

Extrayendo de la web y ficheros remotos

Sistemas Distribuidos

Grado en Ingeniería Informática

Índice

- 1 Extrayendo de la web: scraping
- 2 Acceso al Dropbox
- 3 Acceso al Google Drive

Índice

- 1 Extrayendo de la web: scraping
- 2 Acceso al Dropbox
- 3 Acceso al Google Drive

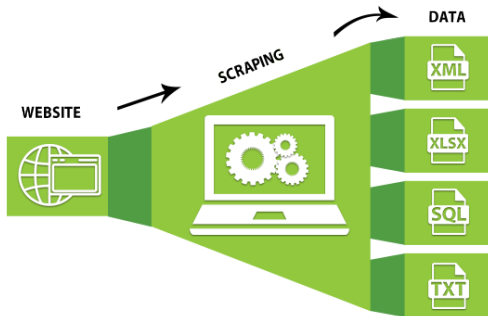
Extrayendo información de la web



La web posee gran cantidad de información

- Tiempo.
- Wikipedia.
- Portales temáticos.
- Datos oficiales: Empleo, ...
- Películas, Hoteles, ...

Extracción



Pasos

- 1 Descargar la página con los datos.
- 2 Analizar la web para identificar los datos interesantes.
- 3 Guardar los datos en otros formatos más útiles.

Análisis de Datos

Análisis de Datos

- HTML (y CSS) guarda información de presentación.
- Identificar qué parte del HTML es la información útil.
- ¿Cómo identificamos lo que nos interesa?

Caso ideal: Etiquetas CSS

- Caso ideal: componente con clase/id CSS.

Estructura DOM

- Usando la estructura de árbol del HTML.

Ejemplo: Etiqueta CSS

Ejemplo

```
<div class="quote">
  <span class="text">"The world as we have created it is
  a process of our thinking. It cannot be changed
  without changing our thinking."</span>

  <div class="author_info">
    by <small class="author">Albert Einstein</small>
    <a href="/author/Albert-Einstein">(about)</a>
  </div>
</div>
```

Ejemplos de acceso

Texto del author ⇒ **span.text**.

Nombre autor ⇒ **small.author**.

Enlace ⇒ **div.author_info a**.

Extrayendo usando CSS

Del CSS

- El CSS identifica componentes para cambiar su aspecto.
- Permite marcar de dos maneras un componente HTML.

Clase Se puede repetir dentro del HTML.

ID No se repite, un único componente.

Nodo padre Indicando su nodo padre.

Notación

`<div class="note">` \Rightarrow **div.note**.

`<h2 id="header">` \Rightarrow **h2#header**.

`<div class="note">...` \Rightarrow **div.note li**.

Extrayendo usando XPath

XPath

- Es una notación avanzada para identificar nodos.
- Permite establecer condiciones.

Notación

node[@atributo=valor] Permite poner condiciones sobre atributo

./div párrafo *div* del contexto actual.

//div *div* que cuelgan del nodo raíz.

Ejemplo

`<h2 id="header">` \Rightarrow `h2[@id="header"]`.

`div div[@class="tag"][1]` \Rightarrow Segundo div de clase 'tag'.

Accediendo a datos o Atributos

Accediendo a datos

- El formato anterior permite recuperar el nodo.
- Luego hay que indicar si se desea el texto o atributos del nodo.

Formato css

Texto \Rightarrow `::text`.

Atributo \Rightarrow `::attr(atributo)`.

Ejemplos

`<div class="node">Textico</div>` \Rightarrow `div.node::text`.

`Enlace` \Rightarrow `a::attr(href)`.

Accediendo a datos o Atributos

Accediendo a datos

- El formato anterior permite recuperar el nodo.
- Luego hay que indicar si se desea el texto o atributos del nodo.

Formato xpath

Texto \Rightarrow `/text()`.

Atributo \Rightarrow `/@atributo`.

Ejemplos

`<div class="node">Textico</div>` \Rightarrow

`div[@class="node"]/text()`.

`Enlace` \Rightarrow `a/@href`.

Obteniendo los valores de una web

No es necesario hacerlo a mano

- Hay herramientas visuales que permiten obtenerlos visualmente.

Ejemplos

- Web Developer de Firefox.
 - Inspector.
 - Portia.

Ejemplo: Captura de componente





Scrapy

- Software en Python para *scraping*.
- Automáticamente descarga y extrae usando CSS o XPath.

Ventajas

- Fácil de usar.
- Muy bien documentado.
- Servicio en la nube
(<https://scrapinghub.com/scrapy-cloud/>).

Instalación de Scrappy

Instalación

```
pip install scrapy  
# Y esperar  
# bajar dependencias faltantes
```

Programas requeridos

- Software **lxml**.
- Librería **openssl**.

Alternativa

- Usar el *virtualenv* incluido en el campus virtual.
- Aún es necesario tener el software instalado.

Conceptos de Scrapy

Spider

- Un spider es una clase que recupera información de una URL concreta.

Proyecto

- Estructura de directorios: *spiders*, configuración, ...
- El comando **scrapy** muestra más opciones si se ejecuta dentro de un proyecto.

Crawl

- scrapy se encarga de bajar las webs asincrónicamente y procesar *los spiders*.

Ejemplo de scrapy

Pasos

- 1 Crear el proyecto.
- 2 Añadir un nuevo *spider*.
- 3 Editar el *spider* para recuperar los datos.
- 4 Ejecutar scrapy para aplicar el *spider* y guardar los resultados.

Formatos de salida

- Formato CSV.
- Formato JSON.
- Formato **JSON Lines** (JSON bien tabulado y sin repetir datos).

Creando la infraestructura

Creando el proyecto

```
$ scrapy startproject milanuncios
```

Creando spiders de bicicletas

```
$ scrapy genspider bicicletas \
    http://milanuncios.com/bicicletas-en-cadiz/
```

Example (Editar el spider)

```
$ vim|nano|emacs spiders/bicicletas.py
```

Ejecutar el spider

```
$ scrapy crawl bicicletas -o bicicletas.csv
$ scrapy crawl bicicletas -o bicicletas.json [-t json | -t jl]
```

Estructura de un spider

Esqueleto

```
import scrapy

class BicicletasSpider(scrapy.Spider):
    name = "bicicletas"
    start_urls = [http://www.milanuncios.com/bicicletas-en-cadiz/]

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

Conceptos

- `start_urls` define la URL de la que traer información.
- `parse` Método que se ejecuta una vez cargado.
- `response` Clase para extraer información.

Interfaz de response

Métodos principales

- `css` Permite recuperar nodo usando formato **CSS**.
- `xpath` Permite recuperar nodo usando formato **XPath**.

¿Nodo o texto?

- Depende de la sintaxis

Extraer

- `extract()` Permite devolver una lista.
- `extract_first()` Permite devolver el primer elemento, excepción si no hay.

Ejemplo: Recuperando bicicletas

Bicicletas

```
def parse(self, response):
    anuncios = response.css("div.aditem-detail")

    for anuncio in anuncios:
        title = anuncio.css("a.aditem-detail-title::text").extract_first()
        text = anuncio.css("div.tx::text").extract_first()
        price = anuncio.css("div.aditem-price::text").extract_first()
        yield {"title": title, "text": text, "price": price}
```

Comentarios

yield Es como *return* pero para métodos asíncronos.

Formato hash Se devuelve como hash, no en el formato de salida.

Verificar Hay que verificar que los CSS y XPath sean correctos para que funcione bien.

Herramienta para verificar

Se puede probar de informa interactiva

```
$ scrapy shell http://...
```

Salida

```
(pytools) daniel@Quixote:~/working/as/disttools$ scrapy shell https://www.milanuncios.com/anuncios-en-cadiz/bicicletas.html
2017-05-09 09:14:15 [scrapy.utils.log] INFO: Scrapy 1.3.3 started (bot: scrapybot)
...
2017-05-09 09:14:15 [scrapy.core.engine] INFO: Spider opened
2017-05-09 09:14:15 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.milanuncios.com/anuncios-en-cadiz/bicicletas.html>
2017-05-09 09:14:16 [traitlets] DEBUG: Using default logger
[s] Available Scrapy objects:
[s]   scrapy       scrapy module (contains scrapy.Request, scrapy.Selector, etc)
[s]   crawler      <scrapy.crawler.Crawler object at 0x7fada796bb70>
[s]   item          {}
[s]   request       <GET https://www.milanuncios.com/anuncios-en-cadiz/bicicletas.html>
[s]   response      <200 https://www.milanuncios.com/anuncios-en-cadiz/bicicletas.html>
[s]   settings      <scrapy.settings.Settings object at 0x7fada796be80>
[s]   spider        <DefaultSpider default at 0x7fada7713518>
[s] Useful shortcuts:
[s]   fetch(url[, redirect=True]) Fetch URL and update local objects (by default)
[s]   fetch(req)                  Fetch a scrapy.Request and update local objects
[s]   shell()                     Shell help (print this help)
[s]   view(response)              View response in a browser
```

Distintas páginas

Límite por paginación

- Múltiples páginas muestran sólo un número por página.
- ¿Cómo podemos evitarlo?

Distintas páginas

Límite por paginación

- Múltiples páginas muestran sólo un número por página.
- ¿Cómo podemos evitarlo?

Múltiples cargas

- Identificar el enlace/botón de más páginas.
- Cargar el enlace al que apunta.

Límite por Paginación

Ejemplo en milanuncios

```
next = response.xpath(//a[text()='Siguiente'])

if next is not None:
    next_page = next.css(:attr(href)).extract_first()
    next_page = response.urljoin(next_page)
    yield scrapy.Request(next_page, callback=self.parse)
```

Comentarios

- El enlace de siguiente siempre puede no existir.
- Usar **urljoin** para concatenar enlaces relativos.
- Uso de Request para responder la página, indicando la página actual.
- Recordar usar yield, es petición asíncrona.

Parámetros

Limitando la salida

- Si quisiese limitar la respuesta.
- ¿Puedo tener un contador?
- ¿Cómo puedo pasar parámetros?

Pasando parámetros

- Por medio del objeto `response.media` se puede guardar información entre peticiones.

Cuidado

- Son métodos asíncronos, no usar variables globales.

Ejemplo completo

Principio (con funciones auxiliares)

```
# -*- coding: utf-8 -*-  
import scrapy  
  
def save_page(page_num, response):  
    response.meta['page'] = page_num  
  
def get_page(response):  
    page_num = response.meta.get('page')  
  
    if not page_num:  
        page_num = 1  
  
    return page_num
```

Ejemplo completo

Spider

```
class BicicletasSpider(scrapy.Spider):
    name = "bicicletas"
    start_urls = [http://www.milanuncios.com/bicicletas-en-cadiz/]
    limits_page = 2

    def parse(self, response):
        anuncios = response.css('div.aditem-detail')

        for anuncio in anuncios:
            title = anuncio.css('a.aditem-detail-title::text').extract_first()
            text = anuncio.css('div.tx::text').extract_first()
            price = anuncio.css('div.aditem-price::text').extract_first()
            yield {'title': title, 'text': text, 'price': price}

        page_num = get_page(response)

        next = response.xpath('//a[text()="Siguiente"]')

        if next is not None and page_num < self.limits_page:
            page_num += 1
            save_page(page_num, response)
            next_page = next.css('::attr(href)').extract_first()
            next_page = response.urljoin(next_page)
            yield scrapy.Request(next_page, callback=self.parse)
```

Documentación

Página oficial <https://scrapy.org/>

Documentación <http://docs.scrapy.org>.

Scrapy Cloud <https://scrapinghub.com/scrapy-cloud/>

Cliente para Scrapy Cloud

<https://pypi.python.org/pypi/shub/>

Índice

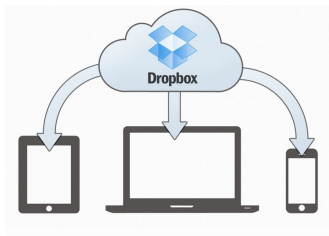
- 1 Extrayendo de la web: scraping
- 2 Acceso al Dropbox
- 3 Acceso al Google Drive

Acceso a Dropbox



Dropbox

- Permite almacenar en la nube.
- Muy usado de forma particular.
- Soporte oficial de acceso desde múltiples idiomas.
- Acceso Python muy sencillo.



Guardar datos usando Dropbox

Librería dropbox (oficial)

- Usaremos la librería oficial **dropbox**.
- Para trabajar necesitamos un Access Token.

Proceso

- 1 Acceder a <https://dropbox.com/developers/apps>.
- 2 Identificarme como usuario Dropbox.
- 3 Crear aplicación.
- 4 Generar Access Token.

Proceso Dropbox: Crear Aplicación



My apps



Daniel Molina

Create app

API v2

My apps

API Explorer

Documentation

HTTP

.NET

Java

JavaScript

Python

Swift

Objective-C

Community SDKs

References

Authentication types

Branding guide

Content hash

Data Ingress guide

Developer guide

OAuth guide

v2 migration guide

Webhooks

Chooser

Saver



DatosRemotos

Status: Development

Permission type: App folder

Proceso Dropbox: Crear Aplicación

Create a new app on the Dropbox Platform

1. Choose an API

- ☒ Dropbox API
For apps that need to access files in Dropbox. [Learn more](#)



- ☐ Dropbox Business API
For apps that need access to Dropbox Business team info. [Learn more](#)



2. Choose the type of access you need

[Learn more about access types](#)

- ☒ App folder – Access to a single folder created specifically for your app.

Mayor seguridad

- ☐ Full Dropbox – Access to all files and folders in a user's Dropbox.

3. Name your app

DatosACompartir

Proceso Dropbox: Generar clave

Development users

Only you

Enable additional users

Permission type

App folder [?](#)

App folder name

DatosACompartir

Change

App key

kiserv6ht1i8apv

App secret

[Show](#)

OAuth 2

Redirect URIs

https:// (http allowed for localhost)

Add

Allow implicit grant [?](#)

Allow

Generated access token [?](#)

Generate

Chooser/Saver domains

example.com

Add

If using the [Chooser](#) or the [Saver](#) on a website, the domain of that site.

Webhooks

Webhook URIs [?](#)

API de Dropbox (V2.0)

Autenticación

```
import dropbox
import tempfile

token = "... "
dbx = dropbox.Dropbox(token)
```

Acceso a la información

- Se accede mediante el objeto dbx (clase Dropbox).

```
user = dbx.users_get_current_account()
```

Usando ficheros

Listado

`files_list_folder(..)` Lista los ficheros de la carpeta ("" para todos).

Devuelve en `.entries` una lista de los ficheros.

File

- Permite obtener todos los datos del usuario.

`name` Nombre.

`size` Tamaño.

... ..

Subiendo y bajando

Bajar fichero

```
dbx.files_download_to_file(nombre_destino, ruta_dropbox)
```

nombre_destino Nombre del fichero en donde guardarlo.

ruta_dropbox Nombre en Dropbox.

Subir ficheros

```
dbx.files_upload(datos, nombre, mute=True)
```

datos Contenido (en binario) del fichero a guardar.

nombre Nombre (con la ruta) del fichero a guardar.

mute Indica si mostrar información por pantalla o no.

Aviso

Las rutas se indican absolutas, pero se guardan relativas a Aplicaciones/NombreAplicación/.

Ejemplo

Subida

```
with open("datos_locales.xls", "rb") as f:  
    data = f.read()  
  
print("Subiendo")  
fname = "/datos_remotos.xls"  
response = dbx.files_upload(data, fname, mute=True)  
print("uploaded2: ", response)
```

Bajada

```
path = "/datos_remotos.xlsx"  
print(path)  
file_temp = tempfile.NamedTemporaryFile(suffix=".xlsx")  
dbx.files_download_to_file(file_temp.name, path)  
print("dropbox:" + path + "->local:" + file_temp.name)
```


Índice

- 1 Extrayendo de la web: scraping
- 2 Acceso al Dropbox
- 3 Acceso al Google Drive

Acceso al Google Drive



Google Drive

- Permite almacenar en la nube.
- Muy usado para editar documentos compartidos.
- Muy buen soporte de ficheros Office.
- Acceso oficial complejo, fácil con librerías externas.

Guardar datos usando Drive

Varias formas


- API oficial.
- Librería oficial.
- Usaremos la librería no-oficial **pydrive**.
- Para trabajar necesitamos un fichero de credenciales.

Proceso

- 1 Acceder a `https://console.developers.google.com`
- 2 Identificarme como usuario de Google/GMail.
- 3 Crear aplicación.
- 4 Habilitar el API correspondiente.
- 5 Escoger el tipo de autenticación.
- 6 Generar Fichero .json con los datos.

Proceso Drive: Crear Aplicación


Seleccionar

 Buscar proyectos y carpetas



Recientes

Todos

Name	ID
✓  [blurred]	[blurred]
 [blurred]	[blurred]
 [blurred]	[blurred]

CANCELAR

ABRIR

Proceso Drive: Habilitar el API

The screenshot displays the Google APIs console interface. At the top, the 'Google APIs' logo is on the left, and a search bar contains 'DatosRemotos' with a dropdown arrow and the text 'Aplicación elegida'. Below the header, the left sidebar shows the 'API Administrador de ...' section with a sub-menu containing 'Panel de control' (highlighted), 'Biblioteca', and 'Credenciales'. The main content area is titled 'Panel de control' and features a red-bordered button labeled '+ HABILITAR API'. Below this, the 'API habilitadas' section states 'Algunas API se habilitan automáticamente'. Two summary cards are shown: 'Tráfico' (Traffic) with the unit 'Solicitudes/segundo' and the message 'There is no traffic for this time period.', and 'Errores' (Errors) with the unit 'Porcentaje de solicitudes' and the message 'There are no errors for this time period.'

Google APIs

DatosRemotos Aplicación elegida

API Administrador de ...

Panel de control

+ HABILITAR API

Panel de control

Biblioteca

Credenciales

API habilitadas

Algunas API se habilitan automáticamente

Tráfico

Solicitudes/segundo

There is no traffic for this time period.

Errores

Porcentaje de solicitudes

There are no errors for this time period.

Proceso Drive: Escoger Google Drive

Biblioteca

APIs de Google

🔍 Buscar en las más de 100 APIs

API populares



APIs de Google Cloud

Compute Engine API
BigQuery API
Cloud Storage Service
Cloud Datastore API
Cloud Deployment Manager API
Cloud DNS API

👇 Más



Aprendizaje automático de Google Cloud

Vision API
Natural Language API
Speech API
Translation API
Machine Learning Engine API



APIs de Google Maps

Google Maps Android API
Google Maps SDK for iOS
Google Maps JavaScript API
Google Places API for Android
Google Places API for iOS
Google Maps Roads API

👇 Más



APIs de G Suite


Drive API

Calendar API
Gmail API
Sheets API
Google Apps Marketplace SDK
Admin SDK

👇 Más



APIs para móviles

Google Cloud Messaging 
Google Play Game Services
Google Play Developer API
Google Places API for Android



APIs de redes sociales

Google+ API
Blogger API
Google+ Pages API
Google+ Domains API



APIs de YouTube

YouTube Data API
YouTube Analytics API
YouTube Reporting API



APIs de publicidad

AdSense Management API
DCM/DFA Reporting And Trafficking API
Ad Exchange Seller API
Ad Exchange Buyer API



Otras API populares

Analytics API
Custom Search API
URL Shortener API
PageSpeed Insights API

Proceso Drive: Crear credencial

API Administrador de ...

Panel de control

Biblioteca

Credenciales

← Google Drive API INHABILITAR

Información general

Cuotas

Integración con la interfaz de Drive

Acerca de esta API

Todas las versiones de API

Todas las credenciales de API

Todos los métodos de API

Tráfico

Por código de respuesta

Solicitudes/s (5 min de media)

No hay dato

Proceso Drive: Crear credencial

API Administrador de ...

Panel de control

Biblioteca

Credenciales

Credenciales Pantalla de autorización de OAuth Verificación de dominio

Crear credenciales Eliminar

Clave de API
Identifica tu proyecto con una simple clave de API para comprobar la cuota y el acceso

ID de cliente de OAuth
Solicita la autorización del usuario para que la aplicación pueda acceder a los datos del usuario

Clave de cuenta de servicio
Permite autenticar a nivel de aplicación y entre servidores mediante cuentas robot

Ayúdame a elegir
Te haremos unas preguntas para decidir qué tipo de credencial puedes usar

Proceso Drive: Crear credencial



Panel de control



Biblioteca



Credenciales

Añadir credenciales al proyecto

1 Averigua qué tipo de credenciales necesitas

Te ayudaremos a configurar las credenciales adecuadas. Puedes saltarte este paso y crear una [clave de API](#), un [ID de cliente](#) o una [cuenta de servicio](#).

¿Qué API estás utilizando?

Determina qué tipo de credenciales necesitas.

Google Drive API

¿Desde dónde llamarás a la API?

Determina qué ajustes necesitas configurar.

Otra IU (por ejemplo, Windows, herramienta de CLI)

¿A qué tipo de datos accederás?

☒ Datos de usuario

Accede a datos pertenecientes a un usuario de Google (con su permiso)

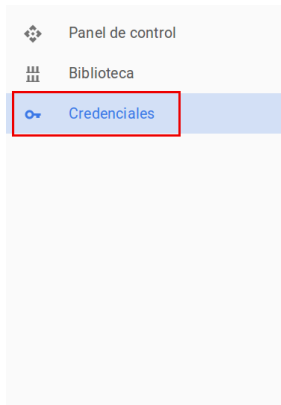
☐ Datos de aplicación

Accede a datos pertenecientes a tu propia aplicación

[¿Qué credenciales necesito?](#)

2 Obtener credenciales

Proceso Drive: Crear credencial



Añadir credenciales al proyecto

-
- ✓ **Averigua qué tipo de credenciales necesitas**
Llamar a Google Drive API desde una plataforma basada en IU
-

2 Crear un ID de cliente de OAuth 2.0

Nombre

cliente

Crear ID de cliente

3 Descargar credenciales

Cancelar

Proceso Drive: Crear credencial

API Administrador de ...

Panel de control

Biblioteca

Credenciales

Credenciales

Añadir credenciales al proyecto

✓

Averigua qué tipo de credenciales necesitas
Llamar a Google Drive API desde una plataforma basada en IU

✓

Crear un ID de cliente de OAuth 2.0
Cliente de OAuth "cliente" creado

3

Descargar credenciales

Client ID

Descarga esta información de credenciales en formato JSON. Siempre estará disponible en la página de credenciales.

Descargar

Lo haré más adelante.

Librería pyDrive

Librería pyDrive

- La idea es **Google Drive fácil**.
- Simplifica la autenticación.
- Simplifica mucho el acceso.

```
$ pip install pydrive
```

Autenticación

```
from pydrive.auth import GoogleAuth  
from pydrive.drive import GoogleDrive
```

```
gauth = GoogleAuth()  
credentials = "mycreds.txt"
```

```
if not os.path.exists(credentials):  
    # Pide confirmacion por la web  
    gauth.LocalWebserverAuth()
```

Uso de Google Drive

Consulta

`ListFile()` Devuelve todos los ficheros.

`ListFile({'q': "})` Permite consultar un tipo concreto.

Multitud de opciones:

<https://developers.google.com/drive/v3/web/search-parameters>

Parámetros de búsqueda

Por nombre \Rightarrow (`name = 'hello'`).

Nombre completo \Rightarrow (`fullName contains "Hola"`).

Por fecha de acceso/modificación .

Usuarios compartidos .

...