Auteur: CAMARA Laby Damaro
Titre: Découvrir Le WebScring & Le Crawling avec Le Framework Scrapy

Crawler En Python

Crawler signifie littéralement **« scanner »**. Autrement dit, il s'agit d'extraire un maximum d'informations possibles d'un site web. Cette analyse permet ainsi de connaître parfaitement la structure d'un site et de résoudre ses problèmes éventuels. Par exemple, une arborescence mal construite, un maillage interne inadéquat ou encore des balises meta dupliquées.

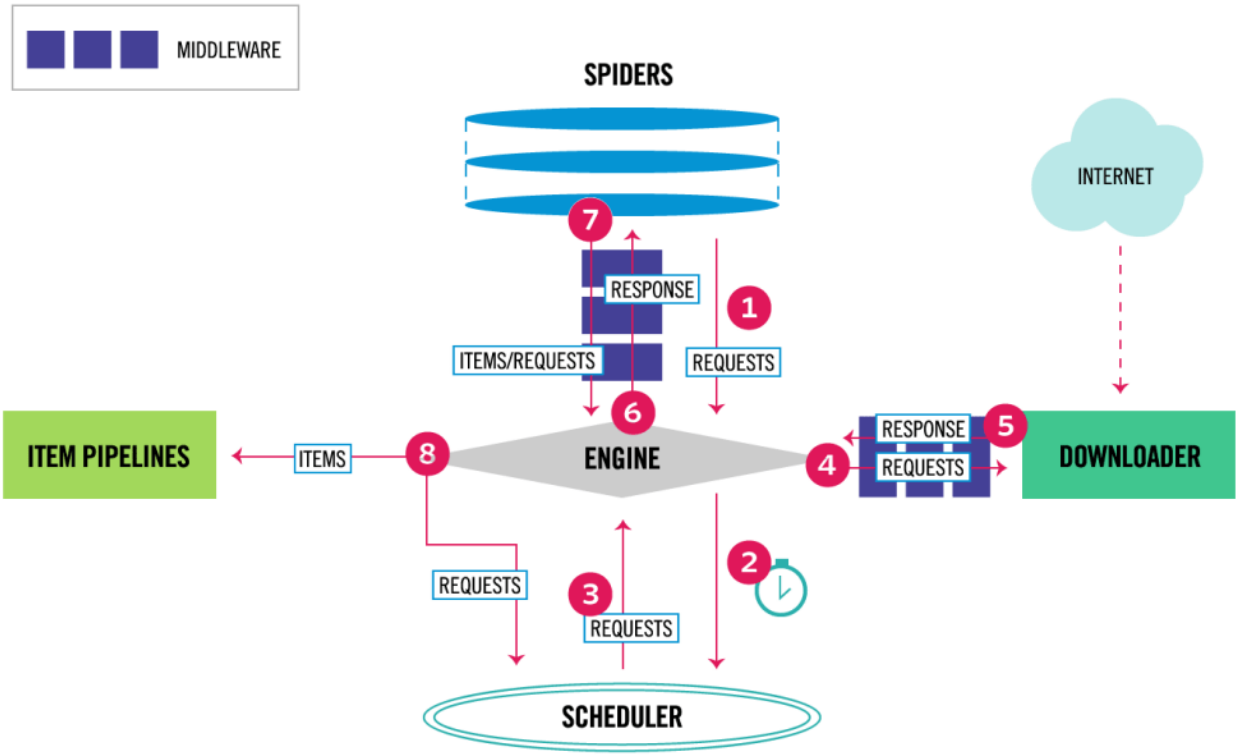
Importance Des Données

L'importance de l'acquisition de données pour le **Data Scientist** n'est plus à démontrer. Le web étant une source intarissable de données de toutes sortes, le web scraping ou web crawling s'est imposé comme une technique incontournable d'acquisition de données. Scrapy est un framework Python permettant de faciliter les tâches de scraping. Dans cet article nous verrons commant utiliser Scrapy pour créer un jeu de données de textes écrits en langage naturel. Nous commencerons par voir les bases de Scrapy puis nous créerons un jeu de données constitué du site jumia.

Qu'est-ce que Scrapy ?

Scrapy est un framework Python qui sert à faire du web scraping. Scrapy est adapté aux grands projets de web scraping. En effet, les projets Scrapy ont une structure assez clair qui facilite la maintenance et le passage à l'échelle. En plus, le framework offre une certaine rapidité due à l'asynchronisme des requêtes (Scrapy utilise Twisted). Bref! Je vous propose de jeter un œil à la structure de Scrapy.

Data flow de Scrapy



Création D'un Projet Crawl

Avant de créer le projet, il faut s'assurer d'avoir Scrapy installé. Utilisez la ligne de commande suivante pour l'installer :

```
pip install scrapy
```

Ou avec conda :

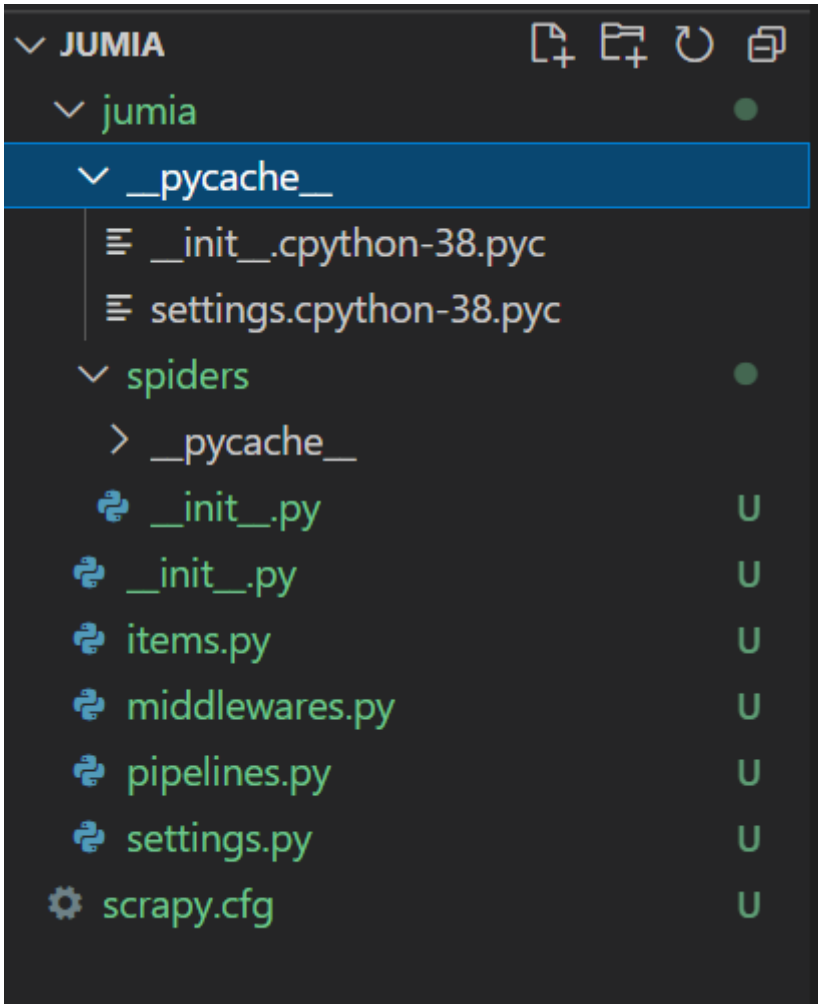
```
conda install -c conda-forge scrapy
```

Maintenant nous pouvons générer notre projet avec la ligne de commande suivante :

```
scrapy startproject jumia
```

"jumia" est le nom du projet, mais on pourrait l'appeler comme bon nous semble.

Un Dossier jumia est créé.



scrapy.py

La racine du projet contient le fichier scrapy.cfg qui est un fichier de configuration qui contient des variables telles que le nom du module qui contient les paramètres du projet et d'autres variables de déploiement.

__init__.py

La racine contient un autre dossier datasets qui contient le projet en lui même. Ce dossier qui est un package Python (d'où le **init**.py) contient le package spiders (qui pour l'heure est vide) ainsi que les modules : **items**, **middlewares**, **pipelines** et **settings**.

Le Fichier middlewares.py

Le module **middlewares** contient les middlewares du projet. Les middlewares pour les spiders et pour le downloader sont créés par défaut. Mais il est possible d'en créer un nouveau.

Le Fichier items.py

Le module **items** définit les modèles des données que doivent respecter les items scrapés. Un item est un objet Python style **“clé-valeur”** qui représente un échantillon élémentaire du jeu de données. Si on considère notre jeu de données comme étant au format **CSV**, un item serait une ligne de ce **CSV**.

Le Fichier pipelines.py

Le module **pipelines** contient les pipelines pour chaque item (modèle de données) défini dans le module items. Ces pipelines permettent de faire des traitements sur l’ensemble des items scrapés. Cela est pratique pour faire du nettoyage de données.

Le Fichier settings.py

Le fichier **settings.py** contient des variables qui sont utilisées par l’engine et le reste du projet.

Création D’Un Item

On commence par créer le modèle de données qu’on veut. On fait cela en créant une classe qui hérite de **“scrapy.Item”** et précisant les 3 champs qu’on souhaite avoir.

```
import scrapy

class ArticleItem(scrapy.Item):
    designation = scrapy.Field()
    image = scrapy.Field()
    prix = scrapy.Field()
```

Création d’un Spider

Le point d’entrée du projet est le dossier **spider**. Nous allons créer un nouveau fichier dans le dossier spiders que l’on va appeler **“article.py“**. Évidemment, vous êtes libre de l’appeler comme vous voulez.

Dans ce fichier, on va créer le spider à proprement dit qui n’est rien d’autre qu’une classe héritant de la classe **Spider** de **scrapy**.

```
    from scrapy import Request, Spider
    from ..items import ArticleItem
    class SpiderArticle(Spider):
        name = "article"
        url = "https://www.jumia.com.tn/"

        def start_requests(self):
            yield Request(url=self.url, callback=self.parse_films)

        def parse_films(self, response):
```

```
            listArticle = response.css("article.pr")
            for article in listArticle:
                designation =
article.css("div.name::text").extract_first()
                image = article.css("img.img").attrib('data-src')
                prix = article.css("div.prc::text").extract_first()

                item = ArticleItem()

                item['designation'] = title
                item['image'] = image
                item['prix'] = prix

                yield item
```

Lancement Du Projet

Pour lancer notre spider et avoir les données scrapées dans un fichier **CSV**, on fait la commande suivante :

```
scrapy crawl article -o article.csv
```

Conclusion

Scrapy est un **framework** Python qui facilite énormément les tâches de **web scraping**. Dans cet article, nous avons vu le principe global de fonctionnement de Scrapy. Puis nous avons vu un exemple de web crawling avec le framework qui nous a permis de créer un jeu de données. Le but de cet article était de faire une introduction au framework Scrapy. Pour une version plus complète du projet de scraping du site **jumia**, cliquez ici.

Scrapy est un outil robuste et assez facile à prendre en main. Si vous faites souvent de l'extraction de données sur le web, cet outil peut sûrement vous simplifier la vie.

Ressources

1. https://github.com/camara94/crawlers
2. https://ledatascientist.com/introduction-au-web-scraping-avec-python/
3. https://docs.scrapy.org/en/latest/topics/selectors.html
4. https://twistedmatrix.com/trac/
5. https://www.datacamp.com/community/tutorials/making-web-crawlers-scrapy-python
6. prjumia.com.tn