



Manuel d'intégration applicative

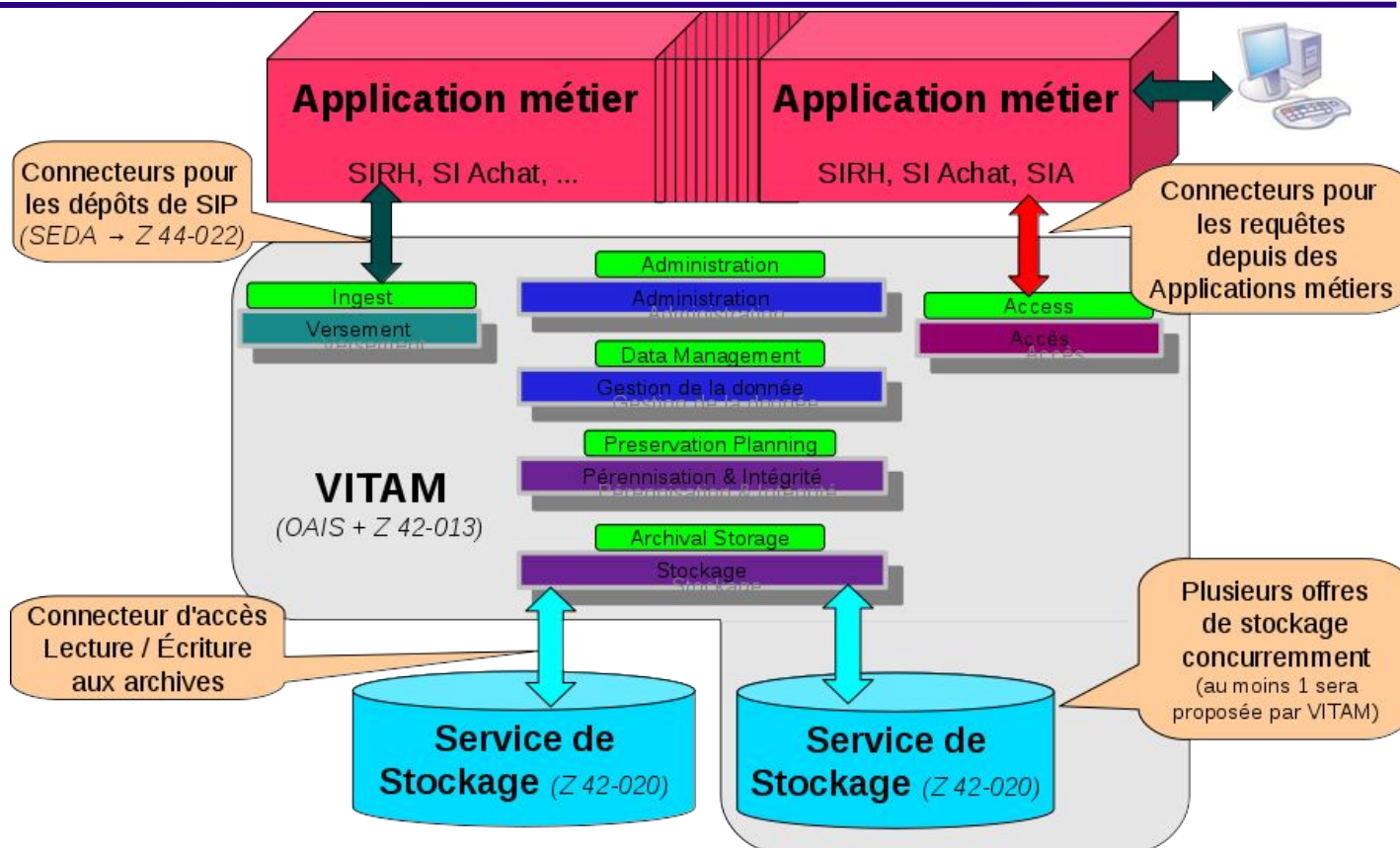
Release 3

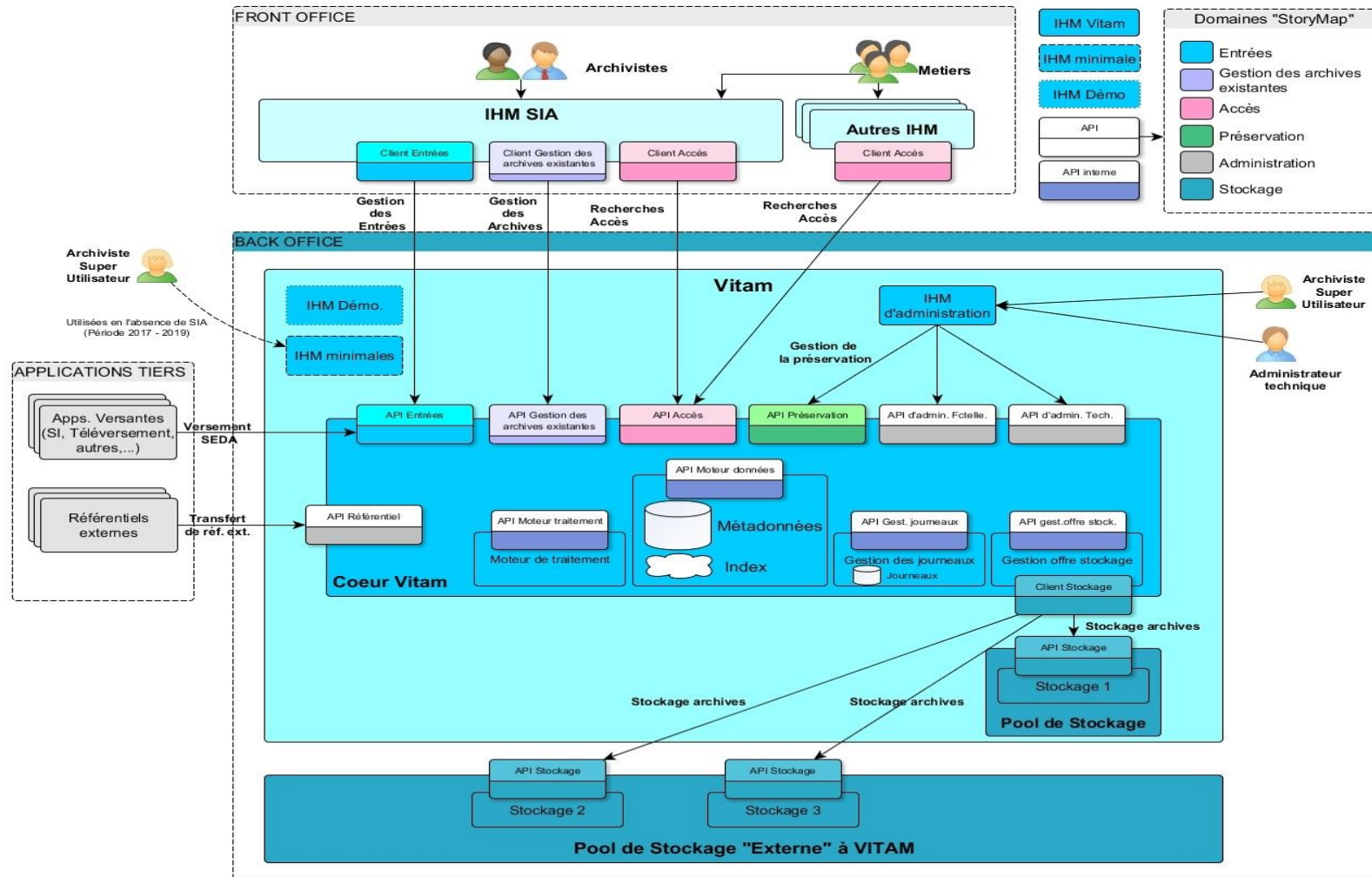
Valeurs Immatérielles Transférées aux Archives pour Mémoire

Plan

- 1) **VITAM**
 - a) **Architecture générale**
 - b) **Architecture des flux**
- 2) **Les API REST de VITAM**
 - a) **Liens documentation**
 - b) **Ingest**
 - c) **Access**
- 3) **DSL**
- 4) **Exemples**
- 5) **Discussions autour des API évolutives v1 et v2, Librairie helper pour la création de DSL ? bib. JS externe ?**

VITAM architecture générale





APIs REST de VITAM

Services

- ❖ ingest-external : opérations d'entrées
- ❖ access-external : accès et journaux d'opérations
- ❖ admin-external : gestion du référentiel et opérations d'administration

Quelques Ressources

- ❖ /ingest-external/v1/ingests
- ❖ /admin-external/v1/formats
- ❖ /access-external/v1/units

Les APIs REST de VITAM - Documentation

- Les API externes de VITAM :

De manière générale

/access-external /v1/ accession-register



Les APIs REST de VITAM - Documentation

RAML

- ❖ <http://www.programmevitam.fr/ressources/Doc0.11.1/raml/externe/introduction.html>
- ❖ <http://www.programmevitam.fr/ressources/Doc0.11.1/raml/externe/ingest.html>
- ❖ <http://www.programmevitam.fr/ressources/Doc0.11.1/raml/externe/access.html>
- ❖ <http://www.programmevitam.fr/ressources/Doc0.11.1/raml/externe/functional-administration.html>

Javadoc

- ❖ <http://www.programmevitam.fr/ressources/Doc0.11.1/javadoc>

Requête - Réponse VITAM

Les requêtes HTTP VITAM ont en commun les attributs suivants :

REQUEST	HEADER	X-Http-Method-Override: GET (optionnel et uniquement si on utilise la méthode POST au lieu de GET) X-Tenant-ID : Integer {0,1,2... }
---------	--------	---

Les réponses HTTP de VITAM ont en commun les attributs suivants :

REQUEST	HEADER	X-Http-Method-Override: GET X-Request-ID : champ alphanumérique unique
---------	--------	---

Vitam
C'est AUSSI By PostMan



Et autres clients..

Vitam By PostMan



Possibilité de gestion des collections

MANAGE ENVIRONNEMENTS

Manage Environments Environment Templates

Edit Environment Bulk Edit

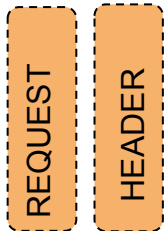
Vitam localhost

<input checked="" type="checkbox"/>	accessServiceUri	https://localhost:8444	⋮	×
<input checked="" type="checkbox"/>	ingestServiceUri	https://localhost:8443	⋮	×
<input checked="" type="checkbox"/>	accessResourcePath	/access-external	⋮	×
<input checked="" type="checkbox"/>	adminResourcePath	/admin-external	⋮	×
<input checked="" type="checkbox"/>	ingestResourcePath	/ingest-external	⋮	×
<input checked="" type="checkbox"/>	serviceVersion	/v1	⋮	×
<input checked="" type="checkbox"/>	adminStatus	/admin	⋮	×
<input checked="" type="checkbox"/>	status	/status	⋮	×
<input checked="" type="checkbox"/>	rulesCollection	/rules	⋮	×
<input checked="" type="checkbox"/>	formatsCollection	/formats	⋮	×
<input checked="" type="checkbox"/>	ingestsCollection	/ingests	⋮	×
<input checked="" type="checkbox"/>	unitsCollection	/units	⋮	×
<input checked="" type="checkbox"/>	operationsCollection	/operations	⋮	×
<input checked="" type="checkbox"/>	unitLifeCyclesCollection	/unitlifecycles	⋮	×

Cancel Update

Vitam By PostMan

POST /ingest-external/v1/ingests

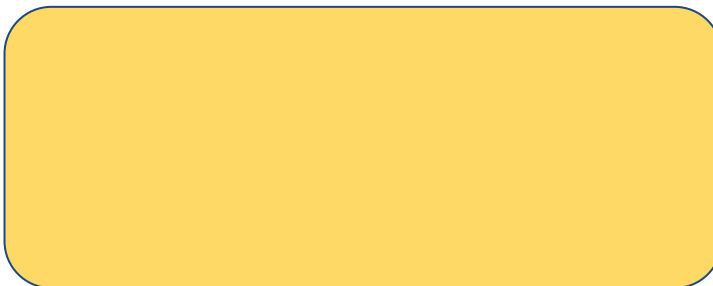


Accept: application/json

Content-Type: application/json

X-Http-Method-Override: GET

X-Tenant-ID: Integer



Vitam By PostMan



HEADER

Accept: application/json

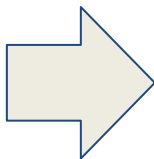
Content-Type: application/json

X-Http-Method-Override: GET

X-Tenant-ID : Integer

BODY

```
{
  "$query": {},
  "$filter": {},
  "$projection": {}
}
```



```
1 {
2   "Shits": {
3     "total": 213,
4     "offset": 0,
5     "limit": 1,
6     "size": 213
7   },
8   "results": [
9     {
10      "_id": "aeaaaaaaaaan7mxablqgakzw3crdeqaaaaa",
11      "evId": "aeaaaaaaaaan7mxablqgakzw3crdeqaaaaa",
12      "evType": "STP_REFERENTIAL_FORMAT_IMPORT",
13      "evDateTime": "2017-01-19T12:48:05.541",
14      "evOldData": null,
15      "evIdProc": "aeaaaaaaaaan7mxablqgakzw3crdeqaaaaa",
16      "evTypeProc": "MASTERDATA",
17      "outcome": "STARTED",
18      "outDetail": null,
19      "outMessg": "Lancement de l'import du référentiel de format",
20      "agId": "{ \"Name\": \"vitan-iaas-app-02\", \"Role\": \"functional-administration\", \"PlatformId\": 425367 }",
21      "agIdApp": null,
22      "agIdAppSession": null,
23      "evIdReq": "aeaaaaaaaaan7mxablqgakzw3crdeqaaaaa",
24      "agIdSubm": null,
25      "agIdOrig": null,
26      "obId": null,
27      "obIdReq": null,
28      "obIdIn": null,
29      "events": [
30        {
31          "evId": "aeaaaaaaaaan7mxablqgakzw3crepaaaaa",
32          "evType": "STP_REFERENTIAL_FORMAT_IMPORT",
33          "evDateTime": "2017-01-19T12:48:08.158",
34          "evOldData": null,
35          "evIdProc": "aeaaaaaaaaan7mxablqgakzw3crdeqaaaaa",
36          "evTypeProc": "MASTERDATA",
37          "outcome": "OK",
38          "outDetail": null,
39          "outMessg": "Succès de l'import du référentiel de format version 88 du fichier de signature PRONOW (DROID_SignatureFile)",
40          "agId": "{ \"Name\": \"vitan-iaas-app-02\", \"Role\": \"functional-administration\", \"PlatformId\": 425367 }",
```

Ingest

Ingests

API de versement (Ingest). Ce point d'entrée permet de chercher ou de créer une transaction de versement. Une transaction d'entrée est volatile, c'est à dire qu'elle disparaîtra dès qu'elle sera terminée. Sa terminaison est liée à la production du rapport et sa récupération par le service de transfert ayant effectué l'entrée.

Crée une transaction d'entrée :

- une requête unique, avec un 'body' contenant toutes les informations dans un ZIP ou un TAR ou un TAR.GZ ou TAR.BZ2 :
 - Métadonnées dans un format SEDA XML ou Json de nom manifest.xml ou manifest.json (json **UNSUPPORTED**)
 - Tous les binaires dans le répertoire "/content"
- d'autres formes pourront être implémentées dans des versions ultérieures (multipart/form-data) avec de multiples requêtes utilisant les sous-collections futures *Units* et *Objects*

/ingests

GET

POST

/ingests/{id_async}

GET

Ingest - Exemples java

/ingest-external/v1/ingests

```
File sipFile = .....
```

```
/* Client auto closeable */
```

```
try (IngestExternalClient client = IngestExternalClientFactory.getInstance().getClient()) {  
    finalResponse = client.upload(new FileInputStream(sipFile), tenantId);  
    final String guid = finalResponse.getHeaderString(GlobalDataRest.X_REQUEST_ID);  
    final InputStream inputStream = (InputStream) finalResponse.getEntity();  
    ....  
    finalResponse.close();  
}
```

```
} ...
```

/ingest-external/v1/ingests/{objectId}/{type} : type = {REPORTS, MANIFESTS}

```
try (IngestExternalClient client = IngestExternalClientFactory.getInstance().getClient()) {  
    Response response = client.downloadObjectAsync(objectId, IngestCollection.REPORTS, tenantId);  
    ...  
    response.close();  
}
```

Access

- ❖ Recherche par **ArchivalAgencyArchiveUnitIdentifier**,
- ❖ Recherche par **producteur** (ex : FRAN_NP00734),
- ❖ Recherche par **titre** (ex : simplification) AND **description** AND **dates** (ex: 2010-2013) AND **producteur** (ex: FRAN_NP00734),
- ❖ Recherche libre : **titre** OR **description**

Access-external

RAML

<http://www.programmevitam.fr/ressources/Doc0.11.1/raml/access/introduction.html>

API qui définit les requêtes pour accéder aux Unités d'archives.

La requête utilise le langage de requête (DSL) de Vitam en entrée et retourne une liste d'Unités d'archives selon le DSL Vitam en cas de succès.

Access-external

/units

ArchiveUnits

API qui définit les requêtes pour accéder aux Unités d'archives. La requête utilise le langage de requête (DSL) de Vitam en entrée et retourne une liste d'Unités d'archives selon le DSL Vitam en cas de succès.

/units

GET

POST

PUT

/units/{idu}

GET

POST

HEAD

PUT

/units/{idu}/object

GET

POST

HEAD

Access-external

/objects

Objects

API qui définit les requêtes pour accéder aux Objets d'archives. La requête utilise le langage de requête (DSL) de Vitam en entrée et retourne une liste d'Objets d'archives selon le DSL Vitam en cas de succès.

/objects

GET

POST

/objects/{ido}

GET

POST

HEAD



Zoom sur DSL

Valeurs Immatérielles Transférées aux Archives pour
Mémoire

Domain specific language - principes

- ❖ Dans le body : langage de requête
 - DSL VITAM
 - SQL : pas de plein texte, parser difficile
 - NoSQL : pas de norme
 - Abstraction indispensable (masquer l'implémentation)
- ❖ Typographie
 - Snake : « `propriete_avec_multiple_noms` »
 - Mais pas « `proprieteAvecMultipleNoms` »
 - Body au format JSON
 - Contient des informations spécifiques à la requête pour la collection
 - Peut contenir une « Query » (DSL)
- ❖ Pagination
 - offset / limit dans la Query
 - Range dans le Header pour les octets d'un fichier binaire
- ❖ Tri
 - orderby dans la Query

Domain specific language - corps de la requête

Projections, Collections, Requêtes (critères=query), Filtres (tri, limite)

SELECT field1, field2 **FROM** table **WHERE** field3 < value **LIMIT** n **SORT** field1 **ASC**

Modèle générique CRUD

Create = POST

data : { champ : valeur, champ : { champ : valeur } }

Read = GET

filter : { limit, offset, orderby }, projection : { field : 0/1, ... }

Update = PUT (avec forme ~ POST) / PATCH

action : { set : { field : value, ... }, inc : { field : value }, ... }

Delete = DELETE

filter : { mult : true/false }

roots = liste des Id de départ (sommet de l'arbre de classement)

HTTP POST/PUT /ressources

```
{
  roots : [ liste Id ],
  queries : [
    { query1 },
    { query2 }
  ],
  filter : { filter },
  data : { data }
}
```

HTTP GET /ressources

```
{
  roots : [ liste Id ],
  queries : [
    { query1 },
    { query2 }
  ],
  filter : { filter },
  projection : { projection }
}
```

HTTP PATCH /ressources

```
{
  roots : [ liste Id ],
  queries : [
    { query1 },
    { query2 }
  ],
  filter : { filter },
  action : { action }
}
```

HTTP DELETE /ressources

```
{
  roots : [ liste Id ],
  queries : [
    { query1 },
    { query2 }
  ],
  filter : { filter }
}
```

- Une Query est exprimée avec des opérateurs (inspiré MongoDB / Elastic)

Domain Specific Language Vitam Query

Query

Les commandes de la Query peuvent être :

Catégorie	Opérateur	Arguments	Commentaire
Accès direct	\$path	identifiants	Accès direct à un noeud
Booléens	\$and, \$or, \$not	opérateurs	Combinaison logique d'opérateurs
Comparaison	\$eq, \$ne, \$lt, \$lte, \$gt, \$gte	Champ et valeur	Comparaison de la valeur d'un champ et la valeur passée en argument
	\$range	Champ, \$lt, \$lte, \$gt, \$gte et valeurs	Comparaison de la valeur d'un champ avec l'intervalle passé en argument
Existence	\$exists, \$missing, \$isNull	Champ	Existence d'un champ
Tableau	\$in, \$nin	Champ et valeurs	Présence de valeurs dans un tableau
	\$size	Champ et taille	Taille d'un tableau
	[n] UNSUPPORTED	Position (n >= 0)	Élément d'un tableau
Textuel	\$term, \$wildcard	Champ, mot clef	Comparaison de champs mots-clefs
	\$match, \$matchPhrase, \$matchPhrasePrefix	Champ, phrase, \$max_expansions (optionnel)	Recherche de phrase
	\$regex	Champ, Expression régulière	Recherche via une expression régulière
	\$search	Champ, valeur	Recherche du type moteur de recherche
	\$fit, \$mit	Champ, valeur	Recherche « More Like This »
Géomatique	\$geometry, \$box, \$polygon, \$center	Positions	Définition d'une position géographique
UNSUPPORTED	\$geoWithin, \$geoIntersects, \$near	Une forme	Recherche par rapport à une forme géométrique

Chaque Query dispose éventuellement d'arguments additionnels pour gérer l'arborescence :

Catégorie	Opérateur	Arguments	Commentaire
Profondeur	\$depth, \$exactdepth	+ ou - n	Permet de spécifier si la query effectue une recherche vers les racines (-) ou vers les feuilles (+) et de quelle profondeur (n), avec une profondeur relative (\$depth) ou exacte (\$exactdepth)
Collection	\$source	units / objects	Permet dans une succession de Query de changer de collection. Attention, la dernière Query doit respecter la collection associée à la requête

DSL - API java et Documentation

❖ Documentation

→ <http://www.programmevitam.fr/ressources/Doc0.11.1/raml/externe/introduction.html>

❖ API java

→ Dans common/common-database-vitam/common-database-public

→ `fr.gouv.vitam.common.database`

→ `fr.gouv.vitam.common.database.builder.request.multiple;`

→ `fr.gouv.vitam.common.database.builder.request.single;`

Access - Exemples

Recherche par **ArchivalAgencyArchiveUnitIdentifier**

Client Java

EndPoint: /access-external/v1/units

```
try (AccessExternalClient client = AccessExternalClientFactory.getInstance().getClient()) {
    Integer tenantId= 0; //à titre d'exemple
    final String selectQuery = "{ \"$query\" : [ { \"$eq\" : { \"#ArchivalAgencyArchiveUnitIdentifier\" : \"20130456/3\" } } ] }";

    final JsonNode queryJson = JsonHandler.getFromString(selectQuery);

    client.selectUnits(queryJson, tenantId);

} catch (InvalidCreateOperationException e) {
    LOG ...
}
```



Access - Exemples

Recherche par **ArchivalAgencyArchiveUnitIdentifier**

/access-external/v1/units

Client Java
Avec
Construction DSI

```
try (AccessExternalClient client = AccessExternalClientFactory.getInstance().getClient()) {  
  
    Select select = new Select() ;  
    Integer tenantId= 0; //à titre d'exemple  
    JsonNode queryDsl = null;  
    final SelectParserSingle parser = new SelectParserSingle();  
        parser.parse(select.getFinalSelect());  
        parser.addCondition(QueryHelper.eq("ArchivalAgencyArchiveUnitIdentifier", "20130456/3"));  
        queryDsl = parser.getRequest().getFinalSelect();  
        client.selectUnits(queryJson, tenantId);  
    } catch (InvalidCreateOperationException e) {  
        LOG ...  
    }  
}
```

Access - Exemples

Recherche par **ArchivalAgencyArchiveUnitIdentifier**

POST /access-external/v1/units



REQUEST

HEADER

X-Http-Method-Override: GET
X-Tenant-ID : 0
Accept:application/json
Content-Type: application/json

BODY

```
{
  "$roots": [],
  "$query": [
    {
      "$and": [
        {
          "$eq": {
            "#ArchivalAgencyArchiveUnitIdentifier": "20130456/3"
          }
        }
      ]
    },
    {
      "$depth": 20
    }
  ]
}
```

Access - Exemples

Recherche par Recherche par producteur (FRAN_NP00734)

Endpoint : </access-external/v1/accession-register>



Client Java

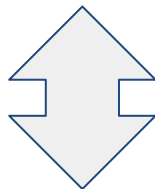
Access - Exemples

Recherche par Recherche par producteur (FRAN_NP00734)

Client Java

/access-external/v1/accession-register
par producteur (FRAN_NP_005568)

```
final AccessExternalClient client =  
    AccessExternalClientFactory.getInstance().getClient();  
final String options =  
  
    "{ \"$query\" : [ { \"$eq\" : { \"OriginatingAgency\" : \"FRAN_NP_005568\" } } ] }";  
Integer tenantId= 0 ;  
  
try (AccessExternalClient client = AccessExternalClientFactory.getInstance().getClient()) {  
    final Map<String, String> optionsMap = JsonHandler.getMapStringFromString(options);  
    final JsonNode query = Dsl.QueryHelper.createSingleQueryDSL(optionsMap);  
    return client.getAccessionRegisterSummary(query, tenantId);  
}
```



Access - Exemples java

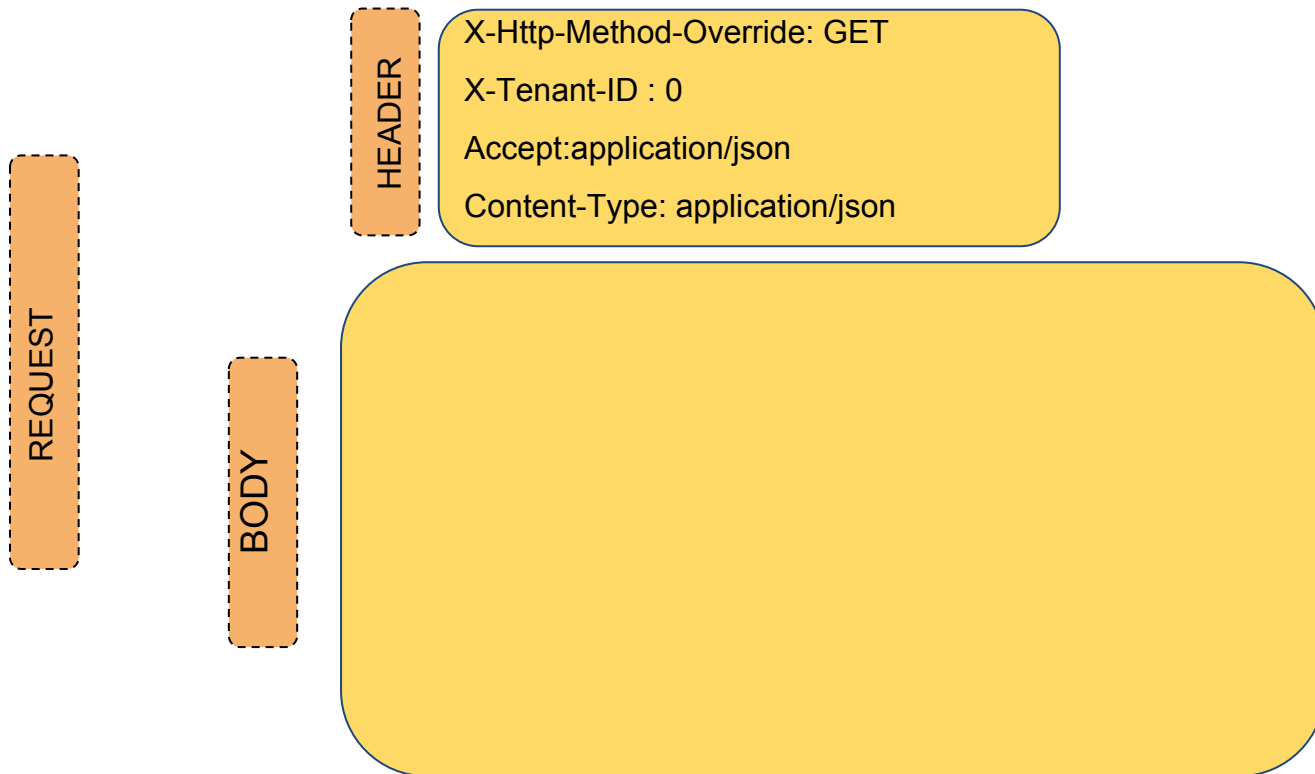
Recherche par Recherche par producteur (FRAN_NP00734)

/access-external/v1/accession-register

Client Java
Avec
Construction DSI

```
try (AccessExternalClient client = AccessExternalClientFactory.getInstance().getClient()) {  
  
    final Select select = new Select();  
  
    select.setQuery(eq("OriginatingAgency", "FRAN_NP_005568"));  
  
    client.getAccessionRegisterSummary(select.getFinalSelect(), tenantId);  
  
} catch (InvalidCreateOperationException e) {  
    LOG ...  
}
```

POST /access-external/v1/accession-register



POST /access-external/v1/accession-register



REQUEST

BODY

```
{
  "$query": {
    "$and": [
      {
        "$eq": {
          "OriginatingAgency": "FRAN_NP_005568"
        }
      }
    ]
  },
  "$filter": {},
  "$projection": {}
}
```

Access - Exemples

- Recherche par titre (ex : simplification) AND description AND dates
-

Endpoint : </access-external/v1//units>

- Recherche libre : **recherche par titre (ex : simplification) AND description AND dates**



Client Java

Access - Exemples

- Recherche par titre (ex : simplification) AND description AND dates

/access-external/v1/units

```
try (AccessExternalClient client =  
AccessExternalClientFactory.getInstance().getClient()) {  
  
final Select select = new Select();  
  
select.setQuery(or(eq("title", "test"),eq("Description", "test")));  
....  
.....  
.....  
  
client.getAccessionRegisterSummary(select.getFinalSelect(), tenantId);  
  
} catch (InvalidCreateOperationException e) {  
LOG ...  
}
```

Client Java

Access - Exemples java

Recherche par titre (ex : simplification) AND description AND dates

[/access-external/v1/units](#)

REQUEST

X-Http-Method-Override: GET

X-Tenant-ID : 0

Accept:application/json

Content-Type: application/json



/access-external/v1/units

REQUEST

BODY

```
{  "$roots": [],
  "$query": [ {
    "$and": [ {
      "$match": {
        "Title": "api"
      } }, {
      "$match": {
        "Description": "Vitam"
      } }, {
      "$or": [
        {
          "$and": [
            {
              "$gte": {
                "TransactedDate": "2016-11-09T23:00:00.000Z"
              } }, {
              "$lte": {
                "TransactedDate": "2016-11-16T22:59:59.000Z"
              } }
            ]
          },
          {
            "$and": [ {
              "$gte": {
                "EndDate": "2016-11-09T23:00:00.000Z"
              } }, {
              "$lte": {
                "StartDate": "2016-11-16T22:59:59.000Z"
              } }
            ]
          }
        ]
      }
    ] }, { "$depth": 20 }
  ],
  "$filter": {
    "$orderby": {
      "TransactedDate": 1
    } },
  "$projection": {
    "$fields": {
      "TransactedDate": 1,
      "#id": 1,
      "Title": 1,
      "#object": 1
    }
  }
}
```



DSL Recherche libre : titre OR description

Endpoint : </access-external/v1//units>



Client Java

Access - Exemples java

Recherche libre : titre OR description

/access-external/v1/units

[Via le client java](#)

```
try (AccessExternalClient client = AccessExternalClientFactory.getInstance().getClient())
{
    final Select select = new Select();
    select.setQuery(or(eq("title", "test"), eq("Description", "test")));
    client.getAccessionRegisterSummary(select.getFinalSelect(), tenantId);
} catch (InvalidCreateOperationException e) {
    LOG ...
}
```

Client Java

Access - Exemples java

Recherche libre : titre OR description

[/access-external/v1/units](#)

HEADER

X-Http-Method-Override: GET

X-Tenant-ID : 0

Accept:application/json

Content-Type: application/json

REQUEST



/access-external/v1/units

REQUEST

BODY

```
{
  "$roots": [],
  "$query": [
    {
      "$or": [
        {
          "$match": {
            "Title": "api"
          }
        },
        {
          "$match": {
            "Description": "Vitam"
          }
        }
      ],
      "$depth": 20
    }
  ],
  "$filter": {
    "$orderby": {
      "TransactedDate": 1
    }
  },
  "$projection": {
    "$fields": {
      "TransactedDate": 1,
      "#id": 1,
      "Title": 1,
      "#object": 1
    }
  }
}
```

