

# Charles - Lesson3

Le parsing/crawling est pseudo-legal

## 1/ Usage des APIs

API : communiquer entre les applis / récupérer les

Premiere question : existe t'il une API disponible ?

Sinon : scraping

Ou : <https://www.kimonolabs.com/> : offrir une API pour les sites Web qui n'ont pas d'API

Intérêt : ne pas maintenir le crawler

une API est assez naturelle : ex "get produit" ou " get titre" etc...  
évite de charger le site avec crawler

exemple : tweet deck

développer par des indépendants pour offrir une API sur tweeter

<https://tweetdeck.twitter.com/>

### **API Rest**

<https://dev.twitter.com/rest/public>

REST : [https://fr.wikipedia.org/wiki/Representational\\_State\\_Transfer](https://fr.wikipedia.org/wiki/Representational_State_Transfer)

Standardisation REST pour l'accès aux ressources

REST = RESTful.

API classiques :

Acronyme : CRUD

Create

Remove

Update

Delete

<http://www.restapitutorial.com/lessons/whatisrest.html>

But : séparer le backend et le front end.

Donc en faisant un appel Get on obtient un retour similaire qqesoit le site

L'API REST retourne du contenu structuré : 2 format : XML et JSON

Ancienne XML. Nouvelle JSON.

## Format JSON :

Format type « Clé , {Valeurs ... Valeurs} »

<http://www.json.org/json-fr.html>

Format JSON :

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        { "value": "New", "onclick": "CreateNewDoc()" },
        { "value": "Open", "onclick": "OpenDoc()" },
        { "value": "Close", "onclick": "CloseDoc()" }
      ]
    }
  }
}
```

**IMPORTANT :** un dictionnaire Python est très proche du format JSON.

intérêt : lire le JSON en une string et le passer sous format texte sur le web

on peut importer la librairie JSON dans python

puis appeler les API pour interroger / travailler avec les string JSON

API You Tube : <https://developers.google.com/youtube/>

<https://developers.google.com/youtube/v3/>

Guide pour s'interfacer avec YouTube Data ou Analytics

Intérêt pour le Site : Permet de maîtriser le contrôle de flux

Limiter le nombre de connexion par jour.

Au coeur des stratégie de monétisation des sites de contenus : permet l'authentification, le suivi d'appel, proposer un tarif

Exemple : <https://www.twilio.com/>

Envoi de SMS lorsqu'une personne fait une action sur le site web

Autre exemple : <https://ifttt.com/>

Glue entre tous les services qui ont des APIs

232 services connectés su iftt

créer une RECIPe entre deux services : If This Then That = IFFT

ex : « IF « un score sur un match sur ESPN » Then « reçoit un SMS

API GitHub : <https://developer.github.com/v3/>

**TODO :** Prochain cours : Utiliser l'API GitHub en Python avec : <https://github.com/PyGithub/PyGithub>

Pour identifier le contenu/format d'un site :  
Le HEADER contient un champ indiquant si c'est du JSON ou du XML

API a utiliser en Python : FLASK  
<http://www.fullstackpython.com/flask.html>  
Import dans python : `from flask import Flask`

Référence : Introduction a Flask <http://blog.miguelgrinberg.com/>

Possibilité de créer la doc automatiquement depuis Flask  
Autre exemple de création de doc : <https://jashkenas.github.io/docco/> pour le JavaScript

Postman est un client REST (ex pour chrome : <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcdcbncdddomop>)

Autre ex : Paw = REST client pour Mac : <https://luckymarmot.com/paw>

Flask va marcher s'il n'y a pas d'authentification imposée par le site server, sinon utiliser le protocole OAuth : <https://fr.wikipedia.org/wiki/OAuth>

## **2/ Utilisation de la librairie PANDAS**

lien : <http://pandas.pydata.org/>

NumPy : introduction du calcul vectoriel en python  
Pandas est basée sur NumPy

Voir Video de 10 minutes

Deux types d'objets importés par Pandas : DataFrame et Series

DataFrame = Tableau  
Series = vecteur / liste

Pandas est l'ensemble des méthodes utilisées sur ces deux objets

=> Pandas étend et enrichi NumPy : plus forcément nécessaire d'utiliser NumPy

**TODO :** Lire les chapitres de Doc Pandas