

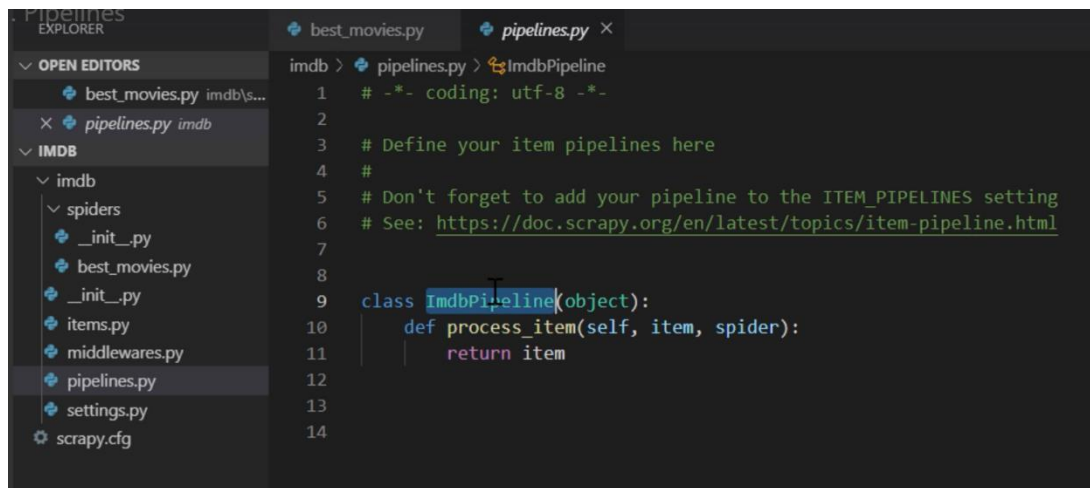
# Working with Pipelines

## Intro

Bu kısımda daha önce yaptığımız projelerden birini alıp, extract edilen dataset'i bir database'e export etmek istiyoruz. Bu amaçla pipelines.py dosyasının içerisini dolduracağız.

Daha önce yapılan IMDB örneği kullanıldığını varsayalım, projenin pipelines.py dosyasına baktığımızda aşağıdaki gibi olduğunu görürüz.

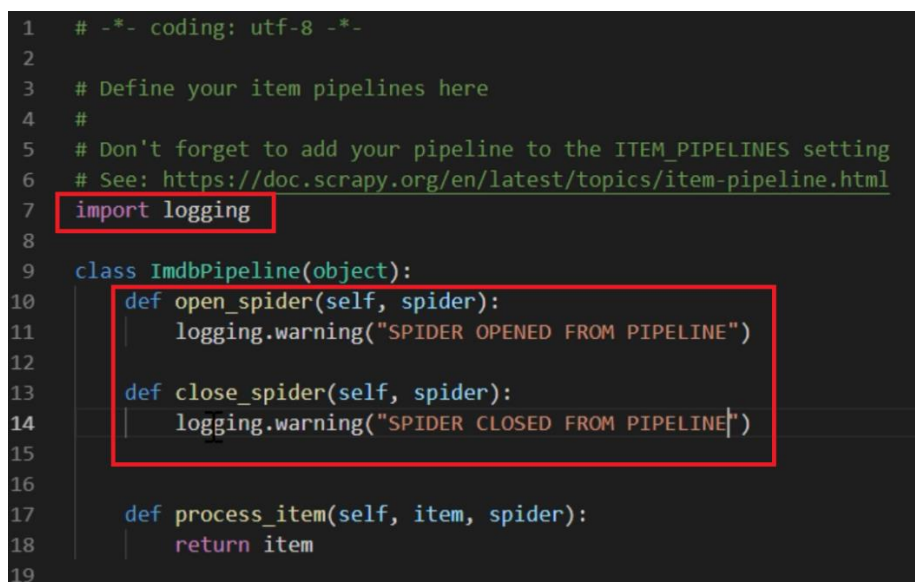
- Burada bir built-in pipeline var ve sadece process\_item isminde tek bir methodu var.
- Bunun da tek yaptığı scrape edilen item'ı return etmek.



```
1  # -*- coding: utf-8 -*-
2
3  # Define your item pipelines here
4  #
5  # Don't forget to add your pipeline to the ITEM_PIPELINES setting
6  # See: https://doc.scrapy.org/en/latest/topics/item-pipeline.html
7
8
9  class ImdbPipeline(object):
10     def process_item(self, item, spider):
11         return item
12
13
14
```

Bu methodun dışında spider'ların kullandığı 2 methoddan daha bahsedilebilir:

**open\_spider()** and **close\_spider()**



```
1  # -*- coding: utf-8 -*-
2
3  # Define your item pipelines here
4  #
5  # Don't forget to add your pipeline to the ITEM_PIPELINES setting
6  # See: https://doc.scrapy.org/en/latest/topics/item-pipeline.html
7  import logging
8
9  class ImdbPipeline(object):
10     def open_spider(self, spider):
11         logging.warning("SPIDER OPENED FROM PIPELINE")
12
13     def close_spider(self, spider):
14         logging.warning("SPIDER CLOSED FROM PIPELINE")
15
16
17     def process_item(self, item, spider):
18         return item
19
```

## Pipeline Methods

**Open\_spider methodu** spider'ımız execution'a başladığı anda bir kez çalışır. Buna karşın **close\_spider methodu** ise spider'ımız execution'ı sonlandırdığı anda bir kez çalışır.

Yani yukarıdaki örnekte, spider'ımız açıldığı ve kapandığı anlarda ekrana console'a warning olarak ilgili yazıyı çıktılardık.

Bunun dışında **process\_item methodu** ise spider'ın yield ettiği her eleman için çağırılacak, bu eleman da item olaca

## Activate this pipeline from the settings.py file

Settings.py dosyasına baktığımızda aşağıdaki gibi commentte bir ITEM\_PIPELINES dictionary'si göreceğiz:

```
62 # scrapy.extensions.telnet.TelnetConsole : None,
63 #}
64
65 # Configure item pipelines
66 # See https://doc.scrapy.org/en/latest/topics/item-pipeline.html
67 # ITEM_PIPELINES = {
68 #     'imdb.pipelines.ImdbPipeline': 300,
69 # }
70
71 # Enable and configure the AutoThrottle extension (disabled by default)
72 # See https://doc.scrapy.org/en/latest/topics/autothrottle.html
73 #AUTOTHROTTLER_ENABLED = True
```

Bu dictionary içerisinde execute etmek istediğim tüm pipeline'ları path'i ile belirtmem gerek.

Örneğin yukarıda bizim yazdığımız pipeline imdb proje klasörünün içindeki pipelines dosyasına ve oradanda ImdbPipeline class'ına ulaşabilirim.

```
# Configure item pipelines
# See https://doc.scrapy.org/en/latest/topics/item-pipeline.html
ITEM_PIPELINES = {
    'imdb.pipelines.ImdbPipeline': 300,
    'imdb.pipelines.FilterDuplicate': 100,
}
```

Örneğin yukarıda iki tane pipeline kullanılıyor, birisi duplicate items'ı filter ediyor buna 100 verdiğimiz için priority bunda, daha sonra ImdbPipeline devreye giriyor.

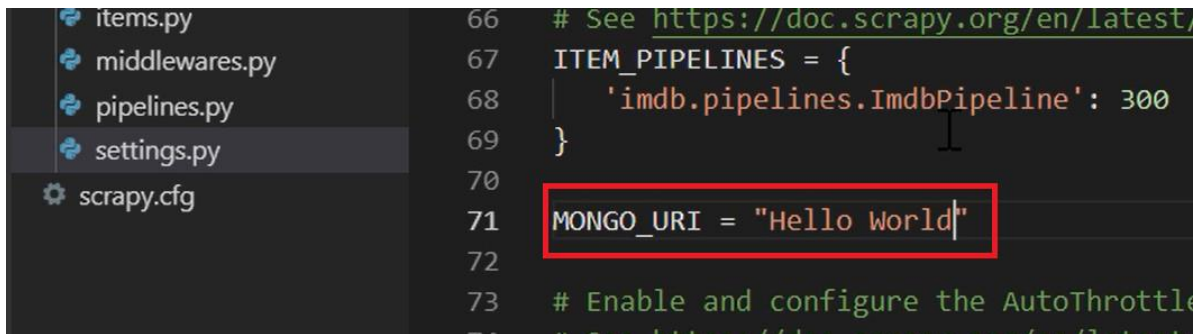
Yani datayı database'e export etmeden önce duplicate items filter ediliyor. Tüm duplicated items filter edildikten sonra, herşeyi database'de store edebilirim.

### Another Pipeline Method

Open spider, close spider ve process item methodlarının yanında from\_crawler() methodundan da bahsedelim, bu bir class methodudur. Argüman olarak cls ve crawler alır.

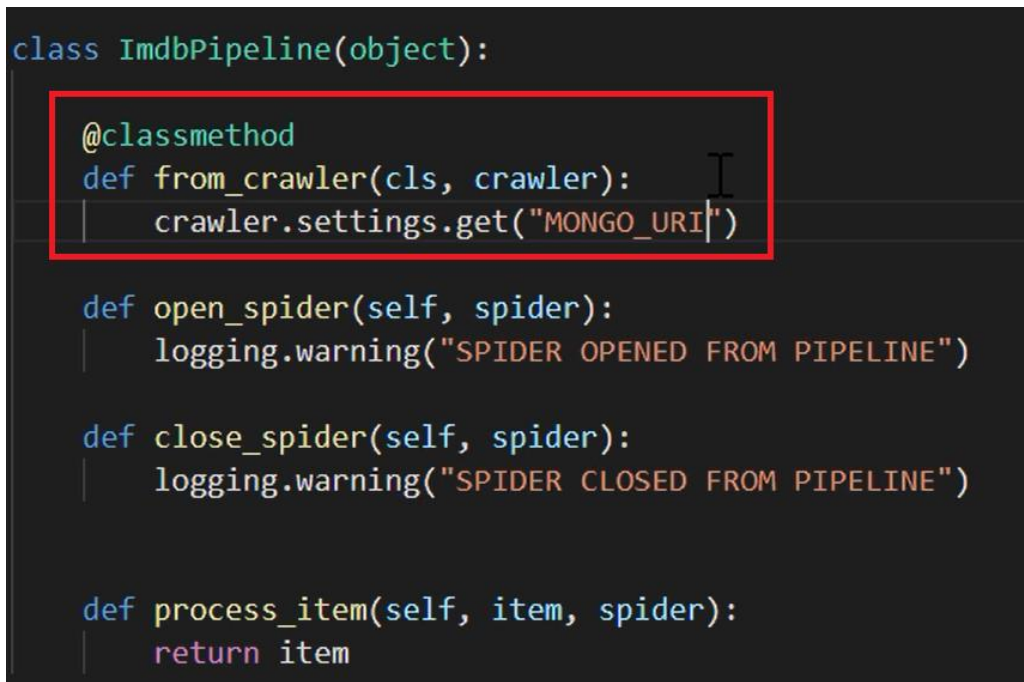
Bu methodun amacı settings.py içerisinde tanımlanan variable'lara pipeline içerisinde ulaşmaktır.

Örneğin settings.py içerisinde aşağıdaki gibi bir MONGO\_URI değişkeni tanımlanmış olsun:



```
66 # See https://doc.scrapy.org/en/latest/
67 ITEM_PIPELINES = {
68     'imdb.pipelines.ImdbPipeline': 300
69 }
70
71 MONGO_URI = "Hello World"
72
73 # Enable and configure the AutoThrottler
74 # See https://doc.scrapy.org/en/latest/
```

Pipeline içerisinde bu değişkene ulaşmak için from\_crawler methodu aşağıdaki gibi kullanılabilir.



```
class ImdbPipeline(object):

    @classmethod
    def from_crawler(cls, crawler):
        crawler.settings.get("MONGO_URI")

    def open_spider(self, spider):
        logging.warning("SPIDER OPENED FROM PIPELINE")

    def close_spider(self, spider):
        logging.warning("SPIDER CLOSED FROM PIPELINE")

    def process_item(self, item, spider):
        return item
```

