



Comment scraper des données avec Scrapy ?

- Configuration de l'environnement
- Création d'un projet
- Configuration du projet
- Repérage des données à scraper
- Création du crawler

1. Configuration de l'environnement

« Les environnements sont des espaces cloisonnés. Ils permettent d'installer des paquets sans que cela n'ait d'impact sur le reste du système. C'est notamment très utile lorsque l'on est confronté au besoin de travailler sur deux projets faisant appel à des versions distinctes d'une même librairie. L'usage des environnements évite alors les conflits de versions. Cette pratique permet de faciliter le travail en équipe. On peut ainsi s'assurer que tous les développeurs disposent du même environnement que celui sur lequel l'application sera déployée en production. »

Source : <https://www.actuia.com/actualite/5-choses-a-savoir-sur-avant-de-commencer-a-programmer-en-python>

Pour créer votre environnement, rendez-vous sur Youtube pour suivre notre tutoriel: <https://youtu.be/pVME6JvdD5g>

2. Création d'un projet

Votre environnement créé, il va falloir installer scrapy. Pour cela, lancez l'anaconda prompt et activez votre environnement avec la commande :

activate nomdelenvironnement (voir vidéo ci-dessus)

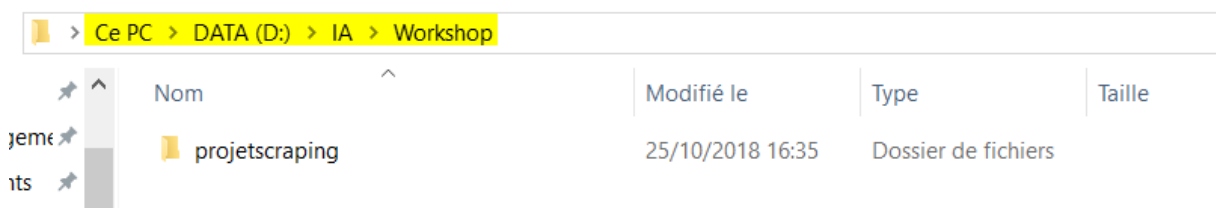
Puis tapez la commande suivante pour installer scrapy (toujours dans votre environnement)

conda install -c conda-forge scrapy

```
(base) D:\IA\Projets\Scaper Workshop IALAB\scraperialab>activate ialab
(ialab) D:\IA\Projets\Scaper Workshop IALAB\scraperialab>conda install -c conda-forge scrapy
Solving environment: /
```

(Vous pouvez voir sur la capture d'écran ci-dessus l'activation de l'environnement ialab). Une fois que scrapy a été installé, vous pouvez vérifier que tout a bien fonctionné en important scrapy dans un éditeur Python.

Vous allez maintenant créer un dossier dans lequel vous installerez python. Dorénavant, toute commande que vous taperez le sera dans votre environnement contenant scrapy.



Dans Anaconda Prompt, rendez-vous dans votre dossier d'installation :

cd D:\IA\Workshop

Puis créez votre projet :

scrapy startproject MonScraper

Ensuite, rendez vous dans le dossier spiders de votre fichier d'installation et créez un template:

scrapy genspider monCrawler <https://www.transfermarkt.fr/fc-paris-saint-germain/startseite/verein/583>

```
(ialab) D:\IA\Projets\Scaper Workshop IALAB\scraperialab>cd D:\IA\Workshop

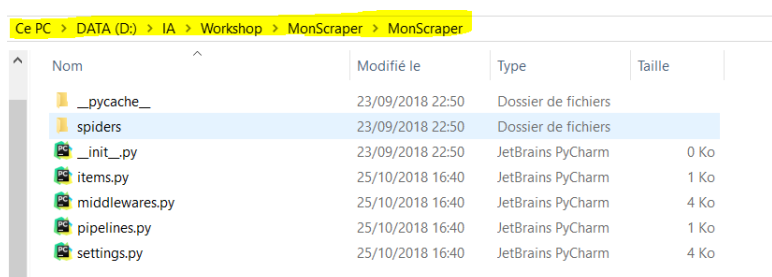
(ialab) D:\IA\Workshop>scrapy startproject MonScraper
New Scrapy project 'MonScraper', using template directory 'd:\programme\anaconda\envs\ialab\lib\site-packages\scrapy\templates\project', created in:
D:\IA\Workshop\MonScraper

You can start your first spider with:
cd MonScraper
scrapy genspider example example.com

(ialab) D:\IA\Workshop>scrapy genspider monCrawler https://www.transfermarkt.fr/fc-paris-saint-germain/startseite/verein/583
Created spider 'monCrawler' using template 'basic'
```

(Si la commande scrapy n'est pas reconnue, vous êtes peut-être dans le mauvais environnement ou vous n'avez peut-être pas correctement installé scrapy).

Si vous retournez dans votre dossier d'installation, le projet a bien été créé.



3. Configuration du projet

Comme vous avez pu le voir, nous avons décidé de scraper un site de statistiques de football, conformément aux projets que nous proposons cette année, à savoir prédire la valeur d'un joueur de football en fonction de ses performances.

Beaucoup de sites possèdent une sécurité pour repousser les scrapers. Nous allons donc devoir configurer notre programme pour passer cette sécurité. Pour cela, nous allons :

- Changer notre User Agent et se faire passer pour un navigateur Mozilla
- Refuser de suivre les redirections pour les bots
- Limiter le nombre de requêtes simultanées à 1
- Limiter la fréquence des requêtes à 1 sec

Pour cela, ouvrez le fichier python settings.py dans votre dossier d'installation.

Remplacez :

- User Agent par 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36'
- ROBOTSTXT_OBEY par False
- CONCURRENT_REQUESTS par 1
- DOWNLOAD_DELAY par 1

(Le tout en enlevant les # à chaque fois)

```
# Crawl responsibly by identifying yourself (and your website) on the user-agent
USER_AGENT = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36'

# Obey robots.txt rules
ROBOTSTXT_OBEY = False

# Configure maximum concurrent requests performed by Scrapy (default: 16)
CONCURRENT_REQUESTS = 1

# Configure a delay for requests for the same website (default: 0)
# See https://doc.scrapy.org/en/latest/topics/settings.html#download-delay
# See also autothrottle settings and docs
DOWNLOAD_DELAY = 1
```

4. Repérage des données à Scraper

Une fois les modifications faites, enregistrez votre fichier settings.py puis fermez Anaconda Prompt. Rouvrez ensuite Anaconda Prompt pour que les modifications soient effectuées. Activez votre environnement puis rendez vous dans votre dossier d'installation et tapez :

scrapy shell

Vous accédez ensuite à la console scrapy. La prochaine ligne de commande à taper est :

fetch('https://www.transfermarkt.fr/fc-paris-saint-germain/startseite/verein/583')

Cela vous permet de scraper l'intégralité du site. Vous obtenez 3 réponses possibles :

- 200 : vous avez réussi à accéder à la page
- 300 : impossible de vous connecter
- 404 : vous avez été repéré, accès impossible


















```
2018-10-25 17:08:29 [scrapy.extensions.telnet] DEBUG: Telnet console listening on 127.0.0.1:6023
[s] Available Scrapy objects:
[s] scrapy scrapy module (contains scrapy.Request, scrapy.Selector, etc)
[s] crawler <scrapy.crawler.Crawler object at 0x000002AB663CADD8>
[s] item {}
[s] settings <scrapy.settings.Settings object at 0x000002AB68AB3940>
[s] Useful shortcuts:
[s] fetch(url[, redirect=True]) Fetch URL and update local objects (by default, redirects are followed)
[s] fetch(req) Fetch a scrapy.Request and update local objects
[s] shelp() Shell help (print this help)
[s] view(response) View response in a browser
In [1]: fetch('https://www.transfermarkt.fr/fc-paris-saint-germain/startseite/verein/583')
2018-10-25 17:08:49 [scrapy.core.engine] INFO: Spider opened
2018-10-25 17:08:50 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.transfermarkt.fr/fc-paris-saint-germain/s
tartseite/verein/583> (referer: None)
```

Nous avons maintenant accès au site, il va donc falloir sélectionner les données.

Pour cela, nous allons utiliser un selector (voir documentation en cliquant sur le lien suivant) : <https://doc.scrapy.org/en/latest/topics/selectors.html>

Je vous propose dans un premier de récupérer l'URL de chaque joueur :

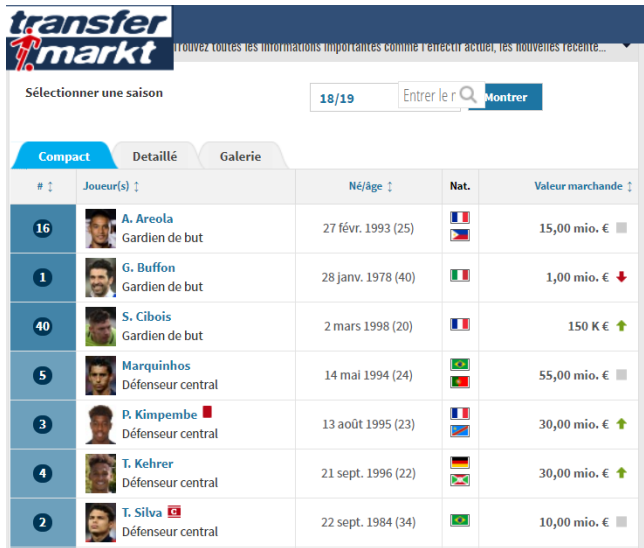
Lien : <https://www.transfermarkt.fr/fc-paris-saint-germain/startseite/verein/583>

Compact					
Detaillé					
Galerie					
#	Joueur(s)	Né/âge	Nat.	Valeur marchande	
16	 Alphonse Areola Gardien de but	27 févr. 1993 (25)		15,00 mio. €	■
1	 Gianluigi Buffon Gardien de but	28 janv. 1978 (40)		1,00 mio. €	↓
40	 Sébastien Cibois Gardien de but	2 mars 1998 (20)		150 K €	↑
5	 Marquinhos Défenseur central	14 mai 1994 (24)		55,00 mio. €	■
3	 Presnel Kimpembe Défenseur central	13 août 1995 (23)		30,00 mio. €	↑
4	 Thilo Kehrer Défenseur central	21 sept. 1996 (22)		30,00 mio. €	↑
2	 Thiago Silva Défenseur central	22 sept. 1984 (34)		10,00 mio. €	■
20	 Layvin Kurzawa Arrière gauche	4 sept. 1992 (26)		20,00 mio. €	↓
14	 Juan Bernat Arrière gauche	1 mars 1993 (25)		10,00 mio. €	↓
	 Stanley N'Soki Arrière gauche				

Comment sélectionner des données ?

Il faut repérer sur la page quelles sont les données dont vous avez besoin, en l'occurrence l'URL renvoyant sur la page de chaque joueur.

Vous avez donc inspecté le code source de la page pour repérer où il se situe.



#	Joueur(s)	Né/âge	Nat.	Valeur marchande
16	A. Areola Gardien de but	27 févr. 1993 (25)		15,00 mio. €
1	G. Buffon Gardien de but	28 janv. 1978 (40)		1,00 mio. €
40	S. Cibois Gardien de but	2 mars 1998 (20)		150 K €
5	Marquinhos Défenseur central	14 mai 1994 (24)		55,00 mio. €
3	P. Kimpembe Défenseur central	13 août 1995 (23)		30,00 mio. €
4	T. Kehrer Défenseur central	21 sept. 1996 (22)		30,00 mio. €
2	T. Silva Défenseur central	22 sept. 1984 (34)		10,00 mio. €

```
<tr class="even">...</tr>
<tr class="odd">...</tr>
<tr class="even selected">
  <td title="Défense" class="zentriert rueckennummer
bg_Abwehr">...</td>
  <td title class="posrela">
    <table class="inline-table" title>
      <tbody>
        <tr>
          <td rowspan="2" class=">...</td>
          <td class="hauptlink">
            <div class="di nowrap">
              <span class="hide-for-small">
                <a class="spielprofil tooltip
tooltipstered" id="181767" href="/
marquinhos/profil/spieler/181767">
Marquinhos</a> == $0
              </span>
            </div>
          </td>
        </tr>
      </tbody>
    </table>
  </td>
</tr>
<tr>...</tr>
</tbody>
</table>
</td>
<td class="hide" itemprop="athlete">Marquinhos</td>
<td class="zentriert">14 mai 1994 (24)</td>
<td class="zentriert">...</td>
<td class="rechts hauptlink">...</td>
</tr>
```

Fonctionnement des sélecteurs :

Les données sont sélectionnées selon leur tag HTML. Voici quelques commandes dont vous pouvez vous inspirer :

- `response.xpath('//h1').extract()` → Sélectionne et extrait le contenu de chaque balise h1
- `response.xpath('//div[@id="not-exists"]/text()')` → Sélectionne le text des div dont l'id est « not-exists »
- Voir documentation ci-dessus pour plus d'informations...

On peut voir dans notre cas que l'url est dans une balise dont la classe est « hide-for-small ».

On peut donc essayer la commande : `response.xpath('//*[@class="hide-for-small"]')`

```
In [5]: response.xpath('//*[@class="hide-for-small"]')
Out[5]:
[<Selector xpath='//*[@class="hide-for-small"]' data='<a name="Logo" href="/" id="logo-home" c>',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="A">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="En provenance de: Juventus Tu">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="G">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="S">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="M">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="P">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="En provenance de: FC Schalke">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="T">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="T">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="L">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="En provenance de: FC Bayern M">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="J">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="Changement interne: FC Paris S">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="S">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="L">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="D">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="Changement interne: FC Paris S">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="C">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="L">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="M">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="A">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="C">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="Changement interne: FC Paris S">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="A">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="J">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="Changement interne: FC Paris S">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="Y">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="N">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="Revenu apres un prêt à: Stoke">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="J">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="En provenance de: AS Monaco d">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="K">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="A">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="Changement interne: FC Paris S">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="M">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="E">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="En provenance de: Stoke City">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="E">',
<Selector xpath='//*[@class="hide-for-small"]' data='<a title="Changement interne: FC Paris S">',
<Selector xpath='//*[@class="hide-for-small"]' data='<span class="hide-for-small"><a title="T">]
```

Super ça fonctionne ! Allons maintenant un peu plus loin en sélectionnant uniquement le lien :

response.xpath('//*[@class="hide-for-small"]/a/@href').extract()

```
In [6]: response.xpath('//*[@class="hide-for-small"]/a/@href').extract()
Out[6]:
['/alphonse-areola/profil/spieler/120629',
'/gianluigi-buffon/profil/spieler/5023',
'/sebastien-cibois/profil/spieler/395251',
'/marquinhos/profil/spieler/181767',
'/presnel-kimpembe/profil/spieler/282041',
'/thilo-kehrer/profil/spieler/228948',
'/thiago-silva/profil/spieler/29241',
'/layvin-kurzawa/profil/spieler/126710',
'/juan-bernat/profil/spieler/126719',
'/stanley-nsoki/profil/spieler/371141',
'/thomas-meunier/profil/spieler/100986',
'/dani-alves/profil/spieler/15951',
'/colin-dagba/profil/spieler/460629',
'/lassana-diarra/profil/spieler/23914',
'/marco-vernatti/profil/spieler/102558',
'/adrien-rabiot/profil/spieler/182913',
'/christopher-nkunku/profil/spieler/344381',
'/antoine-bernedi/profil/spieler/395512',
'/julian-draxler/profil/spieler/85148',
'/yacine-adli/profil/spieler/395236',
'/neymar/profil/spieler/68290',
'/jese/profil/spieler/134936',
'/kylian-mbappe/profil/spieler/342229',
'/angel-di-maria/profil/spieler/45320',
'/moussa-diaby/profil/spieler/395516',
'/edinson-cavani/profil/spieler/48280',
'/eric-maxim-choupo-moting/profil/spieler/45660',
'/timothy-weah/profil/spieler/370846']
```

Nous avons
récupérer

tous
les

liens. Je vous propose maintenant d'aller récupérer quelques informations sur un joueur, pour ensuite tester notre algorithme de scraping à plus grande échelle.

```
fetch('https://www.transfermarkt.fr/fc-paris-saint-germain/startseite/verein/583')
```

Pour récupérer l'âge :

```
response.xpath('//*[@itemprop="birthDate"]/text()').extract_first()
```

Pour récupérer le prénom et le nom :

```
response.xpath('//*[@itemprop="name"]/text()').extract()
```

```
response.xpath('//*[@itemprop="name"]/b/text()').extract()
```

Je vous invite à chercher de votre côté d'autres informations comme le poste, la nationalité, le nombre de buts marqués dans la saison (un peu plus difficile...).

5. Création du crawler

Maintenant que vous savez comment récupérer vos données, il va falloir automatiser le processus de récupération de ces données : c'est le crawling.

Vous allez donc ouvrir le document python que vous avez généré précédemment en créant le template (monCrawler dans mon cas). Il se trouve normalement dans votre dossier d'installation.

```
import scrapy
from scrapy.http import Request

class MoncrawlerSpider(scrapy.Spider):

    name = 'monCrawler'
    allowed_domains = ['www.transfermarkt.fr']
    start_urls = ['https://www.transfermarkt.fr/fc-paris-saint-germain/startseite/verein/583/']
```

Dans un premier temps, vous allez importer Request, juste en dessous de « import scrapy »

```
from scrapy.http import Request
```

Vous allez ensuite changer le allowed_domains en le remplaçant par :

```
allowed_domains = ['www.transfermarkt.fr']
```

/!\ Attention, il se peut que le start_urls possède plusieurs fois « https:// », si c'est le cas, il faut en lancer qu'un seul, sinon votre crawler ne fonctionnera pas.

Dans la fonction parse, vous allez créer une variable urls puis modifier chaque url car nous récupérons des morceaux d'url et il faut les transformer pour arriver à la page « stats détaillés » de chaque joueur.


```
def parse(self, response):

    # On va chercher les urls des joueurs
    urls = response.xpath('//*[@class="hide-for-small"]/a/@href').extract()

    # On transforme les morceaux d'urls récupérés pour qu'ils correspondent à ceux des joueurs

    for url in urls:
        urlcomplet = response.urljoin(url)
        urlcomplet = urlcomplet.replace("profil", "leistungsdaten")
        urlcomplet += "/saison/2018/plus/1#gesamt"

    # On renvoie les urls récupérés vers une nouvelle fonction : parse_data
    yield Request(urlcomplet, callback=self.parse_data)
```

On crée ensuite la fonction `parse_data`, qui va aller scraper les différentes données concernant nos joueurs de foot (âge, nom, prénom...).

```
def parse_data(self, response):
    age = response.xpath('//*[@itemprop="birthDate"]/text()').extract_first()
    prenom = response.xpath('//*[@itemprop="name"]/text()').extract()
    nom = response.xpath('//*[@itemprop="name"]/b/text()').extract()




    yield {"nom": nom,
          "prenom": prenom,
          'age': age}
```

Une fois cette étape terminée, vous pouvez enregistrer et nous allons lancer votre crawler.

Votre fichier py « `monCrawler` » doit être dans le dossier `spiders` de votre dossier d'installation. Si ce n'est pas le cas, déplacez-le dans le bon fichier.

rtage Affichage

Ce PC > DATA (D:) > IA > Workshop > MonScraper > MonScraper > spiders

Nom	Modifié le	Type	Taille
 <code>__pycache__</code>	25/10/2018 22:47	Dossier de fichiers	
 <code>__init__.py</code>	23/09/2018 22:50	JetBrains PyCharm	1 Ko
 <code>monCrawler.py</code>	25/10/2018 22:48	JetBrains PyCharm	2 Ko

Lancez ensuite un nouveau prompt anaconda et rendez-vous dans le dossier `spiders`, puis tapez la commande suivante pour lancer votre crawler :

scrapy crawl monCrawler

En ajoutant `-o data.csv` à cette commande, vous pourrez enregistrer ce fichier dans un fichier csv.

```
(ialab) D:\>cd D:\IA\Workshop\MonScraper\MonScraper\spiders  
(ialab) D:\IA\Workshop\MonScraper\MonScraper\spiders>scrapy crawl monCrawler -o data.csv
```

Une fois le crawling terminé, vous retrouvez les données dans un document texte. Vous avez réussi à scraper des données !!



nom,prenom,age	
Areola,Alphonse ,"	27 févr. 1993
Weah,Timothy ,"	22 févr. 2000
Choupo-Moting,Eric Maxim ,"	23 mars 1989
Cavani,Edinson ,"	14 févr. 1987
Diaby,Moussa ,"	7 juil. 1999
Di María,Ángel ,"	14 févr. 1988
Mbappé,Kylian ,"	20 déc. 1998
Jesé,,"	26 févr. 1993
Neymar,,,"	

Pour accéder à un code un peu plus détaillé, rendez-vous sur notre github :

<https://github.com/matrousseau/transfermarkt-crawler-with-scrapy>