Web Crawlers para Gente Grande com Python e Scrapy

Gileno Alves Santa Cruz Filho

Gileno, quem?


Centro
de Informática
U · F · P · E


PyCursos


PUG-PE
Python User Group Pernambuco

O que são Web Crawlers?

Crawler
X
Scraping

Porque criá-los?

Cotação Dolar

melhorcâmbio.com

CRIAR ALERTA

Dólar Hoje

US$ 1,00

R$ 3,92

DÓLAR COMERCIAL

SELECIONE A CIDADE...

DÓLAR TURISMO

Compare as cotações em sua cidade

Código Fonte

```
91
92  <input type="hidden" id="id-moeda" value="8" />
93  <input type="hidden" id="toggle" value="0" />
94  <input type="hidden" value="3,92" id="taxa-comercial">
95  <input type="hidden" id="taxa-turismo" value="0" />
96  <input type="hidden" id="input-alterado" value="" />
97  <input type="hidden" id="alerta" value="0" />
98
99  <input type="hidden" id="ss" value="0" />
100
```

urllib

```
import urllib.request
import re
url = 'https://www.melhorcambio.com/dolar-hoje'
headers = {
        'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/63.0.3239.132 Safari/5’
}
req = urllib.request.Request(url, headers=headers)
with urllib.request.urlopen(r) as response:
        html = response.read().decode(’utf-8’)
preco = re.findall(r'<input type="hidden" value="(.*)" id="taxa-comercial">', html)[0]
print(preco)
```

Usando Requests

```
import requests
import re
url = 'https://www.melhorcambio.com/dolar-hoje'
headers = {
'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/5’
}
req = requests.get(url, headers=headers)
html = req.text
preco = re.findall(r'<input type="hidden" value="(.*)" id="taxa-comercial">', html)[0]
print(preco)
```

Scrapy

Spiders

```html
<table data-apvhdrid="512">
<tbody>
    <tr data-id="1">
        <td data-itemid="001">Beer</td>
        <td data-quantity="10">10 bottles</td>
        <td data-unitcost="11.00">11.00</td>
        <td data-amount="110.00">110.00</td>
    </tr>
    <tr data-id="2">
        <td data-itemid="002">Vodka</td>
        <td data-quantity="20">20 bottles</td>
        <td data-unitcost="100.00">100.00</td>
        <td data-amount="2000.00">2000.00</td>
    </tr>
</tbody>
</table>
```

```
[
    {"apvhdrid":512, "id":1, "itemid":001, "quantity":10, "unitcost":"10.00", "amount":"110.00"},
    {"apvhdrid":512, "id":2, "itemid":001, "quantity":20, "unitcost":"100.00", "amount":"2000.00"}
]
```

Items (html -> dados estruturados)

Usando Scrapy (dolar_hoje.py)

```
import scrapy
import re
class DolarSpider(scrapy.Spider):
    name = 'dolar_hoje'
    start_urls = ['https://www.melhorcambio.com/dolar-hoje']
    custom_settings = {
        'USER_AGENT': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/63.0.3239.132 Safari/5'
    }
    def parse(self, response):
        html = response.text
        preco = re.findall(
            r'<input type="hidden" value="(.*)" id="taxa-comercial">', html
        )[0]
        self.log(preco)
```

scrapy runspider dolar_hoje.py

scrapy.core.engine] INFO: Spider opened

[scrapy.extensions.logstats] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)

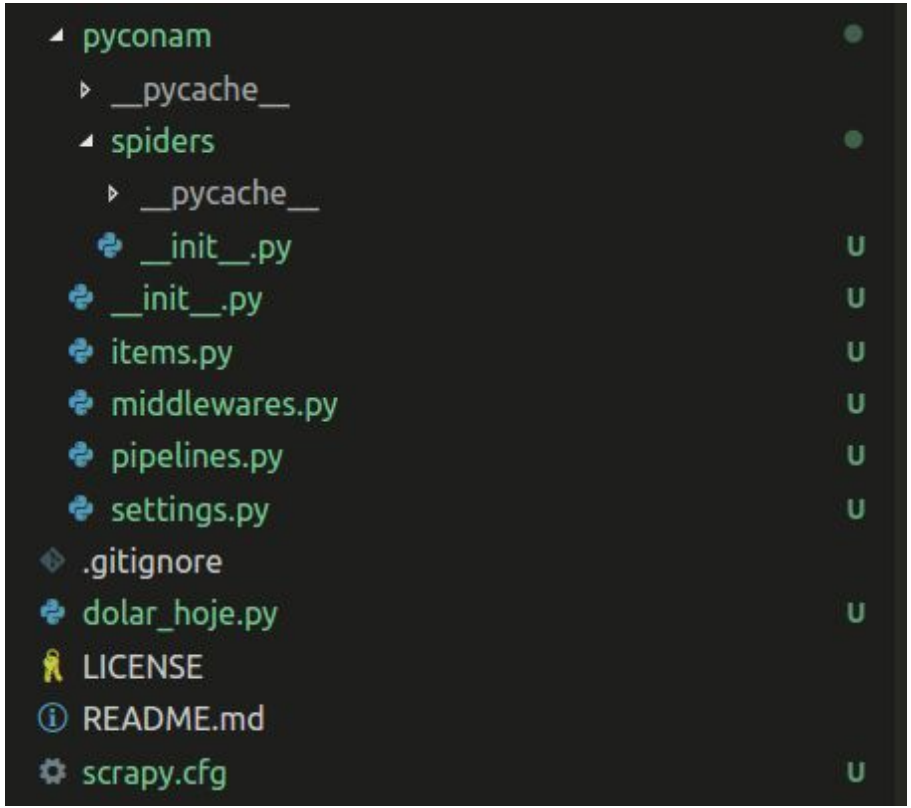
[scrapy.extensions.telnet] DEBUG: Telnet console listening on 127.0.0.1:6023

[scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.melhorcambio.com/robots.txt> (referer: None)

[scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.melhorcambio.com/dolar-hoje> (referer: None)

[dolar_hoje] DEBUG: 3,92

scrapy startproject pynorte



Seletor xpath / css

Usando xpath e css

```
import scrapy
import re


class DolarSpider(scrapy.Spider):

    name = 'dolar_hoje'
    start_urls = ['https://www.melhorcambio.com/dolar-hoje']

    def parse(self, response):
        preco = response.xpath('//input[@id="taxa-comercial"]/@value')
        self.log(preco.extract_first())
        preco = response.css('#taxa-comercial')[0]
        self.log(preco.attrib['value'])
```

scrapy crawl spider

[scrapy.core.engine] INFO: Spider opened

[scrapy.extensions.logstats] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)

[scrapy.extensions.telnet] DEBUG: Telnet console listening on 127.0.0.1:6023

[scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.melhorcambio.com/robots.txt> (referer: None)

[scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.melhorcambio.com/dolar-hoje> (referer: None)

[dolar_hoje] DEBUG: 3,92

[dolar_hoje] DEBUG: 3,92

Items

Retornando Items

```
import scrapy

class VivaRealSpider(scrapy.Spider):

    name = 'vivareal'
    start_urls =
['https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/']

    def parse(self, response):
        for item in response.xpath("//div[contains(@class, 'results-list')]/div"):
            yield {
                'title': item.xpath(".//h2/a/text()").extract_first().strip()
            }
```

scrapy crawl spider

[scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/> (referer: None)

[scrapy.core.scraper] DEBUG: Scraped from <200 https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/>

{'title': 'Apartamento com 3 Quartos à Venda, 82m^2'}

[scrapy.core.scraper] DEBUG: Scraped from <200 https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/>

{'title': 'RIVIERA BOA VIAGEM'}

Item Pipeline

pipelines.py

```
class PyconamPipeline(object):

    def process_item(self, item, spider):
        # Faz alguma limpeza
        # Salva no banco de dados
        return item
```

.... settings.py

```
ITEM_PIPELINES = {
        'pyconam.pipelines.PyconamPipeline': 300,
}
```

yield Request

Criando novas requisições

```
import scrapy
class VivaRealSpider(scrapy.Spider):
    name = 'vivareal'
    start_urls =
['https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/']
    def parse(self, response):
        for item in response.xpath("//div[contains(@class, 'results-list')]/div"):
            href = item.xpath(".//h2/a/@href").extract_first()
            yield scrapy.Request(
                f'https://www.vivareal.com.br{href}', self.parse_detail
            )
    def parse_detail(self, response):
        yield {
            'title': response.xpath("//title/text()").extract_first().strip()
        }
```

scrapy crawl spider

DEBUG: Crawled (200) <GET https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/> (referer: None)

[scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.vivareal.com.br/imoveis-lancamento/maria-olivia-5148/> (referer: https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/)

[scrapy.core.scraper] DEBUG: Scraped from <200 https://www.vivareal.com.br/imoveis-lancamento/maria-olivia-5148/>

{'title': 'Apartamento na Avenida Pedro Paes Mendonça, 200, Boa Viagem em Recife, por R$ 987.000 - Viva Real'}

CrawlSpider

Criando um CrawlSpider

```
import scrapy
from scrapy.spiders import CrawlSpider, Rule
from scrapy.linkextractors import LinkExtractor
class VivarealSpider(CrawlSpider):
    name = 'vivareal_crawl'
    start_urls = ['https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/']
    rules = (
        Rule(
            LinkExtractor(allow='/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/')
        ),
        Rule(
            LinkExtractor(
                allow='/imovel/',
            ), callback='parse_imovel'
        )
    )
    def parse_imovel(self, response):
        yield {
            'title': response.xpath("//title/text()").extract_first()
        }
```

Regras da CrawlSpider

```
rules = (
    Rule(
     LinkExtractor(allow='/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/')
    ),
    Rule(
        LinkExtractor(
            allow='/imovel/',
        ), callback='parse_imovel'
    )
)
```

scrapy crawl spider

DEBUG: Crawled (200) <GET https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/#pagina=> (referer: https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/)

.....

[scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/?pagina=4> (referer: https://www.vivareal.com.br/venda/pernambuco/recife/bairros/boa-viagem/apartamento_residencial/)

Plugins e Settings

settings.py

```
SPIDER_MODULES = ['pyconam.spiders']
NEWSPIDER_MODULE = 'pyconam.spiders'

USER_AGENT = 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/63.0.3239.132 Safari/5'

# Obey robots.txt rules
ROBOTSTXT_OBEY = True

# Configure maximum concurrent requests performed by Scrapy (default: 16)
#CONCURRENT_REQUESTS = 32

DOWNLOAD_DELAY = 0.5
```

plugins

- https://github.com/scrapy-plugins
- Scrapy Splash

Deploy

Algumas opções

- Scrapyd
- ScrapingHub
- SpiderKeeper

Talk is cheap, Show me the code

Obrigado!
Dúvidas?

@gilenofilho
contato@gilenofilho.com.br
https://www.pycursos.com