Les API Web

API : Application Programming Interface

Définition générale : Moyens utilisés par un *logiciel* pour donner accès à certaines fonctionnalités ou ressources

Aujourd'hui : ressources offertes par **des services Web**.

➡️ Moyen d'accéder à des données sur le Web « par programme »
(Web pour les programmes)

Exemple :
L'API Twitter permet d'écrire un client Twitter non officiel

API Rest

REST : REpresentational State Transfer

Modèle le plus utilisé. Défini dans les années 2000. Donne 6 contraintes qui doivent être respectées (Rest sur Wikipedia)

Ici, on se limite à :

- Interrogation d'un serveur qui fournit des informations (et pas de la présentation)
- Pas d'état : l'ordre des requêtes n'a pas d'importance, il n'y a pas de passé, tout est indiqué dans la requête (pas de requête qui règle les résultats en json au lieu de XML, puis une autre qui fait la requête en question).

On peut avoir deux objectifs :

- programmer un serveur Web qui **propose une API**
- **utiliser une API existante**.

Utiliser une API existant

Aujourd'hui, on va plutôt se concentrer sur le point 2 ➡️ quel service ?

- The movie DB
- ~~ISSNDB~~
- Api Nasa
- open notify (ISS)
- Wikipedia query et opensearch (non rest)
- oeis
- Openweathermap
- Openfoodfacts
- Openlibrary
- ... un autre de votre choix

Conditions d'utilisation

- l'accès peut être gratuit ou payant
- avec ou sans authentification
- restrictions (nombre de requêtes pas seconde).

Authentification :

- token (API Key) passé en paramètre
- token (API Key) passé en entête
- authentification plus complexe (Rest ?)

Exemples de requêtes

```
import requests
```

tmdb

```
>>> url = "http://api.themoviedb.org/3/\
       person/85/movie_credits?api_key=XXX"
>>> r = requests.get(url)
>>> resultat = r.json()
>>> resultat["crew"][0]

{'backdrop_path': '/tjvFUGlvqERs9LeRNZcxV2AFLrL.jpg',
 'department': 'Directing',
 'genre_ids': [18],
 'id': 9956,
 'job': 'Director',
 'original_title': 'The Brave',
 'overview': 'A down-on-his-luck American Indian recently
              released from jail is offered the chance to
              "star" as the victim of a snuff film, the
              resulting pay of which could greatly help his
              poverty stricken family.',
 'popularity': 5.946,
 'release_date': '1997-07-30',
 'vote_average': 6,
 'vote_count': 84}
```

Openlibrary

```
import requests
isbn = '0201558025'
url = "http://openlibrary.org/api/books?
       bibkeys=ISBN:{}&format=json&jscmd=data".format(isbn)
r = requests.get(url)
rep = r.json()
for book in rep.values():
    print(book['title'])
    for author in book['authors']:
        print("  ", author['name'])
```

nasa

```
>>> url="https://api.nasa.gov/neo/rest/v1/feed?\
        start_date=2019-04-23&end_date=2019-04-29&api_key=XXX"
>>> r = requests.get(url)
>>> result = r.json()
>>> result["near_earth_objects"]["2019-04-29"][0]

{'absolute_magnitude_h': 22.7,
 'close_approach_data': [{'close_approach_date': '2019-04-29',
   'epoch_date_close_approach': 1556521200000,
   'miss_distance': {'astronomical': '0.391245454',
    'kilometers': '58529488',
    'lunar': '152.1944885254',
    'miles': '36368536'},
   'orbiting_body': 'Earth',
   'relative_velocity': {'kilometers_per_hour': '19627.30460',
    'kilometers_per_second': '5.4520290559',
    'miles_per_hour': '12195.6437952482'}}],
  'id': '3831462',
 'is_potentially_hazardous_asteroid': False,
 'is_sentry_object': False,
 }
```

open notify

```
>>> url = "http://api.open-notify.org/iss-pass.json?\
           lat=45.0&lon=-122.3"
>>> r = requests.get(url)
>>> result = r.json()
>>> result

{'message': 'success',
 'request': {'altitude': 100,
  'datetime': 1556007050,
  'latitude': 45.0,
  'longitude': -122.3,
  'passes': 5},
 'response': [{'duration': 499, 'risetime': 1556026954},
  {'duration': 644, 'risetime': 1556032641},
  {'duration': 629, 'risetime': 1556038454},
  {'duration': 618, 'risetime': 1556044292},
  {'duration': 645, 'risetime': 1556050100}]}
```

wikipedia

```
import requests

url = "https://fr.wikipedia.org/w/api.php?\
       action=opensearch&search=Python"
r = requests.get(url)
result = r.json()
```

Voir aussi à query

oeis

```
>>> url = "https://oeis.org/search?fmt=json\
                                    &q=1,1,2,3,5,8&start=0"
>>> r = requests.get(url)
>>> result = r.json()
>>> result["results"][0]["name"], result["results"][0]["id"]

'Fibonacci numbers: F(n) = F(n-1) + F(n-2)
 with F(0) = 0 and F(1) = 1.', 45

>>> url = "https://oeis.org/A{:06d}/graph?png=1".format(45)
>>> ...
>>>
```