



HAL
open science

Utilisation des APIs de l'entrepôt Recherche Data Gouv

Camille Desiles, Virgile Jarrige, Damien Sans, Dimitri Szabo

► To cite this version:

Camille Desiles, Virgile Jarrige, Damien Sans, Dimitri Szabo. Utilisation des APIs de l'entrepôt Recherche Data Gouv. Printemps de la Donnée 2024, INRAE; Université Haute-Alsace; Université de Strasbourg; INSA; PNDB; AgroParisTech; Université de Lille; Sorbonne Université; Data Terra, 2024, Paris, France. hal-04702536

HAL Id: hal-04702536

<https://hal.inrae.fr/hal-04702536v1>

Submitted on 19 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

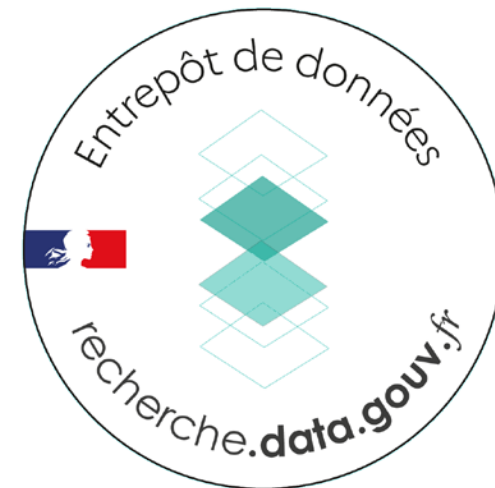


Distributed under a Creative Commons Attribution 4.0 International License

Bonjour et bienvenue !

Cette classe virtuelle démarrera
à xx h

Le support est téléchargeable ici :



Utilisation des APIs de l'entrepôt Recherche Data Gouv

Sessions 2024





⚠️⚠️⚠️ Prérequis ⚠️⚠️⚠️

- Avoir suivi la classe virtuelle **Découverte des APIs**
- Avoir installé le Client API **INSOMNIA**

⚠️⚠️⚠️ Prérequis ⚠️⚠️⚠️

Liste des exercices corrigés

0. CONFIGURATION DU CLIENT
1. LES APPELS
2. RECHERCHER
3. COLLECTIONS
4. GESTION COLLECTIONS
5. JEU DE DONNÉES
6. GESTION JEU DE DONNÉES
7. DÉPÔT DE FICHER(S)
8. TÉLÉCHARGEMENT DE FICHER(S)
9. MÉTRIQUES

Déroulé de la séance

I. INTRODUCTION

II. LES CONCEPTS CLÉS DE L'APPLICATION
DATAVERSE

III. LES POSSIBILITÉS DES APIS DE
RECHERCHE DATA GOUV

IV. UTILISATION PRATIQUE

Ce que cette formation ne couvre pas

- Introduction aux APIs : [voir formation « Découverte »](#)
- Utilisation de l'entrepôt lui-même : [voir formations « Déposer »](#) et [« Administrer »](#)
- Les outils complémentaires (Widget, DVUploader)
- Utilisation des APIs dans un langage de script (ex. R, Python)

Dataverse, moteur de l'entrepôt et du catalogue

116 Installations



Source : <https://dataverse.org/> - 11/04/2024

The
Dataverse
Project 



Projet Open Source



Conçu par Harvard



34 Institutions (18 pays)

dans la gouvernance

(dont INRAE et le CNRS)



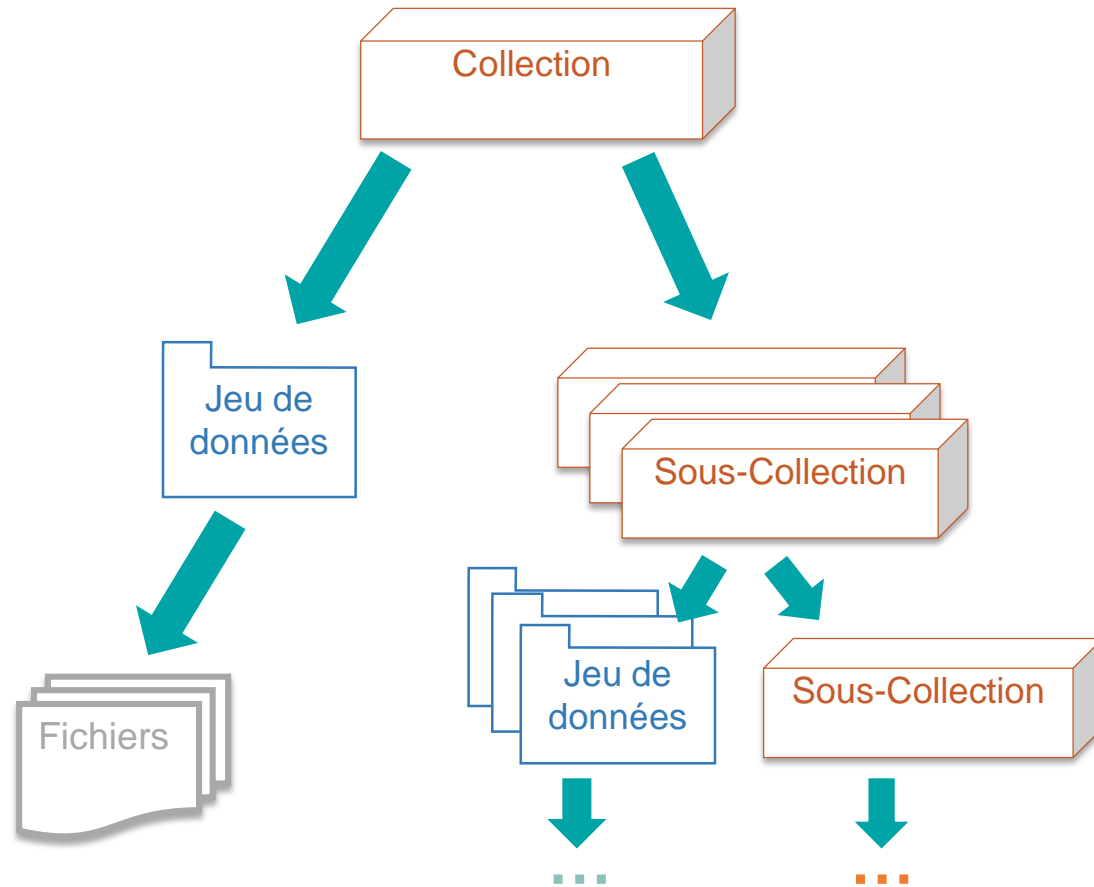
RÉPUBLIQUE
FRANÇAISE recherche.data.gouv.fr

*Liberté
Égalité
Fraternité*

RAPPELS : LES CONCEPTS CLÉS DE L'ENTREPÔT






Les éléments



- **collection (dataverse)** : de **jeux de données** et/ou de **sous-collections** ;
- **jeu de données (dataset)** : rassemble les fichiers et les métadonnées ;
- **fichier (file)** : Un fichier de données ou de documentation (ex. README).

Rappel : les métadonnées

Identifiant pérenne ?	doi:10.57745/PM55G5
Date de publication ?	2023-01-26
Titre ?	One-step synthesis of spin crossover nanoparticles using flow chemistry and supercritical CO2
Point de contact ?	Utiliser le bouton de courriel ci-dessus pour joindre la personne-contact. TOULIN, Stéphane (CNRS)
Auteur ?	Daro, Nathalie (Univ. Bordeaux, CNRS, Bordeaux INP, ICMCB, UMR 5026, F-33600 Pessac, France) - ORCID: 0000-0002-3707-9686 Chastanet, Guillaume (Univ. Bordeaux, CNRS, Bordeaux INP, ICMCB, UMR 5026, F-33600 Pessac, France) - ORCID: 0000-0001-6829-4066 Marre, Samuel (Univ. Bordeaux, CNRS, Bordeaux INP, ICMCB, UMR 5026, F-33600 Pessac, France) - ORCID: 0000-0001-8889-187X Aymonier, Cyril (Univ. Bordeaux, CNRS, Bordeaux INP, ICMCB, UMR 5026, F-33600 Pessac, France) - ORCID: 0000-0003-1775-0716 Vaudel, Tony (Univ. Bordeaux, CNRS, Bordeaux INP, ICMCB, UMR 5026, F-33600 Pessac, France) Afindouli, Luc (Univ. Bordeaux, CNRS, Bordeaux INP, ICMCB, UMR 5026, F-33600 Pessac, France)
Contributeur ?	Data Collector : TOULIN, Stéphane (Univ. Bordeaux, CNRS, Bordeaux INP, ICMCB, UMR 5026, F-33600 Pessac, France) - ORCID: 0000-0002-2087-7403
Producteur ?	Centre National de la Recherche Scientifique (CNRS) https://www.cnrs.fr/  Université de Bordeaux (UBX) https://www.u-bordeaux.fr/  Bordeaux INP https://www.bordeaux-inp.fr/ 
Localisation de la production ?	Pessac
Distributeur ?	Entrepôt-Catalogue Recherche Data Gouv
Description ?	Data set and images from - Publication : Daro N., Vaudel T., Afindouli L., Marre S., Aymonier C. and Chastanet G. One-step synthesis of spin crossover nanoparticles using flow chemistry and supercritical CO2. Chemistry - European Journal 2020 vol.26, n°69, p.16286-16290. DOI : 10.1002/chem.202002322 / HAL : https://hal.science/hal-02945200

- Données décrivant les données
- Facilite la découverte
- Pour **comprendre** les données et les conditions de production

Blocs de métadonnées

- Métadonnées générales :
 - 6 sont **obligatoires** : **Titre, Auteur Nom, Point de contact Courriel, Description Texte, Sujet, Type de données**
 - liens avec d'autres objets (**Lien vers les données, Autre identifiant, Publication associée, Jeu de données associé**)
 - etc.
- Métadonnées spécifiques :
 - *Métadonnées géospatiales*
 - *Métadonnées de sciences humaines et sociales*
 - *Métadonnées de science de la vie*
 - *Métadonnées de revue*
 - *Métadonnées de ressources sémantiques*
 - *Métadonnées d'astronomie et astrophysique*
 - *Métadonnées de workflow de calcul*

Conformes aux standards
DDI , Dublin Core,
DataCite, ISA-TAB

Comment tester ?

- Pour les tests et d'autant plus en écriture :
Utiliser le bac à sable !
- Pour tester la publication, utiliser si possible toujours le même jeu de données
- Si possible, supprimer les tests au fur et à mesure





RÉPUBLIQUE
FRANÇAISE recherche.data.gouv.fr

*Liberté
Égalité
Fraternité*

RAPPELS SUR LES APIS



Un peu de vocabulaire

https://entrepot.recherche.data.gouv.fr/api/v1/search?q=*fq=publicationStatus:Published



1. Protocole : par quoi on échange
2. Serveur : avec qui/quoi on échange
3. Version : permet d'utiliser l'ancienne version d'une API si elle change
4. Endpoint : point d'entrée, quelle commande on effectue
5. Paramètres : avec quelles spécificités

Les méthodes

- Verbes HTTP correspondant à 4 opérations (CRUD)

Opération	Traduction	Verbe
Create	Créer	POST
Read	Afficher	GET
Update	Mettre à jour	PUT
Delete	Supprimer	DELETE

CORS et APIs

**Seules certaines commandes ont CORS activé,
ces commandes sont toutes des lectures (GET),
Il est alors possible de les lancer dans un navigateur ou
script web.**

Dans le cas d'une appli web, pour les autres commandes,
il faut passer par un appel côté serveur.

La documentation des APIs

Guide Dataverse :

<https://guides.dataverse.org/en/5.14/api/>

⚠ bien préciser la version de Dataverse utilisée ⚠

Les différentes APIs y sont documentées séparément

Les bibliothèques

Des bibliothèques dans différents langages de programmation pour construire des programmes.

- R : atom4R, dataverse
- Python : pyDataverse
- Java
- Javascript
- Ruby

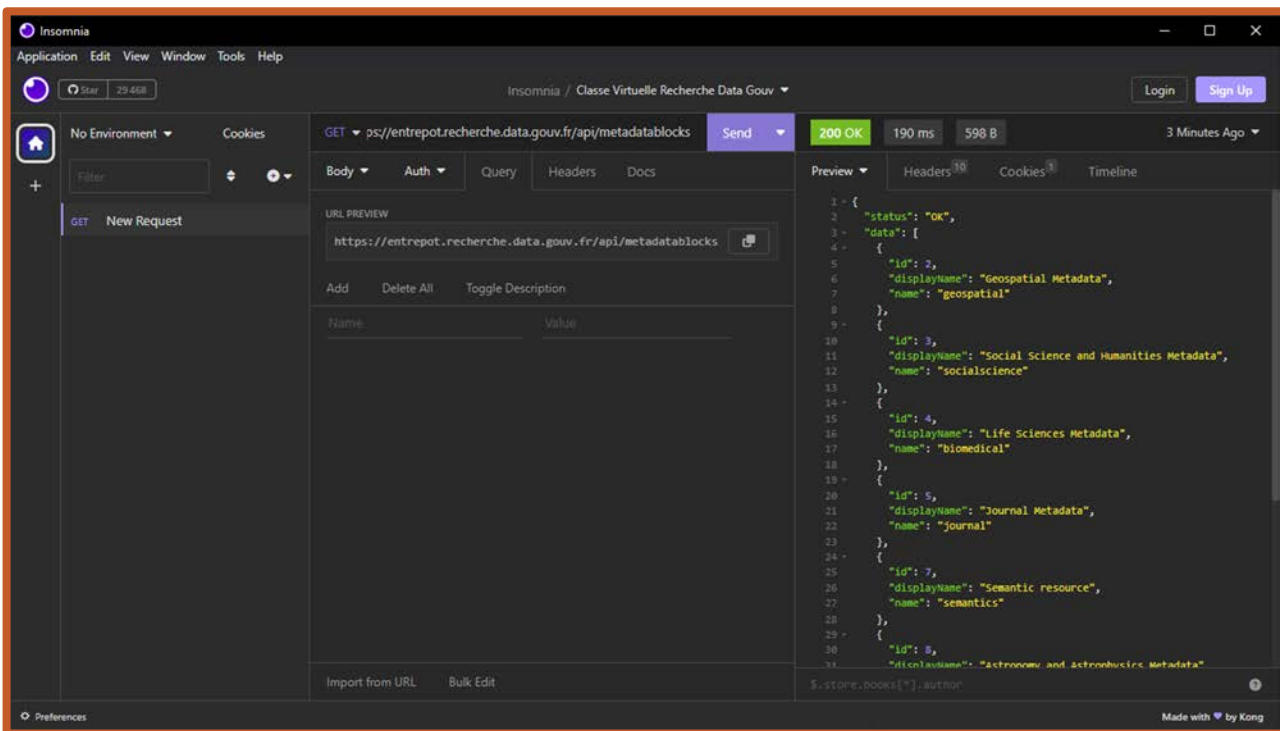
Si le client n'existe/ne convient pas, utiliser des bibliothèques comme « curl » !



Le Plus : intégration

✓ Correspond le plus souvent à l'usage final

Les clients API



Permet des requêtes plus modulaires qu'avec la console

- Par exemple :
<https://insomnia.rest/download>
<https://www.postman.com/>



Le Plus : User Friendly

- ✓ Distinction et configuration des variables, environnements, sauvegarde...



Le Moins : automatisation

- Outil séparé d'une potentielle chaîne de traitement

Convention de nommage

Une accolade = Variable que vous devez modifier

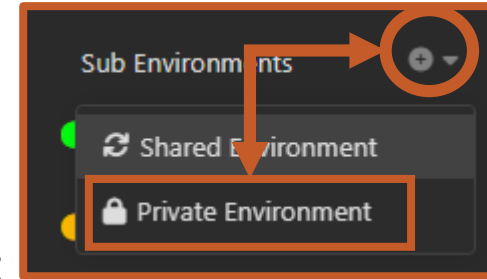
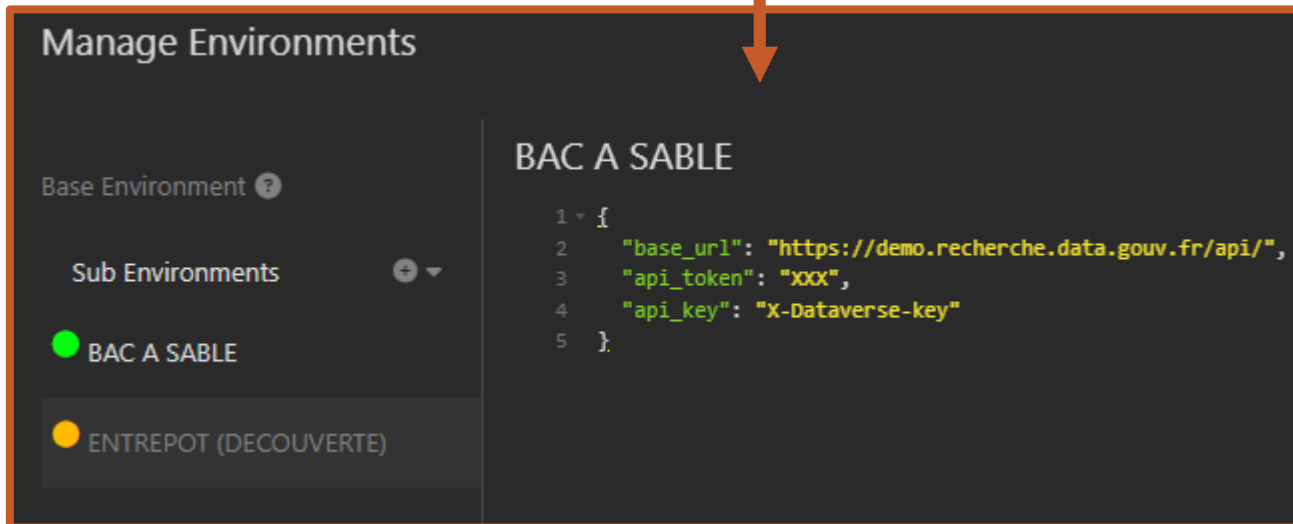
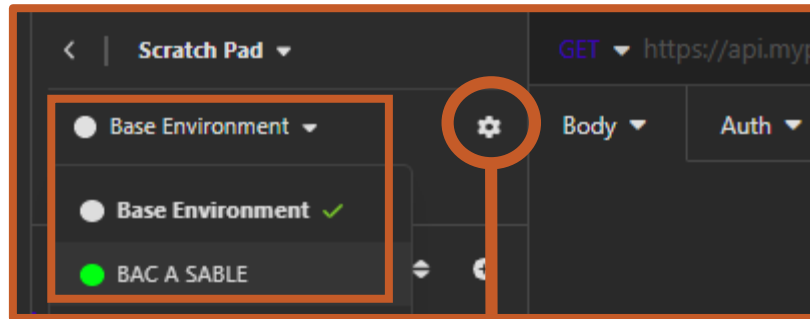
- Exemple : **{alias_de_la_collection}** = l'identifiant de la collection sur laquelle vous souhaitez interagir

Deux accolades = Variable supportée par Insomnia

- Exemple : **{{ _.base_url }}** = variable qui sera stockée dans Insomnia.

Il est possible de copier/coller ces variables

Préparer son projet dans Insomnia



Sub Environments :

- Créer un environnement () :
« BAC A SABLE »
- Créer un environnement
« ENTREPOT »

Initialiser la variable
`{{ _.base_url }}`
pour chaque environnement

- <https://demo.recherche.data.gouv.fr/api/>
- <https://entrepot.recherche.data.gouv.f/api/>

Exercice 1 : Votre premier appel API vers l'entrepôt

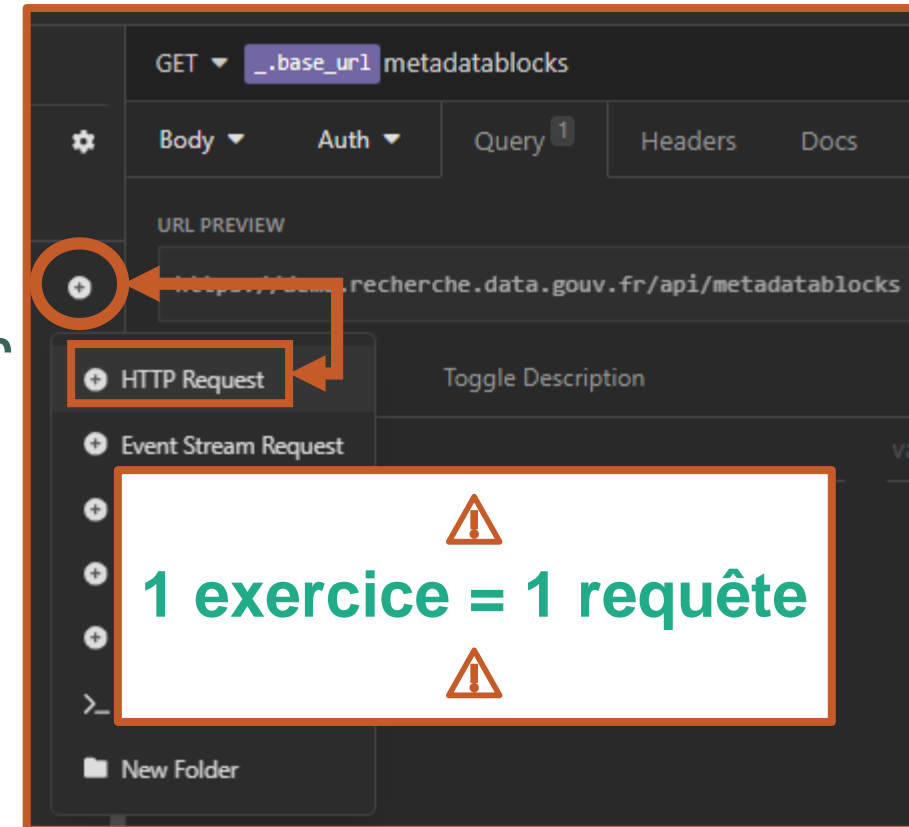
1.a : écrire un appel simple

serveur : bac-à-sable de l'entrepôt

`https://demo.recherche.data.gouv.fr`

verbe : GET

endpoint : `metadatablocks`



Que voyez-vous ?

Exercice 1 : Votre premier appel API vers l'entrepôt

1.a Solution :

`https://demo.recherche.data.gouv.fr/api/metadatablocks`

Ou

`{{ _.base_url }}metadatablocks`

Les blocs de métadonnées s'affichent : leur **identifiant**, leur **nom affiché** et leur nom véritable.

Format d'affichage : `{j s o n}`

Exercice 1 : Quelques commandes supplémentaires

1.b Obtenir le schéma du bloc de métadonnées Générales en adaptant la commande :

`https://demo.recherche.data.gouv.fr/api/metadatablocks/{nom du bloc}`

1.c Réaliser l'appel avec un autre bloc (par exemple « Life Sciences »)

Rappel : la liste des blocs disponibles s'affiche avec l'appel à « metadatablocks »

Testez sur les deux plateformes « BAC A SABLE » et « ENTREPOT » !

Exercice 1 : Quelques commandes supplémentaires

1.b Solution :

```
{{ _.base_url }}metadatablocks/citation
```

1.c Solution :

```
{{ _.base_url }}metadatablocks/{bloc}
```

Exemple, pour « Life Sciences » :

```
{{ _.base_url }}metadatablocks/biomedical
```

NB : ces appels peuvent se faire directement dans un navigateur web

Les différentes APIs

- Search : Recherche
- Native : Fonctionnalités principales
- ⚠ Direct Upload/Replace : Téléversement des fichiers
- Data Access : Téléchargement de fichiers
- Metrics : Statistiques d'utilisation
- SWORD : Dépôt de fichiers et jeux de données
- MyData ⚠ non documentée

Rechercher des jeux de données

API = Search ou MyData

Cas d'usage :

- Recherche (**avancée**), avec ou sans **authentification** (Search)
- Récupérer les données d'un utilisateur (MyData)

NB :

- Pour rechercher sur des métadonnées spécifiques
- Pas toutes les métadonnées renvoyées (utiliser l'API Native)
- ⚠️ Pagination ⚠️ : résultats limités par page

Exercice 2 : Rechercher des données

2.a Tester une nouvelle recherche :

```
{{ _.base_url }}search
```

(utiliser le paramètre : q)

2.b Filtrer sur un champ de métadonnées :

```
{{ _.base_url }}search?q={valeur}&fq={champ}:{valeur}
```

Exercice 2 : Rechercher des données

2.c Limiter le nombre de résultats par page :

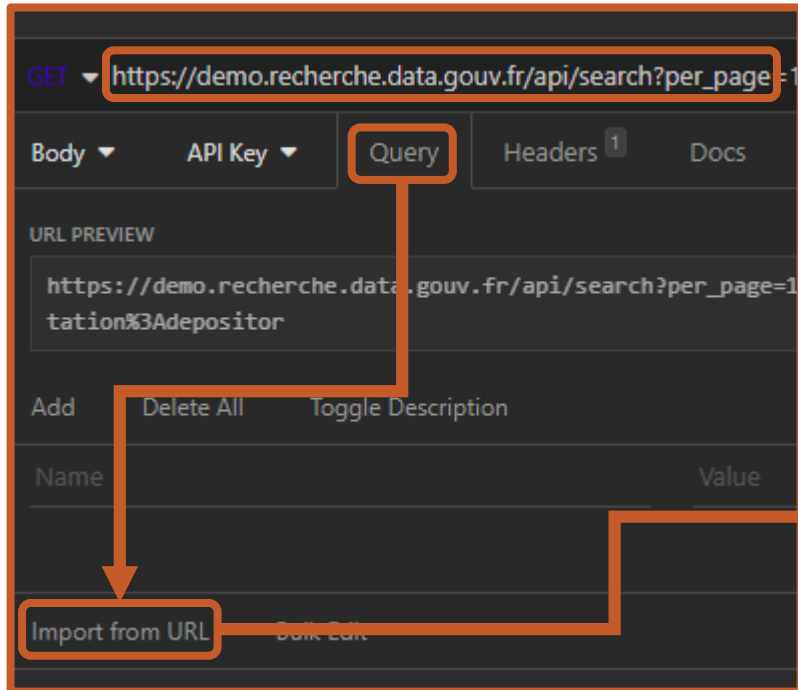
```
{{ _.base_url }}  
search?per_page={##}&q={valeur}&fq={champ}:{valeur}
```

2.d Récupérer un champ additionnel de métadonnées :

```
{{ _.base_url }}  
search?per_page=1&q=tree&fq=title:tree  
&metadata_fields={bloc}:{champ}
```

Bonne pratique avec Insomnia

Transformer les paramètres pour les éditer/(dés)activer facilement :
Onglet « Query » —> Import from URL



GET `https://demo.recherche.data.gouv.fr/api/search?per_page=1&q=*&type=dataset&metadata_fields=citation%3Adepositor`

Body API Key **Query** Headers Docs

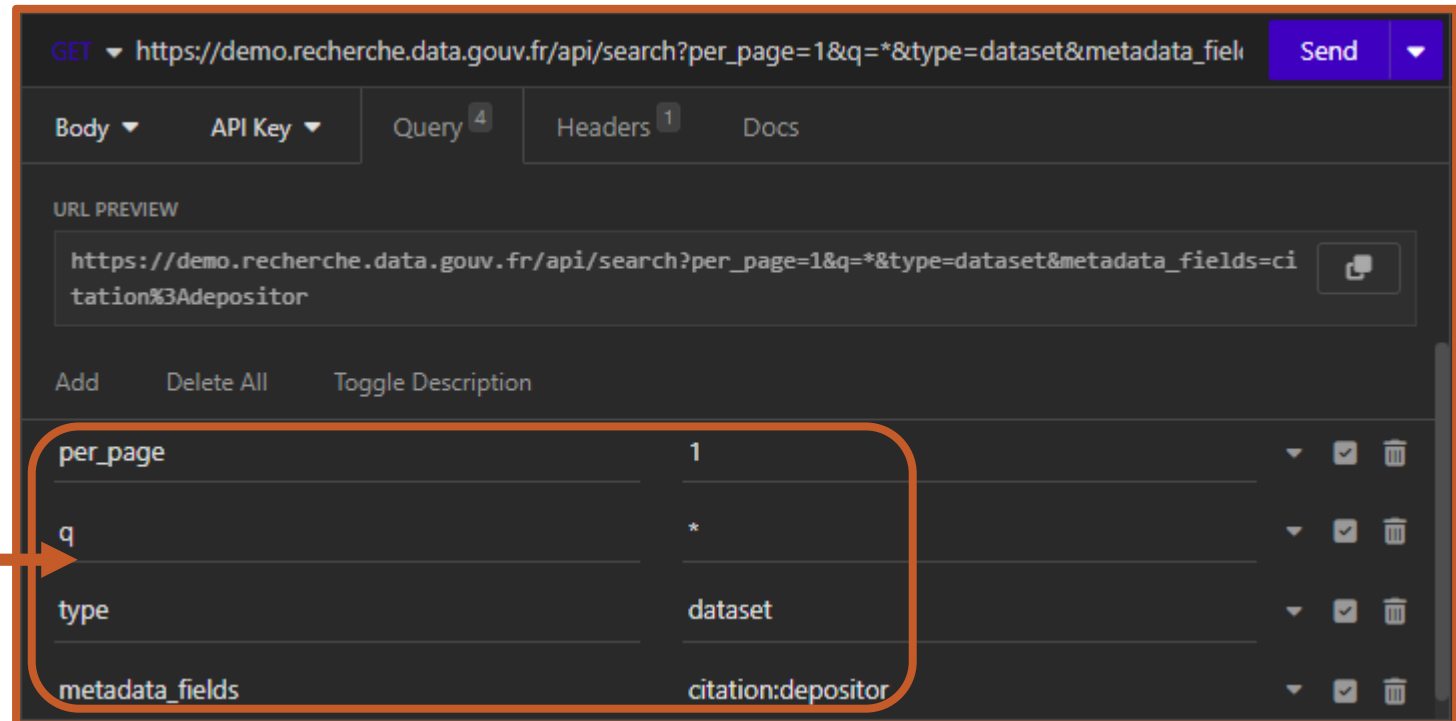
URL PREVIEW

```
https://demo.recherche.data.gouv.fr/api/search?per_page=1&q=*&type=dataset&metadata_fields=citation%3Adepositor
```

Add Delete All Toggle Description

Name	Value
per_page	1
q	*
type	dataset
metadata_fields	citation:depositor

Import from URL Bulk Edit



GET `https://demo.recherche.data.gouv.fr/api/search?per_page=1&q=*&type=dataset&metadata_fields=citation%3Adepositor` Send

Body API Key **Query** Headers Docs

URL PREVIEW

```
https://demo.recherche.data.gouv.fr/api/search?per_page=1&q=*&type=dataset&metadata_fields=citation%3Adepositor
```

Add Delete All Toggle Description

per_page	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
q	*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
type	dataset	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
metadata_fields	citation:depositor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Exercice 2 : Correction

2.a Tester une nouvelle recherche :

```
{{ _.base_url }}search?q=classe
```

2.b Filtrer sur un champ de métadonnées :

```
{{ _.base_url }}search?q=classe  
&fq=title:classe
```

Exercice 2 : Correction

2.c Limiter le nombre de résultats par page :

```
{{ _.base_url }}search?per_page=1  
&q=classe  
&fq=title:classe
```

2.d Récupérer un champ additionnel de métadonnées :

```
{{ _.base_url }}search?per_page=1  
&q=*  
&type=dataset  
&metadata_fields=citation:depositor
```




RÉPUBLIQUE
FRANÇAISE

*Liberté
Égalité
Fraternité*

recherche.data.gouv.fr

L'AUTHENTIFICATION



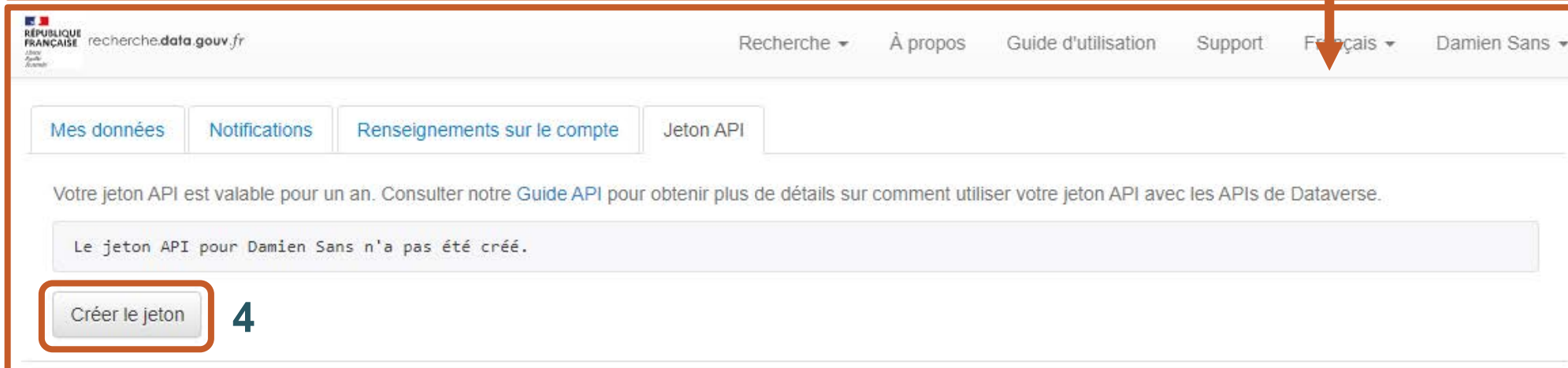
@RechercheDataGv

S'authentifier : le jeton API (1/3 l'obtenir)

- Sur Recherche Data Gouv, 1 se connecter puis :

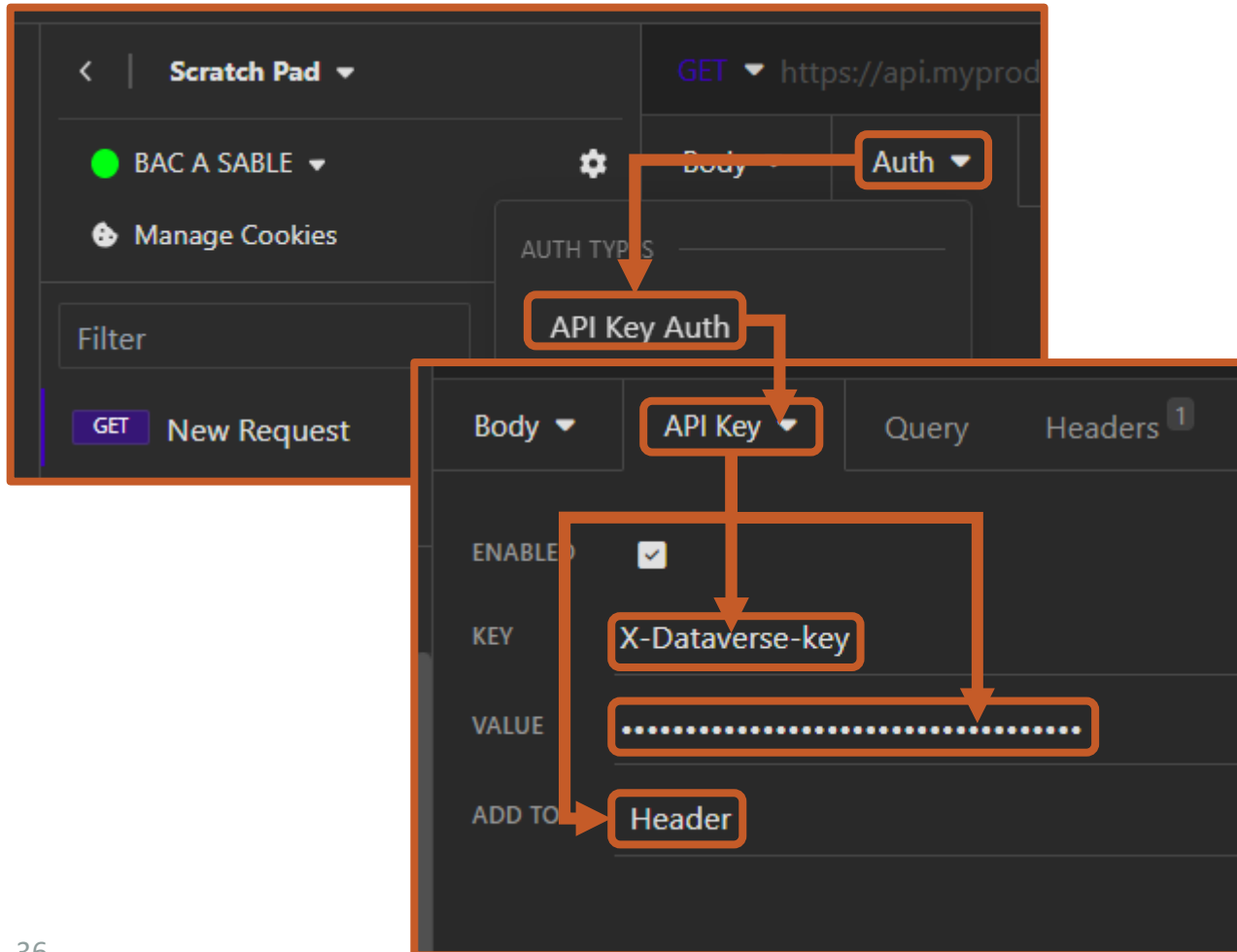


The screenshot shows the top navigation bar of the Recherche Data Gouv website. The user is logged in as 'Damien Sans'. A dropdown menu is open, showing options: 'Mes données', 'Notifications', 'Renseignements sur le compte', 'Jeton API', and 'Se déconnecter'. The 'Jeton API' option is highlighted with a red box and a '3' next to it. A red arrow points from the 'Damien Sans' dropdown to the 'Jeton API' option.



The screenshot shows the 'Jeton API' page. The text reads: 'Votre jeton API est valable pour un an. Consulter notre [Guide API](#) pour obtenir plus de détails sur comment utiliser votre jeton API avec les APIs de Dataverse.' Below this, a message states: 'Le jeton API pour Damien Sans n'a pas été créé.' At the bottom, there is a button labeled 'Créer le jeton' with a '4' next to it, highlighted by a red box.

S'authentifier : le jeton API (2/3 l'utiliser)



Une fois créé, il faut le copier (bouton « Copier dans le presse-papier »).

Le mode de renseignement du jeton dépend ensuite de l'outil utilisé.

- Exemple dans cURL :
en-tête/header* « X-Dataverse-key » :

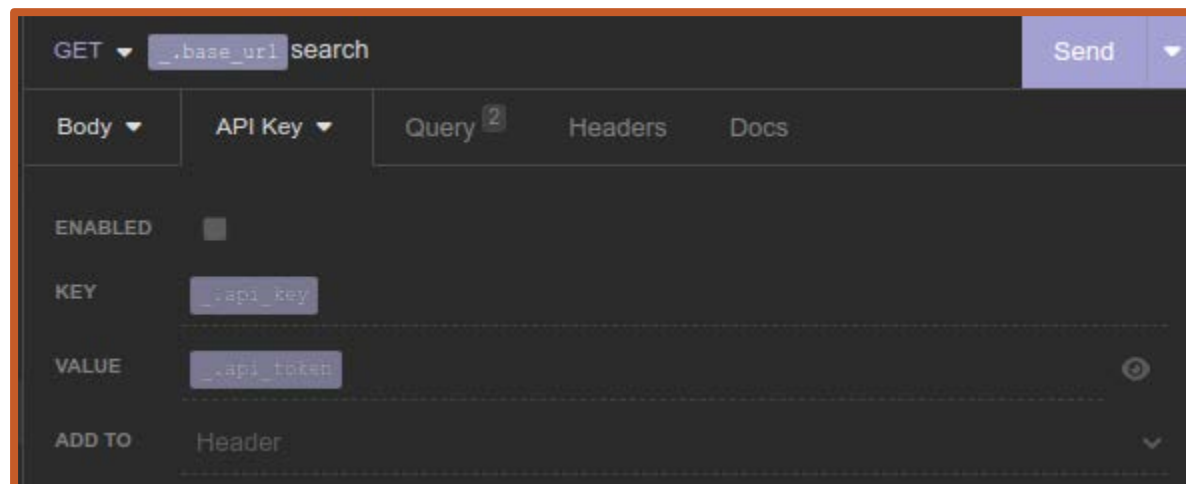
```
curl -H X-Dataverse-key :{jeton}  
-X GET {ma commande}
```

*éléments permettant d'envoyer au serveur des informations supplémentaires.

S'authentifier : Bonne pratique Insomnia

Ajouter les variables d'environnement `{{ _.api_token }}` et `{{ _.api_key }}` pour s'authentifier :

```
"api_token": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",  
"api_key": "X-Dataverse-key"
```



Exercice 2bis : Rechercher des données en étant authentifié

2bis.a : Créer un appel avec l'endpoint « search?q=*&per_page=1 » sur le bac à sable
L'exécuter et noter la valeur de « total_count »

2bis.b : Créer un jeu de données sur le bac à sable depuis l'interface web, dans la collection « Travaux pratiques » – Ne pas le publier !

2bis.c : Ajouter l'authentification pour l'appel que vous avez créé (onglet API Key) et noter la différence pour la valeur de « total_count »

Correction 2bis : Rechercher des données en étant authentifié

2bis.a :

```
{{ _.base_url }}searchsearch?per_page=1&q=*
```

2bis.c :

La valeur de `total_count` est plus élevée lorsque l'on est identifié.
Pourquoi ? Une fois authentifié, les jeux de données en version provisoire auquel notre compte à accès sont ajoutés.

S'authentifier : le jeton API (3/3 le renouveler)

⚠ Le jeton a une **durée de validité d'un an**,
Au delà le jeton est révoqué et il faut en créer un autre.

- Interface web : Bouton « Créer de nouveau le jeton » / « Recreate token »

- API : S'authentifier avec `{{ _api_key }}` et `{{ _api_token }}` en Header

Renouvellement :

Verbe : POST

`{{ _base_url }}users/token/recreate`

Information sur le jeton :

Verbe : GET

`{{ _base_url }}users/token`

Gérer une collection

API = Native

Cas d'usage :

- Créer, modifier et publier une collection
- Gérer les informations, droits et autres possibilités du portail
- Supprimer la collection
- Récupérer les données des Registres des visiteurs

Point d'attention à l'usage :

- Authentification généralement nécessaire
- CORS

Exercice 3 : Créer une sous-collection

3.a Trouver la section « Create a Dataverse Collection » dans « Native API » de la documentation et **télécharger le fichier json** nécessaire à la création d'une collection via l'API <https://guides.dataverse.org/en/5.14/api/native-api.html>

3.b Créer un appel avec l'endpoint « dataverses/{**alias-de-la-collection-travaux-pratiques**} » et sélectionner le verbe « POST »

3.c Copier/coller le contenu du fichier json dans l'onglet **Body** > **JSON** d'Insomnia puis modifiez le pour qu'il corresponde à votre ***collection de rêve***

3.d Créer votre collection dans la collection « Travaux Pratiques » du bac à sable

Correction 3 : Créer une sous-collection

3.a <https://guides.dataverse.org//en/latest/api/native-api.html#create-dataverse-api>

3.b

3.c

3.d

```
POST  Send
JSON API Key Query Headers Docs
1 {
2   "name": "test with insomnia",
3   "alias": "twi",
4   "dataverseContacts": [
5     {
6       "contactEmail": "virgile@unistra.fr"
7     },
8     {
9       "contactEmail": "damien.sans@inrae.fr"
10    }
11  ],
12  "affiliation": "Université de Strasbourg, INRAE",
13  "description": "Test de création de collection avec insomnia",
14  "dataverseType": "UNCATEGORIZED"
15 }
```

Ne pas oublier
l'onglet
API Key
pour faire une
requête
authenticée !

Exercice 4 : Gérer une sous-collection

4.a Désactivez le bloc de métadonnées « journal » sur votre collection précédemment créée

<https://guides.dataverse.org/en/5.14/api/native-api.html#define-metadata-blocks-for-a-dataverse-collection>

4.b Assigner le rôle de curateur sur la collection précédemment créée à vous-même

<https://guides.dataverse.org/en/5.14/api/native-api.html#assign-a-new-role-on-a-dataverse-collection>

4.c Publier la collection

<https://guides.dataverse.org/en/5.14/api/native-api.html#publish-a-dataverse-collection>

Correction 4 : Gérer une sous-collection

4.a

```
POST .base_url dataverses/pbu/metadatablocks 200 OK 89.5 ms 78 B 7 Minutes Ago
```

```
JSON API Key Query Headers 1 Docs Preview Headers 10 Cookies Timeline
```

```
1 [
2   "citation",
3   "geospatial",
4   "socialscience",
5   "astrophysics",
6   "biomedical",
7 ]
```

supprimer la virgule !

```
1 {
2   "status": "OK",
3   "data": {
4     "message": "Metadata blocks of dataverse pbu
5     updated."
6   }
7 }
```

Pour vérifier :

`{{ .base_url }}dataverses/{alias_de_la_collection}/metadatablocks`

Verbe GET

Correction 4 : Gérer une sous-collection

4.b

The screenshot shows a REST client interface. The request is a POST to `dataverses/pbu/assignments/` with a body of `{ "assignee": "@vjarrige", "role": "curator" }`. The response is a 200 OK with a body of `{ "status": "OK", "data": { "assignee": "@vjarrige", "roleId": 7, "_roleAlias": "curator" } }`.

Pour vérifier :

```
{{ _.base_url }}dataverses/{alias_de_la_collection}/assignments  
Verbe GET (sans JSON)
```

4.c

⚠ Pour récupérer un votre UserId ⚠
`{{ _.base_url }}users/:me`

```
{{ _.base_url }}dataverses/{alias_de_la_collection}/actions/:publish
```

Gérer un jeu de données

API = Native ou SWORD

Cas d'usage

- Créer, modifier et publier un jeu de données
- Gérer les métadonnées, droits et autres possibilités du portail
- Supprimer le jeu de données

Point d'attention à l'usage :

- Authentification généralement nécessaire
- CORS

Point d'attention : Les métadonnées spécifiques

Appliquer le schéma pour déposer ou importer des jeux de données

- Exemple **complet (toutes métadonnées remplies)** :

<https://nextcloud.inrae.fr/s/BDf4sAPLe4edN3Z>

- Exemple **limité** :

<https://nextcloud.inrae.fr/s/a9pcG6kGKBGWexT>

⚠ Attention aux évolutions ⚠
(ajout de nouvelles métadonnées) ⚠

Exercice 5 : Créer un jeu de données

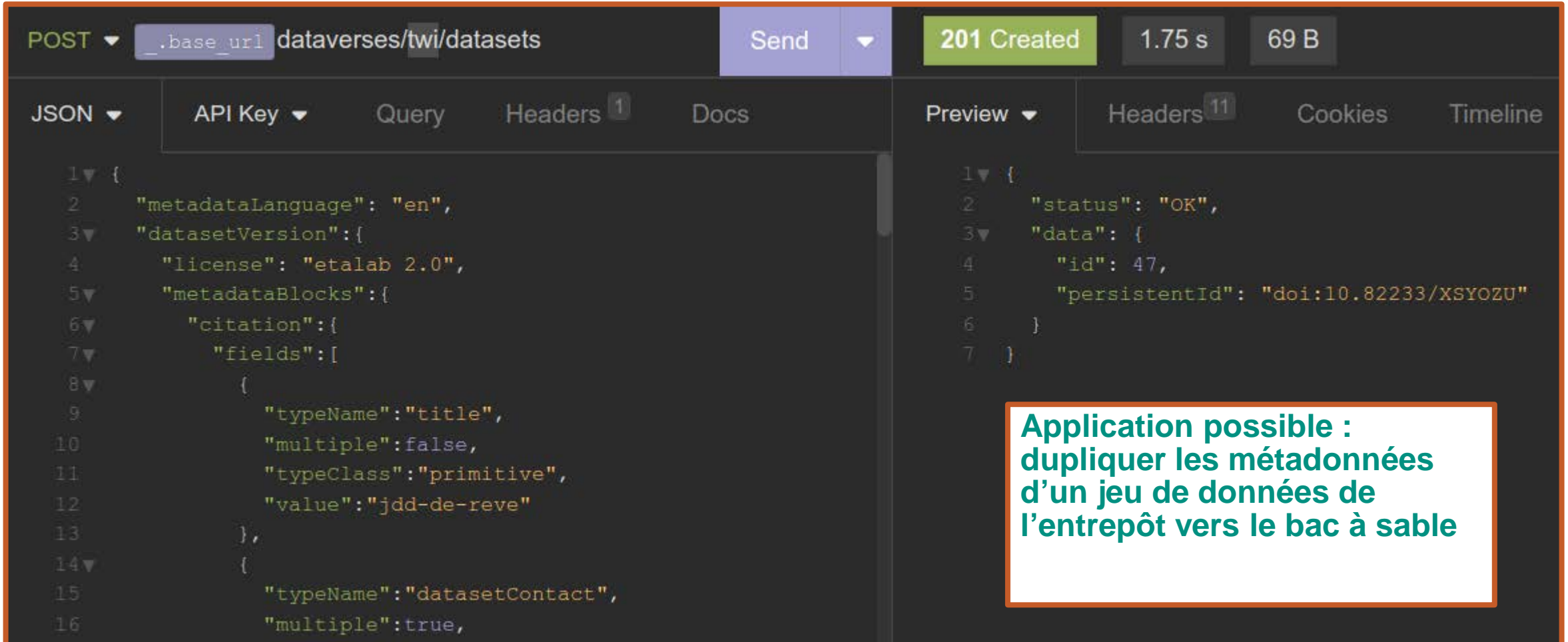
5.a. Créer un appel avec l'endpoint `dataverses/{alias_dataverse}/datasets`
`{alias_dataverse}` étant l'identifiant de votre ***collection de rêve***

5.b : Télécharger le fichier json [nouveau_jdd.json](https://nextcloud.inrae.fr/s/5XLRib2EwXFGPZS)
(<https://nextcloud.inrae.fr/s/5XLRib2EwXFGPZS>)

5.c : Copier/coller le contenu du fichier json de métadonnées dans :
Block > JSON », **modifier l'email du contact, a minima**, et exécuter l'appel.

⚠ Ne pas oublier l'authentification

Correction 5 : Créer un jeu de données



POST Send 201 Created 1.75 s 69 B

JSON API Key Query Headers 1 Docs Preview Headers 11 Cookies Timeline

```
1 {
2   "metadataLanguage": "en",
3   "datasetVersion": {
4     "license": "etalab 2.0",
5     "metadataBlocks": {
6       "citation": {
7         "fields": [
8           {
9             "typeName": "title",
10            "multiple": false,
11            "typeClass": "primitive",
12            "value": "jdd-de-reve"
13          },
14          {
15            "typeName": "datasetContact",
16            "multiple": true,
```

1 {
2 "status": "OK",
3 "data": {
4 "id": 47,
5 "persistentId": "doi:10.82233/XSYOZU"
6 }
7 }

**Application possible :
dupliquer les métadonnées
d'un jeu de données de
l'entrepôt vers le bac à sable**

NOTE : Dupliquer un jeu de données

- Pour dupliquer les métadonnées d'un jeu de données de l'entrepôt vers le bac à sable, il faut avoir les droits suffisants dessus pour récupérer les métadonnées cachées (ex. adresse mail contact, etc.)

Pour récupérer les métadonnées il faut les lister :

```
{{ _.base_url }}datasets/:persistentId?persistentId=doi: {DOI_du_JDD}
```

Et les injecter (copier/coller) dans un nouveau jeu de données dans la collection souhaitée, de la même manière que dans l'exercice 5.

⚠ Attention, les fichiers ne seront pas transférés.

Exercice 6 : Gérer un jeu de données

6.a Publier le jeu de données (version majeur)

<https://guides.dataverse.org/en/5.14/api/native-api.html#publish-a-dataset>

6.b Modifier des métadonnées du jeu de données créé précédemment

<https://guides.dataverse.org/en/5.14/api/native-api.html#edit-dataset-metadata>
avec le fichier d'exemple :

<https://guides.dataverse.org/en/5.14/downloads/cb0fae993277527ea354dc15bc5db864/dataset-edit-metadata-sample.json>

6.c Publier à nouveau le jeu de données (version majeur)

6.d Lister les versions du jeu de données

```
{{ base_url }}datasets/:persistentId/versions?persistentId=doi:{DOI_du_JDD}
```

Correction 6 : Gérer un jeu de données

6.a,c :

```
{{ _base_url }}datasets/:persistentId/actions/:publish?  
persistentId=doi:{DOI_du_JDD}  
&type=major
```

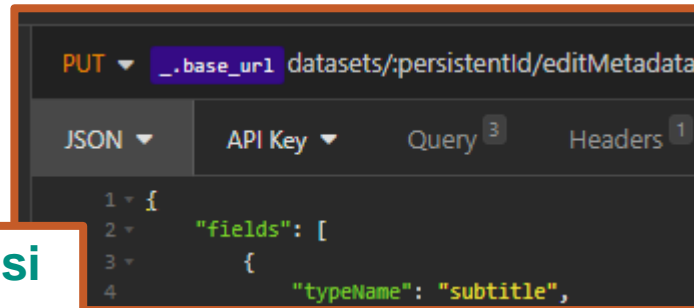
Verbe : POST

6.b :

```
{{ _base_url }}datasets/:persistentId/editMetadata?  
persistentId=doi:{DOI_du_JDD}  
&replace=true
```

Verbe : PUT

Il faut ajouter `replace=true` si on souhaite modifier une valeur



```
PUT _base_url datasets/:persistentId/editMetadata  
JSON API Key Query 3 Headers 1  
1 {  
2   "fields": [  
3     {  
4       "typeName": "subtitle",
```

Gestion des fichiers

API = Native ou Data Access

Cas d'usage

- Créer, remplacer, modifier et supprimer des fichiers
- Administrer (et demander) l'accès
- Télécharger des fichiers ou certaines de leurs variables

Point d'attention à l'usage :

- Authentification généralement nécessaire
- CORS (voir [diapo 46](#))

Exercice 7 : Déposer des fichiers

7.a Déposer un fichier texte depuis votre machine dans votre jeu de données

Verbe : POST

Endpoint : `datasets/:persistentId/add?persistentId=doi:{DOI_du_JDD}`

Dans Body > Multipart Form, ajouter deux paramètres (bouton « Add ») :

- name : file
value : choisir **file** que l'on trouve sous la petite flèche (à droite)
- name : laisser vide
value : laisser vide

Bonus : Métadonnées de fichier

Value :

```
jsonData={"description":"MyDescription.", "directoryLabel": "data/dir1", "categories":["Data"], "restrict":"false", "tabIngest":"false"}
```

Exercice 7bis : Déposer des fichiers

7.b Rendre le fichier accessible en publiant le jeu de données

7.c Lister les fichiers du jeu de données

```
{{ _.base_url }}datasets/:persistentId/versions/{n°_de_version}/files?  
persistentId={DOI_du_JDD}
```

Correction 7 : Déposer des fichiers

7.a

```
{{ _.base_url }}datasets/:persistentId/add?  
persistentId=doi:{DOI_du_JDD}
```

Verbe : POST

7.b :

```
{{ _.base_url }}datasets/:persistentId/actions/:publish?  
persistentId=doi:{DOI_du_JDD}
```

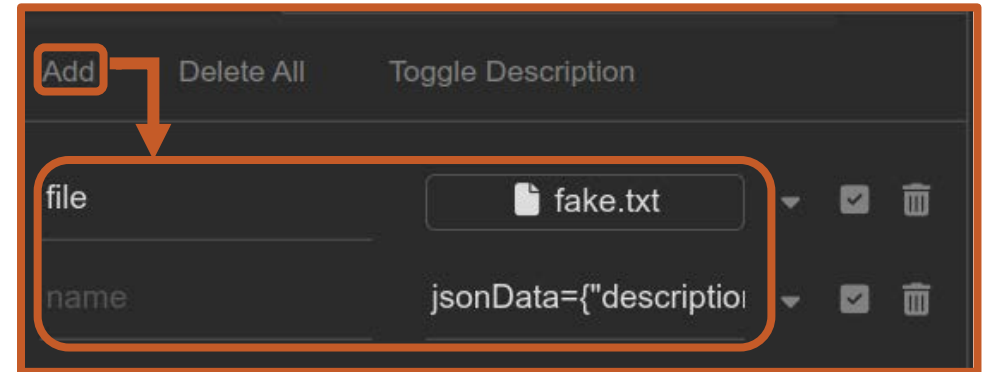
&type=major

Verbe : POST

7.c :

```
{{ _.base_url }}datasets/:persistentId/versions/{n°_de_version}/files?  
persistentId={DOI_du_JDD}
```

Verbe : GET



Exercice 8 : Récupérer des fichiers

8.a Trouver la documentation sur la DATA Access API

<https://guides.dataverse.org/en/5.14/>

8.b Télécharger tous les fichiers du jeu de données 10.57745/EELX9B sur l'entrepôt

8.c Lister les fichiers de ce jeu de données

```
{{ _.base_url }}datasets/:persistentId/?persistentId={DOI_du_JDD}
```

8.d Puis télécharger un seul fichier de ce jeu de données

```
{{ _.base_url }}access/datafile/:persistentId?persistentId={DOI_du_fichier}
```

Correction 8 : Récupérer des fichiers

8.b Télécharger tout

```
{{ _.base_url }}access/dataset/:persistentId/?  
persistentId=doi:10.57745/EELX9B
```

Verbe : GET

8.c Lister

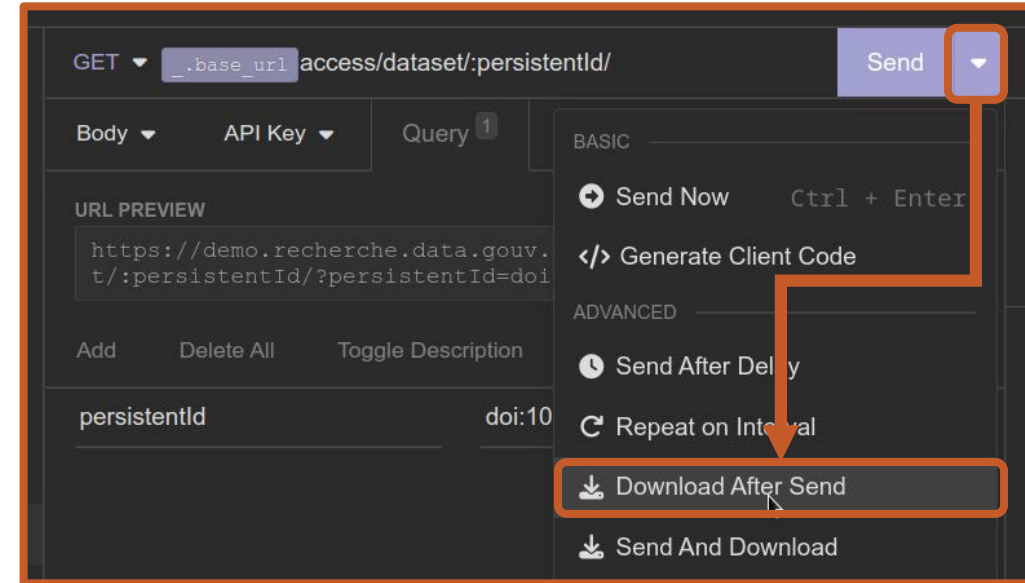
```
{{ _.base_url }}datasets/:persistentId/?  
persistentId=doi:10.57745/EELX9B
```

Verbe : GET

8.d Télécharger un

```
{{ _.base_url }}access/datafile/:persistentId/?  
persistentId=doi:10.57745/UQIFOB
```

Verbe : GET



Métrique

API = Metrics

Cas d'usage

- Récupérer des métriques sur le portail (créations, téléchargements...)
- Récupérer des métriques pour une collection
- Récupérer des métriques d'utilisation pour un jeu de données

Point d'attention à l'usage :

- Données de bac à sable pas forcément renseignées

Exercice 9 : Récupération de données métriques

<https://guides.dataverse.org/en/5.14/api/metrics.html>

9.a Métriques de téléchargements de l'entrepôt

9.b Métriques de téléchargements d'une collection par mois (`monthly`)

9.c Métriques de téléchargements dans une collection par thématique (`bySubject`)

9.d Métriques de téléchargements uniques des fichiers, par mois, dans une collection

Correction 9

9.a Tous téléchargements

```
{{ _.base_url }}info/metrics/downloads
```

Verbe : GET

9.b Téléchargements mensuels dans une collection

```
{{ _.base_url }}info/metrics/downloads/monthly?
```

```
parentAlias={alias_du_dataverse}
```

Verbe : GET

Correction 9.bis

9.c Téléchargements dans une collection par thématique

```
{{ _base_url }}info/metrics/dataverses/bySubject?
```

```
parentAlias={alias_du_dataverse}
```

Verbe : GET

9.d Téléchargements uniques, par mois, dans une collection

```
{{ _base_url }}info/metrics/uniquedownloads/monthly?
```

```
parentAlias={id_de_collection}
```

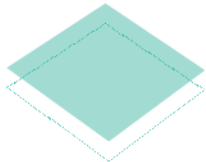
Verbe : GET




**Merci
pour votre écoute**



Classe Virtuelle : Utilisation des APIs

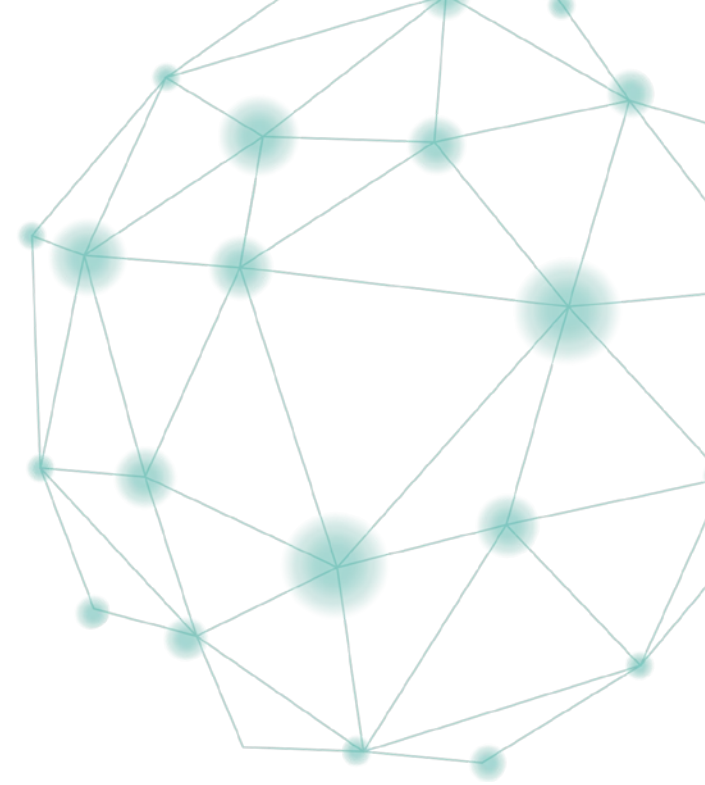


Centre de ressources
entrepôt-catalogue
 Recherche Data Gov

support-recherchedatagouv@inrae.fr



Support réalisé par : Camille Desiles (CNRS-INIST) ; Virgile Jarrige (Unistra) ; Damien Sans (INRAE) ; Dimitri Szabo (INRAE)



Une gouvernance collective

- Partenaires du projet



- Représentants de la communauté ESR



ANNEXE A : API MyData : Documentation

- Endpoint : {serveur}/api/mydata/retrieve
- Paramètres :
 - **⚠ Key = Jeton de l'utilisateur; pour cette API il ne doit pas être passé dans le header (jusqu'à la version 5.14)**
 - **role_ids = Rôles de l'utilisateur, plusieurs valeurs possibles parmi :**
 - 1 = Admin
 - 2 = File Downloader
 - 3 = Dataverse + Dataset Creator
 - 4 = Dataverse Creator
 - 5 = Dataset Creator
 - 6 = Contributor
 - 7 = Curator
 - 8 = Member
 - **dvobject_types = Type d'objet, plusieurs valeurs possibles parmi :**
 - DataFile
 - Dataset
 - Dataverse
 - **published_states = Etat de l'objet, plusieurs valeurs possibles parmi :**
 - Published
 - Unpublished
 - Draft
 - Deaccessioned
 - In+Review
 - **per_page = Nombre de résultats renvoyés par page**