Shelter Dogs Database

FERNANDES MACEDO Gabriella, MATHIOT Raphaël

December 18, 2025

Table of contents

Table of contents

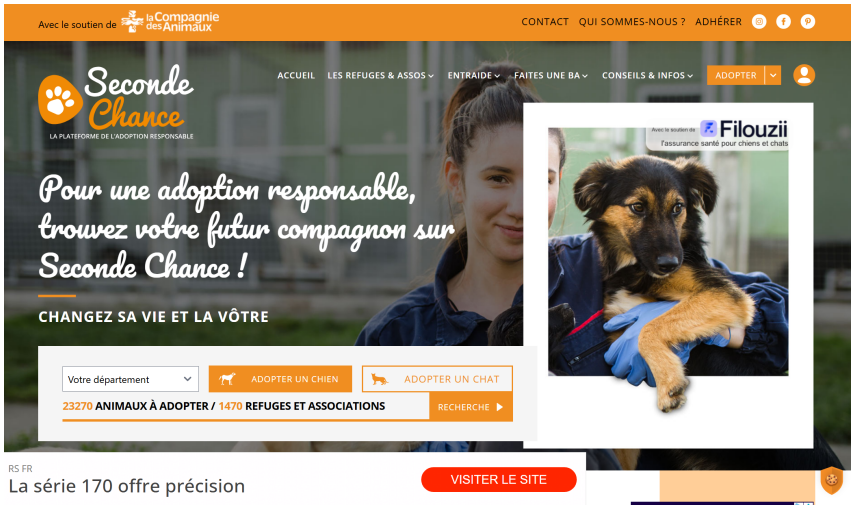
Motivations

- Tens of thousands of dogs are abandoned each year in France.
- Adoption data is fragmented and often incomplete.
- Goal: create a unified, clean dataset of dogs available for adoption.
- Include key info: name, age, breed, size, behavior, compatibility, photos...
- Support research, decision-making and raise awareness about animal adoption.

Websites


```
# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow: /adoption/?search=1&species=**race=*&page=*&seed=*
Disallow: /adoption/*&criteria=*
Disallow: /*?field_refuge_animal_target_id=*
Disallow: /app/wp-json/*
Disallow: /*&post_id=*
Disallow:/tunnel-adoption*
Disallow:/taxonomy*
Disallow: /*field_*
Allow: /adoption/?search=&1race=*
Sitemap: https://www.la-spa.fr/app/sitemaps.xml
```

Figure: https://www.la-spa.fr/




```
User-agent: Slurp
Crawl-delay: 10
User-agent: *
Crawl-delay: 1
User-agent: facebookexternalhit
Disallow:
```

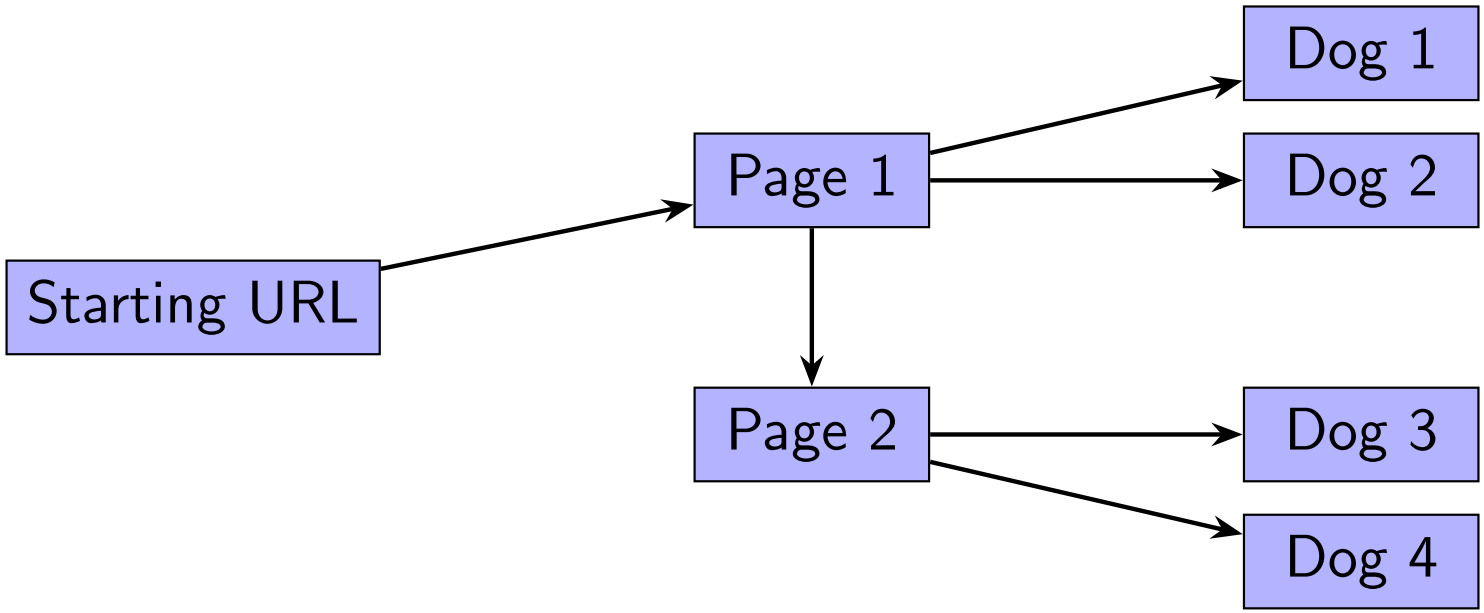
Figure: https://www.secondechance.org/

Table of contents

Scraping Seconde Chance

Seconde Chance is a full-HTML website, which is easy to scrape using `scrapy`.

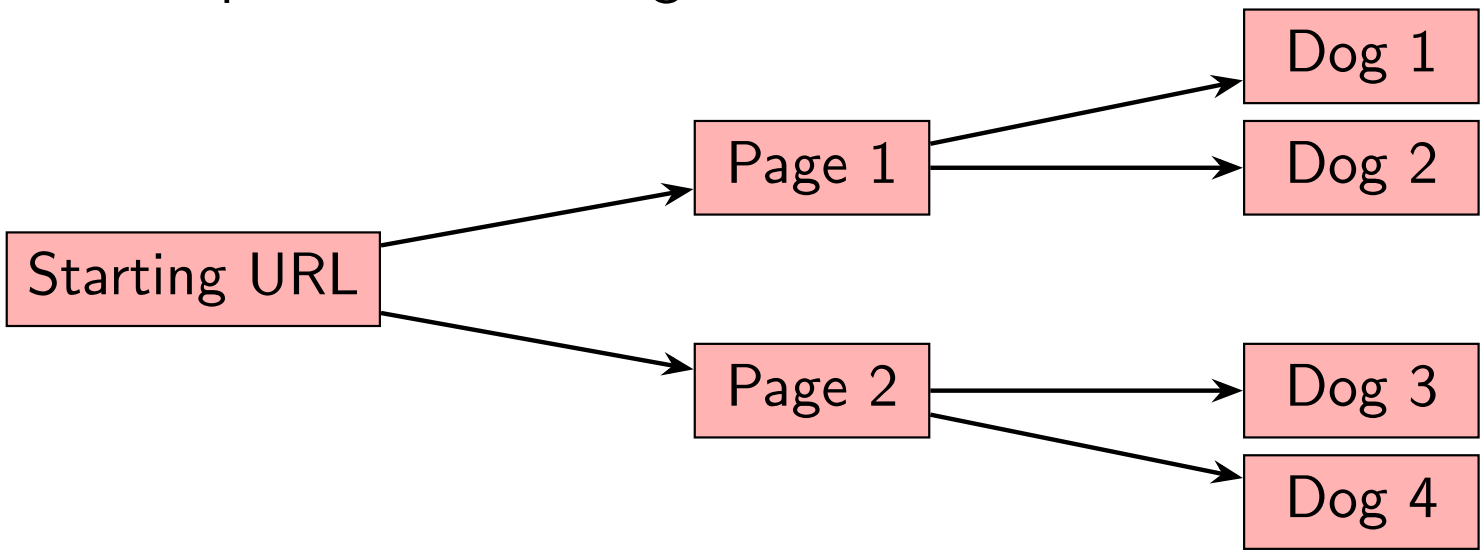
- Starting url : https://www.secondechance.org/animal/dogs
- HTML pages containing URLs pointing to individual HTML pages for each dog.



Scraping SPA

Not as easy as Seconde Chance, since the website uses JavaScript to load dynamically dogs via JSON APIs.

- But we can access this JSON endpoint for each page, which contains IDs of the dog (seed is important !) : https://www.la-spa.fr/app/wp-json/spa/v1/animals/search/?api=1&species=chien&paged=1&seed=224145464626602
- Then, we use the obtained IDs to format the base URL and obtain the JSON endpoint for each dog.



Retrieving information for both websites

Seconde Chance :

```
</div>
    <p>
        <strong>Espèce</strong> : Chien
    </p>
    <p>
        <strong>Type</strong> : CHIEN CROISÉ PETIT
    </p>
    <p>
        <strong>Sexe</strong> : Mâle
    </p>
    <p>
        <strong>Couleur</strong> : Noir et feu
    </p>
    <p>
        <strong>Pelage</strong> : Ras
    </p>
    <p>
        <strong>Âge</strong> : 1 an
    </p>
    <p>
        <strong>Taille</strong> : Petit
    </p>

    <br/>
    <p>
        Dernière mise à jour le 17/12/2025.
    </p>
</div>
```

SPA :


```
"content": {
  "infos": {
    "ID": 184741,
    "title": "Darko",
    "argos_id": "748826",
    "is_liked": false,
    "fad": false,
    "expr": false,
    "sos": false,
    "species": {
      "name": "Chien",
      "url": "/prendre-soin/chiens/les-chiens-et-leurs-races/"
    },
    "races": [
      {
        "name": "Berger allemand",
        "url": "/prendre-soin/chiens/les-chiens-et-leurs-races/le-berger-allemand/"
      }
    ],
    "birthday": "Né(e) le 2024-04-28",
    "age": "junior",
    "sex": "Mâle",
    "colors": [],
    "medias": [
      {
        "type": "image",
        "src": "/app/app/uploads/animals/184741/darko-184741-69317dcb6029b.jpg"
      },
      {
        "type": "image",
        "src": "/app/app/uploads/animals/184741/darko-184741-69317dcc31c4e.jpg"
      }
    ],
    "accepted": {
      "dog": null,
      "cat": null,
      "child": null
    },
```

Data Processing: Cleaning and Normalization

We need the same fields, with the same conventions for both websites. An example of the processing step :

- **Name Cleaning Heuristic:**
  1. Remove punctuation marks and shelter-specific artefacts (for example ids, or categories like HAA, QCN...)
  2. Remove all words belonging to the French dictionary (except if all of them match this condition).
  3. Example : "Adorable THOR, 3 ans" $\rightarrow$ "Thor"

Intermediate Storage and Caching

After collecting data, we need to design a backend to store it and avoid crawling twice the same page :

- Append each dog record in a JSONL file.
- **Cache the visited URLs** to avoid visiting them again.
- If we change our preprocessing, we don't have to crawl again : we can directly start from these JSONL records.
- Use a **sqlite** database to store the final data.

Table of contents

Breeds Data Enrichment

Idea : enrich the information about a dog with **additional data** about its breed.

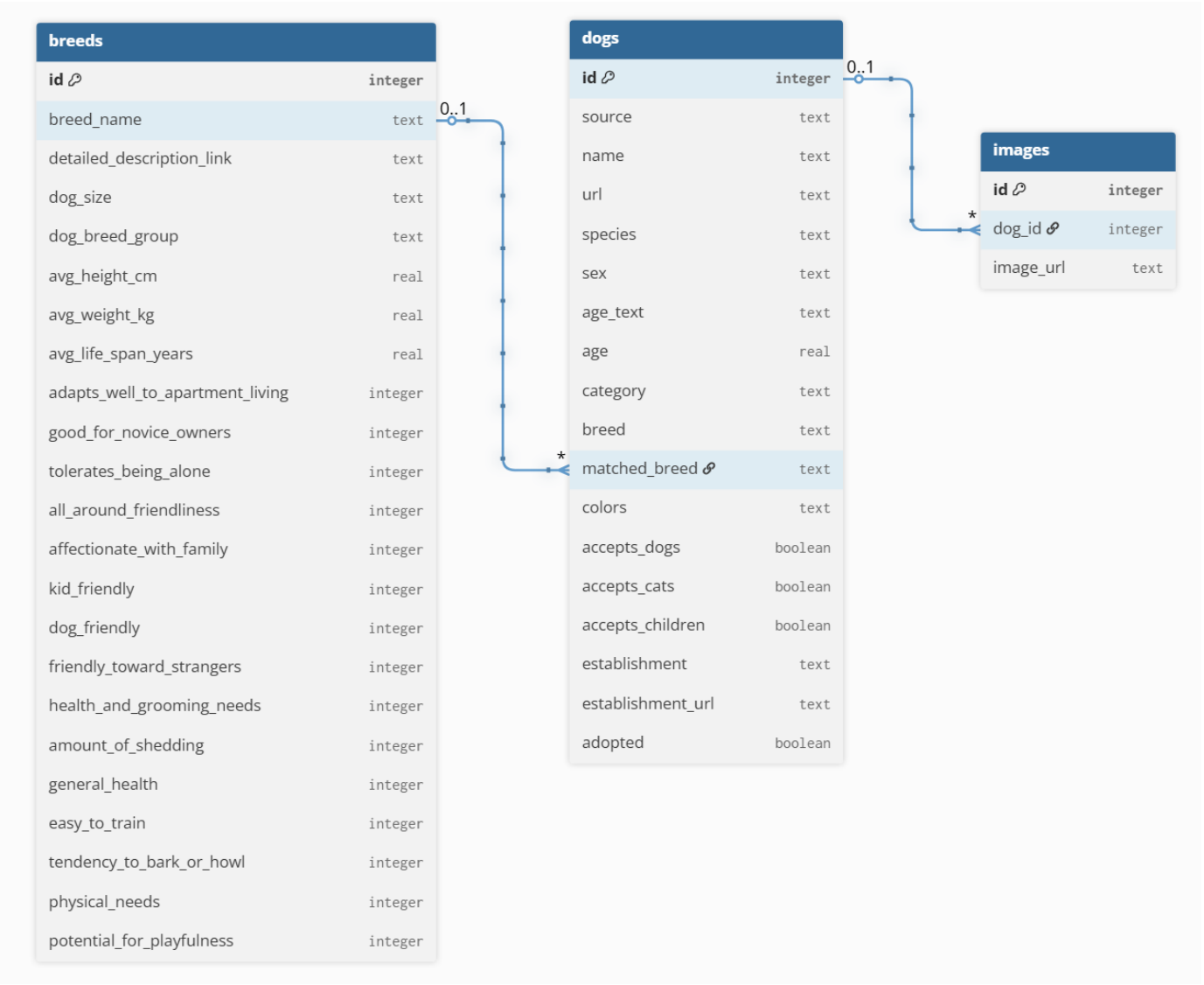
- Possible to do so by scraping data about each breed, for example on `wikipedia`...
- But hard to come up with **normalized statistics** about each breed !
- Use a normalized Kaggle dataset: https://www.kaggle.com/datasets/yonkotoshiro/dogs-breeds?resource=download&select=dogs_cleaned.csv
- Contains 41 features, 31 of which are a **score between 0 and 5**, which can be very useful for statistics.

Updating the database

Problem with our approach : we build our database once and never go back to the online data.

- But adoption lists **fluctuate** very fast !
- Solution : enable a lighter crawl to check if the URLs are **still valid**, to check if the dogs are still registered for adoption.
- Allows us to also keep track of the dogs who have been adopted.

Relational Schema



Field	Missing (%)
id	0.00
source	0.00
name	0.21
url	0.00
adopted	0.00
species	0.00
sex	0.00
age_text	0.25
age	0.17
category	0.00
breed	0.00
matched_breed	72.37
colors	29.63
accepts_dogs	4.21
accepts_cats	6.19
accepts_children	5.02
establishment	0.00
establishment_url	0.00

Percentage of missing values per field

Figure: Relational schema of the created database.

Table of contents

Potential uses

- Machine Learning prediction models on adoption.
- Platform that filters dogs by characteristics and retrieves their breed specificity.
- Statistics to raise awareness about shelter dogs.