

RECSM Summer School: Scraping the web

Pablo Barberá

School of International Relations
University of Southern California

`pablobarbera.com`

Networked Democracy Lab

`www.netdem.org`

Course website:

github.com/pablobarbera/big-data-upf

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...
- ▶ Geographic information, conflict data...

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...
- ▶ Geographic information, conflict data...

These datasets are often provided in an **unstructured format**.

Scraping the web: what? why?

An increasing amount of data is available on the web:

- ▶ Speeches, sentences, biographical information...
- ▶ Social media data, newspaper articles, press releases...
- ▶ Geographic information, conflict data...

These datasets are often provided in an **unstructured format**.

Web scraping is the process of extracting this information automatically and transforming it into a **structured dataset**.

Scraping the web: two approaches

Two different approaches:

1. **Screen scraping**: extract data from source code of website, with html parser and/or regular expressions

Scraping the web: two approaches

Two different approaches:

1. **Screen scraping**: extract data from source code of website, with html parser and/or regular expressions
 - ▶ `rvest` package in R

Scraping the web: two approaches

Two different approaches:

1. **Screen scraping**: extract data from source code of website, with html parser and/or regular expressions
 - ▶ `rvest` package in R
2. **Web APIs** (application programming interfaces): a set of structured http requests that return JSON or XML data

Scraping the web: two approaches

Two different approaches:

1. **Screen scraping**: extract data from source code of website, with html parser and/or regular expressions
 - ▶ `rvest` package in R
2. **Web APIs** (application programming interfaces): a set of structured http requests that return JSON or XML data
 - ▶ `httr` package to construct API requests

Scraping the web: two approaches

Two different approaches:

1. **Screen scraping**: extract data from source code of website, with html parser and/or regular expressions
 - ▶ `rvest` package in R
2. **Web APIs** (application programming interfaces): a set of structured http requests that return JSON or XML data
 - ▶ `httr` package to construct API requests
 - ▶ Packages specific to each API: `weatherData`, `WDI`, `Rfacebook`... Check CRAN Task View on [Web Technologies and Services](#) for more examples

The rules of the game

1. Respect the hosting site's wishes:

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download
 - ▶ Some websites *disallow* scrapers on their `robots.txt` files

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download
 - ▶ Some websites *disallow* scrapers on their `robots.txt` files
2. Limit your bandwidth use:

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download
 - ▶ Some websites *disallow* scrapers on their `robots.txt` files
2. Limit your bandwidth use:
 - ▶ Wait one or two seconds after each hit

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download
 - ▶ Some websites *disallow* scrapers on their `robots.txt` files
2. Limit your bandwidth use:
 - ▶ Wait one or two seconds after each hit
 - ▶ Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download
 - ▶ Some websites *disallow* scrapers on their `robots.txt` files
2. Limit your bandwidth use:
 - ▶ Wait one or two seconds after each hit
 - ▶ Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)
3. When using APIs, read documentation

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download
 - ▶ Some websites *disallow* scrapers on their `robots.txt` files
2. Limit your bandwidth use:
 - ▶ Wait one or two seconds after each hit
 - ▶ Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)
3. When using APIs, read documentation
 - ▶ Is there a batch download option?

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download
 - ▶ Some websites *disallow* scrapers on their `robots.txt` files
2. Limit your bandwidth use:
 - ▶ Wait one or two seconds after each hit
 - ▶ Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)
3. When using APIs, read documentation
 - ▶ Is there a batch download option?
 - ▶ Are there any rate limits?

The rules of the game

1. Respect the hosting site's wishes:
 - ▶ First, check if an API exists or if data are available for download
 - ▶ Some websites *disallow* scrapers on their `robots.txt` files
2. Limit your bandwidth use:
 - ▶ Wait one or two seconds after each hit
 - ▶ Scrape only what you need, and just once (e.g. store the html file in disk, and then parse it)
3. When using APIs, read documentation
 - ▶ Is there a batch download option?
 - ▶ Are there any rate limits?
 - ▶ Can you share the data?

The art of web scraping

Workflow:

1. Learn about structure of website

The art of web scraping

Workflow:

1. Learn about structure of website
2. Build prototype code

The art of web scraping

Workflow:

1. Learn about structure of website
2. Build prototype code
3. Generalize: functions, loops, debugging

The art of web scraping

Workflow:

1. Learn about structure of website
2. Build prototype code
3. Generalize: functions, loops, debugging
4. Data cleaning

Three main scenarios

1. Data in table format



WIKIPEDIA
The Free Encyclopedia

Main menu

Article [Talk](#)

Read [Edit](#) [View history](#)

Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

International court


From Wikipedia, the free encyclopedia

[List of international courts](#) [\[edit \]](#)


Name	Scope	Years active	Subject matter
International Court of Justice	Global	1945–present	General disputes
International Criminal Court	Global	2002–present	Criminal prosecutions
Permanent Court of International Justice	Global	1922–1946	General disputes
Appellate Body	Global	1995–present	Trade disputes within the WTO
International Tribunal for the Law of the Sea	Global	1994–present	Maritime disputes
African Court of Justice	Africa	2009–present	Interpretation of AU treaties
African Court on Human and Peoples' Rights	Africa	2006–present	Human rights
COMESA Court of Justice	Africa	1998–present	Trade disputes within COMESA
ECOWAS Community Court of Justice	Africa	1996–present	Interpretation of ECOWAS treaties
East African Court of Justice	Africa	2001–present	Interpretation of EAC treaties
SADC Tribunal	Africa	2005–2012	Interpretation of SADC treaties

Three main scenarios

2. Data in unstructured format




India English Register for updates 11,072,800 Visitors

 I PAID A BRIBE I DID NOT PAY A BRIBE I MET AN HONEST OFFICER BRIBE HOTLINE ALL REPORTS NEWS **REPORT A BRIBE**

All Reports > I Paid A Bribe

ALL / **I PAID A BRIBE** / BRIBE FIGHTER / HONEST OFFICER / BRIBE HOTLINE


 1 day ago 76 views


POLICE NILO GHUSS (bribe)

Passport | Police Verification for Passport | Paid INR 5,000

Reported on January 17, 2016 from Bankura, West Bengal | Report #89544

What will happen to this country..police mamu's govt income: 30,000 per month. Per day GHUSS income 5000 (per passport verification). Imagine they t...[Read more](#)

 [How to Get a Passport Verified in Ghaziabad](#)


 1 day ago 104 views


Corruption due to vague rules

Police | Traffic Violations | Paid INR 500

Reported on January 16, 2016 from Mumbai, Maharashtra | Report #89509

At Chembur near Eastern Expressway traffic cop stopped me and started checking docs..all was fine buy cup expired..then he pointed out film.. He took...[Read more](#)

 [Things to Know on Traffic Offences and Respective Penalties](#)

 2 days ago 105 views

Bribe collected by Staff of Enrollment agency

Municipal Services | Aadhaar or UID Related | Paid INR 120

Reported on January 16, 2016 from Mysore, Karnataka | Report #89467

UIDAI has to take a stand on fees to be paid to enrolment agencies for processing Aadhaar

FILTER REPORTS

Which city?
All cities

Department
All departments


Bribe Amount
All Amount


SUBMIT


INSPIRE OTHERS WITH YOUR STORY

Manik Tanega, a sports enthusiast, wrote against a custom officer on ipaidabribe.com, for cough up a hefty bribe by a Customs official at Bengaluru airport.

SEE HIS STORY

 Ever Paid A Bribe?

 Report your Bribe Story!

 See action taken.

www.ipaidabribe.com/reports/paid

Three main scenarios

3. Data hidden behind web forms

The screenshot shows the 'MONITOR LEGISLATIVO' website interface. At the top, there is a navigation bar with icons for home, user profile, news, candidates, national assembly, abuses, and contact. Below this is a search bar with the text 'RESULTADOS DE LA CONSULTA'. The search results are displayed under the heading 'DIPUTADOS ENCONTRADOS'. There are six candidate profiles shown, each with a photo, name, and state. The candidates are: Julio Ygarza (Estado: Amazonas), Mauligmer Baloa (Estado: Amazonas), Nirma Guarulla (Estado: Amazonas), José Brito (Estado: Anzoátegui), Chaim Bucarán (Estado: Anzoátegui), and Richard Arteaga (Estado: Anzoátegui). Each profile includes the 'Unidad' logo.

MONITOR LEGISLATIVO

INICIO | PERFIL IDEAL | NOTICIAS | CANDIDATOS | ASAMBLEA NACIONAL | ABUSOS | CONTACTENOS

RESULTADOS DE LA CONSULTA

Seleccione | Partido | BUSCAR

DIPUTADOS ENCONTRADOS

Julio Ygarza
Estado: Amazonas

Mauligmer Baloa
Estado: Amazonas

Nirma Guarulla
Estado: Amazonas

José Brito
Estado: Anzoátegui

Chaim Bucarán
Estado: Anzoátegui

Richard Arteaga
Estado: Anzoátegui

Candidates on 2015 Venezuelan parliamentary election

Three main scenarios

1. Data in **table** format

Three main scenarios

1. Data in `table` format
 - ▶ Automatic extraction with `rvest`

Three main scenarios

1. Data in **table** format
 - ▶ Automatic extraction with `rvest`
2. Data in **unstructured** format

Three main scenarios

1. Data in **table** format
 - ▶ Automatic extraction with `rvest`
2. Data in **unstructured** format
 - ▶ Element identification with `selectorGadget`

Three main scenarios

1. Data in **table** format
 - ▶ Automatic extraction with `rvest`
2. Data in **unstructured** format
 - ▶ Element identification with `selectorGadget`
 - ▶ Automatic extraction with `rvest`

Three main scenarios

1. Data in **table** format
 - ▶ Automatic extraction with `rvest`
2. Data in **unstructured** format
 - ▶ Element identification with `selectorGadget`
 - ▶ Automatic extraction with `rvest`
3. Data hidden **behind web forms**

Three main scenarios

1. Data in **table** format
 - ▶ Automatic extraction with `rvest`
2. Data in **unstructured** format
 - ▶ Element identification with `selectorGadget`
 - ▶ Automatic extraction with `rvest`
3. Data hidden **behind web forms**
 - ▶ Automation of web browser behavior with `selenium`

APIs

API = Application Programming Interface; a set of structured https requests that return data in JSON or XML format.

Types of APIs:

1. **RESTful APIs**: queries for static information at current moment (e.g. user profiles, posts, etc.)
2. **Streaming APIs**: changes in users' data in real time (e.g. new tweets, new FB posts...)

APIs

API = Application Programming Interface; a set of structured https requests that return data in JSON or XML format.

Types of APIs:

1. **RESTful APIs**: queries for static information at current moment (e.g. user profiles, posts, etc.)
2. **Streaming APIs**: changes in users' data in real time (e.g. new tweets, new FB posts...)

Most APIs are **rate-limited**:

- ▶ Restrictions on number of API calls by user/IP address and period of time.

Connecting with an API

Constructing a REST API call:

- ▶ **Baseline URL:**

`https://maps.googleapis.com/maps/api/geocode/json`

- ▶ **Parameters:** `?address=barcelona`

- ▶ **Authentication token:** `&key=XXXXXX`

Response is often in **JSON** format.

Connecting with an API

Constructing a REST API call:

- ▶ **Baseline URL:**

`https://maps.googleapis.com/maps/api/geocode/json`

- ▶ **Parameters:** `?address=barcelona`

- ▶ **Authentication token:** `&key=XXXXXX`

Response is often in **JSON** format.

Authentication:

- ▶ Many APIs require an access key or token
- ▶ An alternative, open standard is called OAuth
- ▶ Connections without sharing username or password, only temporary tokens that can be refreshed
- ▶ `httr` package in R implements most cases (**examples**)

Other APIs

See [CRAN Web Technologies Task View](#)