

Requêtes HTTP : GET et POST

July 5, 2019

1 Présentation de data.gouv.fr

1.1 Différences entre un utilisateur et une organisation

Quand vous créez un compte sur data.gouv.fr, vous devenez utilisateur de la plateforme. Votre compte peut ensuite être ajouté à n'importe quelle organisation. Une organisation est une entité qui peut rassembler plusieurs utilisateurs. Un même utilisateur peut appartenir à plusieurs organisations.

1.1.1 Compte utilisateur

Chaque individu qui contribue sur data.gouv.fr possède un compte utilisateur. Ce compte permet de :

- mettre en ligne des données gratuitement ;
- référencer des réutilisations de données ;
- ouvrir une discussion au sujet d'un jeu de données ;
- suivre les publications d'un autre utilisateur ;
- créer, rejoindre, ou suivre une organisation.

1.1.2 Organisations

Les organisations permettent à plusieurs utilisateurs de publier des données sous le même nom, la même bannière. Les administrateurs d'une organisation peuvent choisir d'ajouter ou de retirer des utilisateurs au sein de leur organisation au moyen de fonctionnalités spécifiques.

Une organisation permet de :

- rassembler des utilisateurs unis par un intérêt commun ;
- publier des jeux de données au nom de l'organisation ;
- devenir un producteur de données certifié par data.gouv.fr.

1.2 Différences entre un jeu de données et une ressource

La plateforme data.gouv.fr permet de créer et de consulter des jeux de données. Un jeu de données se compose de ressources, de la même manière qu'un album est composé de chansons. Dit autrement, un jeu de données peut avoir plusieurs ressources et une ressource appartient nécessairement à un jeu de données.

À chaque jeu de données correspond une page de présentation sur data.gouv.fr, depuis laquelle peuvent être téléchargées les ressources qui y sont associées.

1.2.1 Jeu de données

Un jeu de données comporte :

- un titre (obligatoire) ;
- un sigle (facultatif)
- une description (obligatoire) ;
- une licence (facultative) ;
- une fréquence de mise à jour (obligatoire) ;
- une date de dernière mise à jour (facultative) ;
- des mots-clefs (facultatifs) ;
- un intervalle de couverture temporelle (facultatif) ;
- une zone de couverture spatiale (facultative) ;
- un niveau de granularité spatiale (facultatif).

1.2.2 Ressources

Une ressource comporte :

- un titre (obligatoire) ;
- un type (obligatoire) ;
- une description (facultative) ;
- une date de publication (facultative) ;
- une URL (obligatoire) ;
- une taille en bytes (facultative) ;
- un format (obligatoire) ;
- un type mime (facultatif) ;
- une somme de contrôle (facultative).

1.3 Utilisation de l'API dans les scripts python

Tous les exemples qui suivent sont réalisés avec un compte :

- qui est actif
- dont la clé d'API est my-api-key
- qui est membre d'une organisation.

Les exemples portant sur un jeu de données existant utilisent des identifiants :

- un identifiant pour l'organisation ;
- un identifiant pour un jeu de données ;
- un identifiant pour la ressource.

Tous les exemples Python sont exécutés avec cette convention.

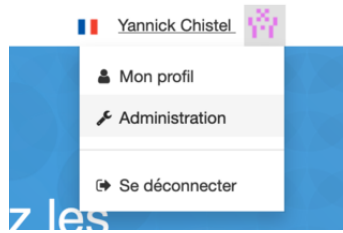
```
import requests # installé avec `pip3 install requests`

# déclaration de constantes
API = 'https://www.data.gouv.fr/api/1'
API_KEY = 'saisir sa clé API présente sur le site data.gouv.fr'
```

```
ORG = 'identifiant organisation'
DATASET = 'identifiant du jeu de données'
RESOURCE = 'identifiant de la ressource'
HEADERS = {
    'X-API-KEY': API_KEY,
}
# requête HTTP avec la librairie Request
response = requests.post(url, json={
    'title': 'Mon titre',
    'description': 'Ma description',
    'organization': ORG,
}, headers=HEADERS)
```

1.4 Pré-requis pour l'activité "requête HTTP - script Python"

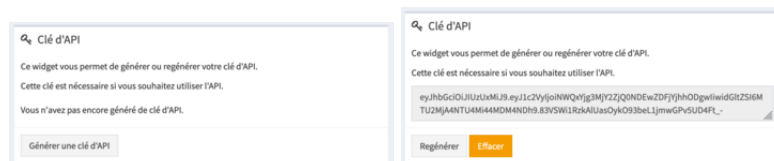
1. Se créer un compte sur le site <https://www.data.gouv.fr/fr/login?next=https%3A%2F%2Fwww.data.gouv.fr/fr>
2. Confirmez votre inscription avec le mail reçu à votre adresse. Vous êtes renvoyé directement sur le site www.data.gouv.fr
3. Déroulez le menu en haut à droite à votre nom et cliquez sur **Administration**.



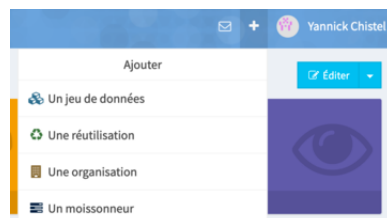
Rendez-vous sur la rubrique **Moi** placée dans la colonne latérale gauche.

La page qui s'affiche recense vos jeu de données et ressources. Elle est pour l'instant vide.

4. En bas de la page, vous devez générer une clé API pour communiquer avec le site dans vos scripts Python.



5. Il ne reste plus qu'à créer son organisation. Sur la gauche du menu à votre nom, en haut à droite, vous avez un signe +.



Cliquer dessus et sélectionnez **Ajouter une organisation** : suivez les étapes et créez votre organisation.

Lorsque l'organisation est créée, vous devez récupérer son **id** dans la page de présentation de l'organisation en cliquant sur le bouton **Détails**.



6.

1.5 Vous êtes prêts pour écrire votre premier script python.

1.6 Création d'un jeu de données

Pour créer un jeu de données, nous allons utiliser l'API de création de jeu de données. ##### En python

```
# déclaration de constantes
API = 'https://www.data.gouv.fr/api/1'
API_KEY = 'saisir sa clé API présente sur le site data.gouv.fr'
ORG = 'identifiant organisation'
url = api_url('/datasets/')
response = requests.post(url, json={
    'title': 'Mon titre',
    'description': 'Ma description',
    'organization': ORG,
}, headers=HEADERS)
```

La réponse en JSON contient les metadonnées du jeu de données créé.
La fiche du jeu de données est maintenant créée et il est maintenant possible d'y ajouter des ressources.

Exemple de mise en pratique du code On dispose d'un compte et d'une organisation sur **data.gouv.fr**.

On va créer un premier jeu de données qui contiendra nos futures ressources en csv, json, odt, pdf,... On renseigne les constantes API, API_KEY et ORG 1. Saisir l'**url** de l'API dans la constante API. 2. Copier et coller la **clé API** de votre compte (encore vide de données) dans la constante API_KEY 3. Coller l'id de l'organisation dans la constante ORG, par exemple : 5d1b922b6f444122f3430e2e.

On va réaliser une requête **POST** en envoyant différentes données: 1. Un titre pour le jeu de données : Premier jeu de données, 2. Une description : les données pour les activités NSI du lycée, 3. L'identifiant de l'organisation.

```
In [2]: import requests
```

```
In [45]: API = 'https://www.data.gouv.fr/api/1'
API_KEY = 'eyJhbGciOiJIUzUxMiJ9.eyJ1c2VyIjoibWQxYjg3MjY2ZjQONDEwZDFjYjhhODgwIiwidGlhZSIsInR5eSI6ImF0dXNpdCI7fQ.'
ORG = '5d1b922b6f444122f3430e2e'
HEADERS = {
    'X-API-KEY': API_KEY,
}
url= 'https://www.data.gouv.fr/api/1/datasets/'
response = requests.post(url, json={
    'title': 'Premier jeu de données',
    'description': 'Les données pour les activités NSI du lycée',
    'organization': ORG,
}, headers=HEADERS)
```

Notre premier jeu de données est créé. On peut contrôler la présence de ce jeu de données sur le site **data.gouv.fr**.

Notre jeu de ressources a un identifiant qu'il faut récupérer, soit en allant sur le site **data.gouv.fr** et en affichant les détails du jeu de données, soit en réalisant une requête **GET** en python.

Procédure pour récupérer l'id du jeu de données avec python On va créer une requête **GET** qui ressemble beaucoup à la requête **POST**. Il faut modifier l'url et enlever les données.

L'url est de la forme : "https://www.data.gouv.fr/api/1/organizations/id_organisation/datasets"

Après la requête, la réponse est stockée dans la variable `response` qui au format **json**.

```
In [47]: API = 'https://www.data.gouv.fr/api/1'
API_KEY = 'eyJhbGciOiJIUzUxMiJ9.eyJ1c2VyIjoibWQxYjg3MjY2ZjQONDEwZDFjYjhhODgwIiwidGltZSI6ImVudD1hY2Z2b6f444122f3430e2e'
ORG = '5d1b922b6f444122f3430e2e'
HEADERS = {
    'X-API-KEY': API_KEY,
}
url= 'https://www.data.gouv.fr/api/1/organizations/5d1b922b6f444122f3430e2e/datasets'
response = requests.get(url, headers=HEADERS)
```

```
In [94]: #import json
          datas=response.json()
          print("id du jeu de données :", (datas['data'][0]['id']))
```

id du jeu de données : 5d1b949f8b4c4139da223cb9

1.7 Ajout d'une ressource dans un jeu de données

Pour créer une ressource, nous allons utiliser l'API création d'une ressource.

Il existe 2 cas de création de ressource :

- avec envoi d'un fichier, dit ressource locale ;
- avec référencement d'un fichier distant, dit ressource distante.

1.7.1 En envoyant un fichier

Nous allons utiliser l'API d'envoi de ressource pour envoyer le fichier.

En python

```
url = api_url('/datasets/{}/upload'.format(DATASET))
response = requests.post(url, files={
    'file': open('/chemin/vers/le/fichier', 'rb'),
}, headers=HEADERS)
```

La ressource est automatiquement créée et il est possible de modifier a posteriori les metadonnées avec l'API de mise à jour de ressource comme décrit plus bas.

```
In [104]: DATASET = '5d1b949f8b4c4139da223cb9'
          url = 'https://www.data.gouv.fr/api/1/datasets/{}/upload'.format(DATASET)
          response = requests.post(url, files={
              'file': open('sites.json', 'rb'),
          }, headers=HEADERS)
```

Remarques:

1. Évitez de lancer plusieurs fois la commande. Elle crée des ressources autant de fois qu'elle a été lancée.
2. Aucun message n'est affiché en retour. Que s'est-il passé ?
Sur le site, aucune ressource n'a été créée !
3. Pour avoir l'entête de la réponse du serveur, on affiche **response.header**
4. Pour connaître le statut de la réponse du serveur, on affiche **response.statut_code**
5. Pour connaître le corps de la réponse du serveur, on affiche **response.content**

```
In [111]: response.headers
```

```
Out[111]: {'Content-Type': 'application/json', 'Server': 'nginx', 'Content-Length': '63', 'Date': 'Mon, 11 Jun 2018 14:00:00 GMT'}
```

```
In [112]: response.status_code
```

```
Out[112]: 405
```

```
In [113]: response.content
```

```
Out[113]: b'{"message": "The method is not allowed for the requested URL."}'
```

Conclusion On observe une erreur de type 405 qui signifie que la méthode POST n'est pas autorisée. Surement une interdiction de téléverser le fichier sur le serveur.

1.7.2 En référençant une URL existante

L'API de création de ressource permet de créer une ressource distante.

Dans notre cas, un fichier csv hébergé sur l'URL <https://url.to/ressource.csv>.

En python

```
url = 'https://www.data.gouv.fr/api/1//datasets/{}/resources/'.format(DATASET)
response = requests.post(url, json={
    'title': 'Mon titre',
    'description': 'Ma description',
    'url': 'https://url.to/ressource.csv',
    'type': 'main',
    'filetype': 'remote',
    'format': 'csv',
}, headers=HEADERS)
```

```
In [121]: url = 'https://www.data.gouv.fr/api/1//datasets/{}/resources/'.format(DATASET)
response = requests.post(url, json={
    'title': 'Annuaire des sites',
    'description': 'Des sites du lycée',
    'url': 'https://leducdenormandie.itslearning.com/data/2362/C198579/sites.json',
    'type': 'main',
    'filetype': 'remote',
    'format': 'json',
}, headers=HEADERS)
```

```
In [122]: response.status_code
```

```
Out[122]: 201
```

```
In [123]: response.content
```

```
Out[123]: b'{"checksum": null, "created_at": "2019-07-04T19:10:01.673884", "description": "Des
```

La ressource a été ajoutée : <https://www.data.gouv.fr/fr/datasets/premier-jeu-de-donnees-2/>

1.8 Conclusion

Cette activité nécessite une bonne connaissance de l'API de www.data.gouv.fr. Des ressources pour vous aider:

1. La librairie python Request : <http://fr.python-requests.org/en/latest/index.html>
2. La documentation sur l'API : <https://doc.data.gouv.fr/api/intro/>
3. Les commandes **curl** et **python** pour interroger l'API : <https://doc.data.gouv.fr/api/dataset-workflow/>
4. La documentation de référence de l'API : <https://doc.data.gouv.fr/api/reference/>

s