

API avancé

Brigitte Bidégaray-Fesquet

Laboratoire Jean Kuntzmann, Grenoble



Journées CasuHAL 2024
13 juin 2024



API avancé

- 1 Introduction
- 2 Les facettes
- 3 Les autres référentiels
 - Les auteurs et leurs structures
 - Les portails et leurs types de documents
 - Domaines et journaux
 - Les financements
- 4 La base de données : le triplestore dataHAL
- 5 Pour finir...

Ce que vous devez déjà savoir

- Le point d'entrée de l'API de recherche HAL
<https://api.archives-ouvertes.fr/search/>
- La structure générale d'une interrogation
 - le champ sur lequel on veut chercher : **q**
 - des filtres sur les réponses : **fq**
 - des facettes (détaillées dans l'atelier «API avancé»)
 - les champs retournés dans la réponse : **fl**
 - l'ordre dans lequel on trie les réponses : **sort**
 - le rang des résultats retournés : **start** et **rows**
 - le format de sortie : **wt** (JSON par défaut)

Ce dont je vais parler

- **Les facettes**
 - un peu abordé dans la formation API : initiation
 - plus détaillé ici
- **Les autres référentiels**
 - leurs champs spécifiques
 - mais on n'en interroge qu'un seul à la fois. . .
 - et pour aller plus loin, soit on a suivi l'atelier Pytnon
 - soit on peut aussi interroger . . .
- **Le triplestore dataHAL**

API avancé

- 1 Introduction
- 2 Les facettes
- 3 Les autres référentiels
 - Les auteurs et leurs structures
 - Les portails et leurs types de documents
 - Domaines et journaux
 - Les financements
- 4 La base de données : le triplestore dataHAL
- 5 Pour finir...

Utilisation de facettes

Un moyen pour filtrer en complément d'une requête est d'utiliser des facettes.

- Pour générer des facettes, il faut ajouter le paramètre **facet=true** à une requête

- Ensuite, on précise

- le champ qui sert pour la facette

facet.field=<champ> :<valeur>

- le type de tri

facet.sort=index (tri lexicographique)

facet.sort=count (tri par nombre d'occurrence)

- Il faut aussi ajouter **&rows=0** (pour n'avoir que les facettes)



Répartition par domaine dans la collection LJK

https://api.archives-ouvertes.fr/search/LJK/?q=*&rows=0
&facet=true&facet.field=level0_domain_s&facet.sort=count

```
{
  "response": {
    "numFound": 6787,
    "start": 0,
    "maxScore": 1,
    "numFoundExact": true,
    "docs": []
  },
  "facet_counts": {
    "facet_queries": {},
    "facet_fields": {
      "level0_domain_s": [
        "math",
        3138,
        "info",
        3115,
        "stat",
        1056,
        "spi",
        584,
        "phys",
        453,
        "sdv",
        291,
        "sdu",
        238,
        "sde",
        224,
```

Raw

Parsed

Préfixe

- On peut aussi préciser
 - le préfixe par lequel commencent les facettes
`facet.prefix=<préfixe>`
 - Répartition par mots-clés commençant par V dans la collection LJK :

`https://api.archives-ouvertes.fr/search/LJK/?q=*.*&rows=0
&facet=true&facet.field=keyword_s&facet.prefix=V`

```
{
  "response": {
    "numFound": 6786,
    "start": 0,
    "maxScore": 1,
    "numFoundExact": true,
    "docs": []
  },
  "facet_counts": {
    "facet_queries": {},
    "facet_fields": {
      "keyword_s": [
        "Vision par ordinateur",
        30,
        "Variational data assimilation",
        15,
        "Visualization",
        12,
        "Variable selection",
        11,

```

Raw Parsed

Terme contenu dans une facette

- On peut aussi préciser
 - un terme présent dans la facette
`facet.contains=<chaîne>`
 - Répartition par mots-clés contenant «*schema*» dans la collection LJK :
`https://api.archives-ouvertes.fr/search/LJK/?q=*&rows=0
&facet=true&facet.field=keyword_s&facet.contains=schema`
 - et on peut préciser d'ignorer la casse :
`facet.contains.ignoreCase=true`

Pivots

- On peut se servir de champs non multi-valués pour servir de pivot

`facet.pivot=<pivot>`

- Répartition par type de documents des types de dépôts dans la collection LJK :

`https://api.archives-ouvertes.fr/search/LJK/?q=*&rows=0
&indent=true&facet=true&facet.pivot=docType_s,submitType_s`

```

  ▾ "docType_s,submitType_s": [
    ▾ {
      "field": "docType_s",
      "value": "COMM",
      "count": 2765,
      ▾ "pivot": [
        ▾ {
          "field": "submitType_s",
          "value": "file",
          "count": 1523
        },
        ▾ {
          "field": "submitType_s",
          "value": "notice",
          "count": 1188
        },
        ▾ {
          "field": "submitType_s",
          "value": "annex"
        }
      ]
    }
  ]

```

Ordre des pivots

Pour un humain l'ordre des pivots compte :

[https://api.archives-ouvertes.fr/search/saga/?q=docType_s:ART
&rows=0&facet=true&facet.pivot=submitType_s,submittedDateY_i](https://api.archives-ouvertes.fr/search/saga/?q=docType_s:ART&rows=0&facet=true&facet.pivot=submitType_s,submittedDateY_i)
&wt=xml

vs.

[https://api.archives-ouvertes.fr/search/saga/?q=docType_s:ART
&rows=0&facet=true&facet.pivot=submittedDateY_i,submitType_s](https://api.archives-ouvertes.fr/search/saga/?q=docType_s:ART&rows=0&facet=true&facet.pivot=submittedDateY_i,submitType_s)
&wt=xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <result name="response" numFound="103365" start="0" maxScore="0.3173635">
    <lst name="facet_counts">
      <lst name="facet_queries"/>
      <lst name="facet_fields"/>
      <lst name="facet_ranges"/>
      <lst name="facet_intervals"/>
      <lst name="facet_heatmaps"/>
    </lst>
    <lst name="facet_pivot">
      <arr name="submitType_s,submittedDateY_i">
        <lst>
          <str name="field">submitType_s</str>
          <str name="value">notice</str>
          <int name="count">67225</int>
        </arr>
        <arr name="pivot">
          <lst>
            <str name="field">submittedDateY_i</str>
            <int name="value">2019</int>
            <int name="count">9804</int>
          </lst>
          <lst>
            <str name="field">submittedDateY_i</str>
            <int name="value">2018</int>
            <int name="count">6212</int>
          </lst>
          <lst>
            <str name="field">submittedDateY_i</str>
            <int name="value">2009</int>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <result name="response" numFound="103364" start="0" maxScore="0.3195819">
    <lst name="facet_counts">
      <lst name="facet_queries"/>
      <lst name="facet_fields"/>
      <lst name="facet_ranges"/>
      <lst name="facet_intervals"/>
      <lst name="facet_heatmaps"/>
    </lst>
    <lst name="facet_pivot">
      <arr name="submittedDateY_i,submitType_s">
        <lst>
          <str name="field">submittedDateY_i</str>
          <int name="value">2019</int>
          <int name="count">12341</int>
        </arr>
        <arr name="pivot">
          <lst>
            <str name="field">submitType_s</str>
            <str name="value">notice</str>
            <int name="count">9804</int>
          </lst>
          <lst>
            <str name="field">submitType_s</str>
            <str name="value">file</str>
            <int name="count">2529</int>
          </lst>
```



Plages de données

- On peut regrouper des résultats en définissant des plages de résultats :
 - champ pour faire la facette : **facet.range**
 - valeur minimale : **facet.range.start**
 - valeur maximale : **facet.range.end**
 - longueur des plages : **facet.range.gap**
- Publications du LJK publiées par paquets de 4 années :

```
http://api.archives-ouvertes.fr/search/LJK/?q=*&facet=true  
&rows=0&facet.range=submittedDateY_i  
&facet.range.start=2007&facet.range.end=2023&facet.range.gap=4  
&wt=xml
```



API avancé

- 1 Introduction
- 2 Les facettes
- 3 Les autres référentiels**
 - Les auteurs et leurs structures
 - Les portails et leurs types de documents
 - Domaines et journaux
 - Les financements
- 4 La base de données : le triplestore dataHAL
- 5 Pour finir...

Les API de HAL

- Il y n'a pas que l'API de recherche HAL.
- Les autres API
 - API SWORD de dépôt sur HAL
 - Les API de recherche dans les référentiels

anrproject

doctype

metadata

author

domain

metadatalist

authorstructure

instance

structure

europeanproject

journal

Le référentiel author

- Lien : <https://api.archives-ouvertes.fr/ref/author>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/author>
- Champs :
[https://api.archives-ouvertes.fr/docs/ref/?resource=author
&schema=fields#fields](https://api.archives-ouvertes.fr/docs/ref/?resource=author&schema=fields#fields)
- Ne retourne pas les mêmes informations que l'API de recherche :
[https://api.archives-ouvertes.fr/search/
?q=authLastName_t:"Bidegaray"](https://api.archives-ouvertes.fr/search/?q=authLastName_t:)
[https://api.archives-ouvertes.fr/ref/author/
?q=lastName_t:"Bidegaray"](https://api.archives-ouvertes.fr/ref/author/?q=lastName_t:)



Essayer d'identifier un auteur

- Un cas simple (car je suis unique) :

https://api.archives-ouvertes.fr/ref/author/?q=Bidegaray-Fesquet&fl=*_s
*_s est un champ dynamique qui retourne toutes sortes d'«identifiants»

- Un cas plus compliqué qui permet d'y voir plus clair :

[https://api.archives-ouvertes.fr/ref/author/?q=lastName_s:Picard
&fq=firstName_s:C*&fl=*_s](https://api.archives-ouvertes.fr/ref/author/?q=lastName_s:Picard&fq=firstName_s:C*&fl=*_s)

[https://api.archives-ouvertes.fr/ref/author/?q=lastName_s:Picard
&fq=firstName_s:Christophe&fl=*_s](https://api.archives-ouvertes.fr/ref/author/?q=lastName_s:Picard&fq=firstName_s:Christophe&fl=*_s)



Le référentiel authorstructure

- Toujours présenté comme un référentiel mais s'interroge à partir de l'API de recherche.
- Lien : <https://api.archives-ouvertes.fr/search/authorstructure>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/authorstructure>
- Trouver les structures de rattachement d'un auteur
https://api.archives-ouvertes.fr/search/authorstructure?firstName_t=Brigitte&lastName_t=Bidegaray&wt=xml
- et en filtrant sur une période
https://api.archives-ouvertes.fr/search/authorstructure?firstName_t=Brigitte&lastName_t=Bidegaray&producedDateY_i=2001&deviation=1&wt=xml



Le référentiel structure

- Lien : <https://api.archives-ouvertes.fr/ref/structure>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/structure>
- Champs :
[https://api.archives-ouvertes.fr/docs/ref/?resource=structure
&schema=fields#fields](https://api.archives-ouvertes.fr/docs/ref/?resource=structure&schema=fields#fields)
- Vous y retrouver au milieu de structures localisées en Corée du Sud
[https://api.archives-ouvertes.fr/ref/structure
?q=country_s:kr&fl=label_s,type_s,valid_s,parentName_s](https://api.archives-ouvertes.fr/ref/structure?q=country_s:kr&fl=label_s,type_s,valid_s,parentName_s)



Le référentiel instance

- Lien : <https://api.archives-ouvertes.fr/ref/instance>
Retourne toutes les instances
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/instance>

Le référentiel doctype

- Lien : <https://api.archives-ouvertes.fr/ref/doctype>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/doctype>
- Champs : le portail **instance_s** et la langue **lang**
- Trouver les types de documents présents dans DUMAS
https://api.archives-ouvertes.fr/ref/doctype?q=instance_s:dumas&fl=label_s

Le référentiel metadata

- Lien : <https://api.archives-ouvertes.fr/ref/metadata>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/metadata>
- Champs :
[https://api.archives-ouvertes.fr/docs/ref/?resource=metadata
&schema=fields#fields](https://api.archives-ouvertes.fr/docs/ref/?resource=metadata&schema=fields#fields)
- Trouver les métadonnées attendues pour un type de document donné
[https://api.archives-ouvertes.fr/ref/metadata
?q=docType_s:ART&fl=label_s](https://api.archives-ouvertes.fr/ref/metadata?q=docType_s:ART&fl=label_s)

Le référentiel metadatalist

- Lien <https://api.archives-ouvertes.fr/ref/metadatalist>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/metadatalist>
- Champs :
[https://api.archives-ouvertes.fr/docs/ref/?resource=metadatalist
&schema=fields#fields](https://api.archives-ouvertes.fr/docs/ref/?resource=metadatalist&schema=fields#fields)
- Trouver tous les types d'annexes
[https://api.archives-ouvertes.fr/ref/metadatalist
?q=metaName_t:typeAnnex&fl=label_s](https://api.archives-ouvertes.fr/ref/metadatalist?q=metaName_t:typeAnnex&fl=label_s)

Le référentiel domain

- Lien : <https://api.archives-ouvertes.fr/ref/domain>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/domain>
- Champs :
[https://api.archives-ouvertes.fr/docs/ref/?resource=domain
&schema=fields#fields](https://api.archives-ouvertes.fr/docs/ref/?resource=domain&schema=fields#fields)
- **Savoir avec quelle finesse vous allez décrire les SHS**
https://api.archives-ouvertes.fr/ref/domain?q=shs&fl=label_s

Le référentiel journal

- Lien : <https://api.archives-ouvertes.fr/ref/journal>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/journal>
- Champs :
[https://api.archives-ouvertes.fr/docs/ref/?resource=journal
&schema=fields#fields](https://api.archives-ouvertes.fr/docs/ref/?resource=journal&schema=fields#fields)
- Trouver les journaux avec «**Probab**» dans **text**
et savoir si c'est une forme valide
https://api.archives-ouvertes.fr/ref/journal?q=Probab&fl=label__s,valid__s

Le référentiel anrproject

- Lien : <https://api.archives-ouvertes.fr/ref/anrproject>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/anrproject>
- Champs :
[https://api.archives-ouvertes.fr/docs/ref/?resource=anrproject
&schema=fields#fields](https://api.archives-ouvertes.fr/docs/ref/?resource=anrproject&schema=fields#fields)
- Trouver la référence d'un projet ANR dont on connaît l'acronyme
[https://api.archives-ouvertes.fr/ref/anrproject
?q=acronym_t:PERSYVAL&fl=reference_s](https://api.archives-ouvertes.fr/ref/anrproject?q=acronym_t:PERSYVAL&fl=reference_s)



Le référentiel europeanproject

- Lien : <https://api.archives-ouvertes.fr/ref/europeanproject>
- Documentation :
<https://api.archives-ouvertes.fr/docs/ref/resource/europeanproject>
- Champs :
[https://api.archives-ouvertes.fr/docs/ref/?resource=europeanproject
&schema=fields#fields](https://api.archives-ouvertes.fr/docs/ref/?resource=europeanproject&schema=fields#fields)
- Chercher des projets de l'appel H2020
[https://api.archives-ouvertes.fr/ref/europeanproject
?q=callId_t:H2020&fl=label_s](https://api.archives-ouvertes.fr/ref/europeanproject?q=callId_t:H2020&fl=label_s)

API avancé

- 1 Introduction
- 2 Les facettes
- 3 Les autres référentiels
 - Les auteurs et leurs structures
 - Les portails et leurs types de documents
 - Domaines et journaux
 - Les financements
- 4 La base de données : le triplestore dataHAL
- 5 Pour finir...

Le triplestore de HAL, dataHAL

Certaines requêtes plus compliquées peuvent nécessiter l'utilisation de plusieurs référentiels ou de l'API de recherche et d'un référentiel et dans ce cas là il est difficile de combiner les recherches sans logiciel adapté.

<https://www.ccsd.cnrs.fr/triplestore-data-hal/>



Documentation

Accès à la présentation du triplestore et aux différents schémas de données

[Consulter »](#)



SPARQL endpoint

Interface d'interrogation de la base de connaissance structurée en RDF

[Consulter »](#)



Téléchargement

Téléchargement de l'ensemble des contenus du triplestore

[Consulter »](#)

source : <https://data.hal.science/> [consulté le 11 juin 2024]

RDF (1/2)

RDF (*Resource Description Framework*) : cadre de description de ressources.

Un document structuré en RDF est un ensemble de triplets, d'où le nom de *triplestore*.

C'est le langage du web sémantique.

Un triplet RDF est constitué

- d'un **sujet** qui représente la ressource à décrire,
- d'un **prédicat** qui représente un type de propriété applicable à cette ressource,
- et d'un **objet** qui représente une donnée ou une autre ressource et est la valeur de la propriété.

RDF (2/2)

- Le prédicat est toujours identifié par un URI (Uniform Resource Identifier, URL ou URN).
- Le sujet et l'objet peuvent aussi être identifiés par un URI ou être des nœuds anonymes.

Vocabulaires et ontologies

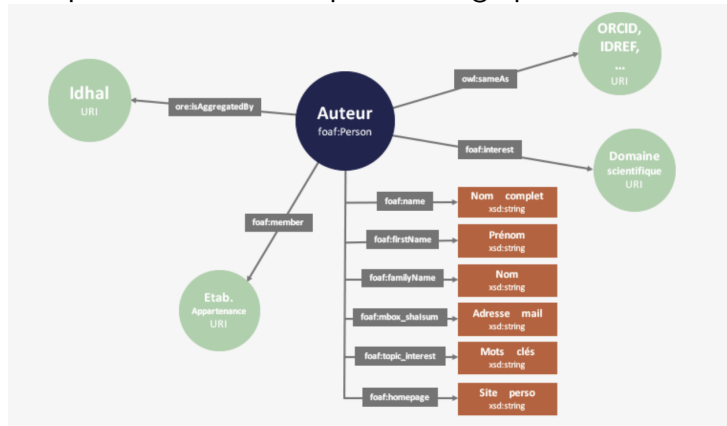
dataHAL utilise les vocabulaires et ontologies suivants :

- **FaBiO** (FRBR-aligned Bibliographic Ontology) : description d'entités publiées ou publiables
- **Bibo** (Bibliographic Ontology) : description de citations et de références bibliographiques
- **Dublin Core** : description de ressources physiques et numériques
- **FOAF** (Friend of a friend) : description de personnes et de leurs relations
- **SKOS** (Simple Knowledge Organization System) : description de systèmes d'organisation des connaissances



Exemple de graphe : Auteur

Chaque référentiel correspond à un graphe.



source : <https://data.hal.science/doc/schema> [consulté le 11 juin 2024]

Une première interrogation



```

select distinct ?s ?o
where {
  ?s a foaf:Person .
  ?s
  <http://www.openarchives.org/ore/terms/isAggregatedBy>
  ?o
}

```

Anatomie d'une interrogation

```
select distinct ?s ?o
where {
  ?s a foaf:Person .
  ?s
  <http://www.openarchives.org/ore/terms/isAggregatedBy>
  ?o
}
```

- **?s** et **?o** sont des variables
- On les appelle comme on veut mais elles sont le sujet et l'objet de la deuxième relation.
- **a** et

<http://www.openarchives.org/ore/terms/isAggregatedBy>
sont des prédicats.



Anatomie d'une interrogation

```
select distinct ?s ?o
where {
  ?s a foaf:Person .
  ?s
  <http://www.openarchives.org/ore/terms/isAggregatedBy>
  ?o
}
```

- **foaf** : peut-être défini comme
prefix foaf : <http://xmlns.com/foaf/0.1/>
- **a** est un raccourci standard pour **rdf:type**
- où **rdf** : peut être défini comme
prefix rdf : <http://www.w3.org/1999/02/22-rdf-syntax-ns#>



Interface d'interrogation en SPARQL

<http://sparql.archives-ouvertes.fr/sparql>

Virtuoso SPARQL Query Editor

[About](#) | [Namespace Prefixes](#) | [Inference rules](#) | [RDF views](#)

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?Concept where {[] a ?Concept} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format:

Execution timeout: milliseconds (values less than 1000 are ignored)

Options:

- Strict checking of void variables
- Log debug info at the end of output (has no effect on some queries and output formats)
- Generate SPARQL compilation report (instead of executing the query)

(The result can only be sent back to browser, not saved on the server, see [details](#).)

Run Query

Reset



Un exemple de post traitement

```
select distinct ?person ?prenom
where {
  ?person a foaf:Person .
  ?person foaf:familyName "Picard" .
  ?person foaf:firstName ?prenom .
}
order by ?prenom
```

API avancé

- 1 Introduction
- 2 Les facettes
- 3 Les autres référentiels
 - Les auteurs et leurs structures
 - Les portails et leurs types de documents
 - Domaines et journaux
 - Les financements
- 4 La base de données : le triplestore dataHAL
- 5 Pour finir...

La documentation

- La documentation de l'API de recherche HAL
<https://api.archives-ouvertes.fr/docs/search>
- La documentation sur les facettes de Apache Solr
<https://cwiki.apache.org/confluence/display/solr/Faceting>
- Les autres référentiels
<https://api.archives-ouvertes.fr/docs/ref>
- La documentation de dataHAL
<https://data.hal.science/doc/schema>

Connaissez-vous Bruno ?

- Vous n'aimez pas taper une longue requête dans un navigateur ?
- Vous ne savez pas comment conserver vos requêtes préférées ?
- Connaissez-vous Bruno ? (un petit outil discuté au GT API)

The screenshot shows the Bruno application interface. On the left, there's a sidebar with 'Collections' and 'HAL-LJK' selected. The main area shows a REST client configuration for a GET request to 'https://api.archives-ouvertes.fr/search/LJK/?q=instStructCountry_sus&fq=publicationDateY_i:2023&wt=xml'. Below the configuration is a table of parameters:

Name	Value	Actions
q	instStructCountry_sus	✓ 🗑️
fq	publicationDateY_i:2023	✓ 🗑️
wt	xml	✓ 🗑️

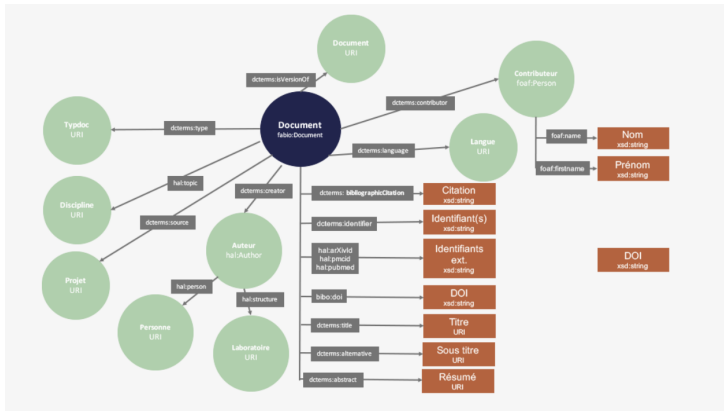
Below the table is a '+ Add Param' button. On the right, the 'Response' tab is active, showing an XML document with 16 lines of code. The response is an XML document with a root element 'response' containing a list of search results. The first result is for Jean-Guillaume Dumas, and the second is for Buy Nguyen, Pedram Akbarian, Trungtin Nguyen, and Nhat Ho.

```

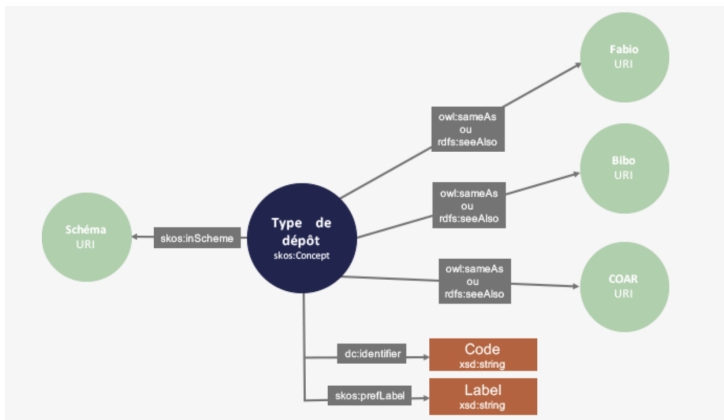
1 <?xml version="1.0" encoding="UTF-8"?>
2 <response>
3   <result name="response" numFound="35" start="0" maxScore
4     =1.64169" numFoundExact="true">
5     <doc>
6       <str name="docid">4026006</str>
7       <str name="label_s">Jean-Guillaume Dumas, Aude M
8         aignan, Clément Fernet, Daniel S. Roche. VESPO: Verified Eva
9         luation of Secret Polynomials: with application to low-stora
10        ge dynamic proofs of retrievability. Privacy Enhancing Techn
11        ologies Symposium, Jul 2023, Lausanne (CH), Switzerland. pp.
12        354--374. &amp;#x27E8;10.56553/gopets-2023-0085&amp;#x27E9;.
13        &amp;#x27E8;hal-03365854v5&amp;#x27E9;</str>
14       <str name="uri_s">https://hal.science/hal-033658
15        54v5</str>
16     </doc>
17     <doc>
18       <str name="docid">4256824</str>
19       <str name="label_s">Buy Nguyen, Pedram Akbarian,
20        Trungtin Nguyen, Nhat Ho. A General Theory for Softmax Gatin
21        g Multinomial Logistic Mixture of Experts. 2023. &amp;#x27E8;
22        ;hal-04256824&amp;#x27E9;</str>
23       <str name="uri_s">https://hal.science/hal-042568
24        24</str>
25     </doc>
26     <doc>
27       <str name="docid">3952063</str>
28       <str name="label_s">Anatoli B. Juditsky, Arkadi
29        Nemirovski, Michael Zibulevsky. Radiation design in computed
30        tomography via convex optimization. 2023. &amp;#x27E8;hal-03

```

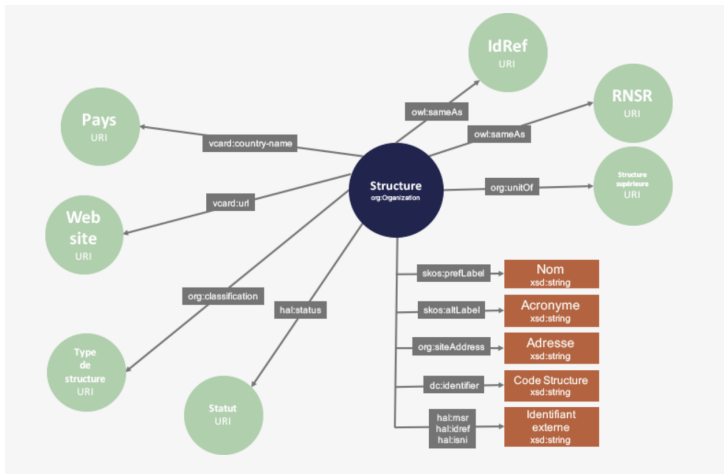

Les autres graphes : Documents



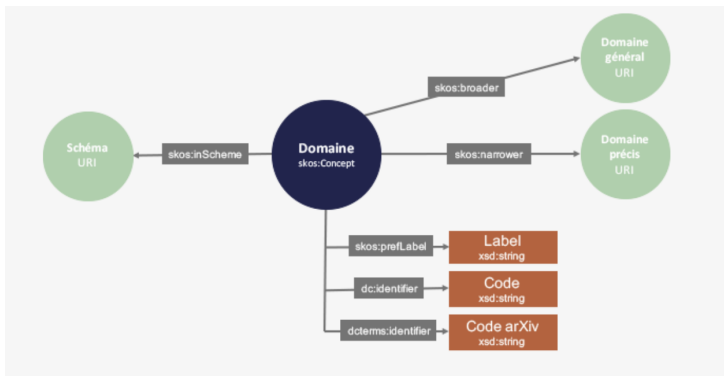
Les autres graphes : Types de dépôt



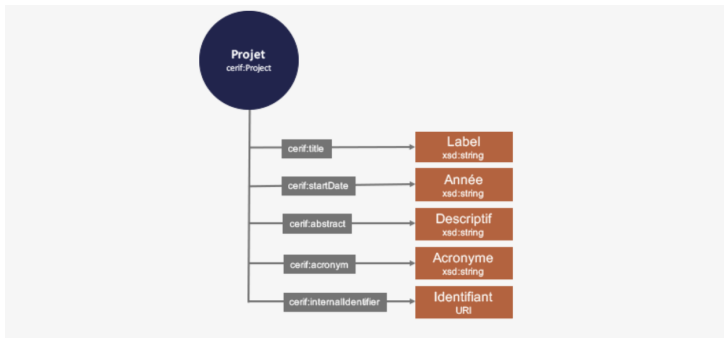
Les autres graphes : Structures



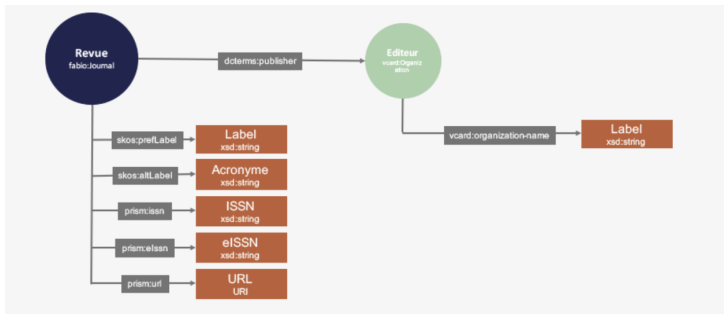
Les autres graphes : Domaines



Les autres graphes : Projets



Les autres graphes : Revues



Les autres graphes : IdHAL

