

GLPI - Utiliser l'API REST de GLPI avec cURL et Python

Présentation de l'activité

Vous avez **expérimenté** une première utilisation de l'**GrasAPI** REST de GLPI avec l'outil **Postman**. Cet outil est très pratique pour **tester** et **déterminer les bons paramètres** à définir dans vos requêtes d'API REST qui **utilisent les protocoles HTTP ou HTTPS**.

Pour rappel il faut s'appuyer sur la **documentation** de l'API pour :

- **connaître** le **point d'entrée** (Endpoint) de l'API c'est à dire **l'URL** à utiliser ;
- connaître les **types de requête** proposés : **GET, POST** mais aussi **DELETE, PUT** et **PATCH** ;
- **interpréter** correctement les **codes retour** de réponse de l'API : **2xx** (succès), **3xx** (redirection) **4xx** (erreur côté client), **5xx** (erreur côté serveur) ;
- savoir **utiliser** les **clés d'API** pour **s'authentifier** et obtenir un **jeton de session** (`session_token`) ;
- **fournir** les **paramètres** nécessaire soit dans **l'URL (Query string)** ou dans **l'entête (Header)** ;
- gérer le **format des données** obtenues qui sont le plus souvent au format **JSON**.

Dans cette activité, vous allez **réutiliser** ce que vous avez expérimenté avec **Postman** pour exécuter des requêtes avec l'utilitaire **cURL** et le langage de programmation **Python**.

Préparer votre environnement Python

Vous allez utiliser la bibliothèque **requests** de Python. Si nécessaire mettez à jour votre environnement python :

- mettre à jour pip
- installer le module requests

```
C:> python -m pip install --upgrade pip
C:> pip install request
```

Obtenir un jeton de session (Session-Token)

Pour obtenir un jeton de session, il faut effectuer une requête **Init session** en renseignant des paramètres dans l'entête (**Headers**) :

- le jeton d'application (**App-Token**) qui a été créé pour vous permettre d'utiliser l'API de GLPI. Demandez à l'enseignant de vous communiquer cet **App-Token**.
- une authentification soit par un couple de login /mot de passe soit en utilisant un jeton d'utilisateur (`user_token`). C'est cette deuxième méthode que vous allez utiliser.

Exécution de la requête avec cURL

```
C:> curl -X GET -H "Authorization: user_token uuu..." -H "App-Token: aaa..."
"http://path/to/glpi/apirest.php/initSession"
{"session_token":"sss..."}
```

Exécution de la requête en Python

- script Python `sessiotoken.py`

```
<code shell> import urllib.parse import requests api = "http://path/to/glpi/apirest.php/initSession"
usertoken = "uuu..."
```

```
App-Token = "aaa..."
```

```
url = mainapi + urllib.parse.urlencode({"key":key, "from":orig, "to":dest}) jsondata = requests.get(url).json() print(json_data)
```

```
C:> curl -X GET -H "Authorization: usertoken uuu..." -H "App-Token: aaa..." "http://path/to/glpi/apirest.php/initSession"
{"sessiontoken":"sss..."} </code>
```

Retour Activité A8

- [A8 - La gestion des configurations avec GLPI](#)

Last update:
2021/03/04 22:02

si7:configuration:glpi_gestionapiestpython /doku.php/si7/configuration/glpi_gestionapiestpython?rev=1614891743

From:
/ - **Les cours du BTS SIO**

Permanent link:
/doku.php/si7/configuration/glpi_gestionapiestpython?rev=1614891743

Last update: **2021/03/04 22:02**

