Introduction aux API avec Python

Objectifs

- Comprendre en profondeur ce qu'est une API, son utilité et comment elle fonctionne dans les architectures modernes.
- Apprendre à interagir avec une API REST via les méthodes principales : **GET**, **POST**, **PUT**, **DELETE**.
- Manipuler les **paramètres d'URL**, comprendre les **API Keys** et la gestion de l'authentification.
- Découvrir comment utiliser Python et la bibliothèque `requests` pour envoyer des requêtes HTTP.
- Expérimenter avec une API réelle et documentée, afin d'acquérir une autonomie dans l'exploration et l'utilisation de nouvelles APIs.

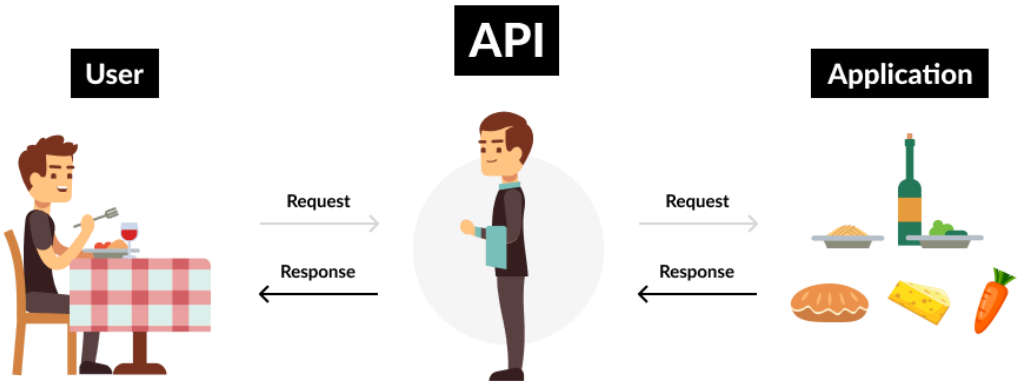
Pré-requis

- Programmation en Python

Qu'est-ce qu'une API ?

Le terme API signifie *Application Programming Interface*, ce que l'on pourrait traduire en français par "interface de programmation d'application". En pratique, une API est un **ensemble de règles et de conventions** qui permettent à deux systèmes de communiquer entre eux. Une API peut être vue comme **un pont entre deux applications**, qui permet d'échanger des informations ou de déclencher des actions.

Pour bien comprendre, prenons une analogie simple : imaginez que vous êtes au restaurant. Le **menu** représente ce que vous pouvez commander (les services offerts), **vous êtes le client**, et **le serveur** est l'API. Vous faites une commande au serveur, qui va la transmettre à la cuisine, et ensuite vous apporter la réponse. Vous ne voyez pas la cuisine (le système interne), vous interagissez uniquement avec l'interface : le serveur.



Dans le contexte du web, une API permet à un logiciel (par exemple une application mobile ou un site web) de demander des informations à un autre système (comme une base de données, un service météo, ou une plateforme de paiement).

APIs REST

La majorité des APIs modernes sont de type **REST** (REpresentational State Transfer). Elles utilisent le protocole HTTP, tout comme les sites web, et s'appuient sur des méthodes standard (GET, POST, PUT, DELETE).

Reprenons notre exemple du restaurant, vous êtes le **client**, l'**API** est le **serveur**, et la **cuisine** est la base de données ou le système interne, les méthodes:

- `GET` : permet de récupérer des informations. Par exemple pour consulter la liste des plats, lire les détails d'un produit ou consulter une commande existant
- `POST` : permet de créer une nouvelle ressource. Par exemple pour passer une commande
- `PUT` : permet de modifier une ressource existante. Par exemple pour modifier une commande passée précédemment ou corriger une information
- `DELETE` : permet de supprimer une ressource. Par exemple pour annuler une commande

Structure d'une requête API

Une requête API constitue une sorte de message structuré. Voici les **composants principaux** d'une requête API:

Composant	Rôle	Exemple
URL	L'adresse du service/API	https://api.thecatapi.com/v1/images/search
Méthode	Type d'action qu'on veut faire	GET, POST, PUT, DELETE
Headers	Informations supplémentaires sur la requête (identité, format, sécurité)	x-api-key: votre_clé Content-Type: application/json

Params	Petites infos dans l'URL pour affiner la requête	?limit=3&type=jpg
Body	Contenu envoyé avec la requête (utile surtout en POST/PUT)	{ "name": "Nouvel élément" } (en JSON)
Authorization	Permet d'accéder à des données protégées	Token, clé API, etc.

Structure d'une réponse API

Une fois la requête envoyée, l'API **répond** avec une **réponse structurée**, souvent au format **JSON**.

Elle contient en général :

Élément	Rôle	Exemple
Code HTTP	Statut de la réponse	200 OK, 201 Created, 400 Bad Request, 401 Unauthorized
Body (corps)	Données retournées	{ "id": 123, "name": "Chat mignon" }
Headers	Infos sur la réponse	Type, durée, quota, etc.

RESSOURCES:

REST API :
https://www.scrapingbee.com/blog/api-for-dummies-learning-api/#what-is-a-rest-api

Premier contact avec une API

Installer la bibliothèque `requests`

Avant de pouvoir interagir avec une API, vous devez installer un outil Python permettant d'envoyer des requêtes HTTP. La bibliothèque `requests` est la plus utilisée pour cela :

```
pip install requests
```

RESSOURCES

● Tutoriel sur requests: https://rtavenar.github.io/poly_python/content/api.html

● Documentation de requests: https://docs.python-requests.org/en/latest/

Exercices 1:

1. En utilisant cette bibliothèque, faite une requête **get** sur l'url https://jsonplaceholder.typicode.com/posts
    a. Stockez le résultat de cette requête dans une variable
    b. Explorez l'objet stocké dans cette variable
2. Que signifie le code "statut"?
3. Dans quel attribut trouve t'on des donnés?

Exercise 2 : The Cat API:

A l'aide de l'API et de la bibliothèque *requests*, écrire pour chaque exercice une fonction en python qui:

1. Récupère une image aléatoire de chat ou de chien
2. Récupère 5 images aléatoires
3. Récupère 3 images aléatoires de type GIF uniquement
4. Récupère la liste complète des races de chats ou de chiens
5. Récupère 3 images de la race "bengal" (ou une autre race si vous travaillez avec les chiens)
6. Envoie un vote positif pour une image de chat ou de chien
7. Supprime le vote que tu viens de créer
8. Fais une requête volontairement incorrecte

9. (Bonus) Envoie une image personnalisée de ton chat ou de ton chien

RESSOURCES

- Présentation CatAPI https://thecatapi.com/
- Liste des requêtes CatAPI (section **Open API Spec doc**): https://developers.thecatapi.com/view-account/ylX4blBYT9FaoVd6OhvR?report=bOoHBz-8t

- Présentation DogAPI https://thedogapi.com/
- Liste des requêtes DogAPI (section **API Reference doc**): https://docs.thedogapi.com/

- Pour aller plus loin, Authentication: https://www.merge.dev/blog/rest-api-authentication