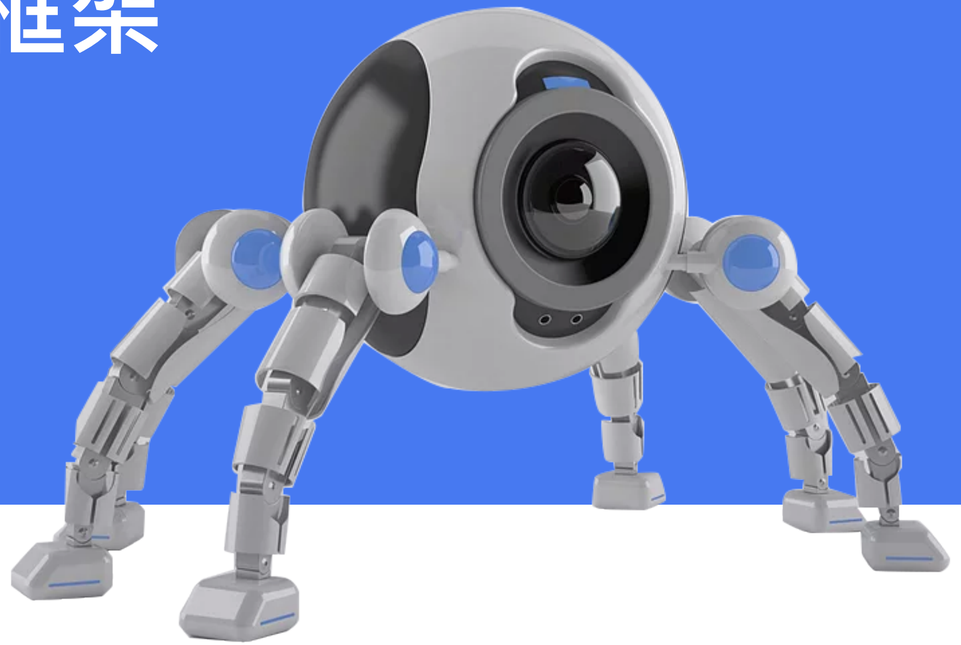
Day 26 SCRAPY 網頁爬蟲框架

Scrapy 爬蟲框架介紹



出題教練：楊鎮銘

python

本日知識點目標

- 了解之前的爬蟲流程與 Scrapy 的差異
- 了解過渡到 Scrapy 的流程

建立 Scrapy 專案

CUPOY

Scrapy 並不僅僅是一個 Python 套件，而是一個**框架**
因此在建立 Scrapy 專案時還要遵循他的專案結構

除了一般 import scrpay 的用法以外，也提供了命令列的功能
我們可以透過 scrapy startproject [專案名稱]
自動產生一個符合框架的檔案結構

Scrapy 框架與套件的差異

CUPOY

前面我們大多使用 requests + BeautifulSoup 套件的組合來實作
但是你必須要自己額外考慮很多爬蟲的細節（甚至你沒想過）

- 先對誰送請求 (Scheduling)
- 資料儲存的 pipeline (ETL)
- etc …

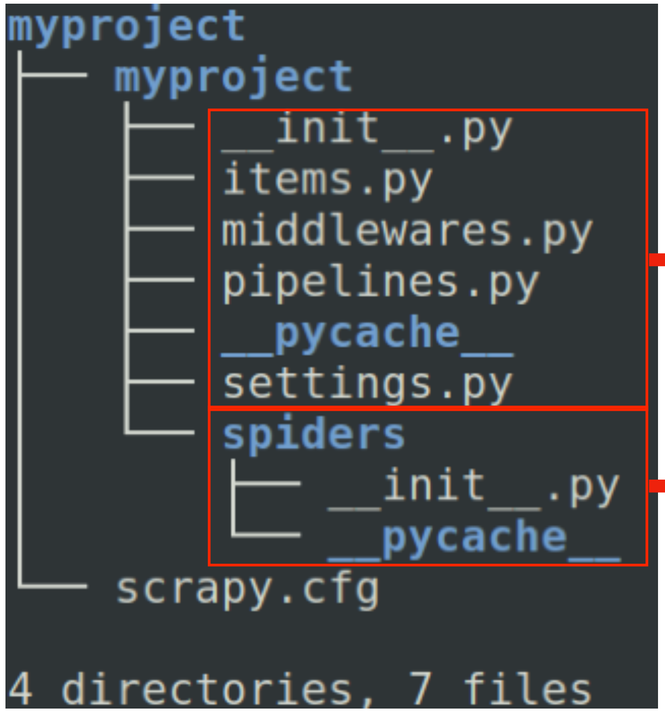
框架基於這些細節都有對應的程式碼，我們可以根據他們定義好的規則去設計撰寫，細節則可以全部交給框架實作

建立 Scrapy 專案

CUPOY

舉例來說，當我們執行 scrapy startproject myproject

會看到 Scrapy 自動建立符合框架的檔案結構



框架重要元件

後面會提到，這邊可以先跳過

主要爬蟲文件的位置

Scrapy 的第一隻爬蟲

同樣地，我們也可以透過命令列來產生出一個符合框架的爬蟲類別

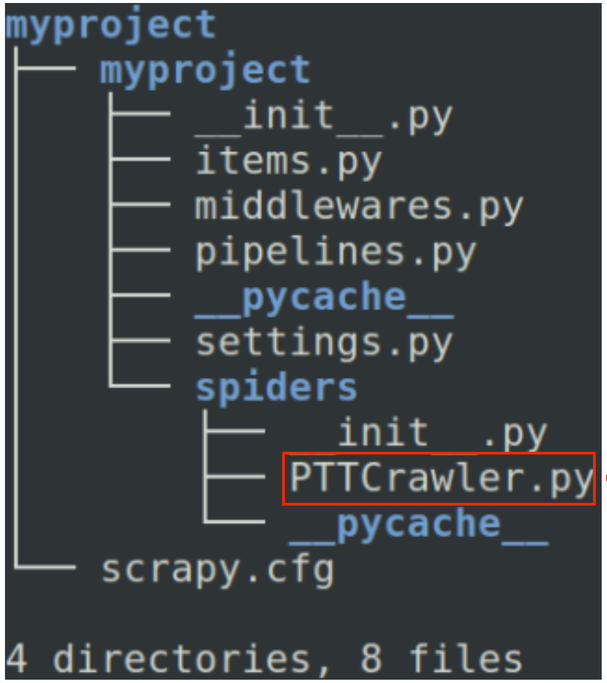
scrapy genspider [爬蟲名稱] [爬蟲目標網址]

Scrapy 的第一隻爬蟲

CUPOY

舉例來說，當我們執行

scrapy genspider PTTCrawler www.ptt.cc




```python
# -*- coding: utf-8 -*-
import scrapy


class PttcrawlerSpider(scrapy.Spider):
    name = 'PTTCrawler'
    allowed_domains = ['www.ptt.cc']
    start_urls = ['http://www.ptt.cc/']

    def parse(self, response):
        pass
```

爬蟲 ID，在整個 project 中必須要是唯一值

Scrapy 請求過程

CUPOY

此時可透過指令開始爬蟲 scrapy crawl PTTCrawler

參考前一頁的程式，這邊其實是因為 scrapy 在背後會呼叫 start_requests 進行爬蟲，為了方便理解我們將其內容寫下來

```
1  # -*- coding: utf-8 -*-¬
2  import scrapy¬
3  ¬
4  ¬
5  class PttcrawlerSpider(scrapy.Spider):¬
6      name = 'PTTCrawler'¬
7      allowed_domains = ['www.ptt.cc']¬
8      start_urls = ['http://www.ptt.cc/']¬
9  ¬
10     def start_requests(self):¬
11         for url in self.start_urls:¬
12             yield scrapy.Request(url=url, callback=self.parse)¬
13 ¬
14     def parse(self, response):¬
15         pass¬
```

這邊會將網址包裝成 scrapy.Request 而不是 requests

這邊先理解他也是在送請求即可

Scrapy 請求過程

```python
# -*- coding: utf-8 -*-
import scrapy


class PttcrawlerSpider(scrapy.Spider):
    name = 'PTTCrawler'
    allowed_domains = ['www.ptt.cc']
    start_urls = ['http://www.ptt.cc/']

    def start_requests(self):
        for url in self.start_urls:
            yield scrapy.Request(url=url, callback=self.parse)

    def parse(self, response):
        pass
```

請求經過包裝後被 scrapy 自動管理 (scrapy.Request)

BeautifulSoup 解析過程在 parse 中執行

5. 解析回應的邏輯

4. 傳送回應

Scary Engie

3. 實際收到回應

2. 實際傳送請求

目標網址

1.準備傳送請求

Scrapy 解析網頁邏輯

```python
# -*- coding: utf-8 -*-
import scrapy


class PttcrawlerSpider(scrapy.Spider):
    name = 'PTTCrawler'
    allowed_domains = ['www.ptt.cc']
    start_urls = ['http://www.ptt.cc/']

    def start_requests(self):
        for url in self.start_urls:
            yield scrapy.Request(url=url, callback=self.parse)

    def parse(self, response):
        pass
```

5. 解析回應的邏輯

（pass 只是先省略實作邏輯）

回傳的 response 物件跟之前 requests 套件的 response 也不一樣了

但是所需功能的使用方式都差不多

- response.url 可以取得我們送出請求的網址
- response.text 都可以取得網頁內容

重要知識點複習

CUPOY

- 了解 Scrapy 在爬蟲專案中變得更加結構化
- 了解如何建構 Scrapy 的專案與爬蟲，並且執行

解題時間
LET'S CRACK IT

請跳出 PDF 至官網 Sample Code & 作業

開始解題

START