Web Crawlers para Gente Grande com Python e Scrapy

Gileno Alves Santa Cruz Filho

Gileno, quem?







Oque são Web Crawlers?

Crawler X Scraping

Scraping

- Extrair Dados Estruturados de algo não estruturado
- HTML
- XML
- IMAGENS (OCR)
- PDF

Crawler

- Visitar Páginas Web
- Seguir regras de visitação
- Não necessariamente precisa extrair informação estruturada (fazer scraping)

Porque criá-los?



Informação == dinheiro

Google

E-Commerce

- Qual o valor que meus concorrentes estão cobrando?
- Quais produtos estão sendo vendidos na loja
 X?
- Qual a variação de preço?
- Produto está disponível?

Imóveis

- Qual preço praticado na região X?
- O apartamento X está abaixo do preço praticado no mercado?
- Que imóveis estão disponíveis na cidade Y?

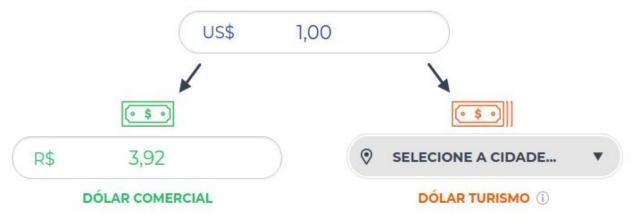
Cotação Dolar







Dólar Hoje



Compare as cotações em sua cidade

Código Fonte

```
'sinput type="hidden" id="id-moeda" value="8" />
'sinput type="hidden" id="toggle" value="0" />
'sinput type="hidden" value="3,92" id="taxa-comercial">
'sinput type="hidden" id="taxa-turismo" value="0" />
'sinput type="hidden" id="input-alterado" value="" />
'sinput type="hidden" id="alerta" value="0" />
'sinput type="hidden" id="alerta" value="0" />
'sinput type="hidden" id="ss" value="o" />
'sinput ty
```

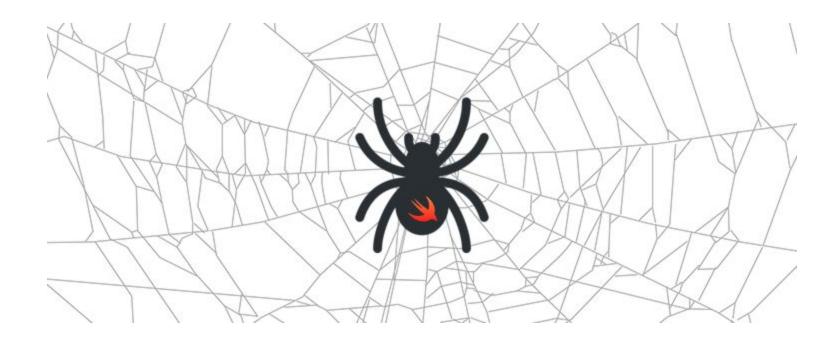
urllib

```
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'
reg = urllib.reguest.Reguest(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'
reg = reguests.get(url, headers=headers)
html = req.text
preco = re.findall(r'<input type="hidden" value="(.*)" id="taxa-comercial">', html)[0]
print(preco)
```

Scrapy



Spiders

```
Beer
            10 bottles
            11.00
            110.00
          Vodka
            20 bottles
            100.00
            2000.00
          {"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)
```

Seletor xpath/css

XPath

- Linguagem de consulta para arquivos XML
- Uma espécie de SQL para arquivos XML (pode ser usado em HTML)

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'])
```

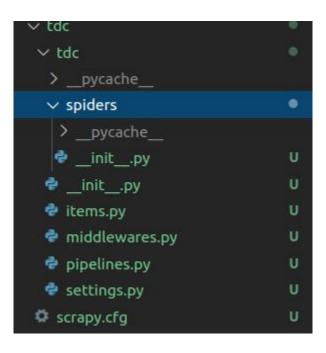
Items

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 Project

scrapy startproject tdc



Item Pipeline

pipelines.py

```
class TdcPipeline(object):
  def process_item(self, item, spider):
    # Faz alguma limpeza
    # Salva no banco de dados
    return item
.... settings.py
ITEM PIPELINES = {
    tdc.pipelines.TdcPipeline: 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()
```

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'
)
```

Plugins e Settings

settings.py

```
SPIDER_MODULES = [tdc.spiders']
NEWSPIDER_MODULE = 'tdc.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 https://github.com/gileno/tdcrecife2019