Scraping the Web

the workshop

José Manuel Ortega

@jmortegac

{codemotion}

Agenda

- Librerías python
- BeautifulSoup
- Scrapy / Proyectos
- Mechanize / Selenium
- Herramientas web / plugins

Repositorio Github

https://github.com/jmortega/codemotion_scraping_the_web

Branch: master codemotion_scraping_the_web / +

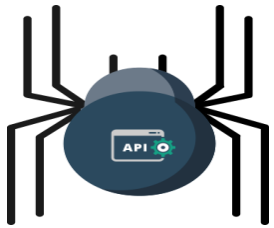
jortega Scraping the web

BeautifulSoup	scraping the web
login_bitbucket	scraping the web
mechanical-soup	scraping the web
mechanize	scraping the web
robobrowser	scraping examples
scrapy	scraping the web
selenium	scraping examples
webscraping	scraping the web
README.md	scraping the web
contenido_Scrapy.md	scraping examples
introduccion_Scrapy.md	scraping examples
json_obtain_ip.py	Scraping the web
recursosadicionales_Scrapy.md	scraping examples
request_duckduckgo.py	Scraping the web
urllib2_codemotion.py	Scraping the web

Técnicas de scraping

- Screen scraping
- Web scraping
- Report mining
- Spider

Webscraping



- Es el proceso de recolección o extracción de datos de páginas web de forma automática.

- Técnica que se emplea para extraer datos usando herramientas de software, está relacionada con la indexación de información que está en la web empleando un robot

- Metodología universal adoptada por la mayoría de los motores de búsqueda.

Python

- http://www.python.org
- Lenguaje de programación interpretado multiparadigma, soporta orientación a objetos, programación imperativa y, en menor medida programación funcional.
- Usa tipado dinámico y es multiplataforma.

Librerías Python

- Requests
- Lxml
- Regular expressions
- Beautiful Soup 4
- Pyquery
- Webscraping
- Scrapy
- Mechanize
- Selenium

Request libraries

- **Urllib2**
- Python ***requests***: HTTP for Humans
  - $ pip install requests

Requests

http://docs.python-requests.org/en/latest

Requests

Star 15,953

Requests is an elegant and simple HTTP library for Python, built for human beings.

Buy Requests Pro

Get Updates

Receive updates on new releases and upcoming projects.

Subscribe to Newsletter

Translations

English
French
German
Japanese
Chinese
Portuguese
Italian

Requests: HTTP for Humans

Release v2.8.1. (Installation)

Requests is an Apache2 Licensed HTTP library, written in Python, for human beings.

Python's standard **urllib2** module provides most of the HTTP capabilities you need, but the API is thoroughly **broken**. It was built for a different time — and a different web. It requires an *enormous* amount of work (even method overrides) to perform the simplest of tasks.

Things shouldn't be this way. Not in Python.

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type":"User"...'
>>> r.json()
{u'private_gists': 419, u'total_private_repos': 77, ...}
```

See similar code, without Requests.

Requests takes all of the work out of Python HTTP/1.1 — making your integration with web services seamless. There's no need to manually add query strings to your URLs, or to form-encode your POST data. Keep-alive and HTTP connection pooling are 100% automatic, powered by urllib3, which is embedded within Requests.

Testimonials

Requests

```
import requests

url = "http://duckduckgo.com/html"
payload = {'q':'python'}
r = requests.get(url, payload)
print r.text.encode('utf-8')
with open("requests_results.html", "w") as f:
    f.write(r.text.encode('utf-8'))
```

Web scraping with Python

1. Download webpage with urllib2, requests
2. Parse the page with BeautifulSoup/lxml
3. Select with XPath or css selectors

Web scraping with Python

- Regular expressions
  - <h1>(.*?)</h1>
- Xpath
  - //h1
- Generar un objeto del HTML (tipo DOM)
  - page.h1

Regular expressions

- **[A-Z]** matches a capital letter
- **[0-9]** matches a number
- **[a-z][0-9]** matches a lowercase letter followed by a number
- **star *** matches the previous item 0 or more times
- **plus +** matches the previous item 1 or more times
- **dot .** will match anything but line break characters \r \n
- **question ?** makes the preceeding item optional

BeautifulSoup

- Librería que permite el parseo de páginas web
- Soporta parsers como lxml,html5lib
- Instalación
  - pip install lxml
  - pip instlal html5lib
  - pip install beautifulsoup4
  - http://www.crummy.com/software/BeautifulSoup

BeautifulSoup

- soup = BeautifulSoup(html_doc,’lxml’)
- Print all: print(soup.prettify())
- Print text: print(soup.get_text())

```
from bs4 import BeautifulSoup
```

BeautifulSoup functions

- **find_all('a')**→Obtiene una lista con todos los enlaces
- **find('title')**→Obtiene el primer elemento <title>
- **get('href')**→Obtiene el valor del atributo href de un determinado elemento
- **(element).text** → obtiene el texto asociado al elemento

```
for link in soup.find_all('a'):
    print(link.get('href'))
```

Extracting links with bs4

https://news.ycombinator.com

```
def get_front_page():
    target = "https://news.ycombinator.com"
    frontpage = requests.get(target)
    if not frontpage.ok:
        raise RuntimeError("Can't access hacker news, you should go outside")
    news_soup = BeautifulSoup(frontpage.text,"lxml")
    return news_soup


def find_interesting_links(soup):
    items = soup.findAll('td', {'align': 'right', 'class': 'title'})
    links = []
    for i in items:
        try:
            siblings = list(i.next_siblings)
            post_id  = siblings[1].find('a')['id']
            link     = siblings[2].find('a')['href']
            title    = siblings[2].text
            links.append({'link': link, 'title': title,'post_id':post_id})
        except Exception as e:
            print e

    return links
```

Extracting links with bs4

https://news.ycombinator.com

```
http://playbook.samaltman.com Startup Playbook (samaltman.com)
http://tpp.mfat.govt.nz/text Text of the Trans-Pacific Partnership (mfat.govt.nz)
https://blog.wearewizards.io/a-lot-of-websockets-in-haskell A lot of websockets in Haskell (wearewizards.io)
https://medium.com/@wwhchung/sr-ed-hurts-canadian-innovation-4fbcf4898d7d Canada's R&D tax credit program hurts R&D in Canada (medium.com)
https://archive.org/details/AtariStarRaidersSourceCode Atari Star Raiders Source Code (1979) (archive.org)
http://blog.fogus.me/2015/11/04/the-100101-method-my-approach-to-open-source/ The 100:10:1 method: my approach to open source (fogus.me)
https://blog.mozilla.org/blog/2015/11/03/firefox-now-offers-a-more-private-browsing-experience/ Firefox Now Offers a More Private Browsing Experience (mozilla.org)
http://www.bloomberg.com/news/articles/2015-11-03/when-a-127-year-old-u-s-industry-collapses-under-china-s-weight 127-Year-Old U.S. Aluminum Industry Collapses Under China's We
http://uploadvr.com/lytro-immerge-vr-light-field-video-camera/ Lytro announces light field VR video camera (uploadvr.com)
http://news.stanford.edu/news/2015/november/robinson-humanities-lecture-110315.html Novelist warns against utilitarian trends in higher education (stanford.edu)
http://on.wsj.com/1iDRXGf Big Banks Lock Horns with Personal-Finance Web Portals (wsj.com)
http://gynvael.coldwind.pl/n/c_cpp_number_to_binary_string_01011010 8-bit number to binary string (gynvael.coldwind.pl)
https://github.com/benoitvallon/react-native-nw-react-calculator A mobile, desktop and website app with the same code (github.com)
http://www.gsb.stanford.edu/insights/what-it-be-owned-warren-buffett What is it like to be owned by Warren Buffett? (stanford.edu)
http://www.nature.com/nature/journal/v527/n7576_supp/full/527S2a.html Big data in genomics: The $1k genome has arrived (nature.com)
http://around.com/let-twitter-be-twitter/ Let Twitter Be Twitter (around.com)
http://publicdomainreview.org/2013/05/16/athanasius-kircher-and-the-hieroglyphic-sphinx/ Athanasius Kircher and the Hieroglyphic Sphinx (2013) (publicdomainreview.org)
http://www.nature.com/news/zombie-physics-6-baffling-results-that-just-won-t-die-1.18685 Zombie physics: Baffling results that won't die (nature.com)
https://www.eff.org/deeplinks/2015/11/hurricane-lte-u-dont-let-wi-fi-get-blown-away Hurricane LTE-U: DonΓÇÖt Let Wi-Fi Get Blown Away (eff.org)
http://savannah.gnu.org/forum/forum.php?forum_id=8398 GNU Guix 0.9.0 released ΓÇô Functional package manager and distribution (gnu.org)
http://www.nytimes.com/2015/11/05/world/australia/australia-penguins-sheepdogs-foxes-swampy-marsh-farmer-middle-island.html Australia Deploys Sheepdogs to Save a Penguin Colony
http://www.networkworld.com/article/3000809/software/its-time-to-update-the-software-update-process.html It's time to update the software update process (networkworld.com)
http://qz.com/540571/baidu-found-chinas-ghost-cities-but-it-is-keeping-their-locations-mostly-a-secret/ Baidu found ChinaΓÇÖs ΓÇ£ghost citiesΓÇ¥ but is keeping their locations
http://www.theguardian.com/world/2015/nov/04/house-bill-end-warrantless-stingray-surveillance-jason-chaffetz Congressman introduces bill to end warrantless Stingray surveillanc
http://decomment.com/ Read deleted comments on Reddit (decomment.com)
http://research.dyn.com/2015/10/the-threat-of-telecom-sabotage/ The Threat of Telecom Sabotage (dyn.com)
http://www.bbc.co.uk/news/magazine-34420194 I found my father living on the street (bbc.co.uk)
https://pl.cs.jhu.edu/projects/big-bang/ The Big Bang project aims to create a typed language with the feel of scripting (jhu.edu)
http://www.wired.com/2015/11/innit-future-kitchen/ A Kitchen That Cooks by Itself (wired.com)
```

Extracting linkedin info with bs4

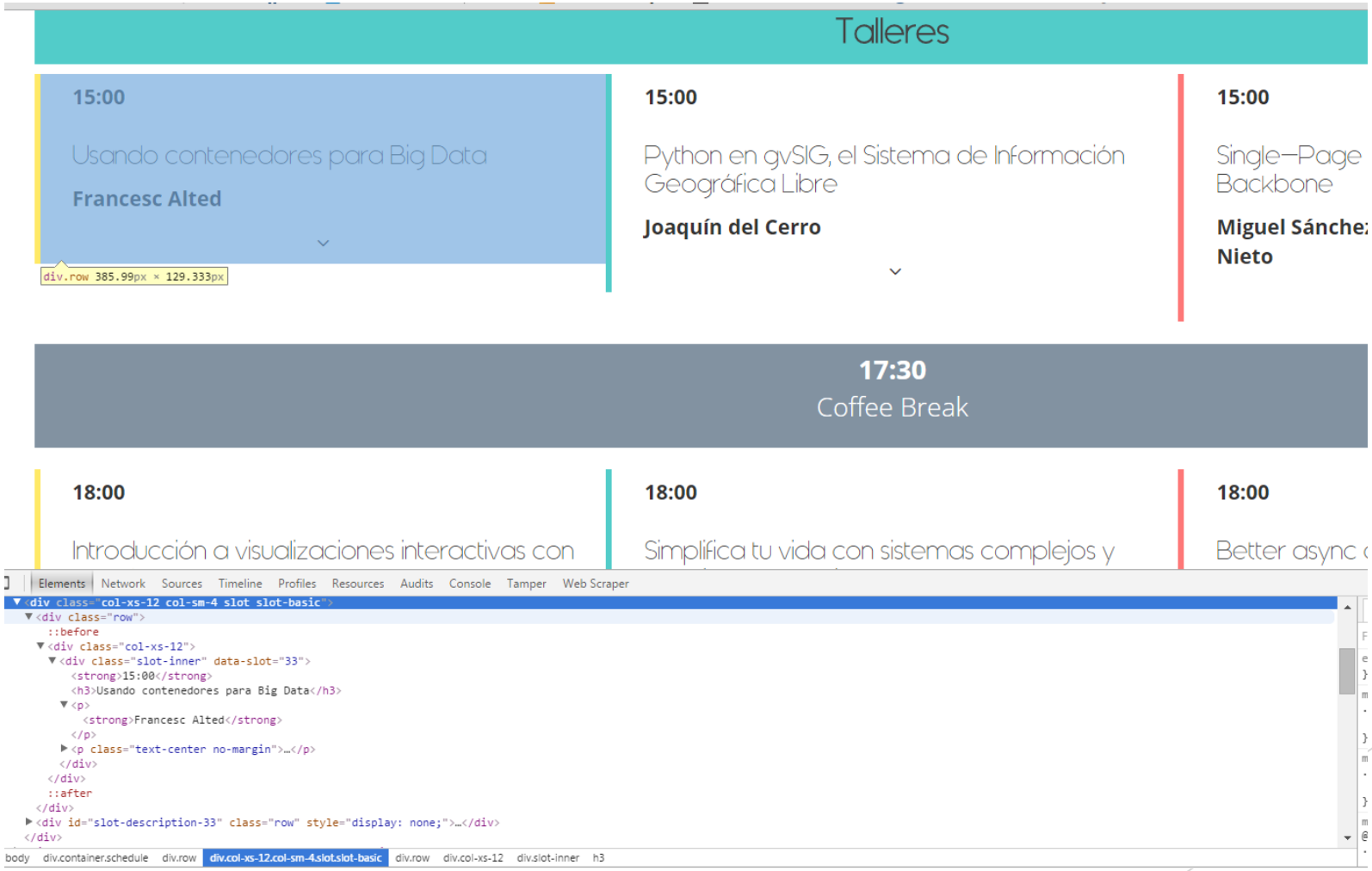
```
EXAMPLE USAGE:
> python linkedin_simple.py google linkedin

'''

from bs4 import BeautifulSoup
import sys, urllib2, random, time, json
import requests

# Get a list of company names from the command line
CO_NAMES = sys.argv[1:]

# Loop over the list of companies
for company in CO_NAMES:

  # Get the html data
  url = "http://www.linkedin.com/company/{}".format(company)

  # Get the HTML contents of the page and put it into BeautifulSoup
  opener = urllib2.build_opener()

  # Create a user agent
  opener.addheaders = [('User-agent', 'Mozilla/5.0 (Macintosh; Intel Mac OS X 1

  # Open the page and read the contents into a variable we can use
  company_page = opener.open(url).read()

  # Create the soup so we can parse it with BS4
  soup = BeautifulSoup(company_page.decode('utf-8', 'ignore'),"lxml")

   # Work with a smaller subset
  node = soup.find(attrs = {"class" : "grid-f"})

  description = soup.find(attrs = {"class" : "basic-info-description"})
```

Extracting linkedin info with bs4

```
Since our founding in 1998, Google has grown by leaps and bounds. From offering search in a single language
b applications for all kinds of tasks┌Çöin scores of languages. And starting from two computer science stude
ld. A lot has changed since the first Google search engine appeared. But some things haven┌ÇÖt changed: our


Specialties

search,
ads,
mobile,
android,
online video



Internet
[u'1600 Amphitheatre Parkway', u'', u'Mountain View,', u'94043', u'United States']
['street-address', 'street-address', 'locality', 'postal-code', 'country-name']
[
  "google",
  {
    "postal-code": "94043",
    "street-address": "",
    "country-name": "United States",
    "locality": "Mountain View,"
  }
]

Founded in 2003, LinkedIn connects the world's professionals to make them more productive and successful. Wi
any, LinkedIn is the world's largest professional network on the Internet. The company has a diversified bus
riptions products. Headquartered in Silicon Valley, LinkedIn has offices across the globe.


Specialties

Online Professional Network,
Jobs,
People Search,
Company Search,
Address Book,
Advertising,
Professional Identity,
Group Collaboration
```

Extraer datos de la agenda de la pycones

http://2015.es.pycon.org/es/schedule

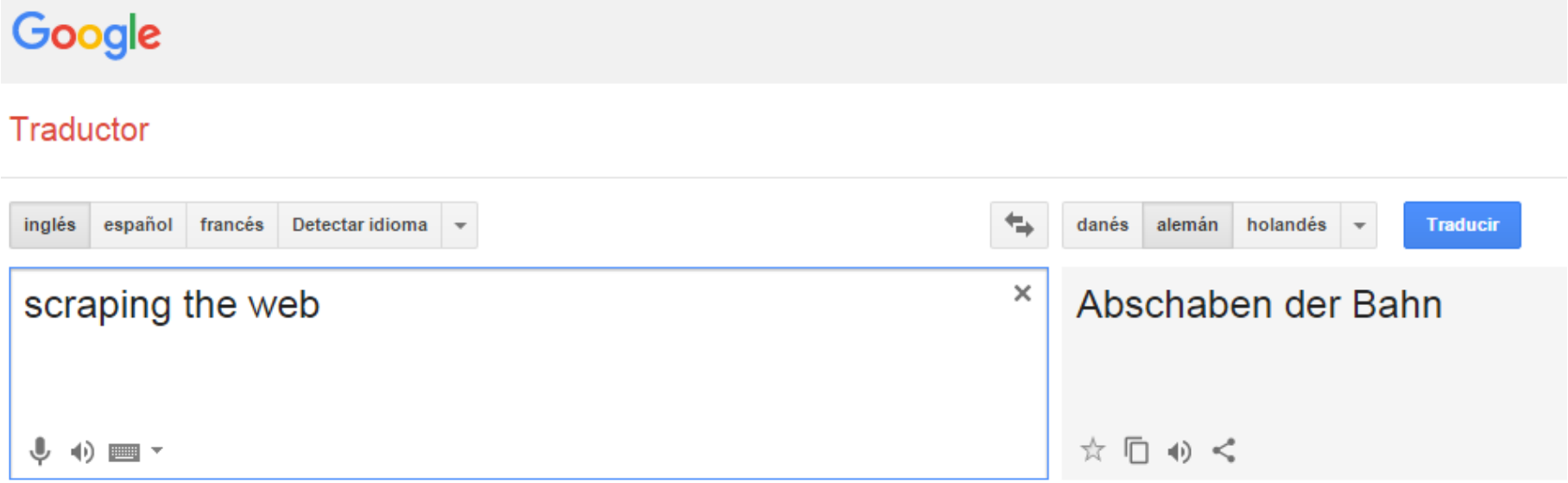


Extraer datos de la agenda de pycones

Beautiful Soup 4

```
    for slot in slots:
        demo = slot.find_all('div', {'class': 'col-xs-12'})
        for slot in demo:
            if slot is not None:
                slot2 = slot.find_all('div', {'class': 'slot-inner'})
                for aux in slot2:
                    speaker = aux.find('p').find('strong')
                    title = aux.find('h3')
                    hour = aux.find('strong')
                description = slot.find('p')
                talk_pycones = {}
                if speaker is not None and title is not None and hour is not None and descriptic
                    talk_pycones['speaker'] = speaker.text.encode('utf-8')
                    talk_pycones['title'] = title.text.encode('utf-8')
                    talk_pycones['hour'] = hour.text.encode('utf-8')
                    talk_pycones['description'] = description.text.encode('utf-8')
                    talks_pycones.append(talk_pycones)

if __name__=="__main__":
    main()

    outfile = open('pycones.json','wb')

    for talk_pycones in talks_pycones:
        print talk_pycones['description']
        print talk_pycones['hour']
        print talk_pycones['title']
        print talk_pycones['speaker']
        line = json.dumps(talk_pycones) + "\n"
        outfile.write(line)
```

Google translate




```
[evaluate google_translate.py]
Language to Convert from : english
Language to Convert to : de
Text to Convert : scraping the web
Converted Text : Abschaben der Bahn
```

Webscraping library

- *pip install webscraping*
- https://bitbucket.org/richardpenman/webscraping/overview
- http://docs.webscraping.com
- https://pypi.python.org/pypi/webscraping

Extraer datos de la agenda de pycones

webscraping

```
#!/usr/bin/env python
#De webscraping se importa download y xpath
from webscraping import download, xpath
import json
import csv

#Se define la instancia Download
D = download.Download()

#get page
html = D.get('http://2015.es.pycon.org/es/schedule/')

index =0
talks_pycones = []

#obtener el div donde se muestra la info de cada conferencia
for row in xpath.search(html, '//div[@class="col-xs-12"]'):

    if index%2 ==0:
        talk = xpath.search(row, '//div[@class="slot-inner"]/h3')

        author = xpath.search(row, '//div[@class="slot-inner"]/p/strong')

        hour = xpath.search(row, '//div[@class="slot-inner"]/strong')

    if index%2 !=0:
        description = xpath.search(row, '/p')

        if talk is not None and author is not None and description is not None and
            talk_pycones ={}
            talk_pycones['talk'] = talk[0]
            talk_pycones['author'] = author[0]
            talk_pycones['description'] = description[0]
            talk_pycones['hour'] = hour[0].encode('utf-8')

            talks_pycones.append(talk_pycones)

    index+=1
```

Meet Scrapy

An **open source** and collaborative framework for **extracting the data you need** from websites. In a fast, simple, yet extensible way.

pypi v1.0.3 | downloads 44k/month | wheel yes | PY3 72% | coverage 82%

Install latest version:

Scrapy 1.0

$ pip install scrapy

PyPI | Ubuntu Package | Tarball | Zip

Scrapy

Build your **own webcrawlers**

Sample Code:

```
$ pip install scrapy
$ cat > myspider.py <<EOF
import scrapy

class BlogSpider(scrapy.Spider):
    name = 'blogspider'
    start_urls = ['http://blog.scrapinghub.com']

    def parse(self, response):
        for url in response.css('ul li a::attr("href")').re(r'.*/\d\d\d\d/\d\d/$'):
            yield scrapy.Request(response.urljoin(url), self.parse_titles)

    def parse_titles(self, response):
        for post_title in response.css('div.entries > ul > li a::text').extract():
            yield {'title': post_title}
EOF
$ scrapy runspider myspider.py
```

Scrapy

- open-source
- Framework que permite crear spiders para ejecutar procesos de crawling de pag web
- Permite la definición de reglas Xpath mediante expresiones regulares para la extracción de contenidos
- Basada en la librería twisted

Scrapy

- Simple, conciso
- Extensible
  - Señales, middlewares
- Rápido
  - IO asíncrona (twisted), parseo en C (libxml2)
- Portable
  - Linux, Windows, Mac
- Bien testeado
  - 778 unit-tests, 80% de cobertura
- Código limpio (PEP-8) y desacoplado
- Zen-friendly / pythónico

Scrapy

- Utiliza un mecanismo basado en expresiones XPath llamado **Xpath Selectors.**
- Utiliza **LXML XPath** para encontrar elementos
- Utiliza **Twisted** para el operaciones **asíncronas**

Ventajas scrapy

- Más rápido que mechanize porque utiliza operaciones asíncronas (emplea Twisted).
- Scrapy tiene un mejor soporte para el parseado del html
- Scrapy maneja mejor caracteres unicode, redirecciones, respuestas gzipped, codificaciones.
- Caché HTTP integrada.
- Se pueden exportar los datos extraídos directamente a csv o JSON.

Scrapy

▶XPath selectors

Expression	Result
nodename	Selects all nodes with the name "nodename"
/	Do selection from the **root**
//	Do selection from current node
.	Select current node
..	Select parent node
@attr	Select attributes of nodes
text()	Select the value of chosen node

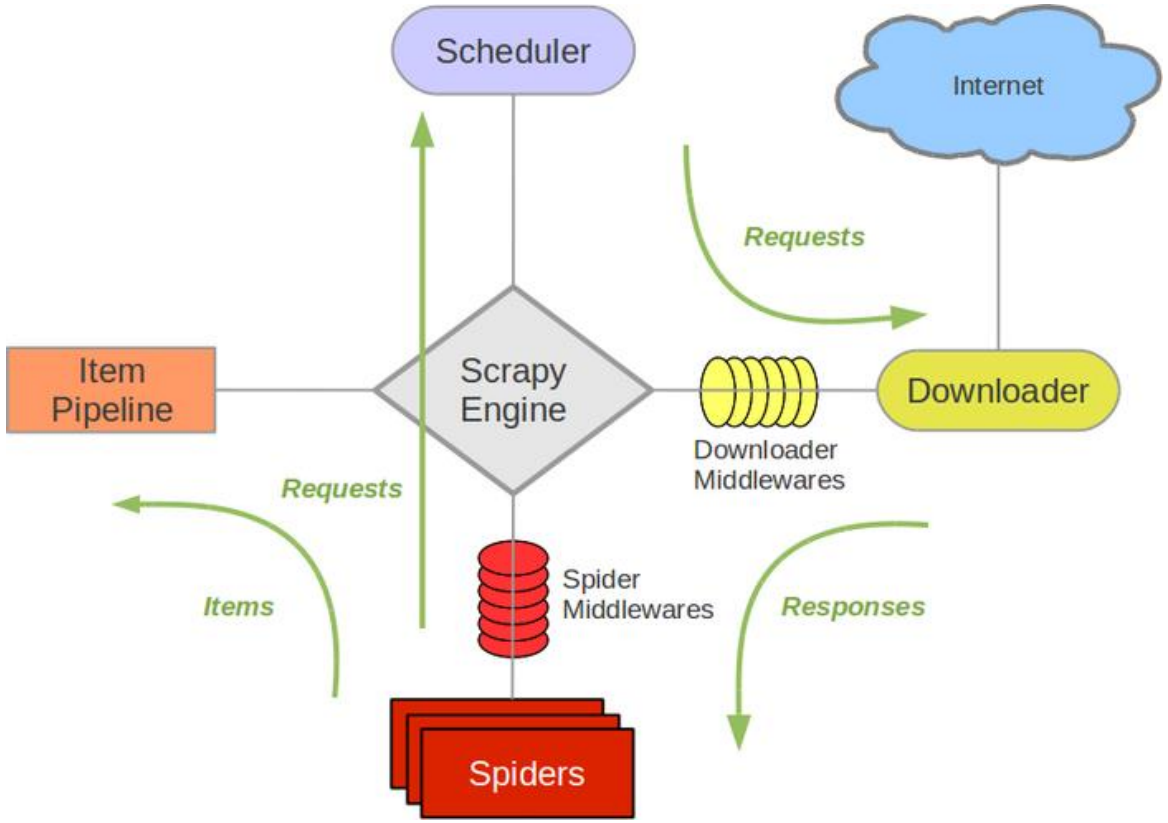
Xpath selectors

Expression	Meaning
name	matches all nodes on the current level with the specified name
name[n]	matches the nth element on the current level with the specified name
/	Do selection from the root
//	Do selection from current node
*****	matches all nodes on the current level
. Or ..	Select current / parent node
@name	the attribute with the specified name
[@key='value']	all elements with an attribute that matches the specified key/value pair
name[@key='value']	all elements with the specified name and an attribute that matches the specified key/value pair
[text()='value']	all elements with the specified text
name[text()='value']	all elements with the specified name and text

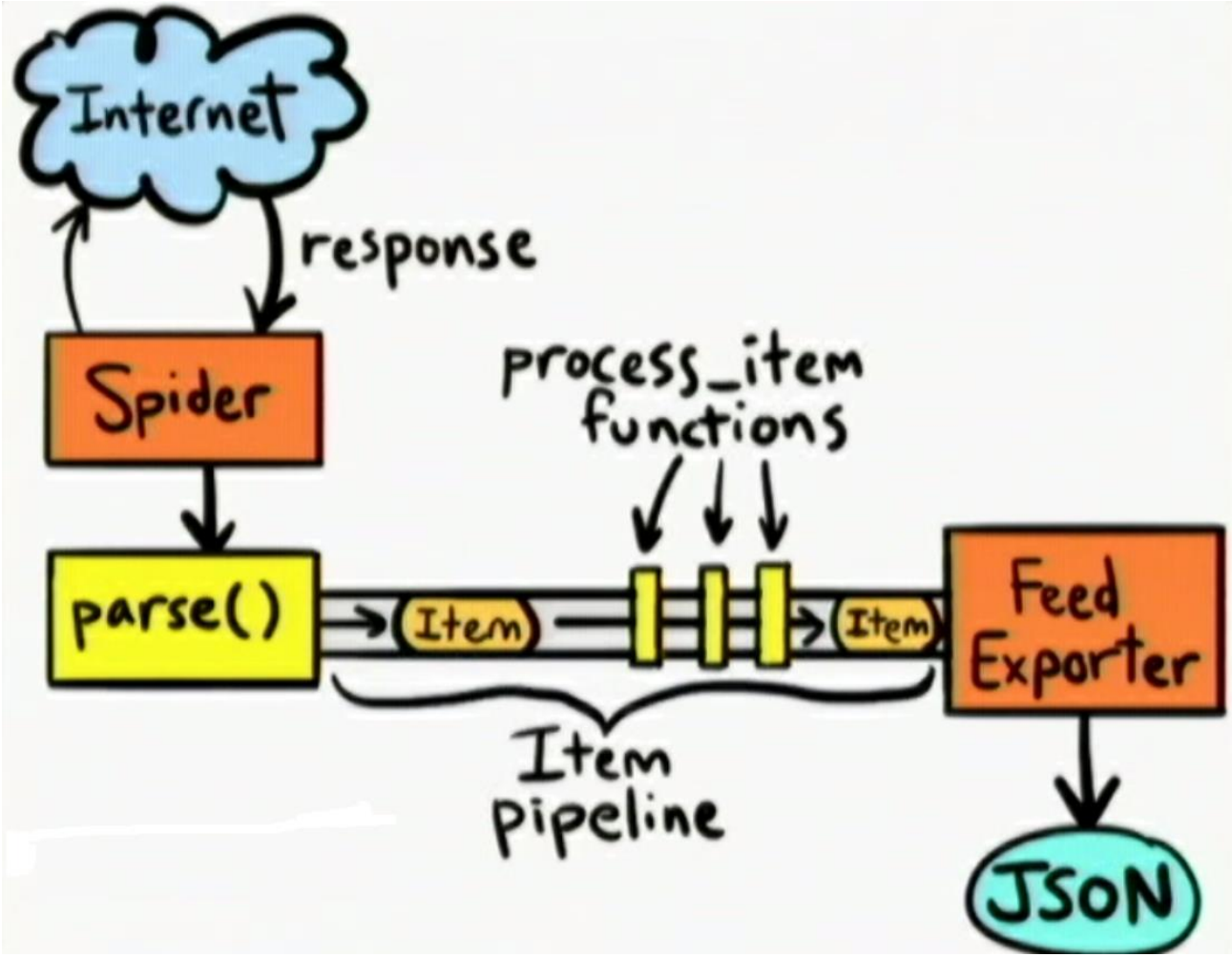
Scrapy

- Cuando usamos Scrapy tenemos que crear un proyecto, y cada proyecto se compone de:
- **Items→** Definimos los elementos a extraer.
- **Spiders→** Es el corazón del proyecto, aquí definimos el procedimiento de extracción.
- **Pipelines→** Son los elementos para analizar lo obtenido: validación de datos, limpieza del código html

Architecture



Architecture



Instalación de scrapy

- Python 2.6 / 2.7
- Lxml
- openSSL
- pip / easy_install

```
$ pip install scrapy
$ easy_install scrapy
```

Instalación de scrapy

```
Collecting scrapy
  Downloading Scrapy-0.24.6-py2-none-any.whl (444kB)
    100% |################################| 446kB 145kB/s
Collecting cssselect>=0.9 (from scrapy)
  Downloading cssselect-0.9.1.tar.gz
Collecting queuelib (from scrapy)
  Downloading queuelib-1.2.2-py2.py3-none-any.whl
Collecting pyOpenSSL (from scrapy)
  Downloading pyOpenSSL-0.15.1-py2.py3-none-any.whl (102kB)
    100% |################################| 106kB 92kB/s
Collecting w3lib>=1.8.0 (from scrapy)
  Downloading w3lib-1.11.0-py2.py3-none-any.whl
Collecting lxml (from scrapy)
  Downloading lxml-3.4.4-cp27-none-win32.whl (3.0MB)
    100% |################################| 3.0MB 35kB/s
Collecting Twisted>=10.0.0 (from scrapy)
  Downloading Twisted-15.2.1-cp27-none-win32.whl (3.2MB)
    100% |################################| 3.2MB 37kB/s
Collecting six>=1.5.2 (from scrapy)
  Downloading six-1.9.0-py2.py3-none-any.whl
Collecting cryptography>=0.7 (from pyOpenSSL->scrapy)
  Downloading cryptography-0.9.1-cp27-none-win32.whl (989kB)
    100% |################################| 991kB 100kB/s
Collecting zope.interface>=3.6.0 (from Twisted>=10.0.0->scrapy)
  Downloading zope.interface-4.1.2.tar.gz (919kB)
    100% |################################| 921kB 72kB/s
Requirement already satisfied (use --upgrade to upgrade): setuptools in c:\python27\lib\site-packages (from cryptography>=0.7->pyOpenSSL->scrapy)
Collecting enum34 (from cryptography>=0.7->pyOpenSSL->scrapy)
  Downloading enum34-1.0.4.tar.gz
Collecting pyasn1 (from cryptography>=0.7->pyOpenSSL->scrapy)
  Downloading pyasn1-0.1.7.tar.gz (68kB)
    100% |################################| 69kB 718kB/s
Collecting idna (from cryptography>=0.7->pyOpenSSL->scrapy)
  Downloading idna-2.0-py2.py3-none-any.whl (61kB)
    100% |################################| 61kB 718kB/s
Collecting ipaddress (from cryptography>=0.7->pyOpenSSL->scrapy)
  Downloading ipaddress-1.0.7-py27-none-any.whl
```

```
pip install scrapy
```

Scrapy Shell (no es necesario crear proyecto)

```
scrapy shell <url>
```

```
[s] Available Scrapy objects:
[s]   crawler    <scrapy.crawler.Crawler object at 0x006AA4D0>
[s]   item       {}
[s]   request    <GET http://2015.es.pycon.org/es/schedule>
[s]   response   <200 http://2015.es.pycon.org/es/schedule/>
[s]   settings   <scrapy.settings.Settings object at 0x0404BDD0>
[s]   spider     <DefaultSpider 'default' at 0x4abf8f0>
[s] Useful shortcuts:
[s]   shelp()           Shell help (print this help)
[s]   fetch(req_or_url) Fetch request (or URL) and update local objects
[s]   view(response)    View response in a browser
```

```
from scrapy.select import Selector
hxs = Selector(response)
Info = hxs.select('//div[@class="slot-inner"]')
```

Scrapy Shell

scrapy shell http://scrapy.org

```
2015-11-05 19:36:27 [scrapy] INFO: Scrapy 1.0.3 started (bot: scrapybot)
2015-11-05 19:36:27 [scrapy] INFO: Optional features available: ssl, http11
2015-11-05 19:36:27 [scrapy] INFO: Overridden settings: {'LOGSTATS_INTERVAL': 0}
2015-11-05 19:36:27 [scrapy] INFO: Enabled extensions: CloseSpider, TelnetConsole, CoreStats, SpiderState
2015-11-05 19:36:28 [scrapy] INFO: Enabled downloader middlewares: HttpAuthMiddleware, DownloadTimeoutMiddleware, UserA
 HttpCompressionMiddleware, RedirectMiddleware, CookiesMiddleware, HttpProxyMiddleware, ChunkedTransferMiddleware, Down
2015-11-05 19:36:28 [scrapy] INFO: Enabled spider middlewares: HttpErrorMiddleware, OffsiteMiddleware, RefererMiddlewar
2015-11-05 19:36:28 [scrapy] INFO: Enabled item pipelines:
2015-11-05 19:36:28 [scrapy] DEBUG: Telnet console listening on 127.0.0.1:6023
2015-11-05 19:36:28 [scrapy] INFO: Spider opened
2015-11-05 19:36:28 [scrapy] DEBUG: Redirecting (302) to <GET http://scrapy.org/> from <GET http://scrapy.org>
2015-11-05 19:36:28 [scrapy] DEBUG: Crawled (200) <GET http://scrapy.org/> (referer: None)
[s] Available Scrapy objects:
[s]   crawler    <scrapy.crawler.Crawler object at 0x003FC4D0>
[s]   item       {}
[s]   request    <GET http://scrapy.org>
[s]   response   <200 http://scrapy.org/>
[s]   settings   <scrapy.settings.Settings object at 0x03F51DF0>
[s]   spider     <DefaultSpider 'default' at 0x49007f0>
[s] Useful shortcuts:
[s]   shelp()           Shell help (print this help)
[s]   fetch(req_or_url) Fetch request (or URL) and update local objects
[s]   view(response)    View response in a browser
2015-11-05 19:36:29 [root] DEBUG: Using default logger
2015-11-05 19:36:29 [root] DEBUG: Using default logger
WARNING: Readline services not available or not loaded.
WARNING: Proper color support under MS Windows requires the pyreadline library.
You can find it at:
http://ipython.org/pyreadline.html

Defaulting color scheme to 'NoColor'

In [1]: response.xpath('//title/text()').extract()
Out[1]: [u'Scrapy | A Fast and Powerful Scraping and Web Crawling Framework']
```

Projecto scrapy

$ scrapy startproject <project_name>

```
tutorial/
    scrapy.cfg
    tutorial/
        __init__.py
        items.py
        pipelines.py
        settings.py
        spiders/
            __init__.py
            ...
```

scrapy.cfg: the project configuration file.

tutorial/:the project's python module.

items.py: the project's items file.

pipelines.py : the project's pipelines file.

setting.py : the project's setting file.

spiders/ : a directory where you'll later put your spiders.

Scrapy europython

http://ep2015.europython.eu/en/events/sessions

EUROPYTHON 2015 SESSIONS

These are the planned sessions for EuroPython 2015 conference. We have more than 200 interesting sessions to attend to, covering a very wide range of topics.

Please note that there may still be some changes to this list in case speakers cannot come and their slots have to be replaced by talks on the waiting list.

Keynotes

- Keynote: Designed for Education: A Python Solution by *Carrie Anne Philbin*
- Keynote: It's Dangerous To Go Alone, Take This: The Power of a Community by *Ola Sitarska*, *Ola Sendecka*

SPONSOR EUROPYTHON

Please help us create an affordable conference for everyone by sponsoring EuroPython!

DID YOU KNOW

The summer music festivals, namely, the

Items

```
import scrapy


class EuropythonItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    title = scrapy.Field()
    author = scrapy.Field()
    description = scrapy.Field()
    date = scrapy.Field()
    tags = scrapy.Field()
```

Crear Spider

- $ scrapy genspider -t basic <YOUR SPIDER NAME> <DOMAIN>

Listado de spiders de un proyecto

- $ scrapy list

Spider

```
class EuropythonSpyderSpider(CrawlSpider):
    name = "europython_spyder"
    allowed_domains = ["ep2015.europython.eu"]
    start_urls = ['http://ep2015.europython.eu/en/events/sessions']

    # Patrón para las entradas que cumplan el formato conference/talks
    rules = [Rule(LxmlLinkExtractor(allow=['conference/talks']),callback='process_response')]

    def process_response(self, response):
        item = EuropythonItem()
        print response
        item['title'] = response.xpath("//div[contains(@class, 'grid-100')]//h1/text()").extract()
        item['author'] = response.xpath("//div[contains(@class, 'talk-speakers')]//a[1]/text()").extract()
        item['description'] = response.xpath("//div[contains(@class, 'cms')]//p//text()").extract()
        item['date'] = response.xpath("//section[contains(@class, 'talk when')]/strong/text()").extract()
        item['tags'] = response.xpath("//div[contains(@class, 'all-tags')]/span/text()").extract()

        return item
```

Pipeline

- ITEM_PIPELINES = [‘<your_project_name>.pipelines.<your_pipeline_classname>']
- pipelines.py

```
class EuropythonPipeline(object):
    def __init__(self):
        self.file = codecs.open('europython_items.json', 'w+b', encoding='utf-8')

    def process_item(self, item, spider):
        line = json.dumps(dict(item), ensure_ascii=False) + "\n"
        self.file.write(line)
        return item

    def spider_closed(self, spider):
        self.file.close()
```

Pipeline SQLite

EuropythonSQLitePipeline

```
db = Database("sqlite", "europython.sqlite", create_db=True)

class EuropythonSession(db.Entity):
        """
        Pony ORM model of the europython session table
        """
        id = PrimaryKey(int, auto=True)
        author = Required(str)
        title = Required(str)
        description = Required(str)
        date = Required(str)
        tags = Required(str)
```

Pipeline SQLite

EuropythonSQLitePipeline

```
# Insert data in database
@db_session
def process_item(self, item, spider):
        # use db_session as a context manager
        with db_session:
                try:
                        strAuthor = str(item['author'])
                        strAuthor = strAuthor[3:len(strAuthor)-2]

                        strTitle = str(item['title'])
                        strTitle = strTitle[3:len(strTitle)-2]

                        strDescription = str(item['description'])
                        strDescription = strDescription[3:len(strDescription)-2]

                        strDate = str(item['date'])
                        strDate = strDate[3:len(strDate)-2]
                        strDate = strDate.replace("[u'", "").replace("']", "").replace("u'", "").replace("',", ",")

                        strTags = str(item['tags'])
                        strTags = strTags.replace("[u'", "").replace("']", "").replace("u'", "").replace("',", ",")

                        europython_session = EuropythonSession(author=strAuthor,title=strTitle,
                                                               description=strDescription,date=strDate,tags=strTags)
```

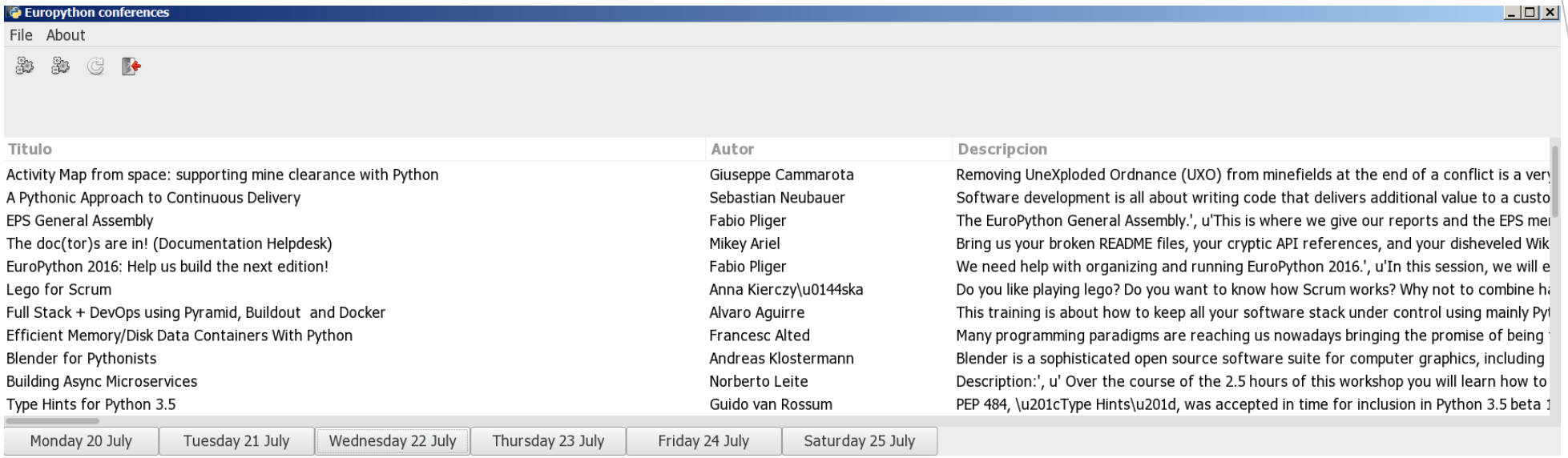
Pipeline SQLite

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: EuropythonSession

	id	author	title	description	date	tags
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Carrie Anne ...	Keynote: Designed for Education: A Pyth...	The problem of introducing childr...	Thursday 23...	python
2	2	Kyran Dale	Data-visualisation with Python and Javas...	To accompany an upcoming O\u...	Tuesday 21 J...	visualization, web, flask, javascript, matplotlib, ...
3	3	Sebastian Bu...	Distributed locks with Python and Redis	Traditional methods of coping with...	Wednesday ...	redis, twisted, concurrency
4	4	Juan Riaza	Dive into Scrapy	Scrapy is a fast high-level screen s...	Tuesday 21 J...	python, scraping, scrapy, open-source
5	5	Pablo Enfeda...	Decorators demystified	Do you know what happens ever...	Thursday 23...	scopes, decorators, namespaces, closures
6	6	Alexys Jacob	Designing a scalable and distributed applic...	One of the key aspect to keep in ...	Wednesday ...	web, management, flask, DevOps, automation...
7	7	Alejandro Ca...	Deja de pegarte con tus servicios; import...	\xbfY si pudieras centrarte en la fu...	Wednesday ...	web, open-source, zookeeper, rabbitmq, fram...
8	8	Ana Balica	Demystifying Mixins with Django	Mixins are a great way to keep an ...	Friday 24 July	mixins, OOP, django
9	9	Fernando Ma...	Data Structures with Python	Data Structures is traditionally a \...	Thursday 23...	education
10	10	Rados\u014...	Code Quality in Python - tools and reasons	Beginner\u2019s guide to Python...	Tuesday 21 J...	python, metrics, automation
11	11	Alexander He...	Data Analysis and Map-Reduce with mon...	The MongoDB aggregation frame...	Wednesday ...	python, mongodb, pymongo, bigdata, analytics
12	12	Sever Banesiu	Distributed Workflows with Flowy	This presentation introduces Flow...	Monday 20 J...	distributed-computing, AWS, SWF, workflow
13	13	Llu\xeds Esq...	CityBikes: bike sharing networks around t...	CityBikes [1] started on 2010 as a...	Wednesday ...	visualization, data-science, nosql, flask, mongo...
14	14	Wojciech Lic...	Continuous Deployment for webapps bas...	When you see users starting to u...	Friday 24 July	Best Practice, case study, django, Tooling, dep...
15	15	Raphael Pierz...	Come to the Dark Side! We have a whole...	(This talk is intended for intermedi...	Thursday 23...	Best Practice, open-source, community, Cooki...
16	16	To be annou...	Recruiting sponsors presentation	Recruiting sponsors presentation.	Tuesday 21 J...	recruiting
17	17	Paul Roeland	Plone help desk	Plone help desk	Tuesday 21 J...	web, CMS, Plone
18	18	Michael Sche...	Ansible helpdesk	For this helpdesk session, I will be ...	Thursday 23...	Operations, system-administration, CLI, yaml, ...
19	19	Fabio Pliger	EuroPython 2016: Help us build the next...	We need help with organizing and...	Wednesday ...	conference, EuroPython, eps
20	20	Fabio Pliger	EPS General Assembly	The EuroPython General Assembl...	Wednesday ...	EuroPython, assembly, eps, GA
21	21	Mikey Ariel	The doc(tor)s are in! (Documentation Hel...	Bring us your broken README file...	Wednesday ...	Best Practice, fun, sphinxdocumentation, Sphi...
22	22	Andreas Dewes	Code is not text! How graph technologies...	Today, we almost exclusively think...	Tuesday 21 J...	Best Practice, visualization, code, graphdataba...
23	23	Juan Riaza	Scrapy Helpdesk	Scrapy is an open source and coll...	Tuesday 21 J...	python, scraping, scrapy
24	24	Harry Percival	TDD for web development, from scratch	The aim is to cover the basics of s...	Friday 24 Jul...	Testing, selenium, tdd, unit-testing, django, py...

Europython project GTK



Ejecución

$ scrapy crawl <spider_name>

$ scrapy crawl <spider_name> -o items.json -t json

$ scrapy crawl <spider_name> -o items.csv -t csv

$ scrapy crawl <spider_name> -o items.xml -t xml

Slidebot

```
$ scrapy crawl -a url="" slideshare
$ scrapy crawl -a url="" speakerdeck
```

slideshare

Speaker Deck

Spider SlideShare

```python
from slidebot.basespider import BaseSpider
from slidebot.items import SlideItem


class SlideshareSpider(BaseSpider):
    name = "slideshare"

    def parse(self, response):
        sel = Selector(response)
        og_url = sel.css('meta[name=og_url]::attr(content)').extract()[0]
        slide_id = og_url.rpartition('/')[2]
        return SlideItem(
            # this urls are already fully qualified
            id=slide_id,
            image_urls=sel.css('.slide_image::attr(data-normal)').extract(),
            url=og_url,
        )
```

Slidebot

```
class SlideImages(FilesPipeline):
    """Downloads slide images."""

    DEFAULT_FILES_URLS_FIELD = 'image_urls'
    DEFAULT_FILES_RESULT_FIELD = 'images'

    def get_media_requests(self, item, info):
        reqs = super(SlideImages, self).get_media_requests(item, info)
        self._load_keys(reqs, item)
        return reqs

    def _load_keys(self, requests, item):
        # Preload file paths into the requests because we use the item data to
        # generate the path.
        for req in requests:
            pr = urlparse_cached(req)
            # filename is last part of the URL path.
            image = pr.path.rpartition('/')[-1]
            req.meta['file_path'] = '/{slide_id}/{image}'.format(
                spider=item['spider'],
                slide_id=item['id'],
                image=image,
            )

    def file_path(self, request, response=None, info=None):
        return request.meta['file_path']
```

Slidebot

```
$ scrapy crawl -a
url="http://www.slideshare.net/jmoc25/testing-android-security"
slideshare
```

testing-android-security-1-638.jpg

testing-android-security-2-638.jpg

testing-android-security-3-638.jpg

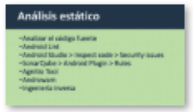
testing-android-security-4-638.jpg

testing-android-security-5-638.jpg

testing-android-security-6-638.jpg

testing-android-security-7-638.jpg

testing-android-security-8-638.jpg



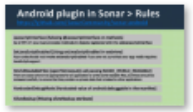
testing-android-security-9-638.jpg

testing-android-security-10-638.jpg

testing-android-security-11-638.jpg

testing-android-security-12-638.jpg

testing-android-security-13-638.jpg



testing-android-security-14-638.jpg

testing-android-security-15-638.jpg

testing-android-security-16-638.jpg

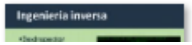
testing-android-security-17-638.jpg

testing-android-security-18-638.jpg

testing-android-security-19-638.jpg

testing-android-security-20-638.jpg

testing-android-security-21-638.jpg



testing-android-security-22-638.jpg

testing-android-security-23-638.jpg

testing-android-security-24-638.jpg

Write CSV / JSON

```
import csv

with open(‘file.csv’,‘wb’) as csvfile:
        writer=csv.writer(csvfile)
        for line in list:
                writer.writerow(line)
```

```
import json

with open(‘file.json’,‘wb’) as jsonfile:
        json.dump(results,jsonfile)
```

Fix encode errors

```
c:\python27\lib\encodings\cp850.pyc in encode(self, input, errors)
     10
     11     def encode(self,input,errors='strict'):
---> 12         return codecs.charmap_encode(input,errors,encoding_map)
     13
     14     def decode(self,input,errors='strict'):

UnicodeEncodeError: 'charmap' codec can't encode character u'\u266b' in position 27: character maps to <undefined>
```

```
myvar.encode("utf-8")
```

Scrapyd

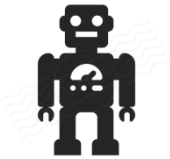
- Scrapy web service daemon

$ pip install scrapyd

- Web API with simple Web UI:
  - http://localhost:6800

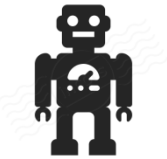
- Web API Documentation:
  - http://scrapyd.readthedocs.org/en/latest/api.html

Mechanize



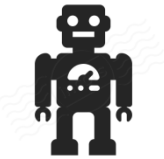
- https://pypi.python.org/pypi/mechanize
- **pip install mechanize**
- **Mechanize permite navegar por los enlaces de forma programática**

Mechanize



```
import mechanize
 # service url
URL = ''
def main():
# Create a Browser instance
b = mechanize.Browser()
# Load the page
b.open(URL)
# Select the form
b.select_form(nr=0)
# Fill out the form
 b[key] = value
# Submit!
return b.submit()
```

Mechanize

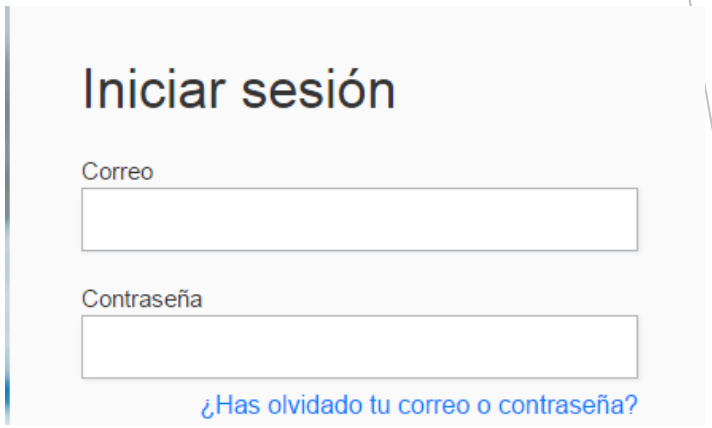


mechanize._response.httperror_seek_wrapper: HTTP Error 403:
request disallowed by robots.txt

browser.set_handle_robots(False)

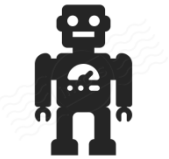
Mechanize netflix login

```python
# Create a Browser
b = mechanize.Browser()

# Disable loading robots.txt
b.set_handle_robots(False)

# Navigate
b.open('https://www.netflix.com/Login?locale=es-ES')

# Choose a form
b.select_form(nr=0)

# Fill it out
b['email'] = 'email'
b['password'] = 'password'

# Stubmit
fd = b.submit()

response =  fd.read()

# ... process the results
soup = BeautifulSoup(response,"lxml")
for link in soup.find_all('a'):
    print(link.get('href'))
```




```
https://www.netflix.com/cookies?locale=es-ES
#launch-evidon
https://www.netflix.com/cookies?locale=es-ES
#launch-evidon
#cookie-disclosure-target
https://www.netflix.com?locale=es-ES
https://www.netflix.com/LoginHelp?locale=es-ES
https://www.netflix.com/rememberme?locale=es-ES
https://www.netflix.com?locale=es-ES
tel:900 971 674
https://www.netflix.com/TermsOfUse?locale=es-ES
https://www.netflix.com/Privacy?locale=es-ES

https://help.netflix.com/support/2101
```

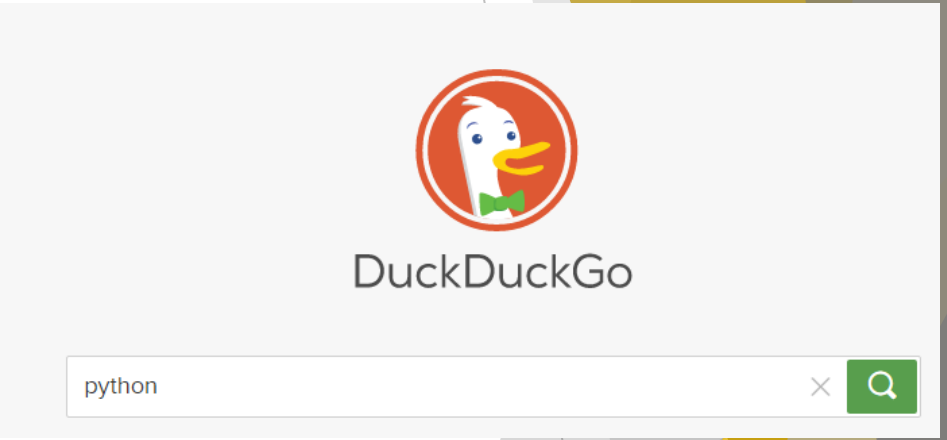
Mechanize utils



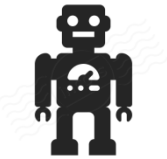
```
def create_browser():
    br = mechanize.Browser()            # Create basic browser
    cj = cookielib.LWPCookieJar()       # Create cookiejar to handle cookies
    br.set_cookiejar(cj)                # Set cookie jar for our browser
    br.set_handle_equiv(True)           # Allow opening of certain files
    br.set_handle_gzip(True)            # Allow handling of zip files
    br.set_handle_redirect(True)        # Automatically handle auto-redirects
    br.set_handle_referer(True)
    br.set_handle_robots(False)         # ignore anti-robots.txt

    # Necessary headers to simulate an actual browser
    br.addheaders = [('User-agent', 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML,
                     ('Accept', 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8'),
                     ('Accept-Charset', 'ISO-8859-1,utf-8;q=0.7,*;q=0.3'),
                     ('Accept-Encoding', 'gzip,deflate,sdch'),
                     ('Accept-Language', 'en-US,en;q=0.8,fr;q=0.6'),
                     ('Connection', 'keep-alive')
                    ]
    return br
```

Mechanize search in duckduckgo

```python
import mechanize

url = "http://duckduckgo.com/html"
br = mechanize.Browser()
br.set_handle_robots(False) # ignore robots
br.open(url)
br.select_form(name="x")
br["q"] = "python"
res = br.submit()
content = res.read()
with open("mechanize_results.html", "w") as f:
    f.write(content)
```



Mechanize extract links



```
import mechanize

br = mechanize.Browser()
response = br.open(url)
for link in br.links():
    print link
```

Alternatives for mechanize

- **RoboBrowser**
  - https://github.com/jmcarp/robobrowser

- **MechanicalSoup**
  - https://github.com/hickford/MechanicalSoup

Robobrowser

- Basada en BeatifulSoup
- Emplea la librería requests
- Compatible con python 3

Dependencies (8)

python *(python-hg)*
python-beautifulsoup4
python-pip *(python-hg)*
python-requests
python-six
python-werkzeug *(python-werkzeug-git)*
python-nose *(check)*
python-setuptools *(make)*

Sources (1)

http://pypi.python.org/packages/source/r/robobrowser/robobrowser-0.5.3.tar.gz

Robobrowser

```python
from robobrowser import RoboBrowser
browser = RoboBrowser()
url = 'https://bitbucket.org/account/signin/'
browser.open(url)

# Get the login form
signin_form = browser.get_form(id='login-form')

# Fill form data
signin_form['username'] = 'USERNAME'
signin_form['password'] = 'PASSWORD'

# Submit form
browser.session.headers['Referer'] = url
signin_form.serialize()
browser.submit_form(signin_form)

#obtain links with beautifulSoup
links = browser.find_all('a')
for link in links:
    #print(link.get('href'))
    if not link['href'].startswith("https"):
        link['href']='https://bitbucket.org'+link['href']
    print link['href']
    print link
    browser.follow_link(link)
    enlaces = browser.find_all('a')
```

Robobrowser

```python
from robobrowser import RoboBrowser

browser = RoboBrowser(history=True)

url = "http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf"
pdf_file_path = "local.pdf"

# do the login (e.g. via a login form)
request = browser.session.get(url, stream=True)

with open(pdf_file_path, "wb") as pdf_file:
    pdf_file.write(request.content)
```

Mechanical soup

```
browser = mechanicalsoup.Browser()

# request github login page. the result is a requests.
#Response object http://docs.python-requests.org/en/latest/user/quickstart/#response-content
login_page = browser.get("https://github.com/login")

# login_page.soup is a BeautifulSoup object http://www.crummy.com/software/BeautifulSoup/bs4/doc/#beautifulsoup
# we grab the login form
login_form = login_page.soup.select("#login")[0].select("form")[0]

# specify username and password
login_form.select("#login_field")[0]['value'] = username
login_form.select("#password")[0]['value'] = password

print login_form.select("#login_field")[0]['value']

# submit form
page = browser.submit(login_form, login_page.url)

counter = page.soup.find('span', class_='counter')
print "\nNumber repositories: " + counter.text
```

```
Number repositories: 30

https://github.com/jmortega/codemotion_scraping_the_web
Description: codemotion_scraping_the_web
Author: jortega
commits 1
branches 1
releases 0
contributors 0

https://github.com/jmortega/python-pentesting
Description: python-pentesting-tool
Author: jortega
commits 7
branches 1
releases 0
contributors
```

Selenium

- Open Source framework for **automating browsers**
- Python-Module
  - http://pypi.python.org/pypi/selenium
- **pip install selenium**
- Firefox-Driver

```
>>> from selenium import webdriver
>>> browser = webdriver.Firefox()
>>> browser.get('http://google.com')
>>> browser.find_element_by_tag_name('title')
<selenium.webdriver.remote.webelement.WebElement ...>
```

Selenium

- Open a browser

```
1  from selenium import webdriver
2  browser = webdriver.Chrome()
```

- Open a Page

```
browser = webdriver.Firefox()
browser.get("http://2015.codemotion.es/agenda.html#5677904553836544")
```

Selenium

- find_element_
  by_link_text(‘text’): find the link by text
  by_css_selector: just like with lxml css
  by_tag_name: ‘a’ for the first link or all links
  by_xpath: practice xpath regex
  by_class_name: CSS related, but this finds all different types that have the same class

Selenium

```
<div id=“myid">...</div>
browser.find_element_by_id(“myid")
```

```
<input type="text" name="example" />
browser.find_elements_by_xpath("//input")
```

```
<input type="text" name="example" />
browser.find_element_by_name(“example")
```

Selenium

```
<div id="myid">
<span class="myclass">content</span>
</div>
browser. find_element_by_css_selector("#myid span.myclass")
```

```
<a href="">content</a>
browser.find_element_by_link_text("content")
```

Selenium

element.click()

element.submit()

Selenium in codemotion agenda

```python
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.common.keys import Keys
import time

browser = webdriver.Firefox()
browser.get("http://2015.codemotion.es/agenda.html#5677904553836544")

for index in range(0,20):
    try:
        a = browser.find_element_by_xpath("//tr["+str(index)+"]/td/p/a").text
        print a
    except Exception:
        pass

browser.close()
```

```
Todo lo que siempre quisiste saber sobre bases de datos distribuídas de alta disponibilidad
Beauty Treatment for your Android Application
Dando amor a los tests
Machine Learning para todos con Azure ML y el proyecto Oxford
Stop making fool of yourself about documentation!
Postgres como base de datos NoSQL
Blues
```

Extraer datos de la agenda de codemotion

```
url1="http://2015.codemotion.es/agenda.html#5677904553836544"
url2="http://2015.codemotion.es/agenda.html#5699289732874240"

urls=[url1,url2]

i=0

talks_codemotion = []

for url in urls:
    print "url:  "+ url
    browser = webdriver.Firefox()
    browser.get(url)

    days= browser.find_elements_by_xpath("//section[2]/div/div/ul/li/a")
    print days[i].text

    for index in range(0,20):
        for columna in range(2,15):
            try:
                hour= browser.find_element_by_xpath("//tr["+str(index)+"]/td")
                talks= browser.find_elements_by_xpath("//tr["+str(index)+"]/td["+str(columna)+"]/p")
                talk_codemotion = {}
                index_aux = 0
                if len(talks)>0:
                    for talk in talks:
                        talk_codemotion['day'] = days[i].text
                        talk_codemotion['hour'] = hour.text
                        if index_aux ==0:
                            talk_codemotion['title'] = talk.text.encode('utf-8')
```

Extraer datos de la agenda de codemotion


```
{"speaker": "JAVIER RAMIREZ", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Todo lo que siempre quisiste saber sobre bases de datos distribu\u(
{"speaker": "CAMILO GALIANA", "day": "27 noviembre", "hour": "10:00-10:45", "title": "El arte de ser vago: Clean Code"}
{"speaker": "MANU DELGADO", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Arquitecturas distribuidas en la nube"}
{"speaker": "RUBEN PERTUSA LOPEZ, MIGUEL EGEA", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Pensando en Big Data: Un paseo por el Modern Dat
{"speaker": "ROBERTO GONZALEZ", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Azure Web Apps - Deep Dive"}
{"speaker": "ELENA TORRO", "day": "27 noviembre", "hour": "10:00-10:45", "title": "El dise\u00f1o de interfaces a trav\u00e9s de los tiempos"}
{"speaker": "MARIO EZQUERRO", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Hardware en el d\u00eda a d\u00eda"}
{"speaker": "ISABEL CABEZAS", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Me gusta que los est\u00e1ndares salgan bien."}
{"speaker": "LUIS CALVO D\u00cdAZ", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Transiciones y animaciones en CSS: que empiece el baile"}
{"speaker": "SILVANO GIL P\u00c9REZ", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Scrum Lego. \u00a1A divertirse \u00e1gilmente!"}
{"speaker": "JOSE MANUEL BEAS", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Taller expr\u00e9s de Planificaci\u00f3n \u00c1gil"}
{"speaker": "JUAN MANUEL SERRANO HIDALGO", "day": "27 noviembre", "hour": "10:00-10:45", "title": "Elimina la corrupci\u00f3n: programaci\u00f3n funcion
{"speaker": "TONI TEBAS, DAVID REGORDOSA AVELLANA", "day": "27 noviembre", "hour": "11:00-11:45", "title": "C\u00f3mo generar una arquitectura cloud aut
```

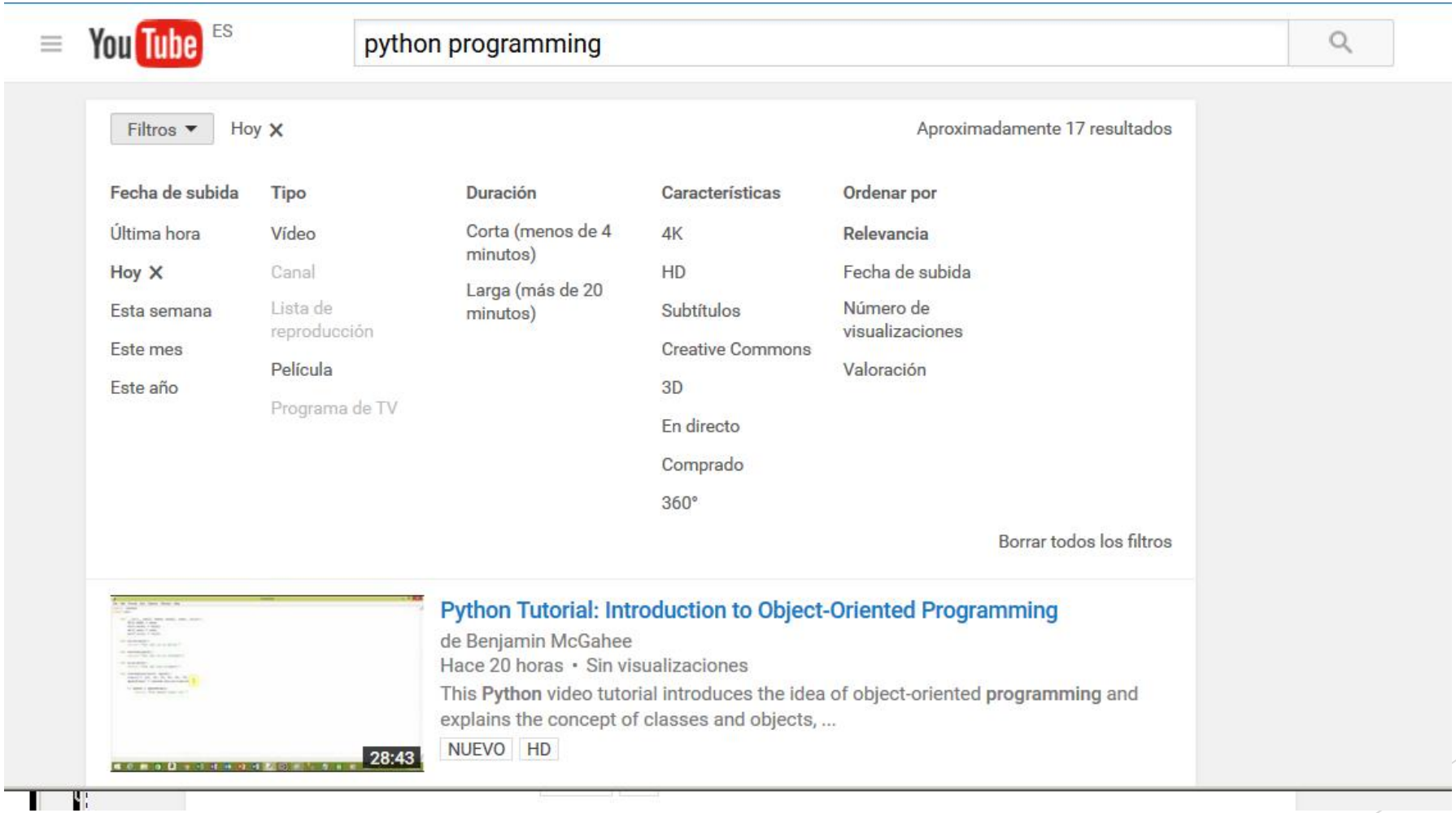


```
28 noviembre,Caminando de Java a Scala en menos de 2 horas,"SERGIO GÃ“MEZ, ABEL RINCÃ“N MATARRANZ",09:30-10:15
28 noviembre,Desarrollar un videojuego mÃ³vil multiplataforma con Cocos2D-X,JON SEGADOR,09:30-10:15
28 noviembre,Spock: testing (in the) Enterprise,FERNANDO REDONDO RAMÃREZ,09:30-10:15
28 noviembre,Cassandra para impacientes,CARLOS ALONSO PÃ‰REZ,10:30-11:15
28 noviembre,World-Class Testing Pipeline in Android,PEDRO VICENTE GÃ“MEZ SÃNCHEZ,10:30-11:15
28 noviembre,De Java a Python en un gestor de dependencias de C y C++: Aventuras y desventuras de una startup,DIEGO RODRIGUEZ-LOSADA,10:30-11:15
28 noviembre,Hackathons on Rails,CODEMOTION MADRID,10:30-11:15
28 noviembre,Coding Culture,SVEN PETERS,10:30-11:15
28 noviembre,Unit testing: el mito de los cero bugs,FERNANDO ESCOLAR,10:30-11:15
```

Selenium Cookies

```
# Go to the correct domain
driver.get("http://www.example.com")

# Now set the cookie. Here's one for the entire domain
# the cookie name here is 'key' and its value is 'value'
driver.add_cookie({'name':'key', 'value':'value', 'path':'/'})
# additional keys that can be passed in are:
# 'domain' -> String,
# 'secure' -> Boolean,
# 'expiry' -> Milliseconds since the Epoch it should expire.

# And now output all the available cookies for the current URL
for cookie in driver.get_cookies():
    print "%s -> %s" % (cookie['name'], cookie['value'])

# You can delete cookies in 2 ways
# By name
driver.delete_cookie("CookieName")
# Or all of them
driver.delete_all_cookies()
```

Selenium youtube



Selenium youtube

```
import random
import time
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.common.keys import Keys

browser = webdriver.Firefox()

browser.get("http://www.youtube.com")
search_bar=browser.find_element_by_id('masthead-search-term')
search_bar.send_keys("python programming")
search_bar.submit()

filter_button = browser.find_element_by_class_name("filter-button-container").find_element_by_tag_name("button")
filter_button.click()
time.sleep(1)
browser.find_element_by_link_text("Hoy").click()
time.sleep(1)

videos = browser.find_elements_by_class_name("yt-uix-tile-link")
videoIndex = random.randint(2,len(videos))
print videos[videoIndex]
videos[videoIndex].click()
```

Kimono

kimono speaker 71 71 + Done

{CODEMOTION} Tickets Agenda Blog Location Codemotion 2014 Sponsoring

Agenda

27 noviembre 28 noviembre

	Track 1	Track 2	Track 3	Track 4	Track 5	Track 6	Track 7	Track 8	Track A	Track B	Track C	Track D
08:00-09:00	REGISTRO											
09:00-09:45	KEYNOTE											
09:45-10:00												
10:00-10:45	Todo lo que siempre quisiste saber sobre bases de datos distribuídas de alta disponibilidad	El arte de ser vago: Clean Code	Hardware en el día a día	Pensando en Big Data: Un paseo por el Modern Data Warehouse	Azure Web Apps - Deep Dive	El diseño de interfaces a través de los tiempos	Arquitecturas distribuidas en la nube	Me gusta que los estándares salgan bien.	Scrum Lego. ¡A divertirse ágilmente!	Transiciones y animaciones en CSS: que empiece el baile	Taller exprés de Planificación Ágil	Elimina la corrupción: programación funcional pura con Scala
	JAVIER RAMIREZ	CAMILO GALIANA	MARIO EZQUERRO	RUBEN PERTUSA LOPEZ, MIGUEL EGEA	ROBERTO GONZALEZ	ELENA TORRO	MANU DELGADO	ISABEL CABEZAS				
10:45-11:00												
11:00-11:45	Beauty Treatment	Seguridad en	Swift 2 para	Cómo Diseñar	Spock: O por qué	npty slot	Web Components +	Deuda Técnica para				

Kimono

1 Collection

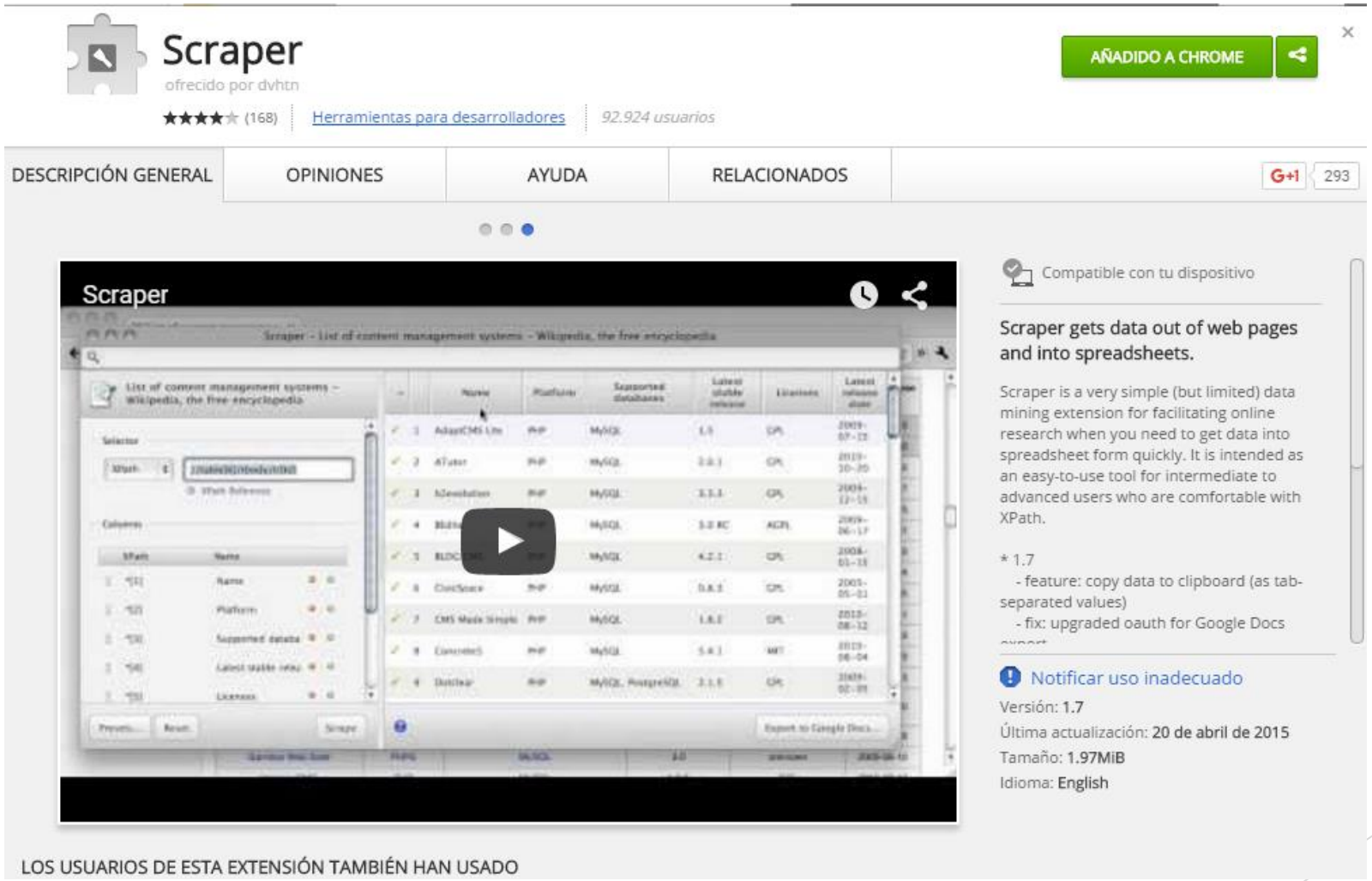
JSON CSV RSS

Download JSON

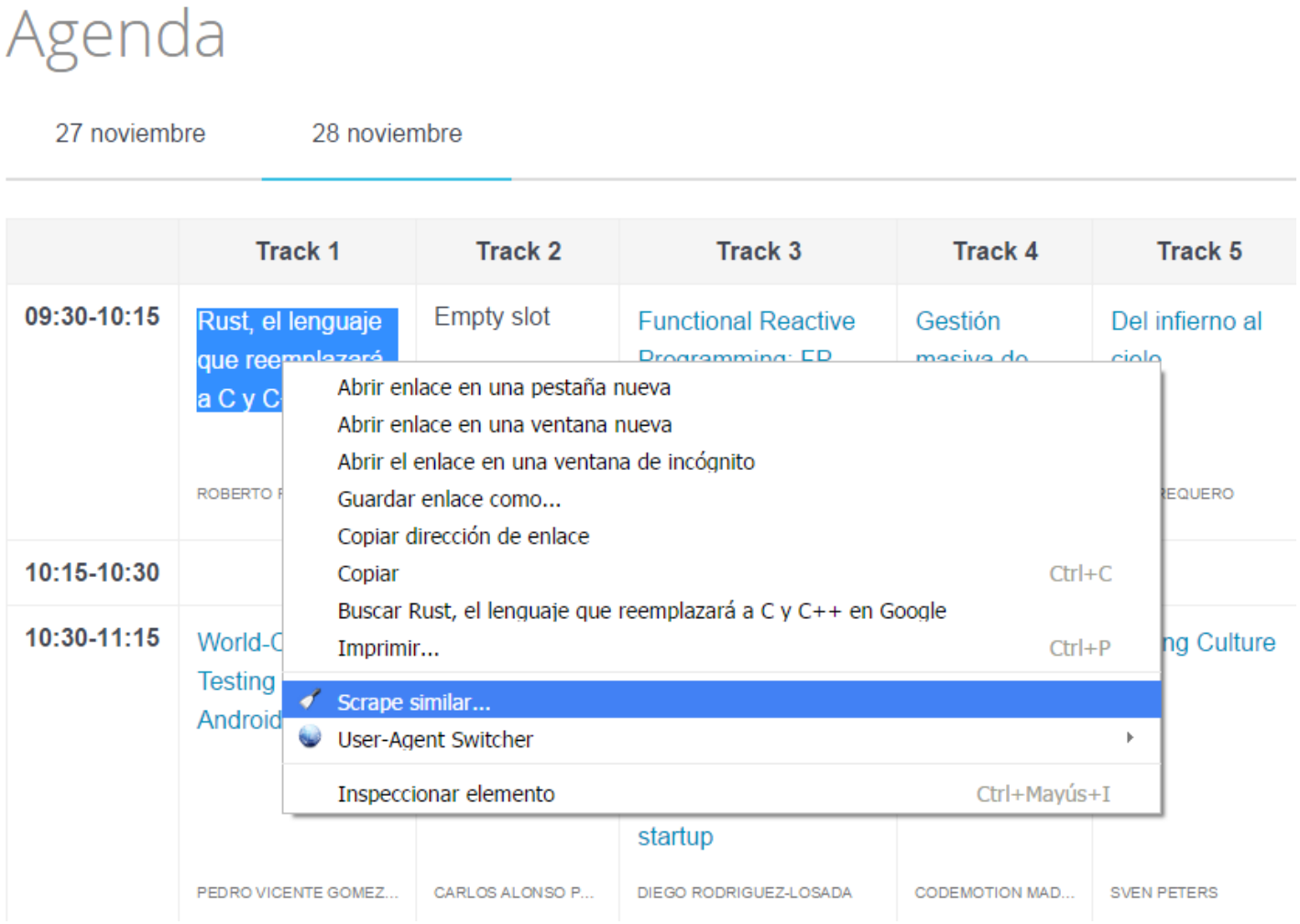
Select All Text


```
{
  "collection1": [
    {
      "talk": {
        "text": "Todo lo que siempre quisiste saber sobre bases de datos distribuídas de alta disponibilidad",
        "href": "http://2015.codemotion.es/agenda.html#/#5677904553836544/48514001"
      },
      "speaker": "javier ramirez"
    },
    {
      "talk": {
        "text": "El arte de ser vago: Clean Code",
        "href": "http://2015.codemotion.es/agenda.html#/#5677904553836544/43004005"
      },
      "speaker": "Camilo Galiana"
    },
    {
      "talk": {
        "text": "Hardware en el día a día",
        "href": "http://2015.codemotion.es/agenda.html#/#5677904553836544/43014002"
      },
      "speaker": "Mario Ezquerro"
    },
    {
      "talk": {
        "text": "Pensando en Big Data: Un paseo por el Modern Data Warehouse",
        "href": "http://2015.codemotion.es/agenda.html#/#5677904553836544/44774005"
      },
      "speaker": "Ruben Pertusa Lopez, Miguel Egea"
```

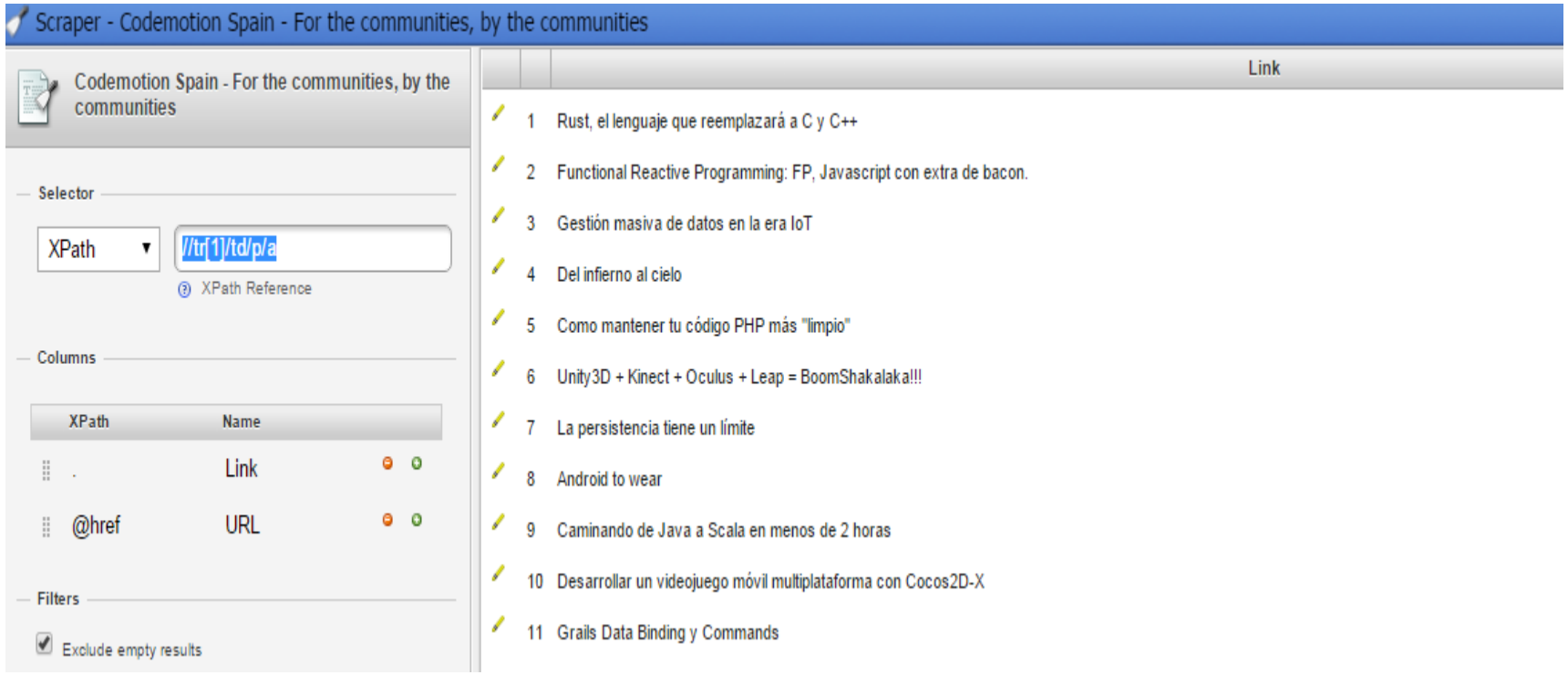
Scraper Chrome plugin



Scraper Chrome plugin



Scraper Chrome plugin



Parse Hub

Home > Projects > * 2015.codemotion.es Project

Design | Settings | Get Data

Project title: 2015.codemotion.es Project

Starting Site: http://2015.codemotion.es/agenda.htm

Starting Template: codemotion

Project Token tlzPitjrYzTMSQlik-ZSdSMe

☑ Load Javascript and images

☐ Rotate IP addresses (Upgrade to Activate)

Max workers: 1

Design | Settings | Get Data

codemotion Options

Select **page**

Quick Select

Group **codemotion_27**

Property **talk**

Property **speaker**

Extract **codemotion_27**

codemotion_28 Options

Extract JSON object $jsonify($JSON.pars

☑ **Use regex** regex

☑ **Extract all occurrences**

Parse Hub

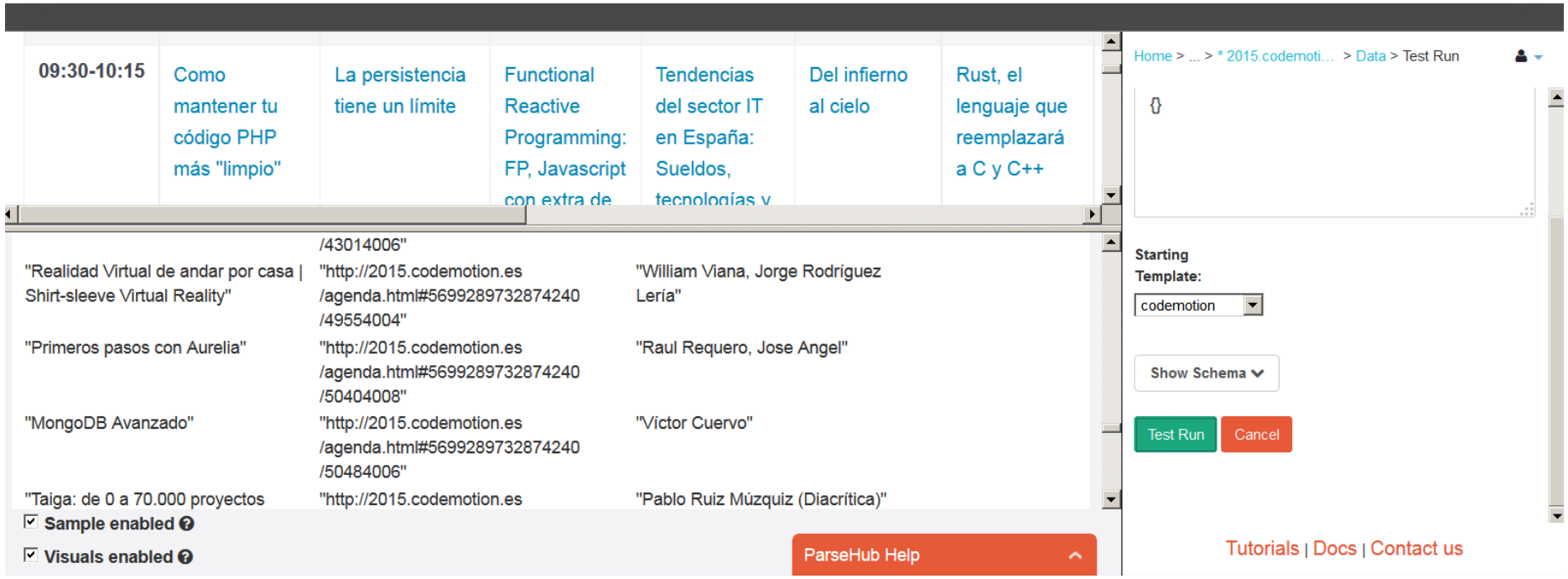
27 noviembre 28 noviembre

	Track 1	Track 2	Track 3	Track 4	Track 5	Track 6	Track 7	Trac
09:30-10:15	Rust, el lenguaje que reemplazará a C y C++ ROBERTO PEREZ	Tendencias del sector IT en España: Sueldos, tecnologías y empleo CODEMOTION MAD...	Functional Reactive Programming: FP, Javascript con extra de bacon. JAVIER ONIELFA	Gestión masiva de datos en la era IoT LUIS GUERRERO	Del infierno al cielo RAUL REQUERO	Como mantener tu código PHP más "limpio" OSCAR VITORES	Unity3D + Kinect + Oculus + Leap = BoomShakalaka!!! TONI RECIO SACRISTA	La persiste tiene un lím EMMA SESMEF
10:15-10:30								


```
  "talk_url":"http://2015.codemotion.es/agenda.html#5699289732874240/49544011"
  "speaker":"Raul Requero"
 },
 {
  "talk":"Como mantener tu código PHP más \"limpio\""
  "talk_url":"http://2015.codemotion.es/agenda.html#5699289732874240/46624004"
  "speaker":"Oscar Vítores"
 },
 {
  "talk":"Unity3D + Kinect + Oculus + Leap = BoomShakalaka!!!"
  "talk_url":"http://2015.codemotion.es/agenda.html#5699289732874240/48524006"
  "speaker":"Toni Recio Sacristà"
 },
 {
```

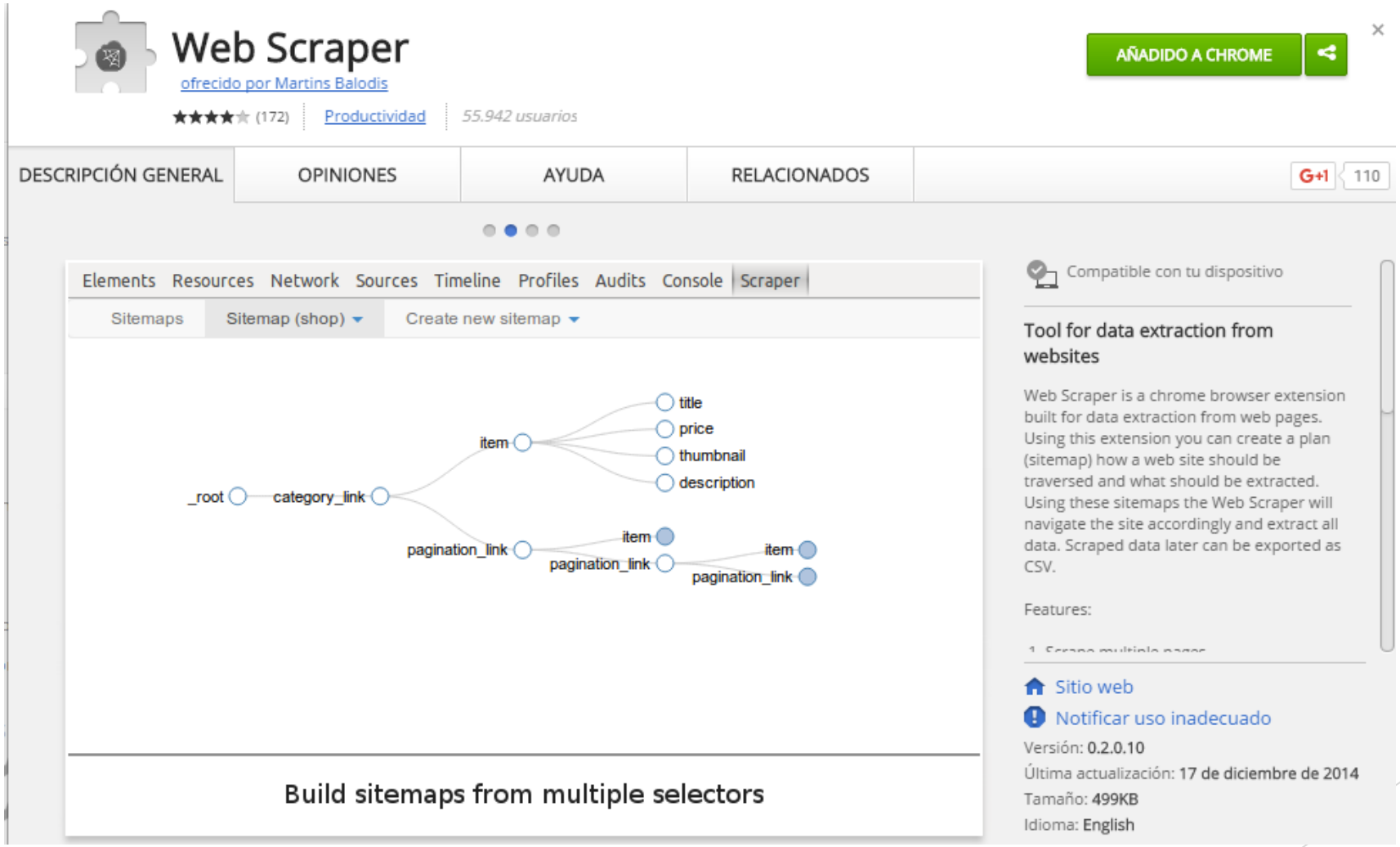

Sample enabled

Parse Hub


09:30-10:15
Como mantener tu código PHP más "limpio"
La persistencia tiene un límite
Functional Reactive Programming: FP, Javascript con extra de
Tendencias del sector IT en España: Sueldos, tecnologías y
Del infierno al cielo
Rust, el lenguaje que reemplazará a C y C++
/43014006"
"Realidad Virtual de andar por casa | Shirt-sleeve Virtual Reality"
"http://2015.codemotion.es /agenda.html#5699289732874240 /49554004"
"William Viana, Jorge Rodríguez Lería"
"Primeros pasos con Aurelia"
"http://2015.codemotion.es /agenda.html#5699289732874240 /50404008"
"Raul Requero, Jose Angel"
"MongoDB Avanzado"
"http://2015.codemotion.es /agenda.html#5699289732874240 /50484006"
"Víctor Cuervo"
"Taiga: de 0 a 70.000 proyectos
"http://2015.codemotion.es
"Pablo Ruiz Múzquiz (Diacrítica)"
Sample enabled
Visuals enabled
ParseHub Help
Home > ... > * 2015.codemoti... > Data > Test Run
{}
Starting Template:
codemotion
Show Schema
Test Run
Cancel
Tutorials | Docs | Contact us

Web Scraper plugin

http://webscraper.io



Web Scraper plugin

Sitemaps | Sitemap (codemotion) | Create new sitemap

root

	Selector	type	Multiple	Parent selectors	Actions
s1	tr.ka-table-tr:nth-of-type(4) a.ka-talk-title	SelectorText	true	_root	Element preview Data preview Edit Delete
s2	tr.ka-table-tr:nth-of-type(6) a.ka-talk-title	SelectorText	true	_root	Element preview Data preview Edit Delete
s3	tr.ka-table-tr:nth-of-type(9) a.ka-talk-title	SelectorText	true	_root	Element preview Data preview Edit Delete
s4	tr.ka-table-tr:nth-of-type(11) a.ka-talk-title	SelectorText	true	_root	Element preview Data preview Edit Delete
s5	tr.ka-table-tr:nth-of-type(13) a.ka-talk-title	SelectorText	true	_root	Element preview Data preview Edit Delete
s6	tr.ka-table-tr:nth-of-type(15) a.ka-talk-title	SelectorText	true	_root	Element preview Data preview Edit Delete

10:00-10:45	Todo lo que siempre quisiste saber sobre bases de datos distribuídas de alta disponibilidad	El arte de ser vago: Clean Code	Hardware en el día a día	Pensando en Big Data: Un paseo por el Modern Data Warehouse	Azure Web Apps - Deep Dive	El diseño de interfaces a través de los tiempos	Arquitecturas distribuidas en la nube	Me gusta que los estándares salgan bien.	Scrum Lego, ¡A divertirse ágilmente!	Transiciones y animaciones en CSS: que empiece el baile	Taller exprés de Planificación Ágil	Elimina la corrupción: programación funcional pura con Scala
	JAVIER RAMIREZ	CAMILO GALIANA	MARIO EZQUERRO	RUBEN PERTUSA LO...	ROBERTO GONZALEZ	ELENA TORRO	MANU DELGADO	ISABEL CABEZAS				

Elements | Network | Sources | Timeline | Profiles | Resources | Audits | Console | Tamper | Web Scraper

Sitemaps | Sitemap (codemotion) | Create new sitemap

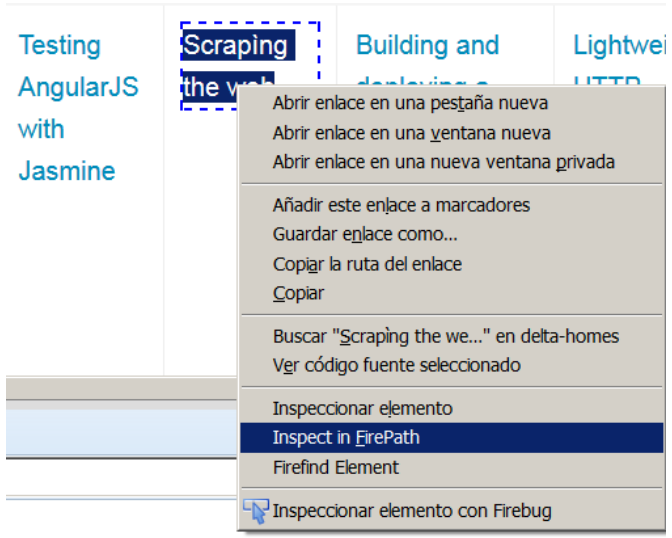
data Preview

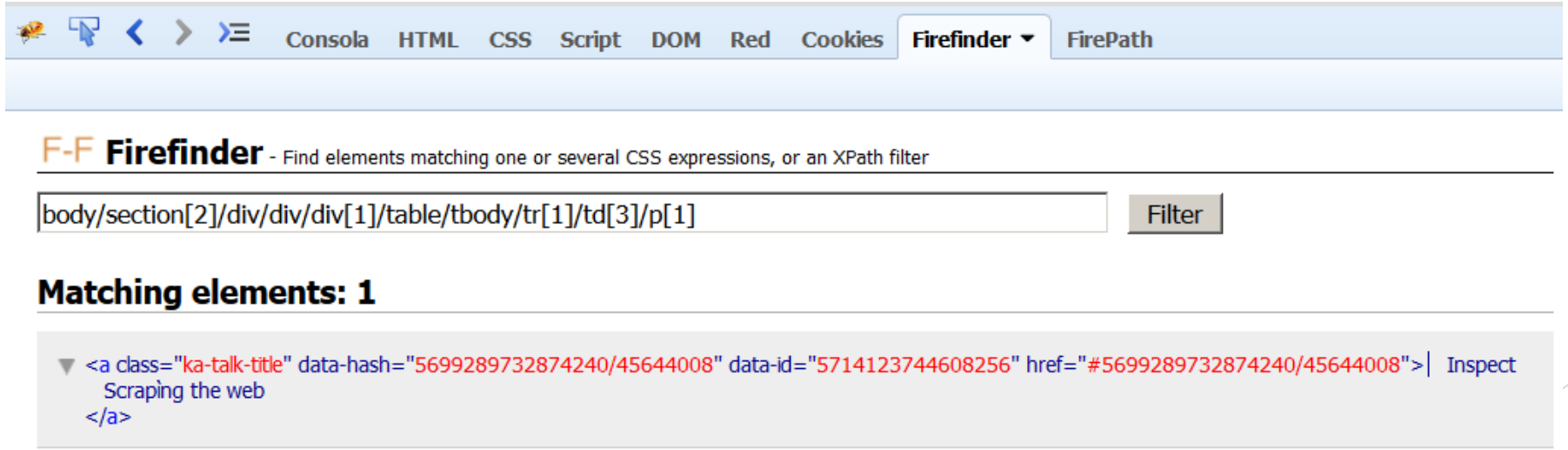
talks1
Todo lo que siempre quisiste saber sobre bases de datos distribuídas de alta disponibilidad
El arte de ser vago: Clean Code
Hardware en el día a día
Pensando en Big Data: Un paseo por el Modern Data Warehouse
Azure Web Apps - Deep Dive

Console | Emulation | Rendering

XPath expressions

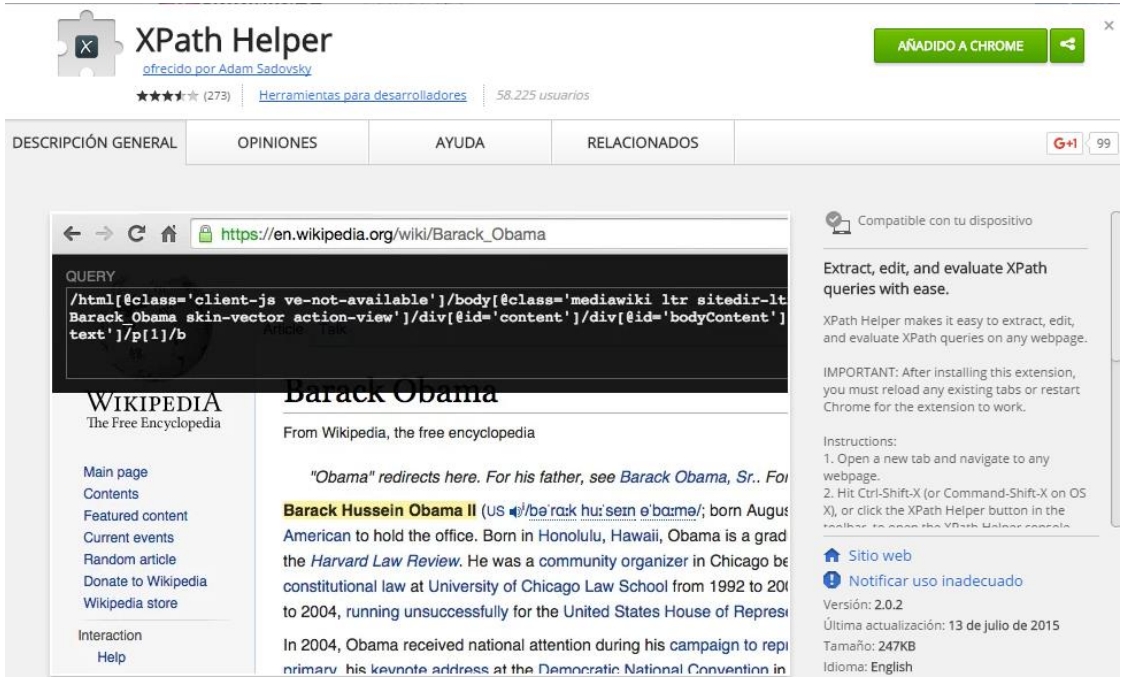
- Plugins para firefox
- FireFinder for FireBug
- FirePath





XPath expressions

- Xpath Helper
- **Mover el mouse + tecla shift**
- **Obtener la expresión xpath de un determinado elemento html**



XPath expressions

QUERY

```
/html/body/section[2]/div[@class='agenda-container']/div[@class='ka']/div[@class='kworkspace']/table[@class='ka-table']/tbody/tr[@class='ka-table-tr'][10]/td[@class='ka-table-td talk'][9]
```

RESULTS (1)

```
Scraping the web
Jose Manuel Ortega
```

14:00-15:30	PAUSA COMIDA											
15:30-16:15	Tu DevOp me da trabajo: Soy auditor de seguridad DANIEL GARCIA (A.K.A. ...	Descubriendo 40 Geo-APIs en 40 minutos RAUL JIMENEZ ORT...	Adding Realtime to your App NACHO MARTIN	Containers - the little Linux feature revolutionizing apps of all sizes RICK OSOWSKI	Proyecto TRRP: Drones con routers, Javascript y C DAVID MELENDEZ	Drilling into Data with Apache Drill TUGDUAL GRALL	Empty slot	Be water with Spark SERGIO GOMEZ	Testing AngularJS with Jasmine SAM LEACH	Scraping the web JOSE MANUEL ORTE...	Building and deploying a distributed application with Docker, Mesos and Marathon JULIA MATEO	Lightweight HTTP applications in the JVM with Groovy and Ratpack ALVARO SANCH...
16:15-16:30												
16:30-17:15	Resolviendo la noche electoral con AWS + Node.js + Angular.js + D3.js + Leafleat.js JAVIER ABADIA	Primeros pasos con Aurelia RAUL REQUERO, JO...	¡EMT Madrid y Apple Watch! Aplicaciones Dependientes de Contexto Siguiente Nivel JORGE RUIZ, ROBERTO ESTE...	Testing Your Spring Boot Application With Selenium MATHILDE RIGABE...	Windows 10: One package for all JOSUE YERAY JULIA...	CoreOS: Tu infrastructura escalable y reproducible LUIS MARTINEZ DE BARTOL...	Realidad Virtual de andar por casa | Shirt-sleeve Virtual Reality WILLIAM VIANA, JORGE ...	Empty slot				

Scraping Hub

- Scrapy Cloud es una plataforma para la implementación, ejecución y seguimiento de las arañas Scrapy y un visualizador de los datos scrapeados.

- Permite controlar las arañas mediante tareas programadas, revisar que procesos están corriendo y obtener los datos scrapeados.

- Los proyectos se pueden gestionan desde la API o a través de su Panel Web.

Scrapy Cloud

http://doc.scrapinghub.com/scrapy-cloud.html

https://dash.scrapinghub.com

```
>>pip install shub
>>shub login
    >>Insert your ScrapingHub API Key:
```

Scrapy Cloud /scrapy.cfg

```
# Project: demo
[deploy]
url =https://dash.scrapinghub.com/api/scrapyd/
#API_KEY
username = ec6334d7375845fdb876c1d10b2b1622
password =
project = 25767
```

Scrapy Cloud

Deploying a Scrapy Spider

NOTE:

You will need the Scrapinghub command line client to deploy projects to Scrapy Cloud, so install it if you haven't done it yet.

The next step is to edit `scrapy.cfg` file of your project and configure Scrapinghub as deployment target:

```
[settings]
default = companies.settings

[deploy]
project = PROJECT_ID
```

`PROJECT_ID` is the numeric project ID which you can find in Scrapinghub URL:

> https://dash.scrapinghub.com/p/PROJECT_ID/...

Then you should put your API key (which you can get from your Account page) in `~/.scrapy.cfg` to authenticate:

```
[deploy]
username = APIKEY
```

Finally, you deploy your spider to Scrapinghub with the following command:

```
$ shub deploy
Server response (200):
{"status": "ok", "project": PROJECT_ID, "version": "1391115259", "spiders": 1}
```

```
Packing version 1446554678
Deploying to Scrapy Cloud project "25767"
{"status": "ok", "project": 25767, "version": "1446554678", "spiders": 1}
Run your spiders at: https://dash.scrapinghub.com/p/25767/
```

Scrapy Cloud

Scrapy Cloud / demo / Spiders / postUGR_spyder

Details | Settings | Settings (raw view)

Details

Name:	postUGR_spyder
Type:	manual
Version:	1446556485
Tags:	No tags Edit
Total Jobs:	1
Custom settings:	

Pending (0) | Running (0) | Completed (1)

Pending Jobs (0)

Job	Spider	Items	Requests	Errors	Log	Wait Time	Added

Running Jobs (0)

Show only jobs with comments (0)

Job	Spider	Items	Requests	Errors	Log	Runtime	Started	Last Activity

Completed Jobs (1)

Show only jobs with comments (0)

Job	Spider	Items	Requests	Errors	Log	Runtime	Finished	Outcome
1/1	postUGR_spyder 1446554678	8	9	0	22	0:00:33	2015-11-03 12:48:17 UTC	finished

Remove | Restart

Schedule Spider

Current version: 1446556485

Spiders

postUGR_spyder

Priority

Normal

Arguments

Schedule

Scrapy Cloud

Filter by Field: Choose field... Choose action... All Items Update

Item 1 2015-11-03 12:48:15 UTC Download Comment

autor	José Antonio Serrano García
categorias	Formación
contenido	Como me gusta mas que , tiene una carencia como gestor de correo y es, no tener un calendario. Pero bueno como todo tiene solución, aquí tambien la tenemos. Lo primero si no tenemos instalado Thunderbird, lo instalamos: sudo apt-get install thunderbird También podemos usar el Synaptic, escribimos thunderbird y lo instalamos. Una vez que esta instalado, pues nos instalamos los siguientes plugings, y , tenemos que instalar ambos. Si estamos usando una versión de 64 bits hay un problema, y es que el plugings Lighting no funciona, pues dice que no funciona en esta arquitectura, no hay problema, nos va mos de nuevo a Synactic y buscamos xul-ext-lighting (lo he probado en Ubuntu, imagino que para las demas distribuciones funcionara igualmente) Una vez instaladas estos plugings ya solo nos queda conectar Calendar a Thunderbird, vamos a explicar como conectarlo. Lo primero abrimos , nos vamos a , y hay seleccionamos , y escogemos el calendario que queramos añadir, en este caso es el de la , y entramos en el. En el apartado de Dirección del calendario: donde esta el icono de XML, con el botón derecho le decimos que copiar la dirección de enlace: Con esta dirección, nos vamos a nuestro thurderbird, y en la pestaña que nos ha creado de Calendar le damos al botón derecho y le decimos Nuevo. Le damos a Siguiente y añadimos el enlace obtenido de Google Calendar Nos pregunta la contraseña para añadirlo. Le ponemos un nombre del Calendario, en este caso OSL y podemos seleccionar un color para diferenciarlo de los demás. Ya podemos nuestro calendario creado y podemos añadir eventos, con aviso a nuestro móvil, por correo o notificación en pantalla. Por supuesto podemos añadir todos los calendarios que queramos.
etiquetas	Formación howtos Calendario correo thunderbird tutorial
titulo	Thunderbird y como tener un calendario

Scraped Fields Hide

_type	8	100%
autor	8	100%
categorias	8	100%
contenido	8	100%
etiquetas	8	100%
titulo	8	100%

Scrapy Cloud

Scrapy Cloud / demo / Spiders / postUGR_spyder / Job 1

Job | Items (8) | Requests (9) | Log (22) | Stats | Reports

Filter by Field: Choose field... Choose action... Update

Request 0 2015-11-03 12:48:00 UTC

URL	http://osl.ugr.es
Duration	1504 ms
Fingerprint	819ee240f5152ef64e20bec5daa0d1f5a985a89c
HTTP Method	GET
Response Size	42993 bytes
HTTP Status	200
Last seen	2015-11-03 13:48:00 UTC

Request 1 2015-11-03 12:48:05 UTC

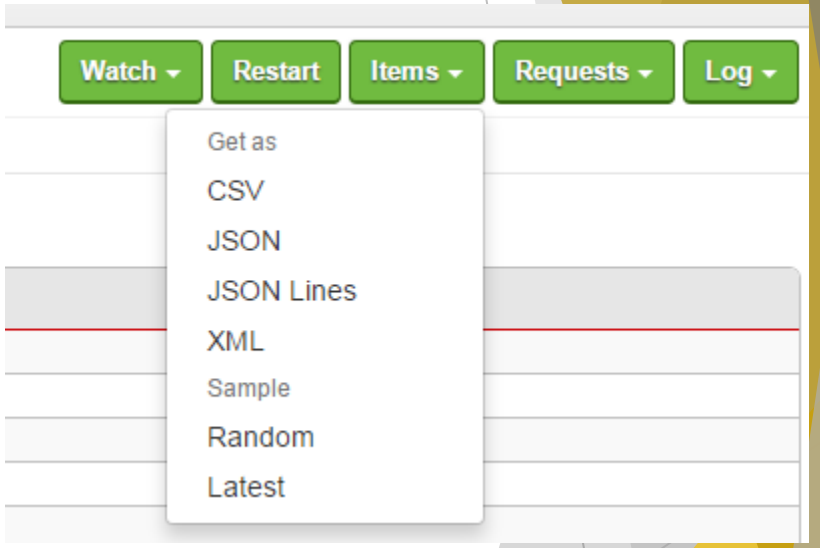
URL	http://osl.ugr.es/2015/10/29/xvi-campana-donacion-de-material-informatico-finali...
Duration	720 ms
Fingerprint	ad4058aa0f70902a69ca4256c91f1f4bea4f14e4
HTTP Method	GET
Parent Request	25767/1/1/0
Response Size	50065 bytes
HTTP Status	200
Last seen	2015-11-03 13:48:05 UTC

Request 2 2015-11-03 12:48:08 UTC

URL	http://osl.ugr.es/2011/03/16/thunderbird-y-como-tener-un-calendario/comment-page...
Duration	764 ms
Fingerprint	285fcc2481ed5c3c28a08751528dc0b071d54b9a
HTTP Method	GET
Parent Request	25767/1/1/0
Response Size	56774 bytes
HTTP Status	200
Last seen	2015-11-03 13:48:08 UTC

Request 3 2015-11-03 12:48:10 UTC

URL	http://osl.ugr.es/2015/06/09/oferta-formativa-primer-cuatrimestre-curso-2015-201...
Duration	756 ms
Fingerprint	94602b9f0bde0c1a6ccaa8b5785c20abee173231
HTTP Method	GET
Parent Request	25767/1/1/0



Scrapy Cloud Scheduling

```
curl -u APIKEY:
https://dash.scrapinghub.com/api/schedule.json -d
project=PROJECT -d spider=SPIDER
```

Running Jobs (1)

Show only jobs with comments (0)

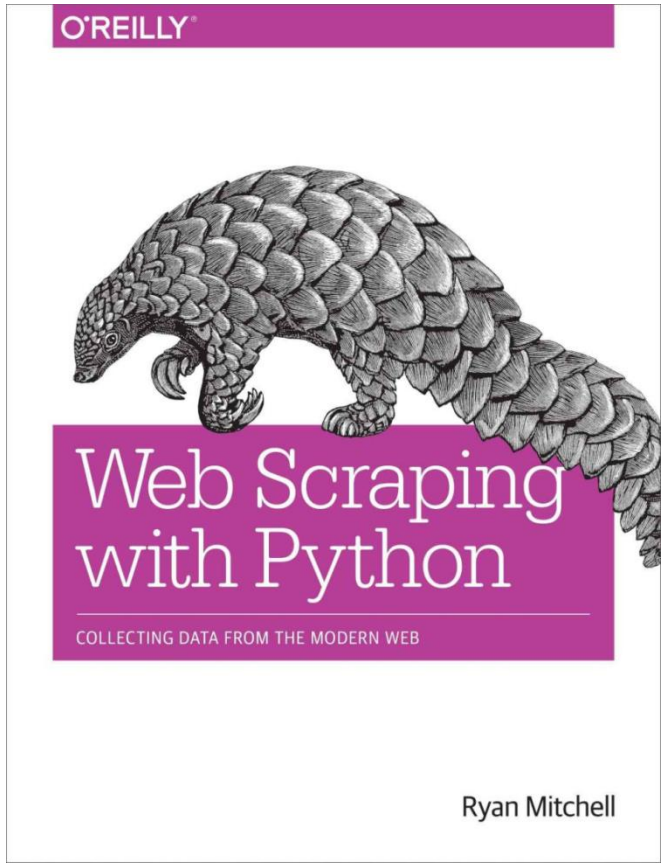
Job	Spider	Items	Requests	Errors	Log	Runtime	Started	Last Activity
1/2	postUGR_spyder 1446556485	0	0	0	0	0:00:22	2015-11-03 13:36:01 UTC	a few seconds ago

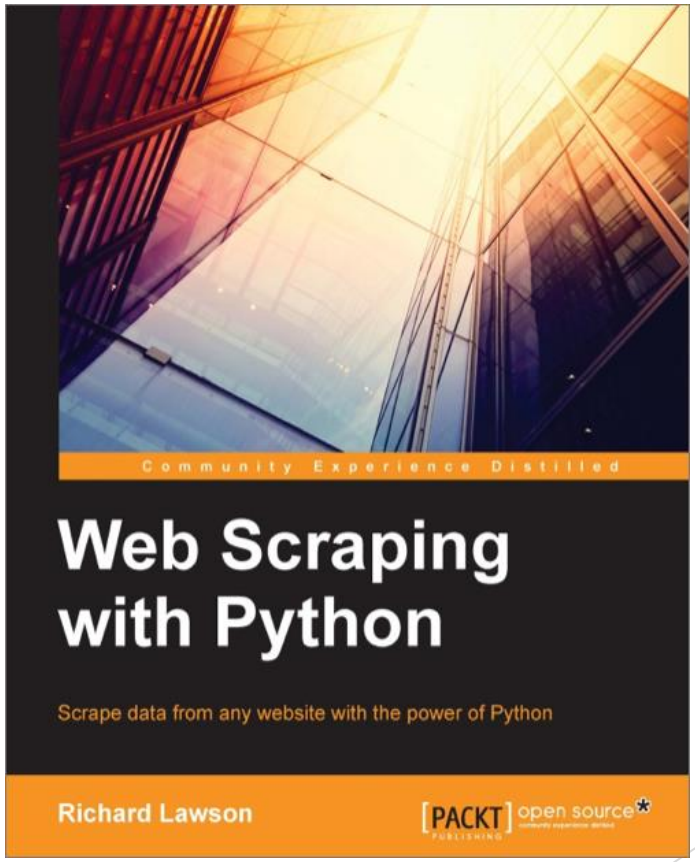
Stop

Referencias

- http://www.crummy.com/software/BeautifulSoup
- http://scrapy.org
- https://pypi.python.org/pypi/mechanize
- http://docs.python-requests.org/en/latest
- http://selenium-python.readthedocs.org/index.html
- https://github.com/REMitchell/python-scraping

Books


O'REILLY
Web Scraping with Python
COLLECTING DATA FROM THE MODERN WEB
Ryan Mitchell


Community Experience Distilled
Web Scraping with Python
Scrape data from any website with the power of Python
Richard Lawson
[PACKT] open source