Scrapy at a glance

Scrapy (/ËˆskreÉªpaÉª/) is an application framework for crawling web sites and extracting structured data which can be used for a wide range of useful applications, like data mining, information processing or historical archival.

Even though Scrapy was originally designed for web scraping, it can also be used to extract data using APIs (such as Amazon Associates Web Services) or as a general purpose web crawler.

Walk-through of an example spider

In order to show you what Scrapy brings to the table, we'll walk you through an example of a Scrapy Spider using the simplest way to run a spider.

Here's the code for a spider that scrapes famous quotes from website https://quotes.toscrape.com, following the pagination:

```
import scrapy


class QuotesSpider(scrapy.Spider):
    name = 'quotes'
    start_urls = [
        'https://quotes.toscrape.com/tag/humor/',
    ]

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'author': quote.xpath('span/small/text()').get(),
                'text': quote.css('span.text::text').get(),
            }

        next_page = response.css('li.next a::attr("href")').get()
        if next_page is not None:
            yield response.follow(next_page, self.parse)
```

Put this in a text file, name it to something like `quotes_spider.py` and run the spider using the :command:`runspider` command:

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, line 45); *backlink***
>
> Unknown interpreted text role "command".

```
scrapy runspider quotes_spider.py -o quotes.jl
```

When this finishes you will have in the `quotes.jl` file a list of the quotes in JSON Lines format, containing text and author, looking like this:

```
{"author": "Jane Austen", "text": "\u201cThe person, be it gentleman or lady, who has not pleasure in a good novel, must be intolerably
{"author": "Steve Martin", "text": "\u201cA day without sunshine is like, you know, night.\u201d"}
{"author": "Garrison Keillor", "text": "\u201cAnyone who thinks sitting in church can make you a Christian must also think that sitting
...
```

What just happened?

When you ran the command `scrapy runspider quotes_spider.py`, Scrapy looked for a Spider definition inside it and ran it through its crawler engine.

The crawl started by making requests to the URLs defined in the `start_urls` attribute (in this case, only the URL for quotes in *humor* category) and called the default callback method `parse`, passing the response object as an argument. In the `parse` callback, we loop through the quote elements using a CSS Selector, yield a Python dict with the extracted quote text and author, look for a link to the next page and schedule another request using the same `parse` method as callback.

Here you notice one of the main advantages about Scrapy: requests are :ref:`scheduled and processed asynchronously <topics-architecture>`. This means that Scrapy doesn't need to wait for a request to be finished and processed, it can send another request or do other things in the meantime. This also means that other requests can keep going even if some request fails or an error happens while handling it.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, line 73); *backlink***
>
> Unknown interpreted text role "ref".

While this enables you to do very fast crawls (sending multiple concurrent requests at the same time, in a fault-tolerant way) Scrapy also gives you control over the politeness of the crawl through :ref:`a few settings <topics-settings-ref>`. You can do things like setting a download delay between each request, limiting amount of concurrent requests per domain or per IP, and even :ref:`using an auto-throttling extension <topics-autothrottle>` that tries to figure out these automatically.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, line 80); *backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, line 80); *backlink***
>
> Unknown interpreted text role "ref".

> **Note**
>
> This is using :ref:`feed exports <topics-feed-exports>` to generate the JSON file, you can easily change the export format (XML or CSV, for example) or the storage backend (FTP or Amazon S3, for example). You can also write an :ref:`item pipeline <topics-item-pipeline>` to store the items in a database.
>
> > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, line 90); *backlink***
> >
> > Unknown interpreted text role "ref".
>
> > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, line 90); *backlink***
> >
> > Unknown interpreted text role "ref".

What else?

You've seen how to extract and store items from a website using Scrapy, but this is just the surface. Scrapy provides a lot of powerful features for making scraping easy and efficient, such as:

- Built-in support for :ref:`selecting and extracting <topics-selectors>` data from HTML/XML sources using extended CSS selectors and XPath expressions, with helper methods to extract using regular expressions.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 105**); ***backlink***
>
> Unknown interpreted text role "ref".

- An :ref:`interactive shell console <topics-shell>` (IPython aware) for trying out the CSS and XPath expressions to scrape data, very useful when writing or debugging your spiders.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 109**); ***backlink***
>
> Unknown interpreted text role "ref".

- Built-in support for :ref:`generating feed exports <topics-feed-exports>` in multiple formats (JSON, CSV, XML) and storing them in multiple backends (FTP, S3, local filesystem)

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 113**); ***backlink***
>
> Unknown interpreted text role "ref".

- Robust encoding support and auto-detection, for dealing with foreign, non-standard and broken encoding declarations.
- :ref:`Strong extensibility support <extending-scrapy>`, allowing you to plug in your own functionality using :ref:`signals <topics-signals>` and a well-defined API (middlewares, :ref:`extensions <topics-extensions>`, and :ref:`pipelines <topics-item-pipeline>`).

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 120**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 120**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 120**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 120**); ***backlink***
>
> Unknown interpreted text role "ref".

- Wide range of built-in extensions and middlewares for handling:
  - cookies and session handling
  - HTTP features like compression, authentication, caching
  - user-agent spoofing
  - robots.txt
  - crawl depth restriction
  - and more
- A :ref:`Telnet console <topics-telnetconsole>` for hooking into a Python console running inside your Scrapy process, to introspect and debug your crawler

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 134**); ***backlink***
>
> Unknown interpreted text role "ref".

- Plus other goodies like reusable spiders to crawl sites from Sitemaps and XML/CSV feeds, a media pipeline for :ref:`automatically downloading images <topics-media-pipeline>` (or any other media) associated with the scraped items, a caching DNS resolver, and much more!

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 138**); ***backlink***
>
> Unknown interpreted text role "ref".

What's next?

The next steps for you are to :ref:`install Scrapy <intro-install>`, :ref:`follow through the tutorial <intro-tutorial>` to learn how to create a full-blown Scrapy project and join the community. Thanks for your interest!

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 146**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\intro\(scrapy-master) (docs) (intro)overview.rst`, **line 146**); ***backlink***
>
> Unknown interpreted text role "ref".