

## Web scraping via une API

Le but de ce troisième travail est de découvrir la collecte de données web non plus de manière générique à l'aide de modules Python dédiés, mais cette fois à l'aide de ce qu'on appelle une API.

Une API, pour *Application Programming Interface* en anglais, ou « interface de programmation d'application » en français, est un ensemble d'outils informatiques (classes, méthodes, fonctions, etc.) fournis dans le but de faciliter le développement d'une application. Dans le cadre du web, une API peut être proposée par un site pour permettre d'interagir avec les données hébergées par le site depuis un programme informatique.

L'intérêt pour un site de fournir une API est de pouvoir contrôler la façon dont les utilisateurs accèdent à ses données. Dans la majorité des cas, la façon dont l'API est codée reste cachée pour l'utilisateur, mais une documentation est fournie pour pouvoir accéder aux fonctionnalités de l'API.

Une API est toujours fournie dans un langage de programmation spécifique. Dans le cadre du web, les API officielles sont le plus souvent proposées en JavaScript. Pour les sites populaires, il existe cependant de très nombreuses API non officielles écrites dans d'autres langages, et codées sous forme de *wrapper* (ou « enveloppe ») autour de l'API officielle : une API Python pourra par exemple faire appel en interne à du code JavaScript fourni par l'API de départ.

Dans ce travail, vous devrez collecter des données depuis un site au choix parmi trois, en utilisant l'API spécifiée par l'énoncé, pour ensuite présenter ces données sous forme de pages web qui seront générées directement par votre code Python. On attend des pages web statiques en HTML, et vous devrez fournir un ou plusieurs fichiers CSS pour la mise en forme de ces pages.

Les trois choix possibles de site web et le travail attendu sont détaillés sur la page suivante. Dans tous les cas, vous devrez proposer un menu en mode texte qui permet d'effectuer les tâches présentées.

## 1. Données sur la NBA depuis basketball-reference.com

API à utiliser :

[https://github.com/vishaalagartha/basketball\\_reference\\_scraper](https://github.com/vishaalagartha/basketball_reference_scraper)

- Données sur les équipes pour la saison en cours

L'utilisateur pourra choisir une équipe dans votre menu, et une page web sera générée avec un récapitulatif des informations sur l'équipe concernée pour la saison en cours : liste des joueurs, résultats, etc.

- Données sur les joueurs

Votre menu devra proposer une liste de 10 joueurs de votre choix, et lorsque l'utilisateur en sélectionnera un, une page web sera générée avec un récapitulatif des informations sur le joueur pour l'ensemble de sa carrière.

## 2. Données sur les sets Lego depuis brickset.com ou rebrickable.com

API à utiliser :

<https://github.com/xxao/brickse>

<https://github.com/xxao/rebrick>

- Données sur les sets Lego pour l'année en cours

L'utilisateur pourra choisir un thème Lego dans votre menu, et une page web sera générée avec la liste des sets sortis cette année pour ce thème.

- Données sur un set spécifique

Votre menu devra proposer une liste de 10 sets Lego de votre choix, et lorsque l'utilisateur en sélectionnera un, une page web sera générée avec un récapitulatif des informations sur le set en question : images, nombre de pièces, etc.

## 3. Données sur les *speedruns* de jeux vidéos depuis speedrun.com

API à utiliser :

<https://github.com/ManicJamie/speedruncompy>

- Données par jeu

Votre menu devra proposer une liste de 10 jeux vidéos de votre choix, et lorsque l'utilisateur en sélectionnera un, une page web sera générée avec un récapitulatif des informations sur les *speedruns* de ce jeu. Vous pourrez vous limiter à la catégorie la plus populaire pour le jeu choisi.

- Données sur les *speedrunners*

Votre menu devra proposer une liste de 10 *speedrunners* de votre choix, et lorsque l'utilisateur en sélectionnera un, une page web sera générée avec un récapitulatif des informations sur la personne en question : liste des jeux pratiqués, temps, classement, etc.