CRAWLEANDO E-COMMERCES COM SCRAPY

BRUNO ANDRADE
DATA ENGINEER @ INFOPRICE

Navegue por
Departamentos


Nossas **Cervejeiras**


Nossas **Geladeiras**


Nossos **Fogões**


Nossas **Lavadoras**


Nossos **Fornos**


Nossas **Coifas**

```python
class ConsulSpider(SitemapSpider):
    name = 'consul'
    sitemap_urls = [
        'http://loja.consul.com.br/sitemap.xml',
    ]
    sitemap_rules = [
        (r'/p$', 'parse_product'),
    ]
    custom_settings = {
        'FEED_FORMAT': 'csv'
    }

    def parse_product(self, response):
        if ('ProductLinkNotFound' in response.request.url) or ('/404?' in response.request.url):
            return None

        data = json.loads(response.xpath('/html/body/script[2]/text()').extract()[0])

        if data['pageCategory'] != 'Product':
            return None

        url = response.request.url

        photo_url = response.xpath('//img[@productindex="0"]/@src').extract()[0]

        yield {
            "nome": data['productName'],
            "preco": data['productPriceTo'],
            "url": url,
            "foto": photo_url
        }
```


csv

WEB SCRAPER
X
WEB CRAWLER

WEB SCRAPING

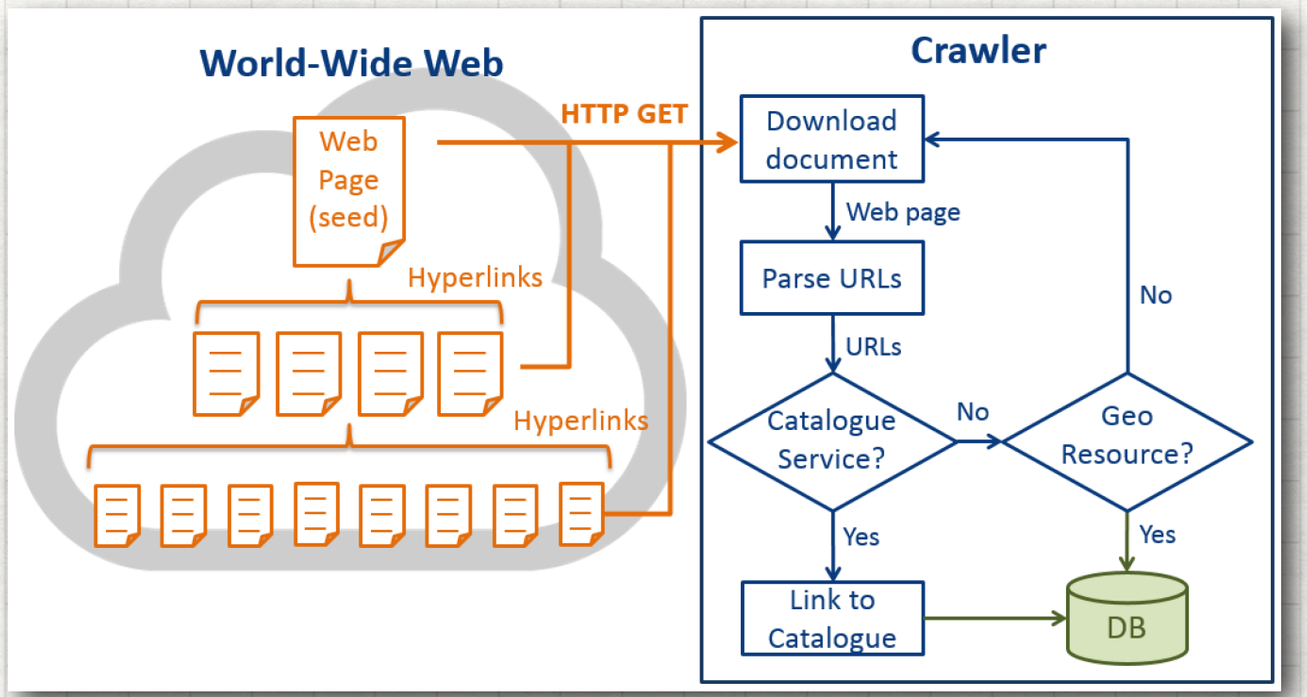
TÉCNICA DE COLETA DE INFORMAÇÕES NÃO ESTRUTURADAS

```
<!DOCTYPE html>
<html>
 <head>
  <title>
   A simple example page
  </title>
 </head>
 <body>
  <p>
   Here is some simple content for this page.
  </p>
 </body>
</html>
```

```
A simple example page
Here is some simple content for this page.
```

WEB CRAWLING

TÉCNICA PARA NAVEGAR E ENCONTRAR AS INFORMAÇÕES A SEREM COLETADAS



POR QUE FAZEMOS ISSO?

WE NEED DATA

OK, E AGORA?

SEU NOVO
MELHOR AMIGO:

SCRAPY!

- CRAWLING

+ SCRAPING

Consul

https://loja.consul.com.br

ROBOTS.TXT

SITEMAP.XML

```
# Disallow all crawlers access to certain pages.
User-agent: *
Disallow: /Account/
Disallow: /Busca/
Disallow: /CallCenter/
Disallow: /CEM/
Disallow: /checkout/
Disallow: /img/
Disallow: /Site/TagDetalhe.aspx
Disallow: /busca
Disallow: /quick-view
Allow: /institucional/
```

https://loja.consul.com.br/robots.txt

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  ▼<sitemap>
    ▼<loc>
       https://loja.consul.com.br/sitemap/sitemap-departments.xml
     </loc>
     <lastmod>2019-07-19</lastmod>
   </sitemap>
  ▼<sitemap>
    ▼<loc>
       https://loja.consul.com.br/sitemap/sitemap-brands.xml
     </loc>
     <lastmod>2019-07-19</lastmod>
   </sitemap>
 ▼<sitemap>
    ▼<loc>
       https://loja.consul.com.br/sitemap/sitemap-products-1.xml
     </loc>
     <lastmod>2019-07-19</lastmod>
   </sitemap>
```


https://loja.consul.com.br/sitemap.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9" xmlns:image="http://www.google.com/sc
  ▼<url>
    ▶<image:image>...</image:image>
    ▼<loc>
       https://loja.consul.com.br/kit-rodape-para-geladeira-w10289734/p
     </loc>
     <lastmod>2019-07-20</lastmod>
     <changefreq>weekly</changefreq>
     <priority>0.4</priority>
   </url>
  ▼<url>
    ▶<image:image>...</image:image>
    ▼<loc>
       https://loja.consul.com.br/kit-correia-sincronizada-consul-w10442114/p
     </loc>
     <lastmod>2019-07-20</lastmod>
     <changefreq>weekly</changefreq>
     <priority>0.4</priority>
   </url>
```


https://loja.consul.com.br/sitemap-products-1.xml

LEI: ROBOTS.TXT

CONVENÇÃO: SITEMAP.XML

```
2    import scrapy
3    import json
4    import re
5    from scrapy.spiders import SitemapSpider
```

```
 7   class ConsulSpider(SitemapSpider):
 8       name = 'consul'
 9       sitemap_urls = [
10           'http://loja.consul.com.br/sitemap.xml',
11       ]
12       sitemap_rules = [
13           (r'/p$', 'parse_product'),
14       ]
15       custom_settings = {
16           'FEED_FORMAT': 'csv'
17       }
18
```

```
18      def parse_product(self, response):
19          if ('ProductLinkNotFound' in response.request.url) or ('/404?' in response.request.url):
20              return None
21
22          data = json.loads(response.xpath('/html/body/script[2]/text()').extract()[0])
23
24          if data['pageCategory'] != 'Product':
25              return None
26
27          url = response.request.url
28
29          photo_url = response.xpath('//img[@productindex="0"]/@src').extract()[0]
30
31          yield {
32              "nome": data['productName'],
33              "preco": data['productPriceTo'],
34              "url": url,
35              "foto": photo_url
36          }
37
```


csv

DÚVIDAS?

MUITO OBRIGADO!

LINKEDIN.COM/IN/BROW1998
BROW1998@GMAIL.COM