TRAITEMENT DE L'INFORMATION

Ophélie Fraisier
ophelie.fraisier@irit.fr

2018 — 2019

WEB SCRAPING

- Définition
- Structure d'un ficher HTML
- Présentation de la bibliothèque Requests
- Présentation de la bibliothèque BeautifulSoup

DÉFINITION DU WEB SCRAPING

- Récolte de données à partir de pages web à l'aide d'un programme
- Peu de contraintes : tout ce qui est visible à l'écran est techniquement récupérable
- Règle de bonne conduite : délai entre les requêtes pour éviter de surcharger le serveur web
  - Risque de blocage de l'adresse IP

- Peut être illégal : certains sites interdisent la récupération automatique de leur contenu
- La qualité du contenu récupéré dépend de la qualité du formatage de la page HTML

STRUCTURE D'UN FICHIER HTML

- Balises
  - Attributs
    - Identifiants (chaque valeur est unique dans le document)
    - Classes

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Proverbes</title>
      <link rel="stylesheet" type="text/css" href="style.css">
   </head>
   <body><!-- éléments visibles à l'écran -->
      <h1 id="titre_principal">Proverbe du jour</h1>
      <div class="proverbe">
         <p lang="en">Once bitten, twice shy.</p>
         <p lang="fr">Chat échaudé craint l'eau froide.</p>
      </div>
   </body>
</html>
```

STRUCTURE D'UN FICHIER HTML : BALISES

`<title>`	Titre de la page
`<link />`	Lien vers une feuille de style
`<meta />`	Métadonnée de la page (mots-clés, encodage, auteur, …)
`<div><span>`	Balises génériques pour organiser le contenu, particulièrement intéressantes lorsqu'elles sont associées à un attribut `class` ou `id`
`<h1> à <h6>`	Titres
`<p>`	Paragraphe
`<a>`	Lien hypertexte
`<img />`	Image
` `	Retour à la ligne
`<abbr>`	Abréviation
`<ul><ol><dl>`	Listes
`<table>`	Tableau
`<form>`	Formulaire
`<style>`	Code CSS

WEB SCRAPING EN PYTHON — REQUESTS

- Bibliothèque **Requests** pour charger les pages web
  - *Doc : http://docs.python-requests.org/*
  - `r = requests.get(url)` : chargement d'une page
  - `r = requests.get(url, params={key: value})` : chargement d'une page avec paramètres dans l'url
  - `r.text` : récupération du contenu
  - `r.json()` : récupération d'un dictionnaire à partir d'un contenu au format JSON
  - `r.status_code` : code retourné par la requête
    - Une requête réussie renvoie le code 200
    - Il est conseillé de gérer le cas où ce code est différent de 200 (success)

WEB SCRAPING EN PYTHON — BEAUTIFULSOUP

- Bibliothèque **BeautifulSoup (bs4)** pour traiter le code HTML
  - *Doc : https://www.crummy.com/software/BeautifulSoup/bs4/doc/*

- `soup = bs4.BeautifulSoup(html_code[, "lxml"])`
  - `lxml` : module lisant le code HTML, argument optionnel

- `soup.select(X)` : renvoie une **liste** de balises ayant les caractéristiques demandées
  - `soup.select("title")` : balises **<title></title>**
  - `soup.select("body a")` : balises **<a>** inclues dans une balise **<body></body>**
  - `soup.select(".sister")` : balises avec **classe « sister »**
  - `soup.select("#link")` : balises ayant comme **id « link »**
  - Combinaisons : `soup.select("body.sister a#link")`

- `for balise in soup.select(X):`
  - `balise.get_text()` : texte de la balise
  - `balise.get("href")` : valeur de l'attribut href de la balise, `None` si la balise n'a pas d'attribut href

WEB SCRAPING EN PYTHON

```
▼ <div class="yt-lockup-dismissable">
  ▶ <div class="yt-lockup-thumbnail contains-addto">…</div>
  ▼ <div class="yt-lockup-content">
    ▼ <h3 class="yt-lockup-title contains-action-menu">
      ▼ <a href="/watch?v=zaD84DTGULo" class="yt-uix-sessionlink yt-ui-ellipsis yt-ui-ellipsis-2 spf-link " data-sessionlink=
      "itct=CIoBEJQ1GAAiEwi1wPTH1L7PAhVRDBwKHTujDuQojh4yCmctaGlnaC1yZWNaD0ZFd2hhdF90b193YXRjaA" title="Police Accountability: Last Week Tonight with John Oliver
      (HBO)" aria-describedby="description-id-645303" dir="ltr">
          "Police Accountability: Last Week Tonight with John Oliver (HBO)"
          ::after
        </a>
        <span class="accessible-description" id="description-id-645303"> - Durée : 19:55.</span>
      </h3>
    ▶ <div class="yt-lockup-byline">…</div>
    ▶ <div class="yt-lockup-meta">…</div>
    ▶ <div class="yt-uix-menu-container yt-lockup-action-menu">…</div>
    </div>
  </div>
  <div class="yt-lockup-notifications-container hid"></div>
  ::after
</div>
```

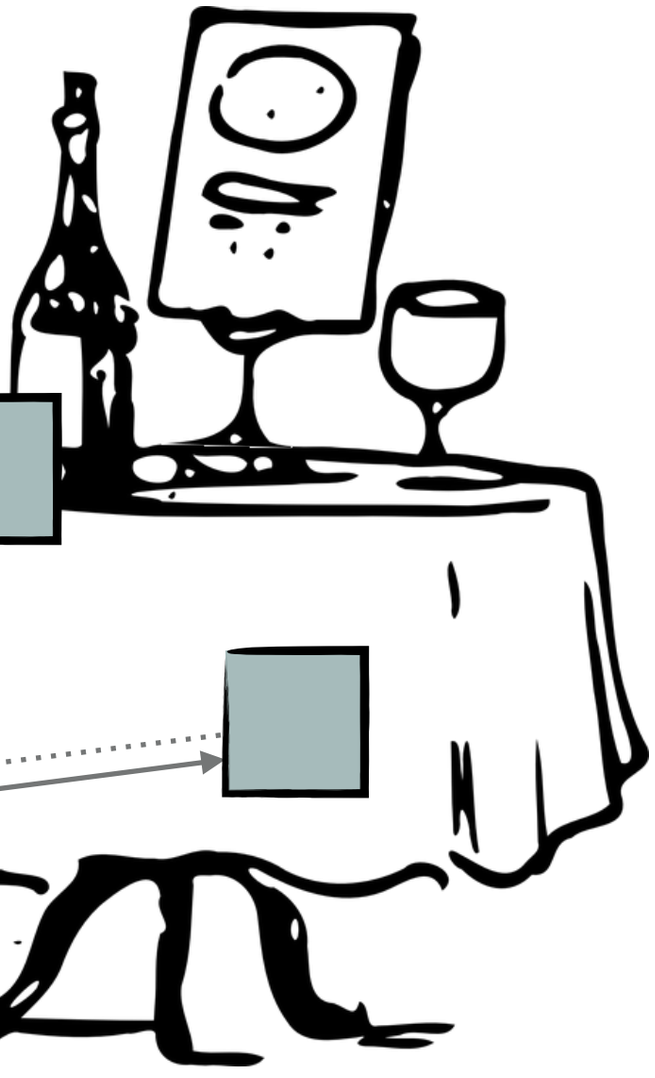

```python
import requests
import bs4 # BeautifulSoup

response = requests.get("http://www.youtube.com")
if response.status_code == 200:
    soup = bs4.BeautifulSoup(response.text, "lxml")
    titles_elements = soup.select("h3.yt-lockup-title a")
    print([a.get("title") for a in titles_elements])
```

API

- Définition d’une API
- API web
- Présentation de l’API Twitter

DÉFINITION D'UNE API



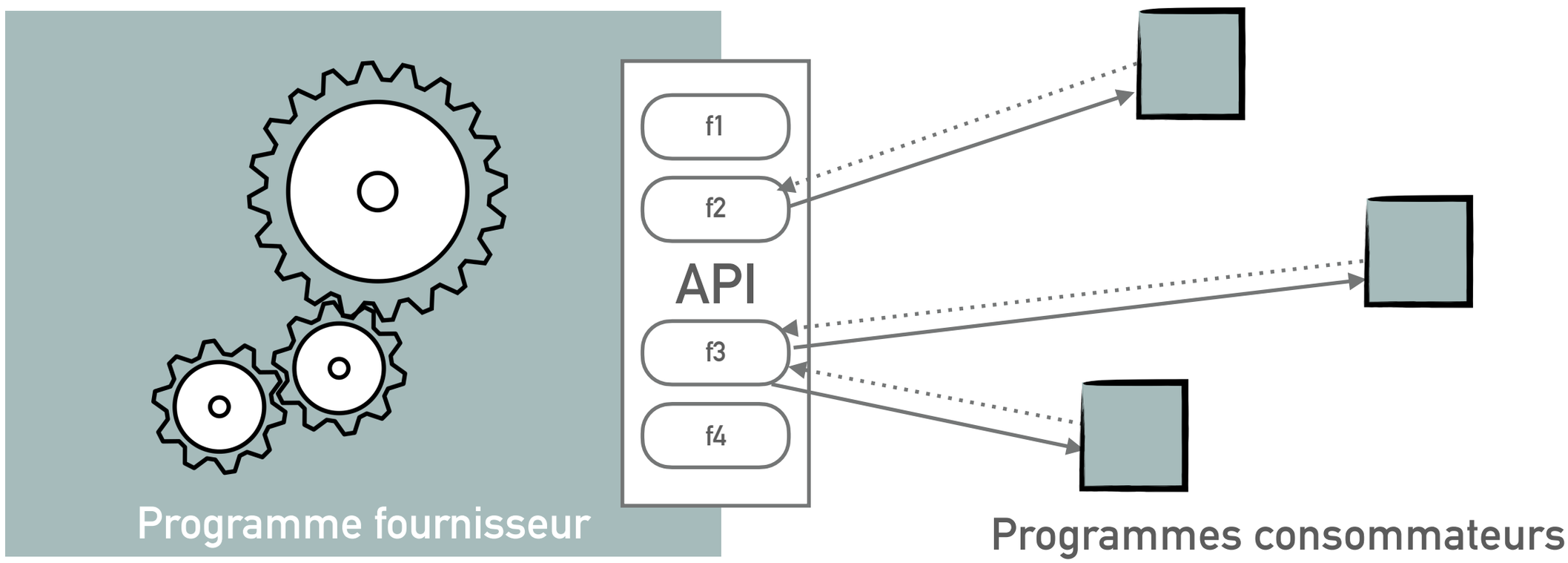
Cuisine

Service

Clients et clientes

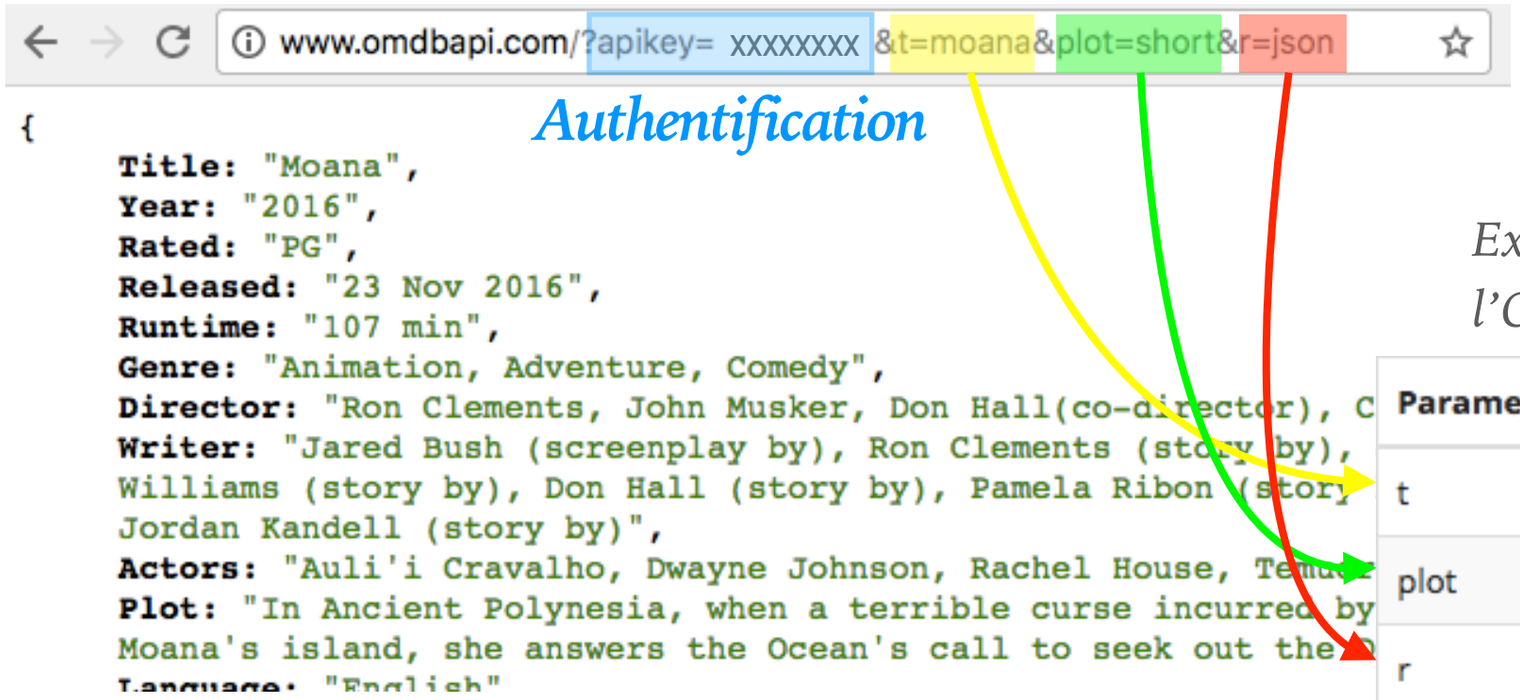
DÉFINITION D'UNE API

- API = Application Programming Interface
- Ensemble de fonctions servant de façade entre un programme fournisseur et des programmes consommateurs
- Les services prodigués par le programme fournisseur sont décrits formellement dans la **documentation** de l'API



API WEB

- Requêtes aux différentes fonctions de l'API par URL
- Données retournées en XML ou JSON
- Largement répandues aujourd'hui
  105 répertoriées en 2005, 9000 en 2013

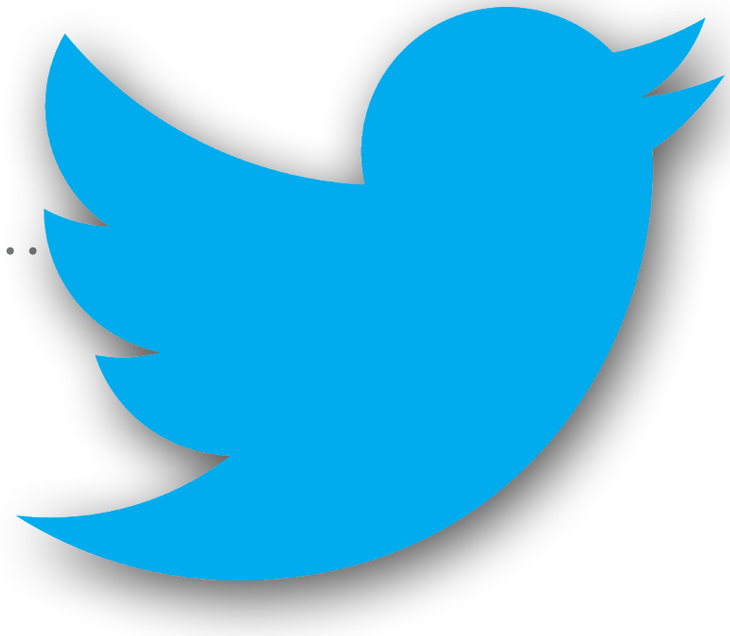


Extrait de la documentation de l'Open Movie Database API

Parameter	Description
t	Movie title to search for.
plot	Return short or full plot.
r	The data type to return.

```
keys = {"apikey": "<clé>", "t": "moana",
        "plot": "full", "r": "json"}
response = requests.get("http://www.omdbapi.com", params=keys)
dict = response.json()
```

API TWITTER



- Twitter : réseau social permettant de
  - Publier de courts messages appelés tweets
  - Suivre des utilisateurs et utilisatrices
  - Retweeter un tweet = le partager à ses propres followers
  - Répondre à un tweet
  - Aimer un tweet

- 2 API web disponibles avec authentification :
  - Streaming API : pour surveiller des tweets en temps réel
  - **REST API : pour faire des requêtes sur les tweets des 7 derniers jours**
- Les données sont renvoyées au format JSON

INTERROGER L'API REST AVEC PYTHON

- Utilisation du package **twitter**
  - *Doc : https://pypi.python.org/pypi/twitter*

```
from twitter import Twitter, OAuth

# informations d'authentification
tokens = OAuth(ACCESS_TOKEN, ACCESS_SECRET,
               CONSUMER_KEY, CONSUMER_SECRET) # à modifier
# connection à l'API
# retry=True : l'application retente la dernière requête
# si les limites de l'API sont atteintes
t = Twitter(auth=tokens, retry=True)
# récupérer les derniers tweets d'un utilisateur
tweets = t.statuses.user_timeline(screen_name="billybob")
# récupérer les tweets contenant un mot-clé
tweets = t.search.tweets(q="#pycon")
```

OBTENIR LES DONNÉES D'AUTHENTIFICATION

1. Créer un **compte Twitter** : https://twitter.com
   Vous devrez renseigner un numéro de téléphone pour pouvoir créer ensuite une application.
2. Créer une **application Twitter** : https://developer.twitter.com
3. Aller dans l'onglet « **Apps** », puis « **Details** » et enfin « **Keys and Tokens** »
4. Créer un nouvel « **Access Token** »
5. Les informations nécessaires sont maintenant disponibles :
   - *Consumer Key (API Key)*
   - *Consumer Secret (API Secret)*
   - *Access Token*
   - *Access Token Secret*

API REST : DOCUMENTATION

Limites d'utilisation	https://developer.twitter.com/en/docs/basics/rate-limits
Recherche de tweets	https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets
Récupération des tweets d'un compte	https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline
Followers d'un compte donné	https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/
Comptes suivis par un compte donné	https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/
Récupération de tweets à partir de leurs identifiants	https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-lookup