Webscraping for Analytics

Vandy Berten

Smals Research

Webinaire 30/06/2020

Smals
ICT for society

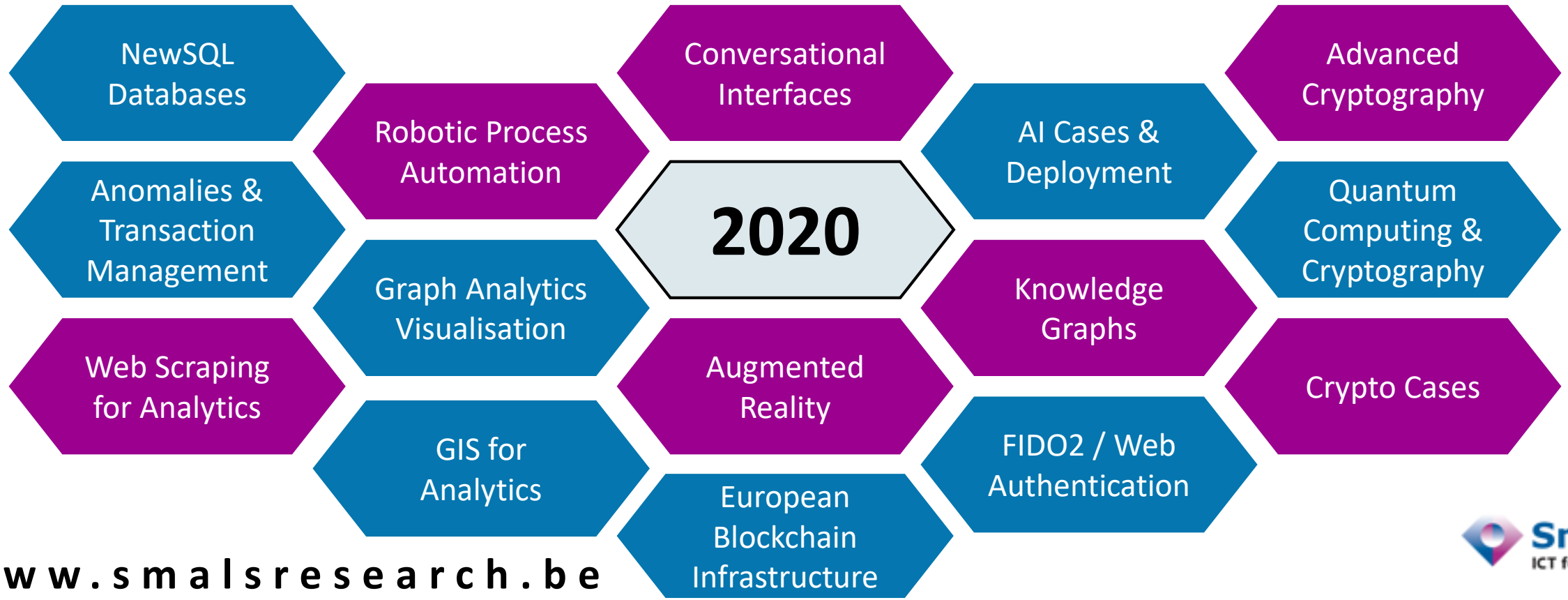
Smals Research

Innovation with new technologies

Consultancy & expertise

Internal & external knowledge transfer

Support for going live



www.smalsresearch.be

Smals
ICT for society

- Définition
- Exemples eGov
- Approches
- Exemple Scrapy
- POC Horeca (Scraping + Entity Linking)
- Aspects juridiques
- Conclusions

- Web scraping : technique d'extraction de contenu sur des sites web, au moyen de scripts/programmes, dans le but de son utilisation dans un autre contexte
- Web crawling/scraping/harvesting
- Sans accès à des APIs dédiées (cf Twitter, Facebook, Google, flux RSS…)

- Le plus gros crawler : Google !

economie

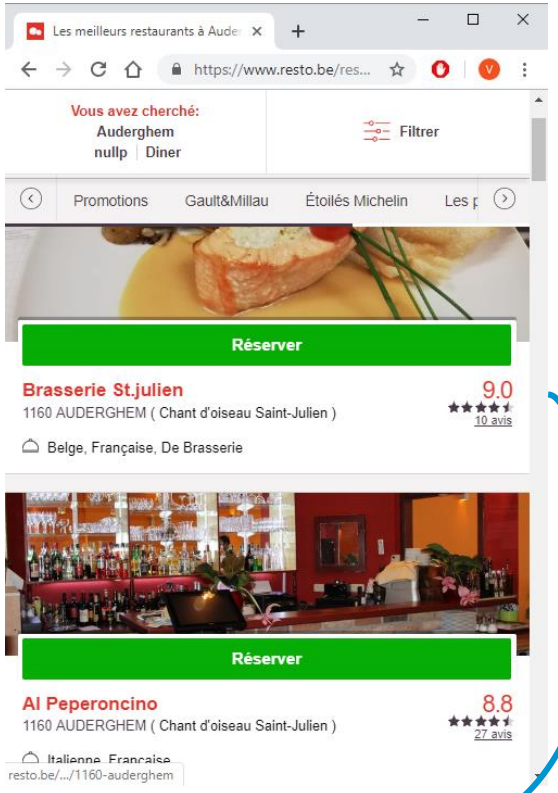
https://statbel.fgov.be/sites/default/files/files/documents/Analyse/FR/Webscraping_fr.pdf

Smals ICT for society

Service Public
Fédéral
FINANCES

- « Uber et Airbnb transmettent les données de leurs utilisateurs au fisc »
- Porte-parole du SPF Finance« […] nous obtenions les données pertinentes concernant l'économie partagée au moyen de scripts, soit des petits programmes développés par le SPF Finances lui-même »

https://moneytalk.levif.be/finance-et-bourse/fiscalite/uber-et-airbnb-transmettent-les-donnees-de-leurs-utilisateurs-au-fisc/article-normal-451629.html



Scrapy

Web Scraping

KBO

Name	Addresse
Chez Léon	Rue des bouchers
Comme chez soi	Place Rouppe
La Villa Lorraine	Avenue Vivier d'Oie
Le Chalet de la Forêt	Drève de Lorraine
…	…

X

KBO	Name	Addresse
0421.614…	Au "Comme chez soi"	Place Rouppe
0414.370…	Villa-Lorraine	Avenue Vivier d'Oie
0455.779…	La sœur du Patron	Chaussée de Wavre
0412.387…	RESTAURANT CHEZ LEON	Rue des bouchers, 18
…	…	…

Entity Linking

Smals ICT for society

- Sites “html-oriented” (resto.be & co) : contenu nécessaire dans le code HTML
- Sites “javascript-oriented” (Google, Facebook...) : contenu nécessaire généré “à la volée”
- API (OpenStreetMap, Google maps…) : contenu direct, structuré

```
<html> (…)
<link href='style.css'
      rel='stylesheet'>
<script src='scripts.js'/>
   <span class="text">
     "My Text"
   </span>
   <img src="logo.png">
</html>
```

Index.html

- Chargement utile : page principale
- Permet d’être très efficace
- Moins lourd pour le serveur, mais … plus détectable
- Exemple de Framework : Scrapy (Python)

logo.png

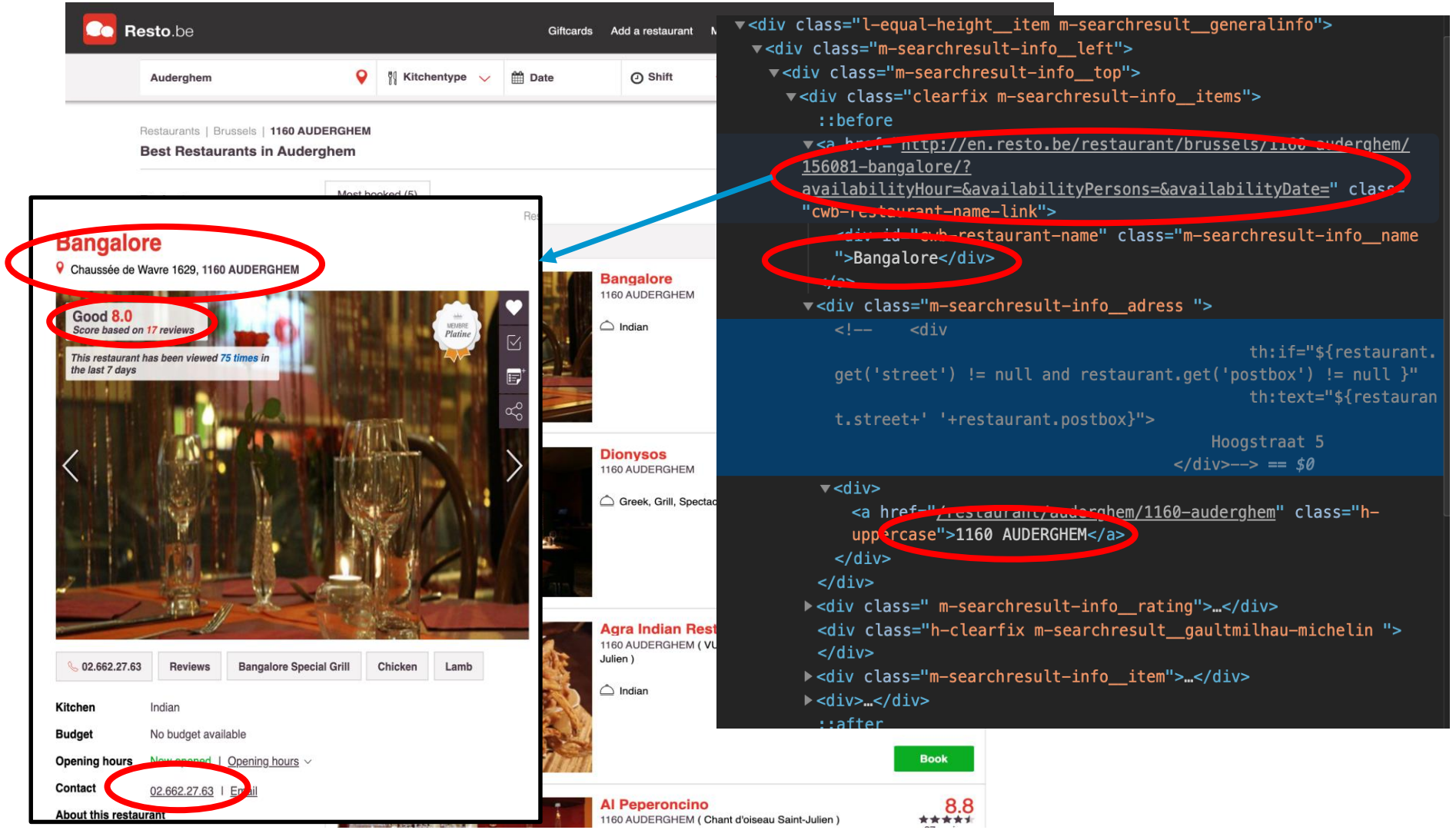
```
function myFunction() {
  document.\
   getElementById("demo").\
   innerHTML="A text";
}
```

scripts.js

```
H1  {
  color: white ;
  border: 1px}
.text {
  color: black ;
  font-family: Arial}
```

style.css

Exemple : resto.be



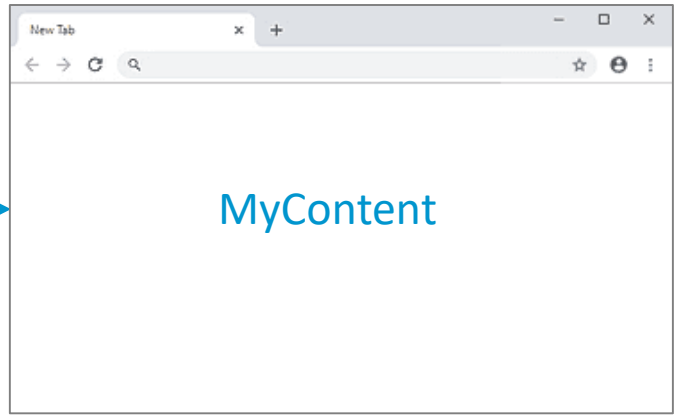
Smals
ICT for society

```
<html> (…)
<script>
function loadContent(){
  document.\
   getElementById(“body").\
   innerHTML=“MyContent";
}
</script>
<body onload=‘loadContent()’/>
</html>
```

Index.html

```
<html> (…)
<body onload=‘loadContent()’>
MyContent
</body>
</html>
```

Dynamic source

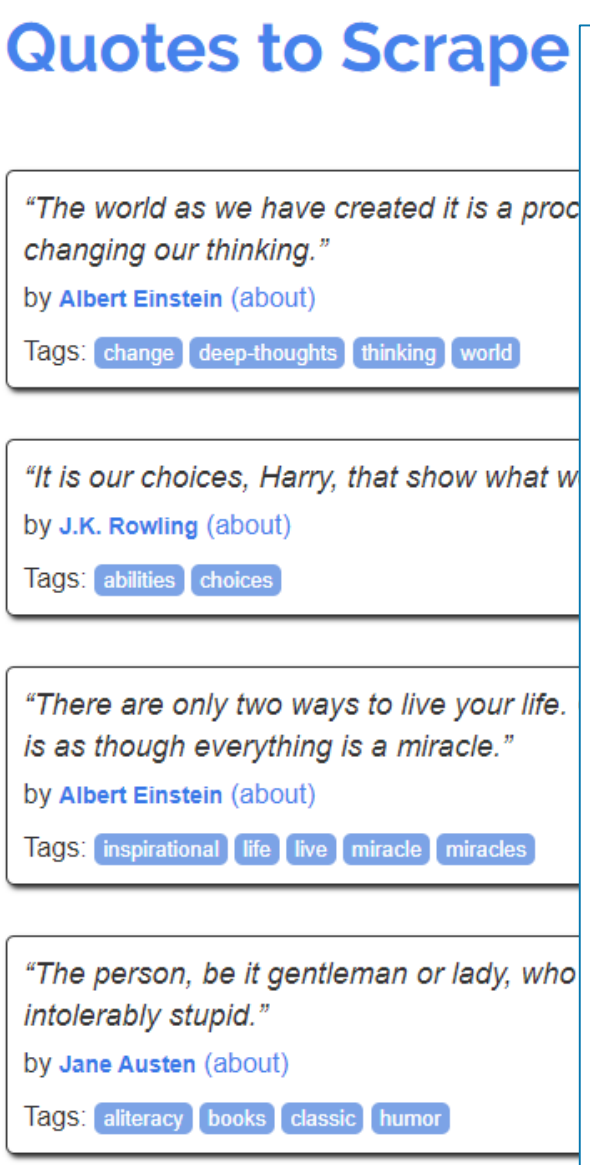


- On doit “simuler” le comportement d’un utilisateur (~RPA) avec un browser
- Possible sans affichage (“headless”)
- Demande beaucoup plus de temps/ressources (chargement complet)
- Souvent difficile pour un script de savoir que la page est “chargée/prête”
- Exemple de Framework : Selenium

Smals ICT for society

- Scrapy : framework en Python pour le webscraping
- À fournir :
  - Un “parseur” (fonction Python) par (type de) page
  - Une (liste d’) URL
- Scrapy :
  - Charge et schedule (en parallèle) les pages → pas de chargement inutile
  - Parse le HTML → On reçoit un object “arborescent”
  - Est hautement paramétrable : débit maximal (délai, # threads…), respect ou non de “robots.txt”, rotation de proxies, …

- Chaque “parseur” est un “générateur” (ou itérateur)
- Générateur : fonction qui génére une liste, mais en créant les éléments au fur et à mesure qu’on les consomme
- Un parseur peut générer :
  - Des données collectées :
    - Typiquement une ligne du tableau résultat
    - Plus généralement, un dictionnaire « {"key1" : "value1", "key2" : "value2",…} »
  - Une URL (+ le parseur associé) qui sera rajoutée au scheduleur



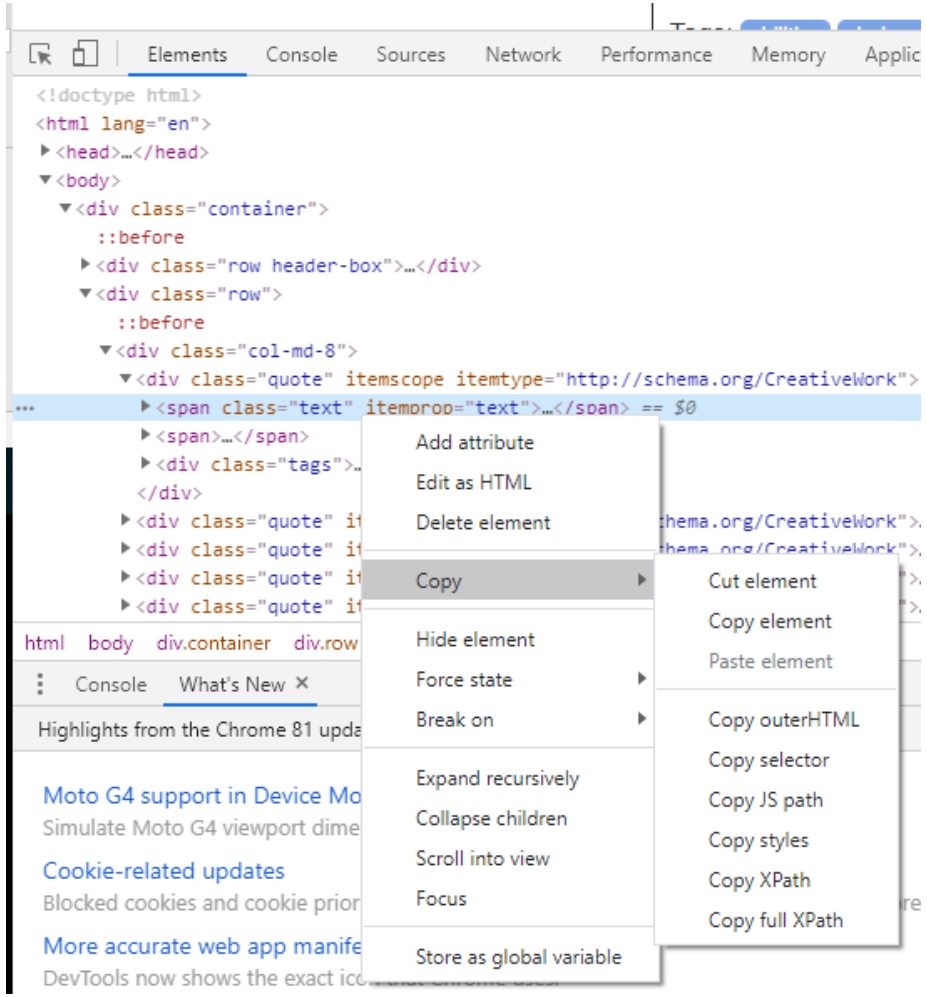

```
<div class="quote" (…)>
    <span class="text" (…)>
        "The world as we have created it (…). "
    </span>
    <span>
        by <small class="author" (…)>Albert Einstein</small> (…)
    </span>
    <div class="tags">(…)</div>
</div>

<div class="quote"(…)>
     <span class="text">"It is our choices, Harry (…)"</span>
     <span>
         by <small class="author" (…)>J.K. Rowling</small> (…)
     </span>
     (…)
</div>
(…)
<li class="next">
          <a href="/page/2/">Next (…)</a>
</li>
```

```
<div class="quote" (…)>

    <span class="text" (…)>
        "The world as we have created it (…). "
    </span>
    <span>
        by <small class="author" (…)>Albert Einstein</small> (…)
    </span>
    <div class="tags">(…)</div>
</div>

</div>
(…)
<li class="next">
        <a href="/page/2/">Next (…)</
</li>
```

```
def parse(self, response):
    for quote in response.css('div.quote'):
        yield {
            'text': quote.css('span.text::text').get(),
            'author': quote.xpath('span/small/text()').get(),
        }

    next_page = response.css('li.next a::attr("href")').get()
    if next_page is not None:
        yield response.follow(next_page, self.parse)
```

Smals ICT for society

- Principale difficulté : identification des tags
  - Souvent bas dans l'arborescence
  - Difficiles à rendre "génériques"
- Deux pistes :
  - Chrome Developper tools
  - Extension Chrome "webscraper.io"



Smals
ICT for society

- POC :
  - Collecter une liste de restaurants
  - Les lier aux infos KBO
- Le POC a été fait en ciblant Bruxelles
- Les données suivantes ont été collectées :
  - Nom(s) et adresse de l'entreprise
  - Numéro de téléphone → quasi que des "02" (fixes)
  - Nombre de visites/commentaires
- Pas de données "personnelles" (?)
- Données collectées sur Resto.be : 1.516 records

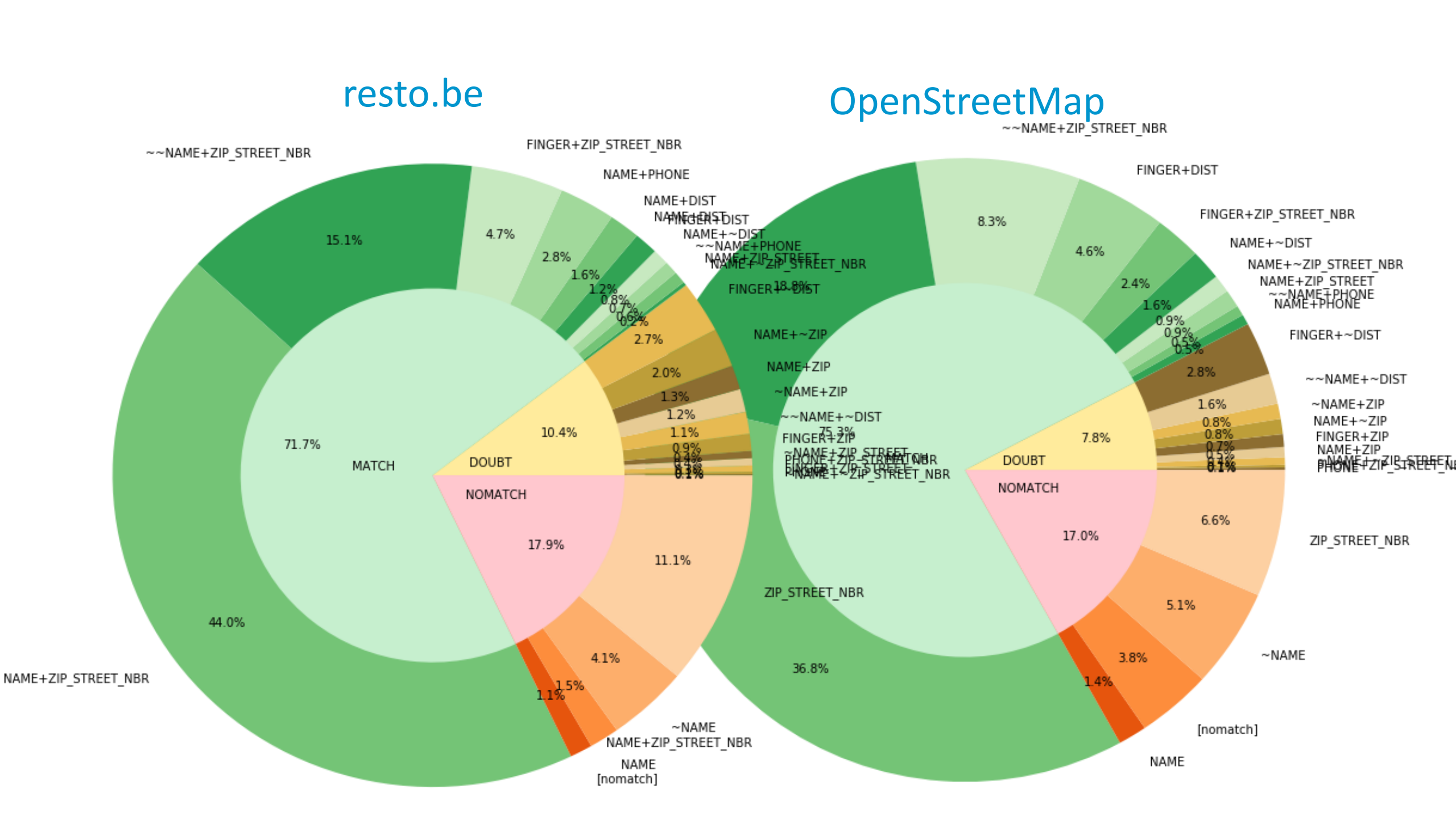
```
import scrapy

class RestoBeSpider(scrapy.Spider):
    name = "RestoBe"
    def start_requests(self):
        urls = ['https://en.resto.be/restaurant/brussels/']
        for url in urls:
            yield scrapy.Request(url=url, callback=self.parse_listpage)

    def parse_listpage(self, response):
        for resto_frame in  response.css("div.m-searchresult"):
            resto_url = resto_frame.css("a.cwb-restaurant-name
                                        link::attr(href)").get().split("?")[0]


            yield scrapy.Request(url=resto_url, callback=self.parse_restopage)
        next_page = response.css("li.m-pagination__button.cwb-next-page
                                                    a::attr(href)").get()
        if next_page:
            yield scrapy.Request(url=next_page, callback=self.parse_listpage)
```

```
def parse_restopage(self, response):
    name  = response.css("h1.m-detail-info__name a::text").get()
    street= response.css("div.m-detail-info__adress span::text").get()
    number= response.css("span.m-detail-info-adres__comma::text").get()
    city  = response.css("a.cwb-city-link::text").get()
    review_score = response.css("span.cwb-average-rating::text").get()

    yield {
       "name": name,
       "addr": {"street": street,
                "number": number,
                "city" : city },
       "review_score": review_score,
       "url": response.url,
     }
```

- On extrait de la KBO une liste “similaire” de
  - Noms (et variantes)
  - Téléphone
  - Adresses (toutes les UTE)
- Avec comme limitations :
  - NACE_CODE “root” = 5
  - Adresse sur ~ Bruxelles (code postal commence par 1)
  - → 410,000 lignes pour 140.000 entreprises
- Idéalement : on compare chaque ligne des deux listes (site web vs KBO)
- Dans la pratique : impossible → technique de “keying”

	Flag	Définition	Exemple
Dénomination	NAME	“Exactement” égal	“Domino Pizza” vs “domino pizza”
	FINGER	“Fingerprint” égal	… vs “Pizza Domino”
	~NAME	Similarité (jaro) ≥ 90%	… vs “Domino Pizzas”
	~~NAME	80 % < Similarité < 90 %	… vs “Dominique Pizzeria”
Adresse	ZIP_STREET_NBR	Correspondance complète	
	ZIP_STREET	Code postal + rue	
	ZIP	Code postal	
	DIST	Coord. géo. proches	
Tél	PHONE	Correspondance	“02 787 57 11” vs “+3227875711”

- **MATCH :**
  - NAME + PHONE
  - NAME + ZIP_STREET
  - ~~NAME + ZIP_STREET_NBR
  - …
- **DOUBT :**
  - ~NAME + ZIP_STREET
  - FINGER + ZIP_STREET
  - …
- **NOMATCH :**
  - ZIP_STREET_NBR
  - NAME
  - Le reste


resto.be
~~NAME+ZIP_STREET_NBR
FINGER+ZIP_STREET_NBR
NAME+PHONE
NAME+DIST
15.1%
4.7%
2.8%
1.6%
1.2%
0.8%
0.7%
0.6%
0.2%
2.7%
2.0%
1.3%
1.2%
1.1%
0.9%
71.7%
MATCH
10.4%
DOUBT
NOMATCH
17.9%
11.1%
44.0%
4.1%
1.5%
1.1%
NAME+ZIP_STREET_NBR
~NAME
NAME+ZIP_STREET_NBR
NAME
[nomatch]
OpenStreetMap
~~NAME+ZIP_STREET_NBR
FINGER+DIST
FINGER+ZIP_STREET_NBR
NAME+~DIST
NAME+~ZIP_STREET_NBR
NAME+ZIP_STREET
~~NAME+PHONE
NAME+PHONE
FINGER+~DIST
~~NAME+~DIST
~NAME+ZIP
NAME+~ZIP
FINGER+ZIP
NAME+ZIP
8.3%
4.6%
2.4%
1.6%
0.9%
0.9%
0.5%
0.5%
2.8%
1.6%
0.8%
0.8%
0.7%
0.5%
0.1%
0.1%
18.8%
NAME+~ZIP
NAME+ZIP
~NAME+ZIP
~~NAME+~DIST
FINGER+ZIP
75.3%
7.8%
DOUBT
NOMATCH
17.0%
6.6%
ZIP_STREET_NBR
5.1%
3.8%
1.4%
~NAME
[nomatch]
36.8%
ZIP_STREET_NBR
NAME

	A	B	C	D	L	N	O	T	U
1717					MATCH	NAME+ZIP_STREET_NBR	Atelier des Degres	Avenue du Prince de Ligne	14
1718	L'atelier En Ville	Rue Haute	64	1000	MATCH	NAME+ZIP_STREET_NBR	Atelier En Ville	Rue Haute	64
1719					MATCH	NAME+ZIP_STREET_NBR	L'ATELIER EN VILLE	Rue Haute	64
1720	L'atelier Européen	Franklinstraat	28	1000	MATCH	NAME+ZIP_STREET_NBR	L ATELIER EUROPEEN	Franklinstraat	28
1721					MATCH	NAME+ZIP_STREET_NBR	L'Atelier europeen	Franklinstraat	28
1722					MATCH	NAME+ZIP_STREET_NBR	L'ATELIER EUROPEEN	Franklinstraat	28
1723	L'atlantide	Rue Franklin	73	1000	MATCH	NAME+ZIP_STREET_NBR	L'ATLANTIDE	Rue Franklin	73
1724	L'auberge Des Maïeurs By La	Parvis Saint Pierre	1	1150	MATCH	~~NAME+ZIP_STREET_NBR	AUBERGE DES MAIEURS	Parvis Saint-Pierre	1
1725	Finca Sprl				MATCH	~~NAME+ZIP_STREET_NBR	L'Auberge des Maieurs by La Finca	Parvis Saint-Pierre	1
1726					MATCH	~~NAME+ZIP_STREET_NBR	L'AUBERGE DES MAIEURS	Parvis Saint-Pierre	1
1727	L'aurige	Avenue de Tervueren	24	1040	MATCH	NAME+ZIP_STREET_NBR	L'AURIGE	Avenue de Tervueren	24 A
1728	L'autre Cantina	Chaussée de Vleurgat	117	1000	DOUBT	FINGER+ZIP	LA CANTINA	Rue du Jardin des Olives	13
1729					DOUBT	FINGER+ZIP	LA CANTINA	Rue Antoine Dansaert	22
1730					DOUBT	FINGER+ZIP	LA CANTINA	Olivetenhof	13
1731					DOUBT	FINGER+ZIP	LA CANTINA	Rue Saint-Gery	31
1732					DOUBT	FINGER+ZIP	LA CANTINA	Rue Antoine Dansaert	22
1733					DOUBT	FINGER+ZIP	LA CANTINA	Antoine Dansaertstraat	22
1734					DOUBT	FINGER+ZIP	LA CANTINA	Antoine Dansaertstraat	22
1735					DOUBT	FINGER+ZIP	LA CANTINA	Sint-Goriksstraat	31
1736	L'autre Orfeo	rue Edith Cavell	15	1180	MATCH	NAME+ZIP_STREET_NBR	L'AUTRE ORFEO	Rue Edith Cavell	15
1743	L'ecailler Du Palais Royal	Rue Bodenbroek	18	1000	MATCH	NAME+ZIP_STREET_NBR	L'Ecailler du Palais Royal	Rue Bodenbroek	18
1744	L'echalote	Rue aux fleurs	8	1000	MATCH	~~NAME+ZIP_STREET_NBR	L ECHALOTTE	Rue aux Fleurs	8
1745	L'ecole D'a	Avenue Marie		1030	NOMATCH	[nomatch]			
1746	L'entree Des Artistes	Grote Zavel	42	1000	DOUBT	FINGER+ZIP	Les Artistes	Avenue de la Brabanconne	40
1747					DOUBT	FINGER+ZIP	Les Artistes	Brabanconnelaan	40
1748					DOUBT	FINGER+ZIP	CAFE LES ARTISTES	Place du Jeu de Balle	19
1749					DOUBT	FINGER+ZIP	CAFE LES ARTISTES	Vossenplein	19
1750	L'epicerie	Rue du Page	66	1150	MATCH	NAME+~ZIP_STREET_NBR	L'EPICERIE	Rue du Page	66
1751	L'equ	Avenue Henri		1140	NOMATCH	~NAME	EQUI	Langeweide	
1752					NOMATCH	~NAME	EQUI	Parvis Saint-Pierre	
1753					NOMATCH	~NAME	EQUI	Parvis Saint-Pierre	
1754					NOMATCH	ZIP_STREET_NBR	DISP	Av. H.	
1755					NOMATCH	~NAME	EQUI	Sint-Pietersvoorplein	1

Sheet1

Smals ICT for society

Disclaimer : je ne suis pas juriste !

- En général : va à l'encontre des CGU
- Exemple resto.be [1]: « Le contenu (…) est protégé par un droit sui generis qui (…) interdi[t] toute extraction et/ou réutilisation (…) de ce contenu. »
- Un problème ?
  - Pour certains : non tant qu'on ne clique pas sur « J'accepte les CGU » (jurisp. USA)
  - Pour d'autres : oui dès qu'on navigue sur le site web
- Selon [2], utilisation condamnée, pas le scraping
- Distinction à faire [3] :
  - Droit pénal : condamne l'introduction frauduleuse
  - Droit de la propriété intellectuelle/de la concurrence : condamme l'utilisation
- GDPR : Mon client en est-il exempt ? Qu'est-ce un donnée personnelle ?

[1] https://www.resto.be/privacy-policy
[2] https://islean-consulting.fr/fr/transformation-digitale/scraping-pages-web-legal/
[3] https://www.actualitesdudroit.fr/browse/tech-droit/start-up/9404/le-web-scraping-une-technique-d-extraction-legale

- Technique accessible, peu de moyens nécessaires
- Demande beaucoup de “bricolage” (surtout en “javascript-oriented”)
- Problème de pérénité : aucune garantie de “stabilité”
- Considérer les aspects juridiques/éthiques dès le début
- Collecter n’est pas finalité, il faut pouvoir exploiter les données

- POC :
  - Collecte : efficace pour l’horeca, secteur très “visible”
  - Reste à prouver pour des secteurs plus “underground” (nail shops, car washes…)
  - Linking : 70-75 % commerces trouvés dans la KBO avec haute fiabilité