

Connexion de vos APIs en OAuth2

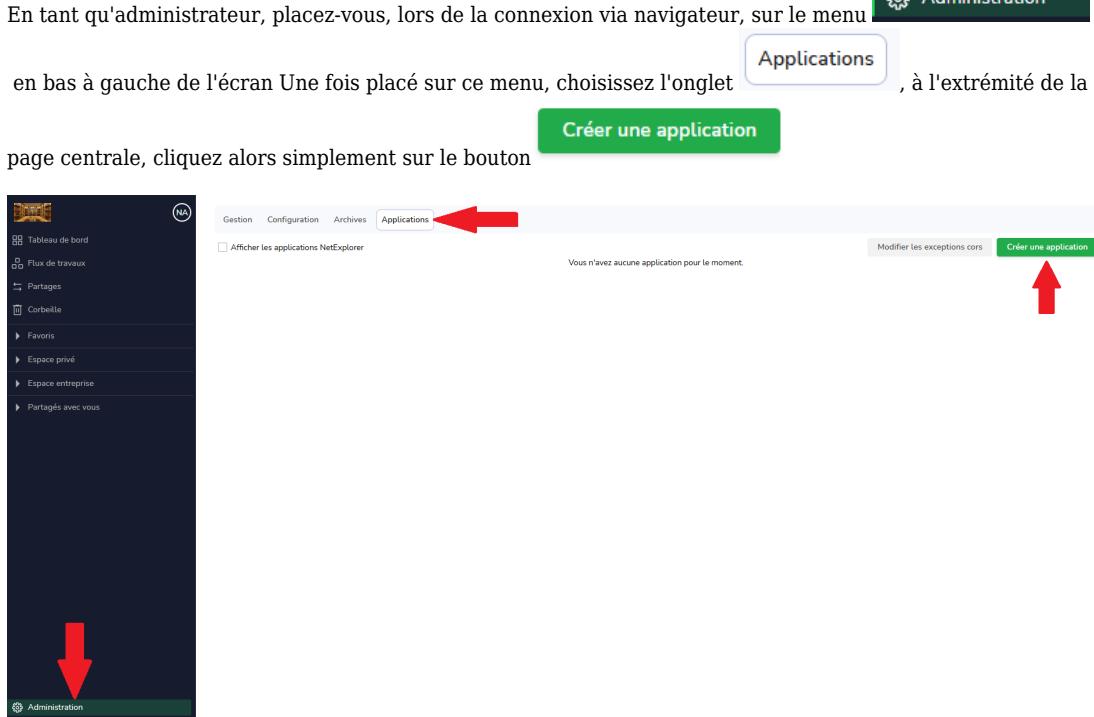
Nicolas ARBOUIN - 2025-09-23 - [Accès aux APIs de la plateforme](#)

Lors de la connexion aux APIs NetExplorer, passez dès maintenant, par une authentification en OAuth2.

Cet article vous explique comment configurer votre application

Connectez vos applications ou scripts, via l'application OAuth2, que votre plateforme NetExplorer vous propose de mettre en place. Vous aurez ainsi la possibilité de choisir le type de connexion utilisée.

Sur votre plateforme



Création de votre application OAuth2

Lors du clic sur le bouton de création, apparaît la fenêtre ci-dessous

Créer une application

X

LOGO **1**

Nom de l'application **2**

Description **3**

Nom de l'éditeur **4**

Site de l'éditeur **5**

Callback URIs **6**

Sélectionnez les scopes **7**

Type d'application **8**

Client

Serveur

Sauvegarder



1. Vous pouvez attribuer un logo à votre application d'authentification en cliquant sur **LOGO**
2. Donnez un nom à votre application OAuth2
3. Ajoutez un description à l'utilisation de votre application
4. Le nom de l'éditeur ou créateur de cette application
5. Le site de l'éditeur ou créateur concerné.
6. Les éventuelles urls de callback à implémenter lors d'une connexion cliente
7. Vous allez, en cliquant sur ce bouton, sélectionner les scopes de droits attribués à votre application (expliqué dans le chapitre au dessous)
8. Cette zone va vous permettre de définir le type d'application OAuth2 utilisée :
 - **Client** : utiliser le profil d'un compte utilisateur existant sur la plateforme
 - Par exemple: permettre un accès à vos applications, aux dossiers sur lesquels l'utilisateur a des droits
 - **Serveur** : créer une application permettant des accès administrateur
 - Par exemple: créer un script d'accès global à la plateforme, automatiser la création d'utilisateurs, uploader des fichiers...

La sélection des scopes

Sélectionnez les scopes

Lors du clic sur le bouton (le point 7 dans le détail ci dessus) vous ouvrez une fenêtre, permettant de donner les droits nécessaires à votre application.

Créer une application

X

[Retour](#)

Tout sélectionner

account

read write

accountsec

read write

annotations

read write

archives

read write

alerts

read write

files

read write

Suivant vos désideratas, vous pourrez alors accorder aux différentes scopes listés dans les scopes (account, annotations, files...) les droits que doit avoir votre application, dès lors qu'un appel API est effectué :

1. En lecture simple (read)
2. En écriture simple (write)
3. Tous les droits (read et write)

Tout sélectionner
Le bouton vous permettant alors de donner les droits globaux à tous les appels.

Une fois les zones remplies, cliquez sur Sauvegarder.

Une fenêtre, montrant 2 paramètres nécessaires, s'affiche après la création de votre application :

Test_OAuth2

X

Attention : notez et sauvegardez votre **client_secret** dès maintenant. Il ne sera plus accessible après cette étape. Le **client_id** restera disponible ultérieurement.

client_id

Za [REDACTED] jbd



client_secret

LU [REDACTED] ElGe



- client_id : vous pourrez retrouver ce paramètre au sein de votre application, le cas échéant.
- client_secret : pensez bien à copier ce paramètre, vous n'y aurez plus accès par la suite.

La connexion de votre application, exemple avec Postman

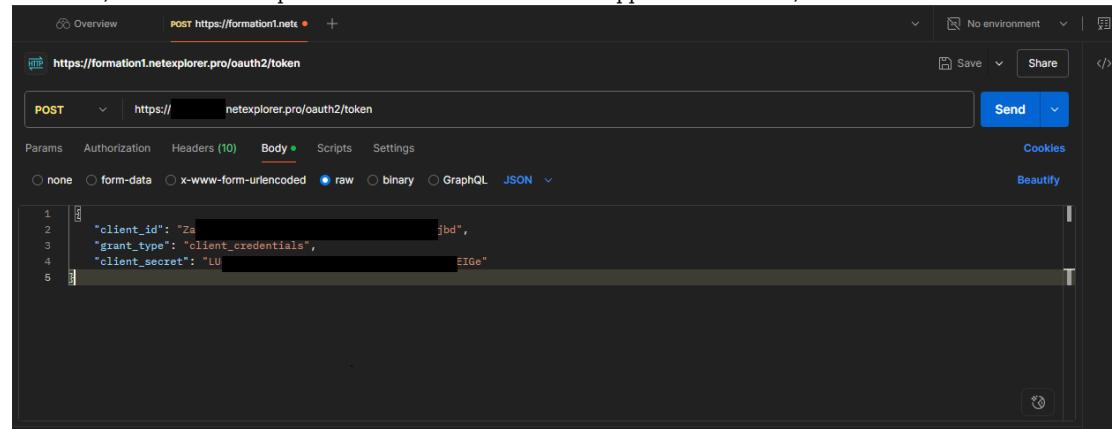
Une fois l'application configurée, vous allez pouvoir appeler l'API d'authentification, au sein de vos éléments de script ou de code.

Nous allons voir ici comment appeler votre authentification via Postman.

Je vais donc passer, comme indiqué dans la documentation API, à cet endroit : <https://api.netexplorer.fr/v3/#authorize> les 3 paramètres nécessaires :

- client_id
- grant_type
- client_secret

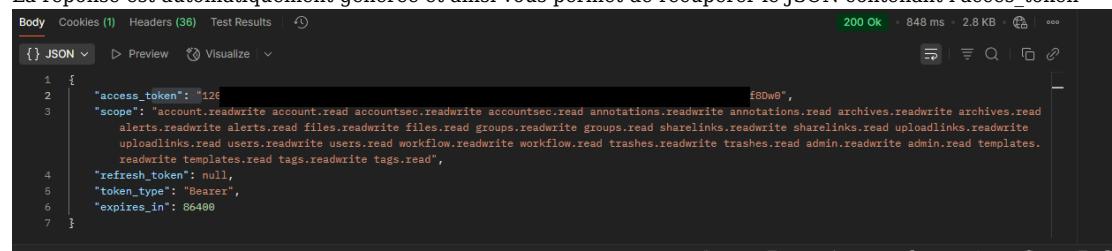
Le but ici, est donc de récupérer un token de connexion de l'application OAuth2, en mode Serveur.



The screenshot shows the Postman interface with a POST request to `https://formation1.netexplorer.pro/oauth2/token`. The request body is a JSON object:

```
1 {  
2   "client_id": "za",  
3   "grant_type": "client_credentials",  
4   "client_secret": "LU"  
5 }
```

La réponse est automatiquement générée et ainsi vous permet de récupérer le JSON contenant l'accès token



The screenshot shows the Postman interface displaying the response to the POST request. The response is a `200 OK` status with the following JSON content:

```
1 {  
2   "access_token": "12e",  
3   "scope": "account.readwrite account.read accountsec.readwrite accountsec.read annotations.readwrite annotations.read archives.readwrite archives.read  
alerts.readwrite alerts.read files.readwrite files.read groups.readwrite groups.read sharelinks.readwrite sharelinks.read uploadlinks.readwrite  
uploadlinks.read users.readwrite users.read workflow.readwrite workflow.read trashes.readwrite trashes.read admin.readwrite admin.read templates.  
readwrite templates.read tags.readwrite tags.read",  
4   "refresh_token": null,  
5   "token_type": "Bearer",  
6   "expires_in": 86400  
7 }
```