": "cm:document:cr:masqueNum",
  "displayName": "test masque",
  "queryName": "cm:document:cr:masqueNum",
  "type": "integer",
  "cardinality": "single",
  "value": null
}
}
}

```

### Création d'un document (**createFolder**)

La création d'un document (AIRS) nécessite les paramètres suivants :

```

cmisaction=createFolder
objectId=<IDENTIFIANT DE L'OBJET PARENT>
propertyId[0]=cmis:objectTypeId
propertyValue[0]=<CODE CMIS DU TYPE DE CONTENU>
propertyId[1]=cmis:baseTypeId
propertyValue[1]=cmis:folder
propertyId[2]=<IDENTIFIANT DE LA PROPRIETE 2>
propertyValue[2]=<VALEUR DE LA PROPRIETE 2>

```

...



Le paramètre **objectId** doit contenir l'identifiant CMIS du flux Airs Dossier dans lequel le document doit être créé.

Voici un exemple :

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-
5/root?cmisaction=createFolder&propertyId[0]=cm:document:cr:crDes&propertyValue[0]=TEST
CREATE
FOLDER&propertyId[1]=cmis:objectTypeId&propertyValue[1]=cm:document:cr&propertyId[2]=cmis:bas
eTypeId&propertyValue[2]=cmis:folder&objectId=ZmxvdzpdUg==
```

La réponse obtenue sera alors un *json* reprenant l'ensemble des propriétés du document AIRS créé.

```
{
  "properties": {
    "cmis:objectId": {
      "id": "cmis:objectId",
      "localName": "cmis:objectId",
      "queryName": "cmis:objectId",
      "type": "id",
      "cardinality": "single",
      "value": "ZG9jdW1lbnQ6NzY3Nzo3Njc3"
    },
    "cmis:objectTypeId": {
      "id": "cmis:objectTypeId",
      "localName": "cmis:objectTypeId",
      "queryName": "cmis:objectTypeId",
      "type": "id",
      "cardinality": "single",
      "value": "cm:document:cr"
    },
    "cmis:baseTypeId": {
      "id": "cmis:baseTypeId",
      "localName": "cmis:baseTypeId",
      "queryName": "cmis:baseTypeId",
      "type": "id",
      "cardinality": "single",
      "value": "cmis:folder"
    },
    "cmis:name": {
      "id": "cmis:name",
      "localName": "cmis:name",
      "queryName": "cmis:name",
      "type": "string",
      "cardinality": "single",
      "value": "Aucune valeur"
    }
  }
}
```

```

"cmis:description": {
  "id": "cmis:description",
  "localName": "cmis:description",
  "queryName": "cmis:description",
  "type": "string",
  "cardinality": "single",
  "value": "Aucune valeur"
},
"cmis:createdBy": {
  "id": "cmis:createdBy",
  "localName": "cmis:createdBy",
  "queryName": "cmis:createdBy",
  "type": "string",
  "cardinality": "single",
  "value": "dossier"
},
"cmis:creationDate": {
  "id": "cmis:creationDate",
  "localName": "cmis:creationDate",
  "queryName": "cmis:creationDate",
  "type": "datetime",
  "cardinality": "single",
  "value": 1679649084000
},
"cmis:lastModifiedBy": {
  "id": "cmis:lastModifiedBy",
  "localName": "cmis:lastModifiedBy",
  "queryName": "cmis:lastModifiedBy",
  "type": "string",
  "cardinality": "single",
  "value": "dossier"
},
"cmis:lastModificationDate": {
  "id": "cmis:lastModificationDate",
  "localName": "cmis:lastModificationDate",
  "queryName": "cmis:lastModificationDate",
  "type": "datetime",
  "cardinality": "single",
  "value": 1679649084000
},
"cmis:changeToken": {
  "id": "cmis:changeToken",
  "localName": "cmis:changeToken",
  "queryName": "cmis:changeToken",
  "type": "string",
  "cardinality": "single",
  "value": "MjQvMDMvMjAyMyAxMDoxMTToyNA=="
},
"cmis:parentId": {
  "id": "cmis:parentId",
  "localName": "cmis:parentId",

```

```

    "queryName": "cmis:parentId",
    "type": "id",
    "cardinality": "single",
    "value": "YmFza2V00jc2Nzc="
  },
  "cmis:path": {
    "id": "cmis:path",
    "localName": "cmis:path",
    "queryName": "cmis:path",
    "type": "string",
    "cardinality": "single",
    "value": "/orga:5/flow:7677/basket:7677/document:7677:7677"
  },
  "cmis:allowedChildObjectIds": {
    "id": "cmis:allowedChildObjectIds",
    "localName": "cmis:allowedChildObjectIds",
    "queryName": "cmis:allowedChildObjectIds",
    "type": "id",
    "cardinality": "multi",
    "value": [
      "cm:attachment",
      "cm:comment",
      "cm:event"
    ]
  },
  "cm:document:cr:crTheme": {
    "id": "cm:document:cr:crTheme",
    "localName": "cm:document:cr:crTheme",
    "displayName": "Thématique",
    "queryName": "cm:document:cr:crTheme",
    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cm:document:cr:dModif": {
    "id": "cm:document:cr:dModif",
    "localName": "cm:document:cr:dModif",
    "displayName": "Date de modification",
    "queryName": "cm:document:cr:dModif",
    "type": "datetime",
    "cardinality": "single",
    "value": 1679649084000
  },
  "cm:document:cr:dCreat": {
    "id": "cm:document:cr:dCreat",
    "localName": "cm:document:cr:dCreat",
    "displayName": "Date de création",
    "queryName": "cm:document:cr:dCreat",
    "type": "datetime",
    "cardinality": "single",
    "value": 1679649084000
  }

```

```

},
"cm:document:cr:crRedacteur": {
  "id": "cm:document:cr:crRedacteur",
  "localName": "cm:document:cr:crRedacteur",
  "displayName": "Rédacteur",
  "queryName": "cm:document:cr:crRedacteur",
  "type": "string",
  "cardinality": "single",
  "value": null
},
"cm:document:cr:crDes": {
  "id": "cm:document:cr:crDes",
  "localName": "cm:document:cr:crDes",
  "displayName": "Désignation",
  "queryName": "cm:document:cr:crDes",
  "type": "string",
  "cardinality": "single",
  "value": "TEST CREATE FOLDER"
},
"cm:document:cr:crResume": {
  "id": "cm:document:cr:crResume",
  "localName": "cm:document:cr:crResume",
  "displayName": "Résumé",
  "queryName": "cm:document:cr:crResume",
  "type": "string",
  "cardinality": "single",
  "value": null
},
"cm:document:cr:crDate": {
  "id": "cm:document:cr:crDate",
  "localName": "cm:document:cr:crDate",
  "displayName": "Date événement",
  "queryName": "cm:document:cr:crDate",
  "type": "datetime",
  "cardinality": "single",
  "value": null
},
"cm:document:cr:tPrior": {
  "id": "cm:document:cr:tPrior",
  "localName": "cm:document:cr:tPrior",
  "displayName": "xml_priority_field_label",
  "queryName": "cm:document:cr:tPrior",
  "type": "string",
  "cardinality": "single",
  "value": null
},
"cm:document:cr:multi": {
  "id": "cm:document:cr:multi",
  "localName": "cm:document:cr:multi",
  "displayName": "test champs multi",
  "queryName": "cm:document:cr:multi",

```



```

    "type": "string",
    "cardinality": "single",
    "value": ""
  },
  "cm:document:cr:corres": {
    "id": "cm:document:cr:corres",
    "localName": "cm:document:cr:corres",
    "displayName": "correspondent",
    "queryName": "cm:document:cr:corres",
    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cm:document:cr:masqueNum": {
    "id": "cm:document:cr:masqueNum",
    "localName": "cm:document:cr:masqueNum",
    "displayName": "test masque",
    "queryName": "cm:document:cr:masqueNum",
    "type": "integer",
    "cardinality": "single",
    "value": null
  }
}
}

```

### Création d'une pièce jointe (**createAttachment**)

La création d'une pièce jointe nécessite les paramètres suivants :

```

cmisaction=createDocument
objectId=<IDENTIFIANT DE L'OBJET PARENT>
propertyId[0]=cmis:objectTypeId
propertyValue[0]=cm:attachment
propertyId[1]=cmis:baseTypeId
propertyValue[1]=cmis:document
propertyId[2]=cmis:contentStreamFileName
propertyValue[2]=<NOM DU FICHIER>
propertyId[2]=cmis:name
propertyValue[2]=<LABEL DE L'ATTACHMENT>
...

```



Le content stream doit être ajouté via une enveloppe multipart.

Voici un exemple :

```

POST http://localhost:8080/dossierweb80/cmis/browser/Dossier-
5/root?cmisaction=createDocument&propertyId[0]=cmis:contentStreamFileName&propertyValue[0]=te
st.pdf&propertyId[1]=cmis:objectTypeId&propertyValue[1]=cm:attachment&propertyId[2]=cmis:base
TypeId&propertyValue[2]=cmis:document&objectId=ZG9jdW1lbnQ6Q1I6RGVybml1cnMgY29tcHRlIHJlbmR1cz

```

```
o3NjQy&propertyId[3]=cmis:name&propertyValue[3]=test piece jointe
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: multipart/form-data; boundary="----=_Part_4_495007518.1679652147738"
MIME-Version: 1.0
Content-Length: 922418
```

La réponse obtenue sera alors un *json* reprenant l'ensemble des propriétés de la pièce jointe créée.

```
{
  "properties": {
    "cmis:objectTypeId": {
      "id": "cmis:objectTypeId",
      "localName": "cmis:objectTypeId",
      "queryName": "cmis:objectTypeId",
      "type": "id",
      "cardinality": "single",
      "value": "cm:attachment"
    },
    "cmis:baseTypeId": {
      "id": "cmis:baseTypeId",
      "localName": "cmis:baseTypeId",
      "queryName": "cmis:baseTypeId",
      "type": "id",
      "cardinality": "single",
      "value": "cmis:document"
    },
    "cmis:objectId": {
      "id": "cmis:objectId",
      "localName": "cmis:objectId",
      "queryName": "cmis:objectId",
      "type": "id",
      "cardinality": "single",
      "value": "YXR0YWNobWVudDpDUjpEZXJuaWVycyBjb21wdGUgcmVuZHVzOjc2NDI6NzAxNA=="
    },
    "cmis:secondaryObjectIds": {
      "id": "cmis:secondaryObjectIds",
      "localName": "cmis:secondaryObjectIds",
      "queryName": "cmis:secondaryObjectIds",
      "type": "id",
      "cardinality": "multi",
      "value": null
    },
    "cmis:name": {
      "id": "cmis:name",
      "localName": "cmis:name",
      "queryName": "cmis:name",
      "type": "string",
      "cardinality": "single",
      "value": "Charte Informatique DIGITECH_20220214.pdf"
    }
  }
}
```

```

"cmis:description": {
  "id": "cmis:description",
  "localName": "cmis:description",
  "queryName": "cmis:description",
  "type": "string",
  "cardinality": "single",
  "value": null
},
"cmis:createdBy": {
  "id": "cmis:createdBy",
  "localName": "cmis:createdBy",
  "queryName": "cmis:createdBy",
  "type": "string",
  "cardinality": "single",
  "value": "Felix Nicolas"
},
"cmis:creationDate": {
  "id": "cmis:creationDate",
  "localName": "cmis:creationDate",
  "queryName": "cmis:creationDate",
  "type": "datetime",
  "cardinality": "single",
  "value": 1671876056000
},
"cmis:lastModifiedBy": {
  "id": "cmis:lastModifiedBy",
  "localName": "cmis:lastModifiedBy",
  "queryName": "cmis:lastModifiedBy",
  "type": "string",
  "cardinality": "single",
  "value": "Felix Nicolas"
},
"cmis:lastModificationDate": {
  "id": "cmis:lastModificationDate",
  "localName": "cmis:lastModificationDate",
  "queryName": "cmis:lastModificationDate",
  "type": "datetime",
  "cardinality": "single",
  "value": 1671876056000
},
"cmis:changeToken": {
  "id": "cmis:changeToken",
  "localName": "cmis:changeToken",
  "queryName": "cmis:changeToken",
  "type": "string",
  "cardinality": "single",
  "value": "YXR0YWNobWVudDpDUjpEZXJuaWVycyBjb21wdGUgcmlVZHVzOjc2NDI6NzAxNA=="
},
"cmis:isImmutable": {
  "id": "cmis:isImmutable",
  "localName": "cmis:isImmutable",

```

```

    "queryName": "cmis:isImmutable",
    "type": "boolean",
    "cardinality": "single",
    "value": true
  },
  "cmis:isLatestVersion": {
    "id": "cmis:isLatestVersion",
    "localName": "cmis:isLatestVersion",
    "queryName": "cmis:isLatestVersion",
    "type": "boolean",
    "cardinality": "single",
    "value": true
  },
  "cmis:isMajorVersion": {
    "id": "cmis:isMajorVersion",
    "localName": "cmis:isMajorVersion",
    "queryName": "cmis:isMajorVersion",
    "type": "boolean",
    "cardinality": "single",
    "value": true
  },
  "cmis:isLatestMajorVersion": {
    "id": "cmis:isLatestMajorVersion",
    "localName": "cmis:isLatestMajorVersion",
    "queryName": "cmis:isLatestMajorVersion",
    "type": "boolean",
    "cardinality": "single",
    "value": true
  },
  "cmis:isPrivateWorkingCopy": {
    "id": "cmis:isPrivateWorkingCopy",
    "localName": "cmis:isPrivateWorkingCopy",
    "queryName": "cmis:isPrivateWorkingCopy",
    "type": "boolean",
    "cardinality": "single",
    "value": true
  },
  "cmis:versionLabel": {
    "id": "cmis:versionLabel",
    "localName": "cmis:versionLabel",
    "queryName": "cmis:versionLabel",
    "type": "string",
    "cardinality": "single",
    "value": "Charte Informatique DIGITECH_20220214.pdf"
  },
  "cmis:versionSeriesId": {
    "id": "cmis:versionSeriesId",
    "localName": "cmis:versionSeriesId",
    "queryName": "cmis:versionSeriesId",
    "type": "id",
    "cardinality": "single",

```

```

    "value": "attachment:CR:Derniers compte rendus:7642:7014"
  },
  "cmis:isVersionSeriesCheckedOut": {
    "id": "cmis:isVersionSeriesCheckedOut",
    "localName": "cmis:isVersionSeriesCheckedOut",
    "queryName": "cmis:isVersionSeriesCheckedOut",
    "type": "boolean",
    "cardinality": "single",
    "value": false
  },
  "cmis:versionSeriesCheckedOutBy": {
    "id": "cmis:versionSeriesCheckedOutBy",
    "localName": "cmis:versionSeriesCheckedOutBy",
    "queryName": "cmis:versionSeriesCheckedOutBy",
    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cmis:versionSeriesCheckedOutId": {
    "id": "cmis:versionSeriesCheckedOutId",
    "localName": "cmis:versionSeriesCheckedOutId",
    "queryName": "cmis:versionSeriesCheckedOutId",
    "type": "id",
    "cardinality": "single",
    "value": null
  },
  "cmis:checkinComment": {
    "id": "cmis:checkinComment",
    "localName": "cmis:checkinComment",
    "queryName": "cmis:checkinComment",
    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cmis:contentStreamLength": {
    "id": "cmis:contentStreamLength",
    "localName": "cmis:contentStreamLength",
    "queryName": "cmis:contentStreamLength",
    "type": "integer",
    "cardinality": "single",
    "value": 0
  },
  "cmis:contentStreamMimeType": {
    "id": "cmis:contentStreamMimeType",
    "localName": "cmis:contentStreamMimeType",
    "queryName": "cmis:contentStreamMimeType",
    "type": "string",
    "cardinality": "single",
    "value": "application/pdf"
  },
  "cmis:contentStreamFileName": {

```

### 3.4.2. Suppression :

*Il existe 3 types d'objets qui sont supprimables via CMIS dans AIRS Dossier :*

- 

La suppression nécessite les paramètres suivants :

Quelques exemples :

## Suppression d'un document

### Suppression d'une pièce jointe

### 3.4. Opération Création/Mise à jour/Suppression

4Nw==

### 3.4.3. Mise à jour des documents :

La mise à jour des documents nécessite les paramètres suivants :

```
cmisaction=update
objectId=<IDENTIFIANT DE L'OBJET>
propertyId[0]=<IDENTIFIANT DE LA PROPRIETE 0>
propertyValue[0]=<VALEUR DE LA PROPRIETE 0>
propertyId[1]=<IDENTIFIANT DE LA PROPRIETE 1>
propertyValue[1]=<VALEUR DE LA PROPRIETE 1>
...
```

Voici un exemple :

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-
5/root?cmisaction=update&objectId=ZG9jdW1lbnQ6Q1I6RGVybml1cnMgY29tcHRlIHJlbmR1czo3NDc2&proper
tyId[0]=cm:document:cr:crDes&propertyValue[0]=CHANGEMENT_DE_VALEUR
```

La réponse obtenue sera alors un *json* reprenant l'ensemble des propriétés du document mis à jour.

```
{
  "properties": {
    "cmis:objectId": {
      "id": "cmis:objectId",
      "localName": "cmis:objectId",
      "queryName": "cmis:objectId",
      "type": "id",
      "cardinality": "single",
      "value": "ZG9jdW1lbnQ6Q1I6RGVybml1cnMgY29tcHRlIHJlbmR1czo3NDc2"
    },
    "cmis:objectTypeId": {
      "id": "cmis:objectTypeId",
      "localName": "cmis:objectTypeId",
      "queryName": "cmis:objectTypeId",
      "type": "id",
      "cardinality": "single",
      "value": "cm:document:cr"
    },
    "cmis:baseTypeId": {
      "id": "cmis:baseTypeId",
      "localName": "cmis:baseTypeId",
      "queryName": "cmis:baseTypeId",
      "type": "id",
      "cardinality": "single",
      "value": "cm:folder"
    }
  }
}
```

```

"cmis:name": {
  "id": "cmis:name",
  "localName": "cmis:name",
  "queryName": "cmis:name",
  "type": "string",
  "cardinality": "single",
  "value": "CHANGEMENT_DE_VALEUR"
},
"cmis:description": {
  "id": "cmis:description",
  "localName": "cmis:description",
  "queryName": "cmis:description",
  "type": "string",
  "cardinality": "single",
  "value": "CHANGEMENT_DE_VALEUR"
},
"cmis:createdBy": {
  "id": "cmis:createdBy",
  "localName": "cmis:createdBy",
  "queryName": "cmis:createdBy",
  "type": "string",
  "cardinality": "single",
  "value": "dossier"
},
"cmis:creationDate": {
  "id": "cmis:creationDate",
  "localName": "cmis:creationDate",
  "queryName": "cmis:creationDate",
  "type": "datetime",
  "cardinality": "single",
  "value": 1670248188000
},
"cmis:lastModifiedBy": {
  "id": "cmis:lastModifiedBy",
  "localName": "cmis:lastModifiedBy",
  "queryName": "cmis:lastModifiedBy",
  "type": "string",
  "cardinality": "single",
  "value": "dossier"
},
"cmis:lastModificationDate": {
  "id": "cmis:lastModificationDate",
  "localName": "cmis:lastModificationDate",
  "queryName": "cmis:lastModificationDate",
  "type": "datetime",
  "cardinality": "single",
  "value": 1674660862000
},
"cmis:changeToken": {
  "id": "cmis:changeToken",
  "localName": "cmis:changeToken",

```



```

    "queryName": "cmis:changeToken",
    "type": "string",
    "cardinality": "single",
    "value": "MjUvMDEvMjAyMyAxNjozNDoyMg=="
  },
  "cmis:parentId": {
    "id": "cmis:parentId",
    "localName": "cmis:parentId",
    "queryName": "cmis:parentId",
    "type": "id",
    "cardinality": "single",
    "value": "YmFza2V0OkNSOkRlcm5pZXJzIGNvbXB0ZSByZW5kdXM="
  },
  "cmis:path": {
    "id": "cmis:path",
    "localName": "cmis:path",
    "queryName": "cmis:path",
    "type": "string",
    "cardinality": "single",
    "value": "/orga:5/flow:CR/basket:CR:Derniers compte rendus/document:CR:Derniers compte rendus:7476"
  },
  "cmis:allowedChildObjectTypeIds": {
    "id": "cmis:allowedChildObjectTypeIds",
    "localName": "cmis:allowedChildObjectTypeIds",
    "queryName": "cmis:allowedChildObjectTypeIds",
    "type": "id",
    "cardinality": "multi",
    "value": [
      "cm:attachment",
      "cm:comment",
      "cm:event"
    ]
  },
  "cm:document:cr:crTheme": {
    "id": "cm:document:cr:crTheme",
    "localName": "cm:document:cr:crTheme",
    "displayName": "Thématique",
    "queryName": "cm:document:cr:crTheme",
    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cm:document:cr:dModif": {
    "id": "cm:document:cr:dModif",
    "localName": "cm:document:cr:dModif",
    "displayName": "Date de modification",
    "queryName": "cm:document:cr:dModif",
    "type": "datetime",
    "cardinality": "single",
    "value": 1674660862000
  }

```

```

},
"cm:document:cr:dCreat": {
  "id": "cm:document:cr:dCreat",
  "localName": "cm:document:cr:dCreat",
  "displayName": "Date de création",
  "queryName": "cm:document:cr:dCreat",
  "type": "datetime",
  "cardinality": "single",
  "value": 1670248188000
},
"cm:document:cr:crRedacteur": {
  "id": "cm:document:cr:crRedacteur",
  "localName": "cm:document:cr:crRedacteur",
  "displayName": "Rédacteur",
  "queryName": "cm:document:cr:crRedacteur",
  "type": "string",
  "cardinality": "single",
  "value": null
},
"cm:document:cr:crDes": {
  "id": "cm:document:cr:crDes",
  "localName": "cm:document:cr:crDes",
  "displayName": "Désignation",
  "queryName": "cm:document:cr:crDes",
  "type": "string",
  "cardinality": "single",
  "value": "CHANGEMENT_DE_VALEUR"
},
"cm:document:cr:crResume": {
  "id": "cm:document:cr:crResume",
  "localName": "cm:document:cr:crResume",
  "displayName": "Résumé",
  "queryName": "cm:document:cr:crResume",
  "type": "string",
  "cardinality": "single",
  "value": "test creat"
},
"cm:document:cr:crDate": {
  "id": "cm:document:cr:crDate",
  "localName": "cm:document:cr:crDate",
  "displayName": "Date événement",
  "queryName": "cm:document:cr:crDate",
  "type": "datetime",
  "cardinality": "single",
  "value": null
},
"cm:document:cr:tPrior": {
  "id": "cm:document:cr:tPrior",
  "localName": "cm:document:cr:tPrior",
  "displayName": "xml_priority_field_label",
  "queryName": "cm:document:cr:tPrior",

```

```

    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cm:document:cr:multi": {
    "id": "cm:document:cr:multi",
    "localName": "cm:document:cr:multi",
    "displayName": "test champs multi",
    "queryName": "cm:document:cr:multi",
    "type": "string",
    "cardinality": "single",
    "value": ""
  },
  "cm:document:cr:corres": {
    "id": "cm:document:cr:corres",
    "localName": "cm:document:cr:corres",
    "displayName": "correspondent",
    "queryName": "cm:document:cr:corres",
    "type": "string",
    "cardinality": "single",
    "value": null
  },
  "cm:document:cr:masqueNum": {
    "id": "cm:document:cr:masqueNum",
    "localName": "cm:document:cr:masqueNum",
    "displayName": "test masque",
    "queryName": "cm:document:cr:masqueNum",
    "type": "integer",
    "cardinality": "single",
    "value": null
  }
}

```

## 3.5. Recherche



L'application **Dossier** offre la possibilité d'effectuer des **queries CMIS** avec quelques limitations car les opérateurs **JOIN**, **IN\_FOLDER** et **IN\_TREE** ne sont pas traités/implémentés en version 7.1.

Dans le but de lancer une recherche (**query**), les paramètres suivants doivent être précisés :

```

cmisselector=query
q=<CRITERES DE RECHERCHE>

```

Quelques exemples :

```

q=SELECT * FROM cm:document:cr WHERE cmis:creationDate >= '28/03/2020'

```

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2?cmisselector=query&succinct=true&q=SELECT%20*%20FROM%20cm:document:cr%20WHERE%20cmis:creationDate%20>=%20'28/03/2020'
```

```
q=SELECT * FROM cm:document:cr WHERE cm:document:cr:crDes = 'Désignation'
```

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2?cmisselector=query&succinct=true&q=SELECT%20*%20FROM%20cm:document:cr%20WHERE%20cm:document:cr:crDes%20=%20'Désignation'
```

```
q=SELECT * FROM cm:document:cr WHERE cm:document:cr:crDes LIKE 'Désign%'
```

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2?cmisselector=query&succinct=true&q=SELECT%20*%20FROM%20cm:document:cr%20WHERE%20cm:document:cr:crDes%20LIKE%20'Désign%'
```

```
q=SELECT * FROM cm:document:cr WHERE cm:document:cr:crDes LIKE 'Désign%' AND cmis:creationDate >= '28/03/2020'
```

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2?cmisselector=query&succinct=true&q=SELECT%20*%20FROM%20cm:document:cr%20WHERE%20cm:document:cr:crDes%20LIKE%20'Désign%'%20AND%20cmis:creationDate%20>=%20'28/03/2020'
```

```
q=SELECT * FROM cm:document:cr WHERE cmis:creationDate IN ('04/05/2020', '07/05/2020')
```

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2?cmisselector=query&succinct=true&q=SELECT%20*%20FROM%20cm:document:cr%20WHERE%20cmis:creationDate%20IN%20('04/05/2020',%20'07/05/2020')
```

```
q=SELECT * FROM cm:document:cr WHERE cmis:creationDate NOT IN ('04/05/2020', '07/05/2020') ORDER BY cmis:creationDate DESC
```

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2?cmisselector=query&succinct=true&q=SELECT%20*%20FROM%20cm:document:cr%20WHERE%20cmis:creationDate%20NOT%20IN%20('04/05/2020',%20'07/05/2020')%20ORDER%20BY%20cmis:creationDate%20DESC
```

## 3.6. Récupération du contenu d'un document

La récupération du contenu d'un document s'effectue en précisant les paramètres

```
cmisselector=content
objectId=<IDENTIFIANT DU DOCUMENT>
```

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2/root?objectId=YXR0YWNobWVudDpDUiNEZXJuaWVycyBjb21wdGUgcmluZHVzIzQ2MzMjNDI1Mg%3D%3D&cmisselector=content
```

Le corps (**body**) de la réponse sera alors le document *streamé*.

## 3.7. Bonnes pratiques

### 1. Réponses 'succinctes'

Les réponses, au format *JSON*, peuvent être verbeuses (i.e. représentant un important volume de données).

Ces réponses peuvent être abrégées/réduites en précisant le paramètre **succinct=true**.

En reprenant l'exemple du chapitre [Section 3.3, "Récupération d'éléments"](#)

```
http(s)://<hostname>:<port>/<context>/cmis/browser/Dossier-2/root/ZmxvdzpdUg%3D%3D?cmisselector=children&succinct=true
```

la réponse sera alors :

```
{
  "objects": [
    {
      "object": {
        "succinctProperties": {
          "cmis:objectId": "ZG9jdW1lbnQ6Q1IjRGVybmlldnMgY29tcHRlIHJlbmR1cyM0Njky",
          "cmis:objectTypeId": "cm:document:cr#cr",
          "cmis:baseTypeId": "cmis:folder",
          "cmis:name": "jjj",
          "cmis:description": "jjj",
          "cmis:createdBy": "dossier",
          "cmis:creationDate": 1589880601000,
          "cmis:lastModifiedBy": "dossier",
          "cmis:lastModificationDate": 1590482813000,
          "cmis:changeToken": "MjYvMDUvMjAyMCAxMDo0Njo1Mw==",
          "cmis:parentId": "ZmxvdzpdUg%3D%3D",
          "cmis:path": "/flow:CR/basket:CR#Derniers compte rendus/document:CR#Derniers compte rendus#4692",
          "cmis:allowedChildObjectIds": [
            "cm:attachment",
            "cm:comment",
            "cm:event"
          ],
          "cm:document:cr:crTheme": "Entreprise Libérée",
          "cm:document:cr:crDes": "Compte-rendu",
          "cm:document:cr:crRedacteur": "FÉLIX Nicolas",
          "cm:document:cr:crDate": 1589925600000
        }
      }
    }
  ]
}
```

```
}  
}  
,  
...,
```

## 4. Dossier CMIS serveur

### 4.1. Arborescence / object model

Dossier présente une représentation CMIS comme il suit.

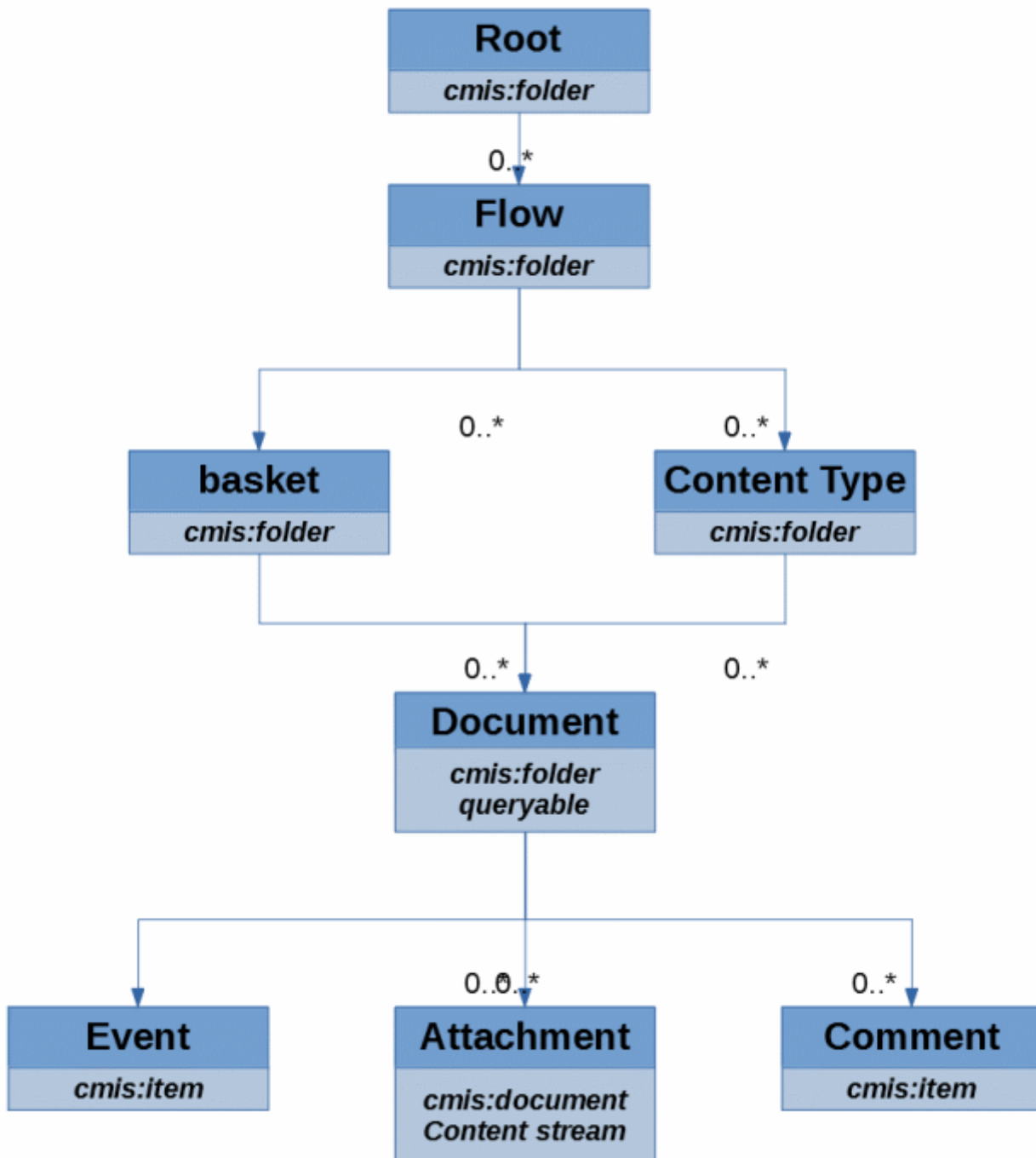
1. Dossier

CMIS

data

model

1 repository par organisation



## 4.2. Authentification

L'authentification au service **CMIS Dossier** est effectuée via une **Basic HTTP Authentication**, c'est-à-dire que le couple **identifiant/mot de passe** est transmis dans l'entête *http* encodé en **base64**



Cela sous-entend que pour une utilisation sur un réseau publique, le protocole **HTTPS / TLS** doit être mis en place.

## 4.3. Limitations

L'implémentation de la norme **CMIS** dans **Dossier** n'est pas exhaustive, la norme définissant de nombreuses fonctionnalités optionnelles.

Voici les principales fonctionnalités **non** implémentées (en 7.1 de **Dossier**):

- *Versioning*
- Modifications des *permissions* et droits d'accès
- Accès au *Change logs*
- *policies*

De plus, les opérateurs suivants, définis dans la norme **CMIS-SQL** (voir liste et description [ici](#)), ne sont pas implémentés

- **JOIN**
- **IN\_FOLDER** et **IN\_TREE**

## 4.4. Préconisations



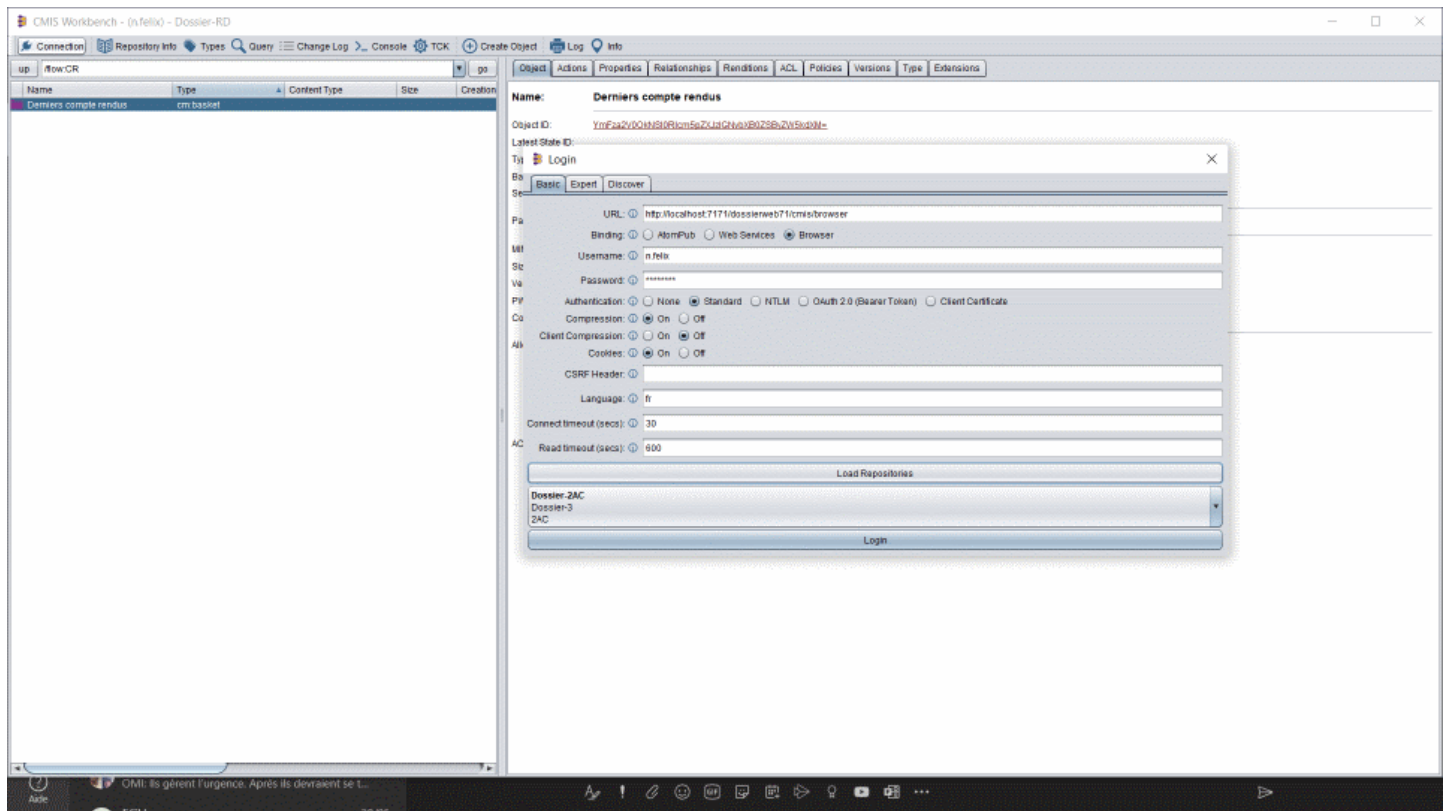
Comme déjà évoqué [ici](#), pour des raisons de performances et simplicité, le protocole **Browser Binding** est préconisé.



## 5. Clients CMIS

### 5.1. Apache Chemistry / opencmis

Plusieurs clients [CMIS](#) (1.1) gratuits sont disponibles, dont le [CMIS workbench](#) fournit par le projet [Apache Chemistry](#).



Chaque intégrateur/développeur peut utiliser l'une des [librairies disponibles](#) pour implémenter son propre client, en suivant notamment les différents guides disponibles [ici](#) et [là](#), ainsi que les [exemples](#) à disposition.

D'autres librairies existent également en dehors du contexte [Apache Chemistry](#), comme celle-ci [CmisJS](#).

### 5.2. Dossier CMIS client.

[Dossier](#) fournit, à partir de sa version 7.1, un client [CMIS Java](#), basé sur le projet [Apache Chemistry](#).

Celui-ci abstrait les principales actions utilisées.



La librairie, **dossier-cmis-client.jar** (actuellement en version **1.0.1**), n'est pas disponible sur un référentiel [maven](#) publique.  
Veuillez nous contacter pour obtenir celle-ci.

Vous trouverez :

- dans le chapitre [suivant](#) des exemples concrets.
- en annexe [l'interface principale](#)

- également en annexe [les dépendances](#) nécessaires à l'utilisation de la librairie **dossier-cmis-client.jar**.

## 5.3. Dossier CMIS : Cas d'utilisation



Toutes les opérations nécessitent au préalable une connexion à un repository

```
public class CmisConnection {
    /** The repository id. */
    private static final String REPOSITORY_ID = "myRepoID";

    private Session session = null;

    private IDossierCmisClient cmisClient;

    /**
     * constructor
     *
     * @param appUrl dossier URL
     */
    public CmisConnection(String appUrl) {
        super();
        cmisClient = new DossierBrowserCmisClient(url);
    }

    /**
     * Main method.
     */
    public static void main(String[] args)
        throws Exception {
        String dossierUrl = args[0];
        String login = args[0];
        String pwd = args[0];

        CmisConnection con = new CmisConnection(dossierUrl);
        Session session = con.connect2Repo(login, pwd);
    }

    /**
     * connect to repository
     *
     * @param login login
     * @param pwd password
     * @return token Session
     * @throws Exception ...
     */
    public Session connect2Repo(String login, String pwd)
        throws Exception {
        Session session = null;
        try {
```

```
// This call will work if there is only one repository !
session = cmisClient.createSession(login, pwd);
// else call
// session = cmisClient.createSession(REPOSITORY_ID, login, pwd);
}
catch(...){
}
return session;
}
```

Si **Dossier** contient plus d'un **repository** (par défaut, 1 est défini par **organisation**), il vous faut :

- spécifier l' **ID** du repository cible
  - soit parce que vous le connaissez
  - soit en interrogeant le serveur **CMIS** et en récupérant la liste des **repositories**

```
/**
 * get available repository IDs
 *
 * @param login login
 * @param pwd password
 * @return repo IDs
 * @throws Exception ...
 */
public List<String> getRepositoryIDs(String login, String pwd)
    throws Exception {
    try {
        List<Repository> repositories = cmisClient.getRepositoriesCmisInfo(login, pwd);
        if(repositories != null) {
            return repositories.stream().map(RepositoryInfo::getId).collect(Collectors.toList())
        }
    }
    catch(...){
    }
    return Collections.emptyList();
}
```

### 5.3.1. Récupération d'un objet

```
public Map<String, String> getObjectIDPropertiesAsString(Session session, String objectId)
    throws Exception{

    try{
        CmisObjectWrapper obj = cmisClient.getObjectById(session, objectId);
        return obj.getProperties().entrySet().stream().collect(Collectors.toMap(Map.Entry:
:getKey, e -> e.getValue().getValueAsString()));
    }
}
```

```

        catch(...){
        }
    }

```

### 5.3.2. Récupération des enfants d'un objet

```

public List<CmisObjectWrapper> getObjectChildren(Session session, String objectId)
    throws Exception{

    try{
        CmisObjectWrapper obj = cmisClient.getObjectById(session, objectId);

        return cmisClient.getAllChildren(obj.getCmisObject());
    }
    catch(...){
    }
}

```

### 5.3.3. Récupération du contenu binaire d'un document

```

public ContentStream getContentStream(Session session, String objectId)
    throws Exception{

    try{
        // attention, ContentStream#getStream, si utilisé, doit être fermé !
        return cmisClient.getContentStream(session, objectId);
    }
    catch(...){
    }
}

public String downloadContentStream(Session session, String objectId)
    throws Exception{

    try{
        // renvoie le chemin absolu vers le fichier téléchargé
        return cmisClient.downloadContentStream(session, objectId);
    }
    catch(...){
    }
}

```

il est également possible d'obtenir un contenu **partiel** du contenu

```

public ContentStream getPartialContentStream(Session session, String objectId, long offset,
long length)
    throws Exception{

```

```
try{
    // attention, ContentStream#getStream, si utilisé, doit être fermé !
    return cmisClient.getContentStream(session, objectId, offset, length );
}
catch(...){
}
}
```

## TODO Rendition à ajouter

### 5.3.4. queries

```
public List<CmisQueryResultWrapper> query(Session session, String query)
    throws Exception{

    return cmisClient.queryObjects(session, query);
}
```

### 5.3.5. Suppression d'un document

```
public void deleteObject(Session session, String objectId)
    throws Exception {

    try {
        cmisClient.deleteObject(session, objectId);
    }
    catch(...) {
    }
}
```

### 5.3.6. Mise à jour d'un document

```
public void updateObject(Session session, String objectId)
    throws Exception {

    try {
        String testString = "TEST UPDATE ";
        Map<String, Object> properties = new HashMap<>();
        properties.put("cm:document:cr:crDes", testString);
        properties.put("cm:document:cr:crDate", new Date(531658800));
        cmisClient.updateObject(session, objectId, properties);
    }
    catch(...) {
    }
}
```

### 5.3.7. Création d'un commentaire

```
public void createComment(Session session, String objectId)
    throws Exception {

    try {
        String testString = "TEST COMMENT ";
        Map<String, Object> properties = new HashMap<>();
        properties.put("cmis:name", testString);
        properties.put("cmis:objectId", "cm:comment");
        properties.put("cmis:baseTypeId", "cmis:item");
        ObjectId newObjectId = cmisClient.createObject(session, objectId, properties, null);
    }
    catch(...) {
    }
}
```

### 5.3.8. Création d'un document AIRS

```
public void createFolder(Session session, String objectId)
    throws Exception {

    try {
        String testString = "TEST CREATE FOLDER";
        Map<String, Object> properties = new HashMap<>();
        properties.put("cmis:objectId", "cm:document:cr");
        properties.put("cmis:baseTypeId", "cmis:folder");
        properties.put("cm:document:cr:crDes", testString);
        properties.put("cm:document:cr:crTheme", "172");
        properties.put("cm:document:cr:crRedacteur", "97");
        properties.put("cm:document:cr:crDate", new Date(531658800));
        ObjectId newObjectId = cmisClient.createObject(session, "ZmxvdzpdUg==", properties, null);
    }
    catch(...) {
    }
}
```

### 5.3.9. Création d'une pièce jointe

```
public void updateObject(Session session, String objectId)
    throws Exception {

    try {
        String testString = "TEST_CREATE_DOCUMENT";
        Map<String, Object> properties = new HashMap<>();
        properties.put("cmis:objectId", "cm:attachment");
    }
}
```

```
properties.put("cmis:baseTypeId", "cmis:document");
properties.put("cmis:versionLabel", testString);
properties.put("cmis:contentStreamFileName", "nomDeFichier.pdf");
properties.put("cmis:name", testString);
Path contentStream = Paths.get("C:/files/1.pdf");
ObjectId newObjectId = cmisClient.createObject(session,
"ZG9jdW11bnQ6Q1I6RGVybml1cnMgY29tcHRlIHJlbmR1czo3NjQy", properties, contentStream);
}
catch(...) {
}
}
```

## 6. Annexes

### 6.1. Interface client Dossier

*Interface IDossierCmisClient*

```
package com.digitech.dossier.cmis.client;

import java.io.IOException;
import java.nio.file.Path;
import java.util.List;
import java.util.Map;
import java.util.Set;
import org.apache.chemistry.opencmis.client.api.*;
import org.apache.chemistry.opencmis.commons.data.ContentStream;
import org.apache.chemistry.opencmis.commons.data.RepositoryCapabilities;
import org.apache.chemistry.opencmis.commons.data.RepositoryInfo;

import com.digitech.dossier.cmis.client.exception.CmisClientException;
import com.digitech.dossier.cmis.client.exception.InvalidArgumentException;
import com.digitech.dossier.cmis.client.exception.ObjectNotFoundException;
import com.digitech.dossier.cmis.client.typedef.CmisObjectWrapper;
import com.digitech.dossier.cmis.client.typedef.CmisQueryResultWrapper;
import com.digitech.dossier.cmis.client.typedef.CmisTypeDefinitionWrapper;

/**
 * Dossier Cmis Client interface
 */
public interface IDossierCmisClient {

    /**
     * inject basic credential (and default ones)
     *
     * @param login user login
     * @param pwd user password
     */
    void setCredentials(String login, String pwd);

    /**
     * inject language (and default one)
     *
     * @param language expected user language
     */
    void setLanguage(String language);

    /**
     * create CMIS session on 1st repository, using already set credentials
     * If more than one repository exist, an exception will be thrown
     */
}
```



```

* @return {@link Session}
* @throws CmisClientException in case of any exception
*/
Session createSession()
    throws CmisClientException;

/**
 * create CMIS session on 1st repository, using already set credentials
 *
 * @param repoId repository Id
 * @return {@link Session}
 * @throws CmisClientException in case of any exception
 */
Session createSession(String repoId)
    throws CmisClientException;

/**
 * create CMIS session on 1t repository
 * If more than one repository exist, an exception will be thrown
 *
 * @param login login
 * @param pwd password
 * @return {@link Session}
 * @throws CmisClientException in case of any exception
 */
Session createSession(String login, String pwd)
    throws CmisClientException;

/**
 * create CMIS session on specific repository (id)
 *
 * @param repoId repository Id
 * @param login login
 * @param pwd password
 * @return {@link Session}
 * @throws CmisClientException in case of any exception
 */
Session createSession(String repoId, String login, String pwd)
    throws CmisClientException;

/**
 * create CMIS session on specific repository (id)
 *
 * @param login login
 * @param pwd password
 * @param repoId repository Id
 * @param language language
 * @return {@link Session}
 * @throws CmisClientException in case of any exception
 */
Session createSession(String repoId, String login, String pwd, String language)

```

```

        throws CmisClientException;

/**
 * get available repositories, using credentials already specified by previous
 *setCredentials* call
 *
 * @return {@link Repository}
 * @throws CmisClientException in case of any exception
 */
List<Repository> getRepositoriesCmisInfo()
    throws CmisClientException;

/**
 * get available repositories, using specific credentials
 *
 * @param login login
 * @param pwd password
 * @return {@link RepositoryInfo}
 * @throws CmisClientException in case of any exception
 */
List<Repository> getRepositoriesCmisInfo(String login, String pwd)
    throws CmisClientException;

/**
 * get cmis repository information, related to the repository user is connected to
 *
 * @param session valid CMIS server session
 * @return {@link Repository}
 * @throws InvalidArgumentException if input {@link Session} is invalid
 * @throws CmisClientException in case of any exception
 */
RepositoryInfo getRepositoryCmisInfo(Session session)
    throws InvalidArgumentException, CmisClientException;

/**
 * get repository capabilities
 *
 * @param session valid CMIS server session
 * @return {@link RepositoryCapabilities}
 * @throws InvalidArgumentException if input {@link Session} is invalid
 * @throws CmisClientException in case of any exception
 */
RepositoryCapabilities getRepositoryCapabilities(Session session)
    throws InvalidArgumentException, CmisClientException;

/**
 * check if current repository enabled Query
 *
 * @param session valid CMIS server session
 * @return true if it is supported
 * @throws InvalidArgumentException if input {@link Session} is invalid

```

```

* @throws CmisClientException      in case of any exception
*/
boolean isSupportQuery(Session session)
    throws IllegalArgumentException, CmisClientException;

/**
 * get type definition from its ID
 *
 * @param session valid CMIS server session
 * @param typeId type ID
 * @return {@link ObjectType}
 * @throws CmisClientException      if input {@link Session} is invalid
 * @throws IllegalArgumentException in case of any exception
 */
ObjectType getCmisTypeDefinition(Session session, String typeId)
    throws CmisClientException, IllegalArgumentException;

/**
 * load type definitions
 *
 * @param session valid CMIS server session
 * @return list of {@link ObjectType}
 * @throws CmisClientException      if input {@link Session} is invalid
 * @throws IllegalArgumentException in case of any exception
 */
List<ObjectType> getCmisTypeDefinitions(Session session)
    throws CmisClientException, IllegalArgumentException;

/**
 * get type definition from its ID
 *
 * @param session valid CMIS server session
 * @param typeId type ID
 * @return {@link CmisTypeDefinitionWrapper}
 * @throws CmisClientException      if input {@link Session} is invalid
 * @throws IllegalArgumentException in case of any exception
 */
CmisTypeDefinitionWrapper getWrappedTypeDefinition(Session session, String typeId)
    throws CmisClientException, IllegalArgumentException;

/**
 * load type definitions
 *
 * @param session valid CMIS server session
 * @return list of {@link CmisTypeDefinitionWrapper}
 * @throws CmisClientException      if input {@link Session} is invalid
 * @throws IllegalArgumentException in case of any exception
 */
List<CmisTypeDefinitionWrapper> getWrappedTypeDefinitions(Session session)
    throws CmisClientException, IllegalArgumentException;

```

```

/**
 * get root folder
 *
 * @param session valid CMIS server session
 * @return root {@link Folder}
 * @throws CmisClientException in case of any exception
 * @throws InvalidArgumentException if input {@link Session} is invalid
 */
Folder getRootFolder(Session session)
    throws CmisClientException, InvalidArgumentException;

/**
 * get folder children, iterable result
 *
 * @param parent {@link Folder} to get children from
 * @return iterable children
 * @throws CmisClientException in case of any exception
 * @throws InvalidArgumentException if parent is not valid
 */
Iterable<CmisObject> getChildren(Folder parent)
    throws CmisClientException, InvalidArgumentException;

/**
 * get folder children, iterable result
 *
 * @param parent folder to get children from
 * @return iterable children
 * @throws CmisClientException in case of any exception
 * @throws InvalidArgumentException if parent is not valid or not a {@link Folder}
 */
Iterable<CmisObject> getChildren(CmisObject parent)
    throws CmisClientException, InvalidArgumentException;

/**
 * get all folder children
 *
 * @param parent {@link Folder} to get children from
 * @return iterable children
 * @throws CmisClientException in case of any exception
 * @throws InvalidArgumentException if parent is not valid
 */
List<CmisObjectWrapper> getAllChildren(Folder parent)
    throws CmisClientException, InvalidArgumentException;

/**
 * get all folder children
 *
 * @param parent folder to get children from
 * @return iterable children
 * @throws CmisClientException in case of any exception
 * @throws InvalidArgumentException if parent is not valid or not a {@link Folder}

```

```

*/
List<CmisObjectWrapper> getAllChildren(CmisObject parent)
    throws CmisClientException, InvalidArgumentException;

/**
 * get object by its id
 *
 * @param session valid CMIS server session
 * @param id      object id
 * @return {@link CmisObjectWrapper}
 * @throws CmisClientException in case of any exception
 * @throws InvalidArgumentException if input {@link Session} is invalid
 * @throws ObjectNotFoundException if object has not been found
 */
CmisObjectWrapper getObjectById(Session session, String id)
    throws CmisClientException, InvalidArgumentException, ObjectNotFoundException;

/**
 * get object by its id
 *
 * @param session      valid CMIS server session
 * @param id            object id
 * @param filteredMetadata expected returned metadata (by default all)
 * @return {@link CmisObjectWrapper}
 * @throws CmisClientException in case of any exception
 * @throws InvalidArgumentException if input {@link Session} is invalid
 * @throws ObjectNotFoundException if object has not been found
 */
CmisObjectWrapper getObjectById(Session session, String id, Set<String> filteredMetadata)
    throws CmisClientException, InvalidArgumentException, ObjectNotFoundException;

/**
 * get object by its path
 *
 * @param session valid CMIS server session
 * @param path    object path
 * @return {@link CmisObjectWrapper}
 * @throws CmisClientException in case of any exception
 * @throws InvalidArgumentException if input {@link Session} is invalid
 */
CmisObjectWrapper getObjectByPath(Session session, String path)
    throws CmisClientException, InvalidArgumentException;

/**
 * get object by its path
 *
 * @param session      valid CMIS server session
 * @param path          object path
 * @param filteredMetadata expected returned metadata (by default all)
 * @return {@link CmisObjectWrapper}
 * @throws CmisClientException in case of any exception

```

```

* @throws IllegalArgumentException if input {@link Session} is invalid
*/
CmisObjectWrapper getObjectByPath(Session session, String path, Set<String>
filteredMetadata)
    throws CmisClientException, IllegalArgumentException;

/**
 * query objects
 *
 * @param session valid CMIS server session
 * @param query SQL-92 compliant query
 * @return list of results (wrapped to {@link CmisQueryResultWrapper})
 * @throws CmisClientException in case of any exception
 * @throws IllegalArgumentException if input {@link Session} is invalid
 */
List<CmisQueryResultWrapper> queryObjects(Session session, String query)
    throws CmisClientException, IllegalArgumentException;

/**
 * query objects
 *
 * @param session valid CMIS server session
 * @param query SQL-92 compliant query
 * @param pageSize page size
 * @return list of results (wrapped to {@link CmisQueryResultWrapper})
 * @throws CmisClientException in case of any exception
 * @throws IllegalArgumentException if input {@link Session} is invalid
 */
List<CmisQueryResultWrapper> queryObjects(Session session, String query, int pageSize)
    throws CmisClientException, IllegalArgumentException;

/**
 * query objects, and get IDs
 *
 * @param session valid CMIS server session
 * @param query SQL-92 compliant query
 * @return list of objects ids
 * @throws CmisClientException in case of any exception
 * @throws IllegalArgumentException if input {@link Session} is invalid
 */
List<String> queryObjectIDs(Session session, String query)
    throws CmisClientException, IllegalArgumentException;

/**
 * query objects, and get IDs
 *
 * @param session valid CMIS server session
 * @param query SQL-92 compliant query
 * @param pageSize page size
 * @return list of objects ids
 * @throws CmisClientException in case of any exception

```

```

* @throws IllegalArgumentException if input {@link Session} is invalid
*/
List<String> queryObjectIDs(Session session, String query, int pageSize)
    throws CmisClientException, IllegalArgumentException;

/**
 * get content URL
 *
 * @param session valid CMIS server session
 * @param objectId object id to get document url from
 * @return url
 * @throws CmisClientException in case of any exception
 * @throws IllegalArgumentException if input {@link Session} is invalid
 * @throws ObjectNotFoundException if object has not been found
 */
String getContentStreamUrl(Session session, String objectId)
    throws CmisClientException, IllegalArgumentException, ObjectNotFoundException;

/**
 * get document content Stream.
 * Careful, if using {@link ContentStream#getStream()}, you should take care of closing it
 *
 * @param session valid CMIS server session
 * @param objectId object id to get document url from
 * @return see {@link ContentStream}
 * @throws CmisClientException in case of any exception
 * @throws IllegalArgumentException if input {@link Session} is invalid
 * @throws ObjectNotFoundException if object has not been found
 */
ContentStream getContentStream(Session session, String objectId)
    throws CmisClientException, IllegalArgumentException, ObjectNotFoundException;

/**
 * download content URL
 *
 * @param session valid CMIS server session
 * @param objectId object id to get document url from
 * @param offset offset (in bytes) to start document reading (first offset will then be
skipped)
 * @param length byte number read
 * @return see {@link ContentStream}
 * @throws CmisClientException in case of any exception
 * @throws IllegalArgumentException if input {@link Session} is invalid
 * @throws ObjectNotFoundException if object has not been found
 * @throws IOException if any I/O exception occurred
 */
ContentStream getContentStream(Session session, String objectId, long offset, long length)
    throws CmisClientException, IllegalArgumentException, ObjectNotFoundException,
IOException;

/**

```

```

* download content URL
*
* @param session valid CMIS server session
* @param objectId object id to get document url from
* @return full downloaded file path
* @throws CmisClientException in case of any exception
* @throws InvalidArgumentException if input {@link Session} is invalid
* @throws ObjectNotFoundException if object has not been found
* @throws IOException if any I/O exception occurred
*/
String downloadContentStream(Session session, String objectId)
    throws CmisClientException, InvalidArgumentException, ObjectNotFoundException,
IOException;

/**
* download content URL
*
* @param session valid CMIS server session
* @param objectId object id to get document url from
* @param targetFolder folder to download file to
* @return full downloaded file path
* @throws CmisClientException in case of any exception
* @throws InvalidArgumentException if input {@link Session} is invalid
* @throws ObjectNotFoundException if object has not been found
* @throws IOException if any I/O exception occurred
*/
String downloadContentStream(Session session, String objectId, Path targetFolder)
    throws CmisClientException, InvalidArgumentException, ObjectNotFoundException,
IOException;

/**
* delete an object
*
* @param session valid CMIS server session
* @param objectId object id to get document url from
* @throws CmisClientException in case of any exception
* @throws InvalidArgumentException if input {@link Session} is invalid
* @throws ObjectNotFoundException if object has not been found
*/
void deleteObject(Session session, String objectId)
    throws CmisClientException, InvalidArgumentException, ObjectNotFoundException;

/**
* update an object
*
* @param session valid CMIS server session
* @param objectId object id to get document url from
* @param propertiesMap properties to update (cmis property code, property value) {@link
Map}
* @throws CmisClientException in case of any exception
* @throws InvalidArgumentException if input {@link Session} is invalid

```



```

* @throws ObjectNotFoundException if object has not been found
*/
void updateObject(Session session, String objectId, Map<String, ?> propertiesMap)
    throws CmisClientException, InvalidArgumentException, ObjectNotFoundException;
}

```

## 6.2. librairie dossier-cmis-client

Liste des dépendances de **dossier-cmis-client**

```

com.digitech.airs:dossier-cmis-client:jar: 1.0.1
+- org.apache.chemistry.opencmis:chemistry-opencmis-client-impl:jar:1.1.0:compile
| +- org.apache.chemistry.opencmis:chemistry-opencmis-client-api:jar:1.1.0:compile
| +- org.apache.chemistry.opencmis:chemistry-opencmis-commons-api:jar:1.1.0:compile
| +- org.apache.chemistry.opencmis:chemistry-opencmis-commons-impl:jar:1.1.0:compile
| | \- org.codehaus.woodstox:woodstox-core-asl:jar:4.4.1:compile
| | \- org.codehaus.woodstox:stax2-api:jar:3.1.4:compile
| +- org.apache.chemistry.opencmis:chemistry-opencmis-client-bindings:jar:1.1.0:compile
| +- org.apache.cxf:cxf-rt-transport-http:jar:3.3.5:compile
| | \- org.apache.cxf:cxf-core:jar:3.3.5:compile
| | +- org.glassfish.jaxb:jaxb-runtime:jar:2.4.0-b180830.0438:compile
| | | +- javax.xml.bind:jaxb-api:jar:2.2.2:compile
| | | | +- javax.xml.stream:stax-api:jar:1.0-2:compile
| | | | \- javax.activation:activation:jar:1.1:compile
| | | +- org.glassfish.jaxb:txw2:jar:2.4.0-b180830.0438:compile
| | | +- com.sun.istack:istack-commons-runtime:jar:3.0.7:compile
| | | +- org.jvnet.staxex:stax-ex:jar:1.8:compile
| | | +- com.sun.xml.fastinfoset:FastInfoset:jar:1.2.15:compile
| | | \- javax.activation:javax.activation-api:jar:1.2.0:compile
| | +- com.fasterxml.woodstox:woodstox-core:jar:5.0.3:compile
| | +- org.apache.ws.xmlschema:xmlschema-core:jar:2.2.5:compile
| | \- jakarta.xml.bind:jakarta.xml.bind-api:jar:2.3.2:compile
| | \- jakarta.activation:jakarta.activation-api:jar:1.2.1:compile
| \- org.apache.cxf:cxf-rt-ws-policy:jar:3.0.12:compile
| +- wsdl4j:wsdl4j:jar:1.6.3:compile
| \- org.apache.neethi:neethi:jar:3.0.3:compile
+- org.apache.commons:commons-lang3:jar:3.9:compile
+- org.apache.commons:commons-collections4:jar:4.4:compile
+- commons-io:commons-io:jar:2.6:compile
+- commons-codec:commons-codec:jar:1.13:compile
+- org.slf4j:slf4j-api:jar:1.7.30:compile

```