Nikita Andreev Sabina Askerova Anna Krysta Caio Rocha

Job market

data acquisition

Data acquisition, extraction, and storage course (2024-2025)

Prof. Pierre Senellart

Table Of Contents

{01} Web crawling and scraping

Resources and tools used

Challenges encountered

{02} Data cleaning

Data format

Challenges encountered

{03} Storage solution

Elaborate on what you want to discuss.

{04} Final analysis

Elaborate on what you want to discuss.

{01} Web crawling and scraping

Sites considered:

welcometothejungle.com
upwork.com
linkedin.com
indeed.com

Job APIs are almost never free.

😞

Challenges Encountered

Robots.txt Compliance

All sites disallow job scraping

```
`User-agent : *
disallow: /me/*
disallow: /settings/*
disallow: /users/*
disallow: */jobs?query=*`
```

welcometothejungle.com/robots.txt

```
`User-agent : *
# Directories
Disallow: /att/
Disallow: /att-old/
Disallow:
/freelancers/public/api/
Disallow: /messages/
Disallow: /*/jobs/search*
Disallow: /search/profiles/*
Disallow: /catalog-images/*`
```

www.upwork.com/robots.txt

Challenges Encountered

CSRF Token Issues

(a unique, secret, and unpredictable value that is generated by the server-side application and shared with the client.)

Sites concerned: Upwork, LinkedIn, Indeed

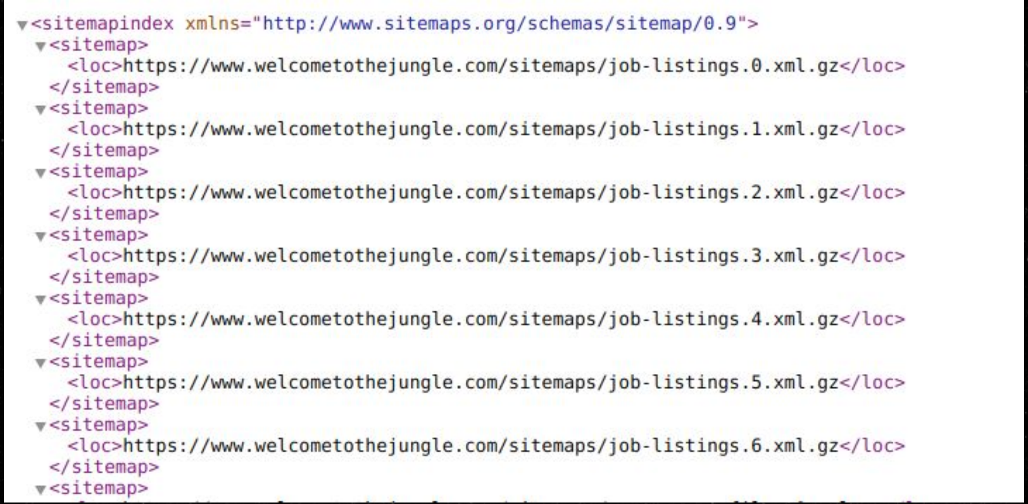
- Couldn’t retrieve from HTML or Cookie
- Changes at every page query and session and page refresh

cookie:

```
visitor_id=7…000; spt=e100e574- … 6a2; G_ENABLED_IDPS=google;
_tt_enable_cookie=1; _ttp=cFIJsN…aqg6pze;
_cq_duid=1.1724497332.DDo…th7nm; __pdst=e84…c177;
IR_PI=522e6f7d-6208…97334653;
OptanonAlertBoxClosed=2024-11-08T07:38:06.710Z;
_ga=GA1.2.756219045.1724497334; ftr_ncd=6;
recognized=username; company_last_accessed=d…42;
country_code=FR; cookie_prefix=; cookie_domain=.upwork.com;
__cflb=02Di…bcGx5iHtF;
_cfuvid=DqX8RDziG5…s-1733986834830-0.0.1.1-604800000;
_upw_ses.5831=*; _cq_suid=1.173…kEOPF;
visitor_gql_token=oauth2v2_c498e5483…37a93cb;
XSRF-TOKEN=ef3e…6e7c828b0;
```

*CSRF token needed for login to the sites, impossible to use except for browser automation tools which are slower

Challenges Encountered



```
▼<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
 ▼<sitemap>
    <loc>https://www.welcometothejungle.com/sitemaps/job-listings.0.xml.gz</loc>
  </sitemap>
 ▼<sitemap>
    <loc>https://www.welcometothejungle.com/sitemaps/job-listings.1.xml.gz</loc>
  </sitemap>
 ▼<sitemap>
    <loc>https://www.welcometothejungle.com/sitemaps/job-listings.2.xml.gz</loc>
  </sitemap>
 ▼<sitemap>
    <loc>https://www.welcometothejungle.com/sitemaps/job-listings.3.xml.gz</loc>
  </sitemap>
 ▼<sitemap>
    <loc>https://www.welcometothejungle.com/sitemaps/job-listings.4.xml.gz</loc>
  </sitemap>
 ▼<sitemap>
    <loc>https://www.welcometothejungle.com/sitemaps/job-listings.5.xml.gz</loc>
  </sitemap>
 ▼<sitemap>
    <loc>https://www.welcometothejungle.com/sitemaps/job-listings.6.xml.gz</loc>
  </sitemap>
 ▼<sitemap>
```

www.welcometothejungle.com/sitemaps/index.xml.gz

Sitemap Limitations:

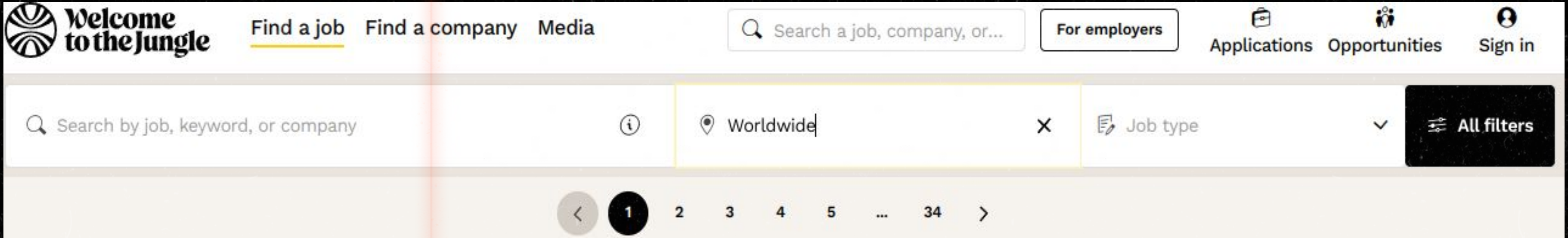
- Incomplete links for jobs and companies
- Compressed Format
- Multiple Sitemaps

Reverse Engineering API

Attempt: Reverse engineering the API for more data

Outcome: Limited info available :(

(34×30 job postings through filter search on main page)



Data Acquisition Process

Step 01

Crawling companies using Katana Go

```
katana -u https://www.welcometothejungle.com/en/companies -d 3 -o companies.txt
```

Discovered very useful links:

https://www.welcometothejungle.com/en/directory/x

https://www.welcometothejungle.com/en/directory/y

https://www.welcometothejungle.com/en/directory/z

Why we chose Katana Go for crawling?

- Concurrency
- Speed: Go is a compiled language
- Ease of use in terminal

github.com/projectdiscovery/katana

Data Acquisition Process

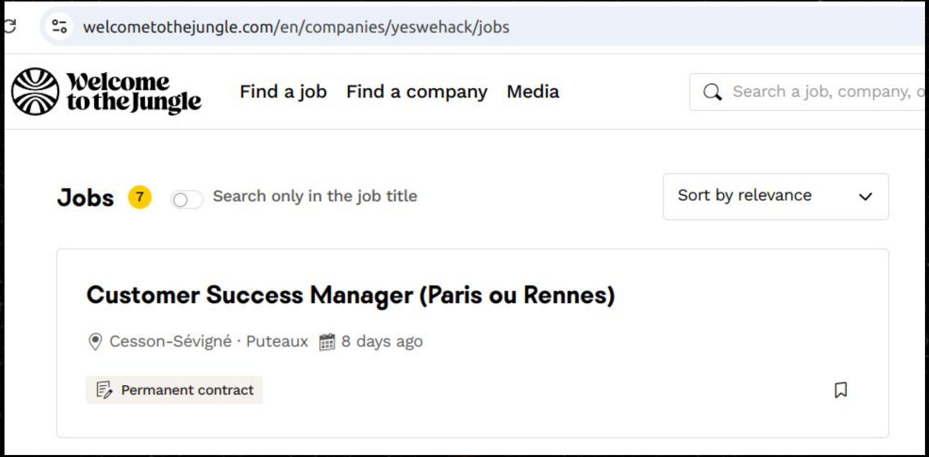
Follow and scrape companies and jobs using Scrapy

Step 02

welcometothejungle.com/en/companies/yeswehack/jobs

Link complies with robots.txt (disallow: */jobs?query=*) because no query.

Hooray!



welcometothejungle.com/en/companies/yeswehack/jobs

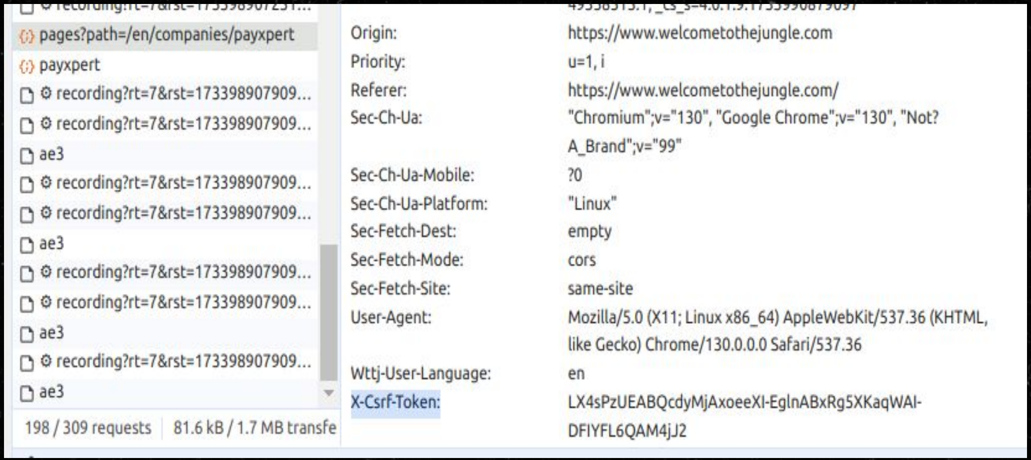
Customer Success Manager (Paris ou Rennes)

Cesson-Sévigné · Puteaux 8 days ago

Permanent contract

Data Acquisition Process

Step 02



```
pages?path=/en/companies/payxpert
payxpert
⚙ recording?rt=7&rst=173398907909...
⚙ recording?rt=7&rst=173398907909...
ae3
⚙ recording?rt=7&rst=173398907909...
⚙ recording?rt=7&rst=173398907909...
ae3
⚙ recording?rt=7&rst=173398907909...
⚙ recording?rt=7&rst=173398907909...
ae3
⚙ recording?rt=7&rst=173398907909...
ae3
198 / 309 requests    81.6 kB / 1.7 MB transfe
```

```
Origin:                 https://www.welcometothejungle.com
Priority:               u=1, i
Referer:                https://www.welcometothejungle.com/
Sec-Ch-Ua:              "Chromium";v="130", "Google Chrome";v="130", "Not?
                        A_Brand";v="99"
Sec-Ch-Ua-Mobile:       ?0
Sec-Ch-Ua-Platform:     "Linux"
Sec-Fetch-Dest:         empty
Sec-Fetch-Mode:         cors
Sec-Fetch-Site:         same-site
User-Agent:             Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
                        like Gecko) Chrome/130.0.0.0 Safari/537.36
Wttj-User-Language:     en
X-Csrf-Token:           LX4sPzUEABQcdyMjAxoeeXI-EglnABxRg5XKaqWAI-
                        DFIYFL6QAM4jJ2
```

Reverse Engineering to scrape companies data was impossible because of CSRF token problem again :(

Data Acquisition Process

Step 03

Scrape general info about companies using Scrapy

Being nice

```
ROBOTSTXT_OBEY = True # Obey robots.txt rules
CONCURRENT_REQUESTS = 8 # being nice
DOWNLOAD_DELAY = 3 # being nice
CONCURRENT_REQUESTS_PER_DOMAIN = 4 # being nice
```

Used css selectors unique for all pages

```
company_item["website"] =
response.css('div[data-testid="showc
ase-header-website"]
a::attr(href)').get()

job_items =
response.css('ul[data-testid="search
-results"] >
li[data-testid="search-results-list-
item-wrapper"]')
```

Jobs.json

Key	Data Type	Description
company_name	String	Name of the company.
job_title	String	Title of the job position.
location	String	Location where the job is based.
posted_date	String (Date)	Date and time when the job was posted (ISO 8601 format).
contract_type	String	Type of employment contract.
remote_status	String	Remote work policy.
job_link	String (URL)	Relative link to the job posting on the website.

Companies.json

Key	Data Type	Description
name	String	Name of the company.
sector	String	Industry sectors.
website	String (URL)	Company's website link.
year_of_founding	String (Year)	Year of founding.
employees	String (Number)	Number of employees.
gender_breakdown	Object	Gender statistics.
average_age	String (Number)	Average age of employees.
social_links	Object	Object containing social media links.
text_blocks	Object	Descriptive text fields.

{02} Data cleaning

- Attribute unique IDs
- Remove companies without name
- Split sector string into a list of sectors the company operates in
- Identification of the language of the company description

Challenges Encountered

Named Entity Recognition (NER) on the company description field. Tried to extract:

- Organizations
- Locations
- Persons
- Skills

Challenges:

- Quality of the NER model
- Extracting custom skills
- Storing semi-structured data in the relational database

{03} Storage solution – SQLite

Table: companies

Column Name	Data Type	Description
id	INTEGER	Primary Key, unique identifier for each company.
name	TEXT	Name of the company.
sector	TEXT	Sector(s) in which the company operates (comma-separated).
website	TEXT	Website URL of the company.
year_of_founding	TEXT	Year the company was founded.
employees	INTEGER	Number of employees in the company.
gender_breakdown_women	INTEGER	Percentage of female employees.
gender_breakdown_men	INTEGER	Percentage of male employees.
average_age	INTEGER	Average age of employees.
social_links_facebook	TEXT	Facebook link for the company.
social_links_linkedin	TEXT	LinkedIn link for the company.
social_links_twitter	TEXT	Twitter link for the company.
social_links_youtube	TEXT	YouTube link for the company.
presentation	TEXT	Company description or presentation text.
looking_for	TEXT	Details about what the company is looking for in candidates.
good_to_know	TEXT	Additional information about the company.

{03} Storage solution – SQLite

Table: jobs

Column Name	Data Type	Description
id	INTEGER	Primary Key, unique identifier for each job.
company_id	INTEGER	Foreign Key referencing companies(id), linking job to its company.
job_title	TEXT	Title of the job.
location	TEXT	Location of the job posting.
posted_date	TEXT	Date the job was posted.
contract_type	TEXT	Type of contract (e.g., temporary, part-time, internship).
remote_status	TEXT	Remote working status (e.g., a few days at home, fully-remote).
job_link	TEXT	URL link to the detailed job posting.

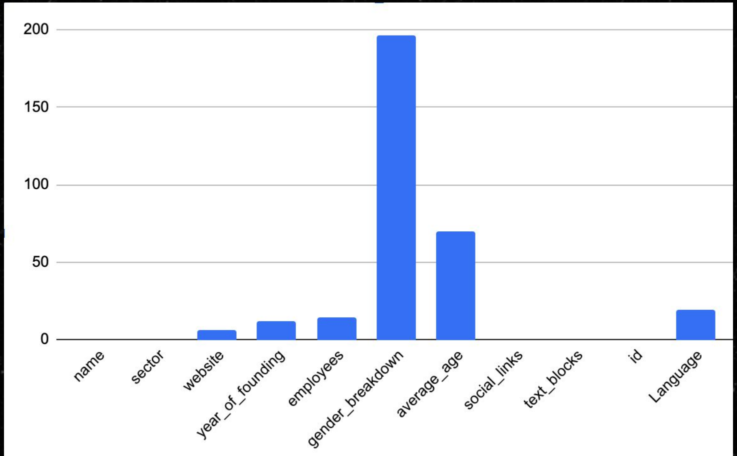
Relationships
- One-to-Many: companies.id → jobs.company_id

{03} Storage solution – SQLite

- Data stored in a single file
- Relational database
- Portable and easy to integrate for practical usage
- Data used in our project has relations and is structured which indicates to use SQLite

{04} Analysis of the quality of the final dataset

- 203 companies were filtered out (no name provided)
- 675 companies in the final dataset
- Only 1 company duplicated was found out
- No columns with all-NaN values
- Almost 10% missing values in age info
- Gender breakdown is missing for 196 companies (with 52%/48% male to female employees ratio)



Missing values for each column in companies table

{04} Analysis of the final dataset’s biases

- Average founding year is 2006 + 2463 employees on average (so, probably not much fresh startups)

- Average employees age is 32 (not much internship/new grad openings)

- Male/Female ratio is 58%/42% (and also no info for 196 companies)

- Most popular sectors: Cloud Services, Software and AI

Thank you for your attention guys 😄