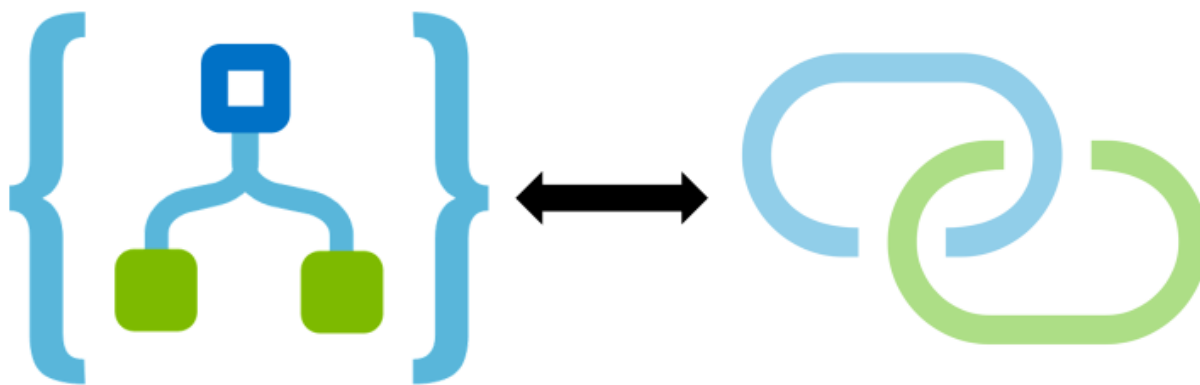


VERSION 1.0  
MAY 15, 2022



# MISE EN PLACE D'UNE API EN PYTHON

ETUDE DE CAS

**AUTEUR :** BAPTISTE VANDENBORGHT  
BAPTISTE.VANDENBORGHT@GMAIL.COM

## TABLE DES MATIERES

1.	Contexte .....	3
2.	Énumération du besoin client .....	3
3.	Objectifs techniques .....	3

## 1. CONTEXTE

Le client est une fédération sportive, comptant des dizaines de milliers d'adhérents, et qui regroupe principalement des activités liés à la promotion, la réalisation et l'encadrement d'évènements sportifs sur tout le territoire national.

Le client dispose déjà d'un site internet public, c'est-à-dire un site accessible librement sur Internet via une recherche Google. Celui-ci dispose de plein de pages web différentes, parlant de la fédération, des activités proposées et des prochains évènements.

**Ce site est réalisé sur WordPress.**

Jusqu'à présent, les quelques 10 000 adhérents disposaient également d'un extranet, c'est-à-dire un espace privé (en opposition au site public), sur lequel ils peuvent notamment récupérer leur documents légaux, et aussi pour pouvoir contacter la fédération.

## 2. ÉNUMÉRATION DU BESOIN CLIENT

Le client souhaite mettre en place un système de réservation et de consultation d'évènements sportif sur leur extranet. Le problème, c'est qu'une telle fonctionnalité n'est pas possible sur WordPress et ils ne peuvent pas le développer dessus.

Par conséquent, le client a fait appel à une boîte externe (qui n'est pas vous) pour développer le système événementiel. Cette nouvelle plateforme de réservation sera disponible sur une autre adresse URL, hébergée sur une autre infrastructure, et sera principalement développée en Django.

En revanche, il faut que les +10 000 adhérents puissent se connecter à cette nouvelle plateforme avec leurs comptes adhérents du site WordPress.

**Dans le cadre de l'étude de cas, votre mission sera de réaliser le module Python qui servira d'interconnexion entre la base utilisateurs WordPress et la nouvelle plateforme.**

Une fois votre module réalisé, la boîte qui développe la plateforme pourra utiliser celui-ci pour faire la passerelle entre son système et WordPress pour que les utilisateurs puissent se connecter sans avoir à recréer un compte utilisateur sur la plateforme de réservation.

## 3. OBJECTIFS TECHNIQUES

L'objectif de cette mission est de livrer un ou plusieurs fichiers Python qui disposera d'une fonction `authenticate(username, password)` pour permettre au développeur de l'utiliser afin de permettre à ses utilisateurs de se connecter. Cette fonction retourne un **Boolean** (**True** or **False**).

D'un point de vue algorithmie, cette fonction devra :

- 1) Se connecter à l'API WordPress avec le compte utilisateur adhérent passé en paramètre
- 2) Récupérer les champs du prénom, nom et adresse de cet adhérent pour les afficher via **print** (afin de le retourner en base de données plus tard, mais vous ne développerez pas cette fonction)
- 3) Retourne **True** si l'utilisateur WordPress s'est bien connecté (username/password), sinon **False**

Voici les quelques directives :

- Vous devez utiliser exclusivement Python 3 (version que vous voulez), pas de Python 2
- Pour réaliser à bien la connexion avec WordPress, vous devez concevoir une classe Python **WordPressAPI** qui disposera des 2 méthodes suivantes, parmi d'autres évidemment :
  - o **connect(self)** : Tentera une connexion à l'API REST WordPress en POST sur la ressource token pour récupérer le jeton de connexion de l'utilisateur. Cette méthode ne retourne rien mais mettra à jour le Header de l'Authorization pour envoyer des requêtes authentifiées à WordPress
  - o **get(self, endpoint)** : Permettra de lancer une requête GET sur la ressource WordPress passée en paramètre endpoint. Cette méthode retourne le contenu de la réponse sous format JSON, si la requête a été en succès évidemment (donc avec un statut 200)
- Vous devrez prendre en compte les erreurs HTTP retournées par WordPress et avoir une gestion de celles-ci (pour éviter des Exception, notamment)
- Pour tester votre produit, vous devez vous créer gratuitement un site WordPress avec un compte utilisateur à tester (<https://wordpress.com/fr/free/>) à nous fournir après pour valider l'étude de cas
- Vous devez utiliser et ajouter le plugin **WordPress JSON Web Tokens (JWT)** sur votre site WordPress afin de pouvoir vous connecter sur celui-ci et utiliser l'API WordPress
- Vous vous aiderez des documentations suivantes pour réaliser à bien le projet
  - o REST API WordPress : <https://developer.wordpress.org/rest-api/>
  - o Plugin WordPress Json Web Tokens (JWT): <https://wordpress.org/plugins/jwt-authentication-for-wp-rest-api/>
- Vous devrez utiliser le module Python **requests** (à installer via PIP) pour faire vos requêtes HTTP

**Vous avez absolument le droit (et on vous le conseille même) de nous poser des questions si vous êtes bloqué(e) ou que vous ne comprenez pas quelque chose dans ce projet : vous ne serez pas jugé(e) sur le fait d'y arriver tout(e) seul(e), mais plutôt sur votre capacité à vous adapter, à répondre au besoin du client, et à faire appel habilement aux ressources que vous avez à disposition pour coder le mieux possible.**

Même si le produit ne fonctionne pas, mais que vous pouvez nous démontrer et expliquer la logique, cela sera pris en compte de manière positive.

Bonne chance et bon code !