prog_datasci_5_api_tutorial_scrapy

April 21, 2020

1 Fundamentos de programación

1.1 Tutorial de creación de expresiones *xpath*

---

2 Páginas web y HTML

En este tutorial veremos los conceptos básicos de páginas web que nos ayudaran a desarrollar *crawlers* con scrapy, en especial, a desarrollar el *parser*, que es la parte del *crawler* que selecciona, de cada página, qué datos nos interesa guardar.

El esqueleto de una página web está escrito en HTML (*Hypertext Markup Language*), un lenguaje de marcas que permite construir documentos estructurados mediante la definición de diferentes elementos. Los elementos están delimitados por *tags* (`<tag_name>`) y cada *tag* tiene una funcionalidad o propósito asociado. Por ejemplo, `<header>` se utiliza para definir el encabezado de la página, mientas que `<body>` se utiliza para definir el cuerpo. Existen diferentes *tags* con funcionalidad diversa: `<a>`, `<img>`, `<p>`, ... Por otro lado, los *tags* pueden tener propiedades asociadas, como su identificador (`id`) o su clase (`class`).

En primer lugar, empezaremos viendo cuál es la correspondencia entre los elementos que podemos ver en una página y su código HTML. Para ello, utilizaremos las herramientas de desarrollador del navegador. En segundo lugar, una vez identificado el código HTML que nos interesa, veremos cómo podemos derivar las expresiones **xpath** para seleccionarlo, además de cómo podemos refinar estas expresiones y probarlas.

3 Herramientas de desarrollador del navegador

La mayoría de navegadores modernos, sino todos, viene con herramientas de desarrollador que nos permiten, entre otras cosas, inspeccionar el código HTML de las páginas que visitamos.

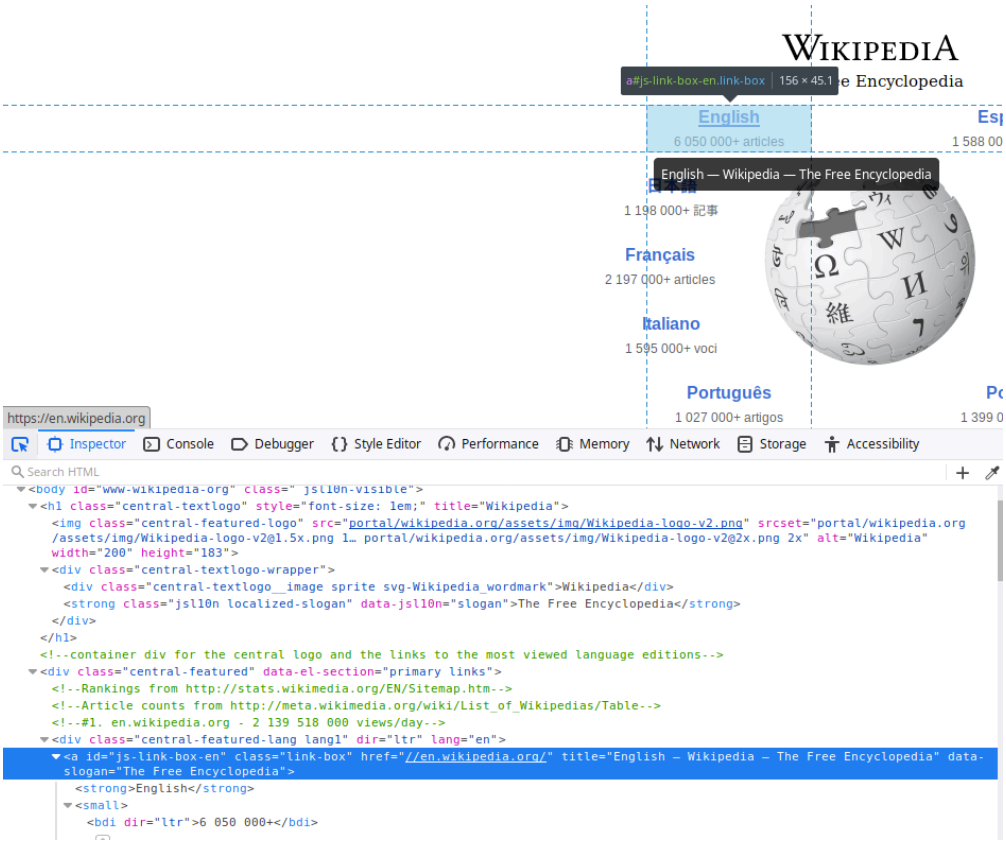
Nosotros utilizaremos Mozilla Firefox, ya que lo tenemos disponible en la máquina virtual, pero podríamos conseguir resultados similares con otros navegadores.

3.1 Inspeccionar código

Para inspeccionar código haremos clic derecho en la página que queremos inspeccionar y seleccionaremos **Inspect Element** (o pulsaremos Q).

Como resultado, se desplegará una nueva sección con diferentes herramientas. La que nos interesa es la de la pestaña **Inspector** (primera de la izquierda).

Con el inspector abierto, podremos ver que en el recuadro inferior izquierdo ha aparecido el código HTML de la página, y que podemos interactuar con el (movernos por el código, abrir y cerrar *tags*, y hasta modificar partes del código). Además, **podemos seleccionar secciones de la web para ver a qué parte corresponden**. Para hacerlo, haremos clic sobre el selector de elementos (icono del cuadro con una flecha, o presionando CTRL+SHIFT+C) y seleccionaremos el elemento que nos interese:



Veremos que una parte del código se ha seleccionado. Esta es la parte correspondiente a la sección sobre la que hemos hecho clic. Podemos ver, por ejemplo, cuál es el **id** del elemento seleccionado, es decir, el campo que lo identifica dentro de la web. Además, podemos ver dentro de qué *tags* se encuentra el elemento. En nuestro caso:

html/body/div[class="central-featured"]/div[class="central-featured-lang lang1"]/a[id="js-link-box-en"]

Por otro lado, podemos ver que este *tag* es `<a>`, que se utiliza para crear enlaces a otras direcciones web. En nuestro caso, el *link* nos lleva a la página de Wikipedia en inglés: el *link* asociado al *tag* es **//en.wikipedia.org/**.

Finalmente, podemos ver que la clase a la que pertenece este elemento es **link-box** (campo *class*):

4 Scrapy shell

Obtener el *xpath* para unos elementos en concreto, a la primera, puede ser complicado. Para poder probar nuestras expresiones hasta encontrar cuál se adapta mejor a nuestras necesidades podemos utilizar la *scrapy shell*. Para obtener la información de una página web con *scrapy shell* deberemos ejecutar `scrapy shell <URL>`, en nuestro caso:

```
scrapy shell wikipedia.org
```

Al abrir la *shell*, scrappy ha creado automáticamente algunos objetos sobre la página descargada. Entre ellos se encuera el objeto `response`, que contiene el *parsing* de la página, y que utilizaremos para probar las expresiones *xpath*.

A continuación veremos cómo utilizar *scrapy shell* para obtener la misma información que hemos obtenido con las herramientas de desarrollador del navegador.

Empezaremos por un ejercicio más sencillo para acostumbrarnos a la sintaxis:

Imaginemos que queremos obtener el título de la página de Wikipedia, es decir, el nombre que vemos en la pestaña del navegador cuándo abrimos una web.

Esta información se encuentra, siempre, en el *tag title* del encabezado (*tag head*) de la web. Para obtener la información crearemos un *xpath* a partir de la respuesta que nos ha devuelto *scrapy shell* (objeto `response`).

```
In [1]: response.xpath('/html/head/title')
Out[1]: [<Selector xpath='/html/head/title' data='<title>Wikipedia</title>'>]
```

Como podemos ver, los *tags* se especifican como si fueran directorios (separados por `/`). En nuestro caso estábamos buscando `/html/head/title`.

Si quisiéramos conseguir todos los *tags title* dentro de la página, podríamos hacerlo con:

```
In [2]: response.xpath('//title')
Out[2]: [<Selector xpath='//title' data=<title>Wikipedia</title>'>]
```

El resultado sería exactamente el mismo, ya que debería haber un único título por página. Lo mismo se aplicaría a cualquier *tag*, `response.xpath(//<tag_name>)` nos devolvería todos los *tags* de tipos `<tag_name>` que contenga la página.

Obtengamos ahora todos los *links* a apartados de Wikipedia en diferentes idiomas basándonos en lo que hemos visto anteriormente con las herramientas de desarrollador. Podemos empezar por obtener el *tag* al *link* en inglés, de la misma forma que habíamos hecho anteriormente, y continuar desde ahí.

Para obtener el *tag*, podemos basar nuestro *xpath* en el *path* que hemos creado con las herramientas de desarrollador: fijaros que para hacer referencia a una propiedad (*id*, *class*, ...) debemos utilizar @ y poner el valor entre comillas (`@class=myClass`).

```
In [3]: response.xpath('/html/body/div[@class="central-featured"]/div[@class="central-featured-
Out[3]: [<Selector xpath='/html/body/div[@class="central-featured"]/div[@class="central-featur
```

Si quisiéramos obtener el contenido de este *tag* para ver la propiedad *href* (que contiene el *link* que buscábamos) podríamos añadir `@href` al final de nuestro *xpath*:

```
response.xpath('/html/body/div[@class="central-featured"]/div[@class="central-featured-lang lan
```

Finalmente, si quisiéramos obtener los diferentes *links* de Wikipedia mostrados en la página actual, podríamos aprovecharnos del hecho que estos comparten la misma clase, como hemos podido ver anteriormente:

```
In [5]: response.xpath('//a[@class="link-box"]')
Out[5]:
[<Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-en" href="//en.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-es" href="//es.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-ja" href="//ja.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-de" href="//de.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-fr" href="//fr.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-ru" href="//ru.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-it" href="//it.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-zh" href="//zh.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-pt" href="//pt.wikipe'>,
 <Selector xpath='//a[@class="link-box"]' data='<a id="js-link-box-pl" href="//pl.wikipe'>]
```

Si no conociéramos exactamente el contenido de la propiedad que estamos buscando, pero conociéramos una subcadena de esta (como en el caso de `central-featured-lang lang1`), podríamos utilizar el método `contains`:

```
In [6]: response.xpath('//div[contains(@class, "central-featured-lang")]')
Out[6]:
[<Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
 <Selector xpath='//div[contains(@class, "central-featured-lang")]' data='<div class="central-
```