Tutorial de Scrapy

Laboratório Hacker de Campinas

- **Hackerspace**
- Espaço aberto e comunitário para que entusiastas de tecnologia possam desenvolver seus projetos
- Mantido pelos frequentadores:
  - Mensalidades (direito a chave)
  - Compras (camisetas, bebidas, etc)
  - Doações
    - Dinheiro / Equipamentos / Tempo

Laboratório Hacker de Campinas

- Encontros comunidades (Python, Ruby, Blockchain, ...)
- Oficinas (IoT, FreeCAD, OpenWRT, ...)
- Produção de Cerveja
- Churrascos
- Projetos nas mais diversas áreas (eletrônica, marcenaria, agricultura,...)

Web Scraping

Extrair dados **estruturados** de fontes de dados **não estruturadas** (tipicamente páginas web)

Casos de Uso

1. Pesquisas com dados governamentais
2. Monitorar o que estão falando do meu produto
3. Monitorar os produtos dos concorrentes
4. Ofertas de emprego, imóveis, bens de consumo
5. Análise de redes sociais

Scrapy

https://scrapy.org/

An open source and collaborative framework for extracting the data you need from websites.

In a fast, simple, yet extensible way.

Instalando o Scrapy

```
renne@capivara:code$ python3 -m venv .venv
renne@capivara:code$ source .venv/bin/activate
(.venv) renne@capivara:code$ pip install scrapy

...

(.venv) renne@capivara:code$ scrapy version
Scrapy 1.5.1
```

Hello World! Meu primeiro Spider!

```python
#hackerspaces.py
class HackerspaceListSpider(scrapy.Spider):
  name = 'hackerspace-list'
  start_urls = [
    'https://wiki.hackerspaces.org/'
    'List_of_ALL_Hacker_Spaces',
  ]

  def parse(self, response):
    for row in response.css('table tr'):
      yield {
        'hackerspace': row.css(
          '.Hackerspace *::text').get(),
        'country': row.css(
          '.Country *::text').get(),
        'status': row.css(
          '.Hackerspace-status *::text').get(),
        'url': row.css(
          '.Website a::attr(href)').get()
      }
```

```
$ scrapy runspider hackerspaces.py -o hackerspaces.csv
2018-08-03 16:31:02 [scrapy.utils.log] INFO: Scrapy 1..
2018-08-03 16:31:02 [scrapy.utils.log] INFO: Versions:.
2018-08-03 16:31:02 [scrapy.crawler] INFO: Overridden .
2018-08-03 16:31:02 [scrapy.middleware] INFO: Enabled .
['scrapy.extensions.corestats.CoreStats',
 'scrapy.extensions.telnet.TelnetConsole',
 'scrapy.extensions.memusage.MemoryUsage',
 'scrapy.extensions.feedexport.FeedExporter',
 'scrapy.extensions.logstats.LogStats']
2018-08-03 16:31:02 [scrapy.middleware] INFO: Enabled .
...
 'item_scraped_count': 101,
 'log_count/DEBUG': 103,
 'log_count/INFO': 8,
 'memusage/max': 51961856,
 'memusage/startup': 51961856,
 'response_received_count': 1,
 'scheduler/dequeued': 1,
 'scheduler/dequeued/memory': 1,
 'scheduler/enqueued': 1,
 'scheduler/enqueued/memory': 1,
 'start_time': datetime.datetime(2018, 8, 3, 19, 34, 2.
2018-08-03 16:34:23 [scrapy.core.engine] INFO: Spider .
```

```
$ scrapy runspider hackerspaces.py -o hackerspaces.csv
```

```
$ scrapy runspider hackerspaces.py -o hackerspaces.json
```

```
$ scrapy runspider hackerspaces.py -o hackerspaces.jl
```

```
$ scrapy runspider hackerspaces.py -o hackerspaces.xml
```

Um Spider básico

```
class MySpider(scrapy.Spider):
  name = 'spider_name'

  def start_requests(self):
    yield [
      scrapy.Request(
        'http://example.com',
        callback=self.parse
      )
    ]

  def parse(self, response):
    self.logger.info('Passei por aqui!')
```

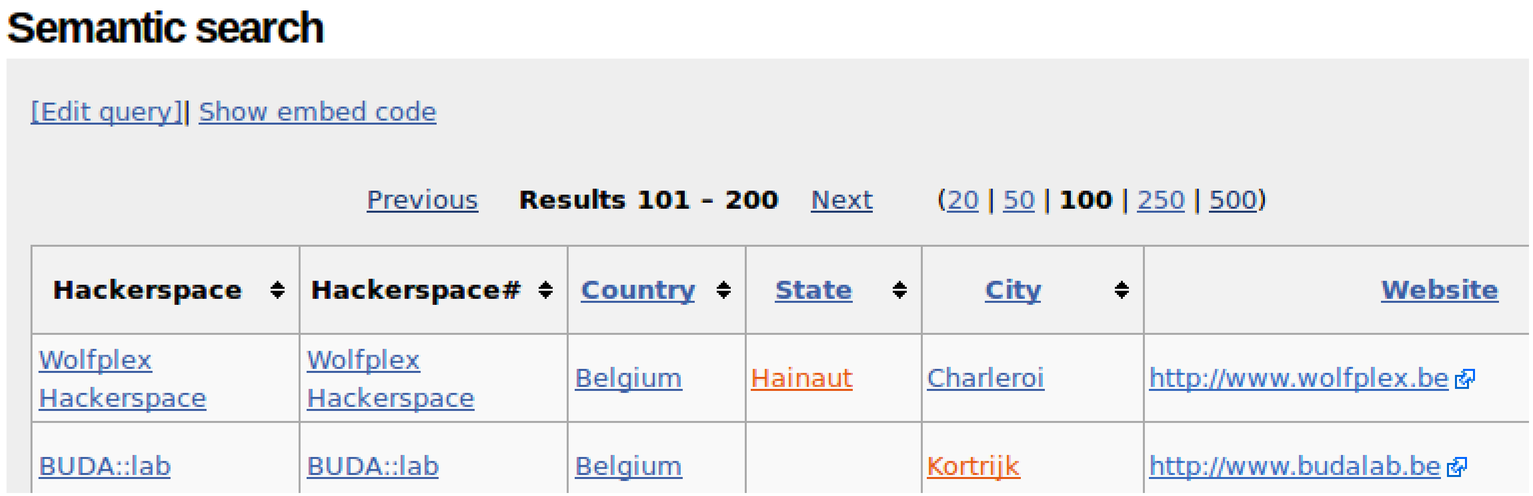
Qual o resultado?

```
def parse(self, response):
  for row in response.css('table tr'):
    yield {
      'hackerspace': row.css(
        '.Hackerspace *::text').get(),
      'country': row.css(
        '.Country *::text').get(),
      'status': row.css(
        '.Hackerspace-status *::text').get(),
      'url': row.css(
        '.Website a::attr(href)').get()
    }
  further_results = response.xpath(
    '//a[contains(text(), "further")]//@href')
  if further_results:
    yield scrapy.Request(
      response.urljoin(
        further_results.get()
      )
    )
```

Ko-Lab	Ko-Lab	K
Shortcut	Shortcut	S
ELECTRONICS::lab	ELECTRONICS::lab	E
G-Hive	G-Hive	G
Hackerspace Brussels	Hackerspace Brussels	H
... further results		

This page was last modified on 26 August 2013, at 14:48. This page has be

```
$ scrapy runspider hackerspaces.py -o hackerspaces.csv
2018-08-03 16:31:02 [scrapy.utils.log] INFO: Scrapy 1..
2018-08-03 16:31:02 [scrapy.utils.log] INFO: Versions:.
2018-08-03 16:31:02 [scrapy.crawler] INFO: Overridden .
2018-08-03 16:31:02 [scrapy.middleware] INFO: Enabled .
['scrapy.extensions.corestats.CoreStats',
 'scrapy.extensions.telnet.TelnetConsole',
...
 'item_scraped_count': 201,
 'log_count/DEBUG': 204,
 'log_count/INFO': 8,
 'memusage/max': 51937280,
 'memusage/startup': 51937280,
 'request_depth_max': 1,
 'response_received_count': 2,
 'scheduler/dequeued': 2,
 'scheduler/dequeued/memory': 2,
 'scheduler/enqueued': 2,
 'scheduler/enqueued/memory': 2,
 'start_time': datetime.datetime(2018, 8, 3, 19, 34, 2.
2018-08-03 16:34:23 [scrapy.core.engine] INFO: Spider .
```



Semantic search

[Edit query]| Show embed code

Previous   **Results 101 - 200**   Next   (20 | 50 | **100** | 250 | 500)

Hackerspace	Hackerspace#	Country	State	City	Website
Wolfplex Hackerspace	Wolfplex Hackerspace	Belgium	Hainaut	Charleroi	http://www.wolfplex.be
BUDA::lab	BUDA::lab	Belgium		Kortrijk	http://www.budalab.be

```
further_results = response.xpath(
  "//a[contains(text(), 'further')]//@href|"
  "//a[contains(text(), 'Next')]//@href")
```

Como extrair dados da página?

- Seletores CSS https://www.w3.org/TR/selectors/
- Seletores XPath https://www.w3.org/TR/xpath/all/

```
(.venv) renne@capivara:code$ pip install ipython
(.venv) renne@capivara:code$ ipython
Python 3.6.4 (default, Mar 26 2018, 15:25:21)
Type 'copyright', 'credits' or 'license' for more infor
IPython 6.5.0 -- An enhanced Interactive Python. Type '

In [1]: from parsel import Selector

In [2]: with open('product_list.html') as code:
   ...:     response = Selector(text=code.read())
   ...:

In [3]:
```

```
response.css('h1')
```

```
response.css('ul#offers')
```

```
response.css('.product')
```

```
response.css('ul#offers .product')
```

```
response.css('ul#offers .product a::attr(href)')
```

```
response.css('ul#offers .product *::text')
```

```
response.css('ul#offers .product p::text')
response.xpath('//h1')
```

```
response.xpath('//h1[2]')
```

```
response.css('//ul[@id="offers"]')
```

```
response.xpath('//li/a/@href')
```

```
response.xpath('//li/text()')
```

```
response.xpath('//li//text()')
```

```
response.xpath('//p/text()')
response.xpath(
  '//ul[@id="offers"]//li[@class="product"]'
)
```

```
response.xpath(
  '//ul[@id="offers"]//li[contains(@class, "product")]'
)
```

```
response.xpath(
  '//li[@class="ad"]/following-sibling::li'
  '[@class="product"]').getall()
```

http://quotes.toscrape.com/

Queremos obter todas as informações disponíveis sobre as citações disponíveis nesta página (citação, nome do autor, URL para detalhes do autor e lista de tags da citação).

Começamos criando um projeto Scrapy:

```
$ scrapy startproject quotes
$ cd quotes
$ scrapy genspider default_quotes quotes.toscrape.com
```

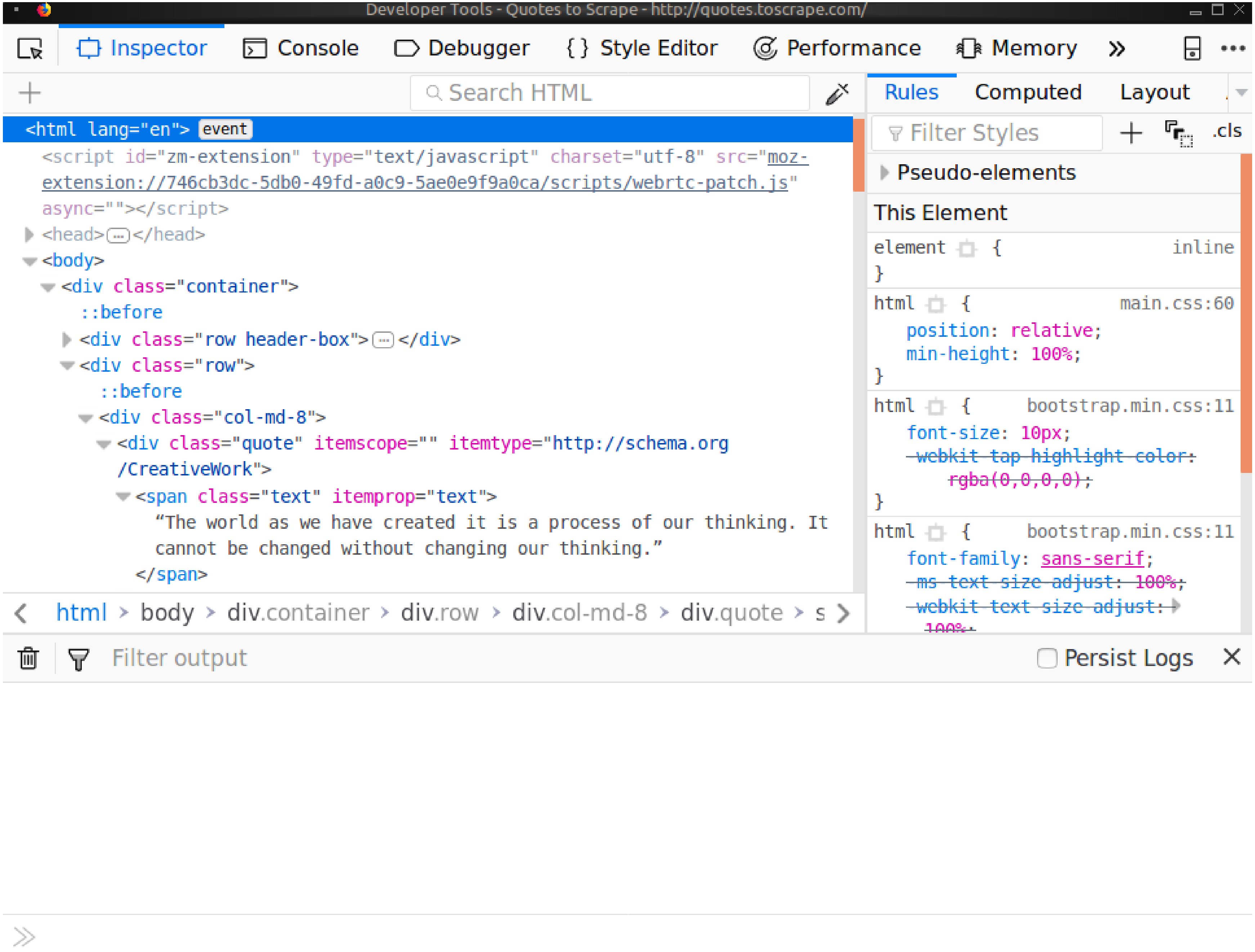
```python
# -*- coding: utf-8 -*-
import scrapy


class DefaultQuotesSpider(scrapy.Spider):
    name = 'default_quotes'
    allowed_domains = ['quotes.toscrape.com']
    start_urls = ['http://quotes.toscrape.com/']

    def parse(self, response):
        pass
```

Ferramentas de uso diário

- Developer Tools
- `scrapy shell <URL>`



```
<html lang="en"> event
  <script id="zm-extension" type="text/javascript" charset="utf-8" src="moz-
  extension://746cb3dc-5db0-49fd-a0c9-5ae0e9f9a0ca/scripts/webrtc-patch.js"
  async=""></script>
  <head>…</head>
  <body>
    <div class="container">
      ::before
      <div class="row header-box">…</div>
      <div class="row">
        ::before
        <div class="col-md-8">
          <div class="quote" itemscope="" itemtype="http://schema.org
          /CreativeWork">
            <span class="text" itemprop="text">
              “The world as we have created it is a process of our thinking. It
              cannot be changed without changing our thinking.”
            </span>
```

```
element {                                   inline
}
html {                                 main.css:60
   position: relative;
   min-height: 100%;
}
html {                        bootstrap.min.css:11
   font-size: 10px;
   -webkit-tap-highlight-color:
       rgba(0,0,0,0);
}
html {                        bootstrap.min.css:11
   font-family: sans-serif;
   -ms-text-size-adjust: 100%;
   -webkit-text-size-adjust:
    100%;
```

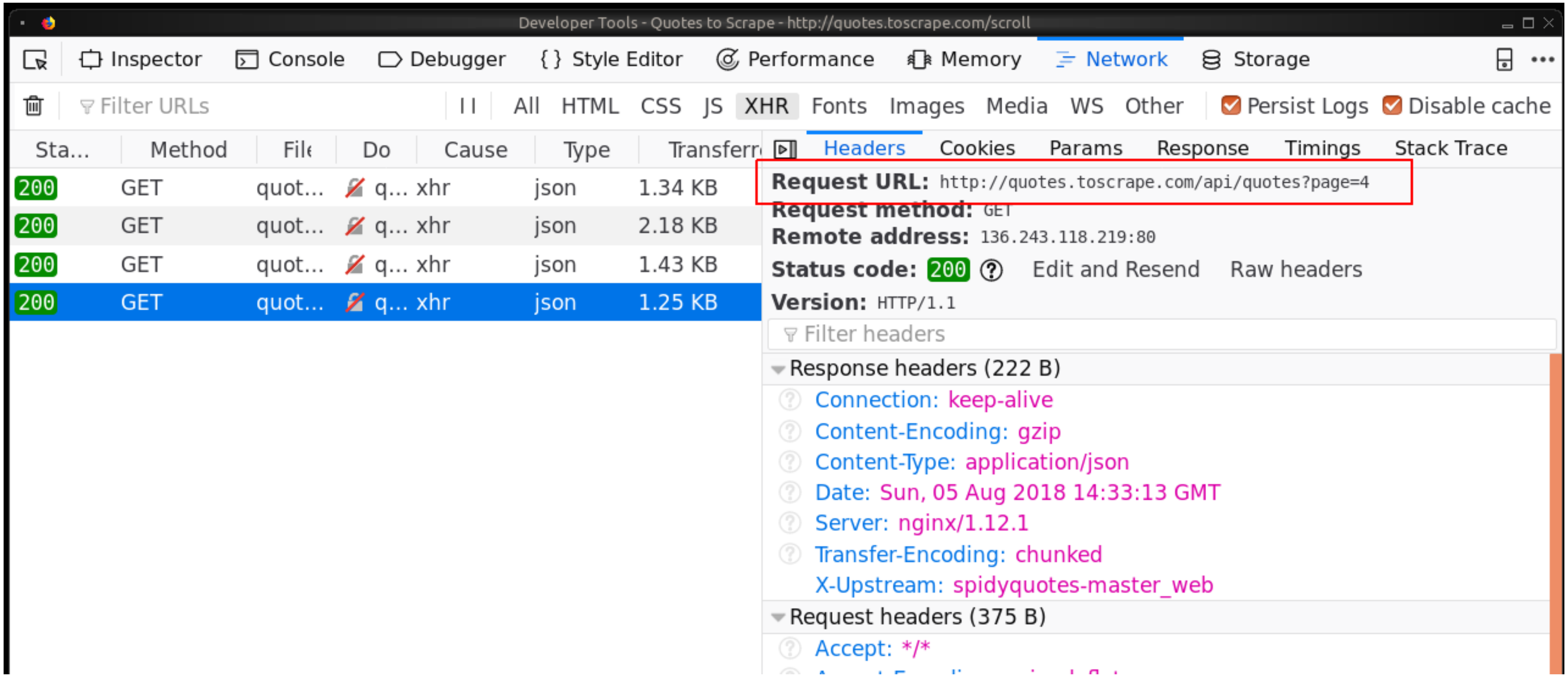
```
$ scrapy shell http://quotes.toscrape.com/
```

10
11
12
9
1
8
2
7
3
4
5
30
60
40
10
30
20

http://quotes.toscrape.com/scroll

Página com scroll infinito.

```
$ scrapy genspider scroll_quotes quotes.toscrape.com
```



Developer Tools - Quotes to Scrape - http://quotes.toscrape.com/scroll

☑ Persist Logs ☑ Disable cache

Sta…	Method	Fil	Do	Cause	Type	Transferr
200	GET	quot…	q…	xhr	json	1.34 KB
200	GET	quot…	q…	xhr	json	2.18 KB
200	GET	quot…	q…	xhr	json	1.43 KB
200	GET	quot…	q…	xhr	json	1.25 KB

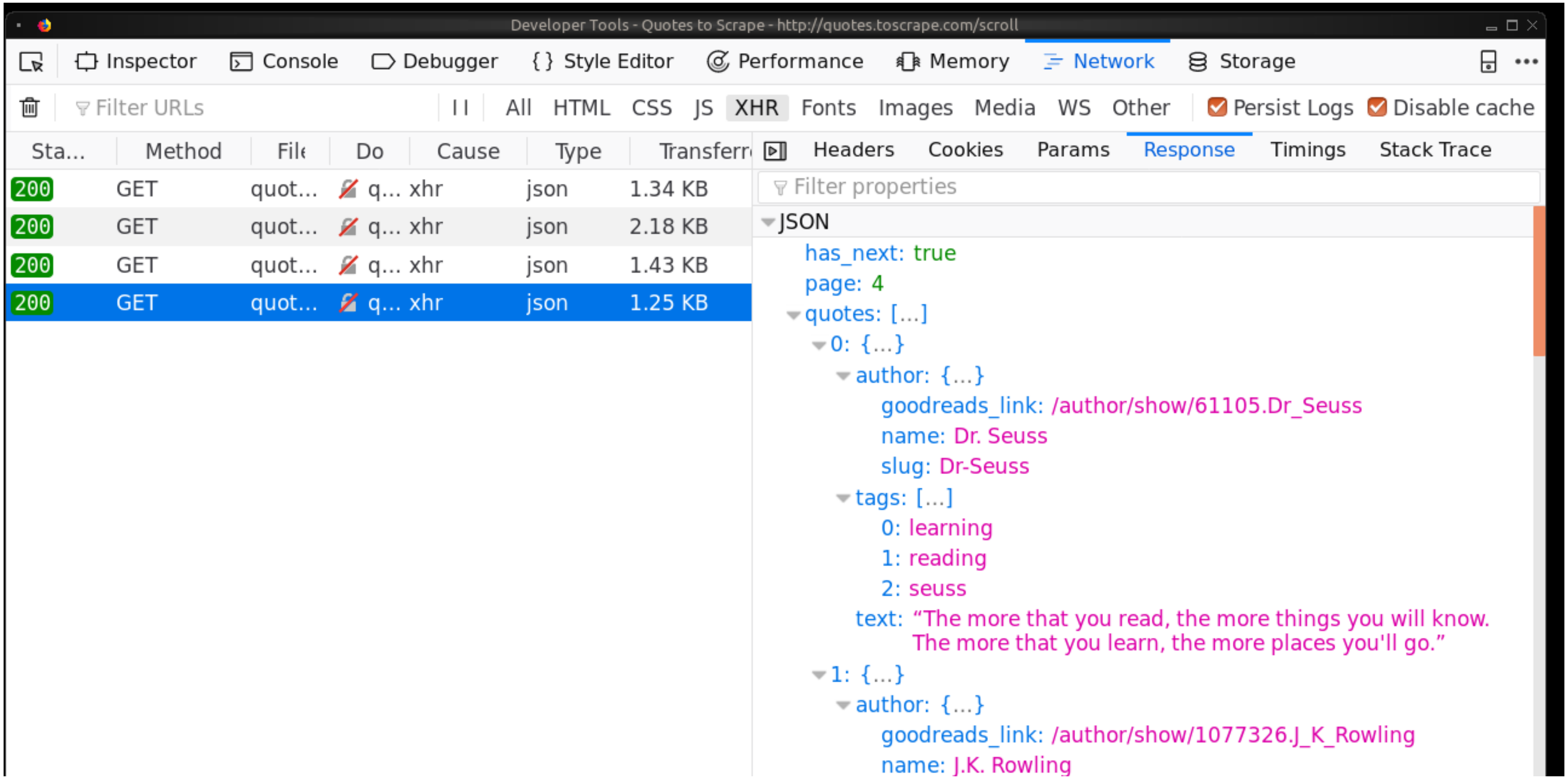
Request URL: `http://quotes.toscrape.com/api/quotes?page=4`
Request method: GET
Remote address: 136.243.118.219:80
Status code: 200
Version: HTTP/1.1

Response headers (222 B)

Connection: keep-alive
Content-Encoding: gzip
Content-Type: application/json
Date: Sun, 05 Aug 2018 14:33:13 GMT
Server: nginx/1.12.1
Transfer-Encoding: chunked
X-Upstream: spidyquotes-master_web

Request headers (375 B)

Accept: */*



Sta...	Method	File	Do	Cause	Type	Transferr
200	GET	quot...	q...	xhr	json	1.34 KB
200	GET	quot...	q...	xhr	json	2.18 KB
200	GET	quot...	q...	xhr	json	1.43 KB
200	GET	quot...	q...	xhr	json	1.25 KB

```
JSON
  has_next: true
  page: 4
  quotes: […]
    0: {…}
      author: {…}
        goodreads_link: /author/show/61105.Dr_Seuss
        name: Dr. Seuss
        slug: Dr-Seuss
      tags: […]
        0: learning
        1: reading
        2: seuss
      text: "The more that you read, the more things you will know.
             The more that you learn, the more places you'll go."
    1: {…}
      author: {…}
        goodreads_link: /author/show/1077326.J_K_Rowling
        name: J.K. Rowling
```

```
import json

class ScrollQuotesSpider(scrapy.Spider):

  (...)

  def parse(self, response):
    data = json.loads(response.body)
```

http://quotes.toscrape.com/login

Página com acesso restrito por login.

```
$ scrapy genspider login_quotes quotes.toscrape.com
```

```
yield scrapy.FormRequest(
  url,
  formdata,
  callback,
)
```