Manuel Méthodologique

Projet individuel - Techniques Web

Camille REY
Master TAL - IM
2020/2021

Sujet : extraction d'information depuis applications MPA/SPA et visualisation des données sur une application WEB

Lien vers l'application web
Lien vers le dépôt GitHub

Le but de ce projet est d'appliquer les techniques d'extraction de données depuis le web vues en cours. Sur la base de deux sites imposés, nous devons extraire des données et les mettre en valeur pour convaincre des investisseurs (imaginaires) de soutenir deux projets : un outil de recherche de logements écologiques et une plateforme de dictionnaires collaboratifs en ligne pour les langues peu dotées. Un troisième site de choix libre peut également être scrapé pour compléter le projet.

I. Logements écologiques - NH Hotel 4

1.Choix des données 4

2.Extraction des données 5

a.Vérification des conditions 5

b.Extraction des différentes données 5

3.Visualisation des données 10

a.Présentation - carte interactive 10

b.Diagrammes et statistiques 11

c.Petit moteur de recherche 12

II. Dictionnaires collaboratifs - NTeaLan / Leo Dict 13

1.Choix des données 13

2.Extraction des données 14

a.Vérification des conditions 14

b.Extraction des entrées du dictionnaire 14

c.Extraction du nombre d'entrées pour chaque dictionnaire 16

d.Plateforme de comparaison : LEO 17

3.Visualisation des données 18

a.Présentation d'un article pour chaque dictionnaire 18

b.Statistiques sur les données des entrées 19

c.Comparaison avec statistiques de LEO 19

III. Détails techniques et installation/lancement 19

1.Installation 19

2.Lancement 20

a.Lancement de l'application Web 20

b.Lancement des scripts d'extraction 21

I. Logements écologiques - NH Hotel

1. Choix des données

Pour mettre en valeur les données de la plateforme NH Hotel, et plus précisément le potentiel des logements écologiques, j'ai décidé d'extraire des informations spécifiques. Pour tous les hôtels (357 hôtels) de la plateforme, j'ai extrait les informations suivantes :

- nom de l'hôtel
- ville
- pays
- continent
- lien
- longitude
- latitude
- nombre de chambres
- nombre d'étoiles
- note moyenne
- nombre d'avis laissés
- certificat ISO (possède ou non)
- mention Green Leader (possède ou non)
- mention eco-friendly (possède ou non)

Ces informations seront exploitées dans la partie « 3. Visualisation », et leur choix justifié.

2. Extraction des données

a. Vérification des conditions

Avant de procéder à l'extraction des données, j'ai scrupuleusement vérifié que la plateforme autorisait l'extraction des informations qui m'intéressaient. Pour cela j'ai vérifié dans le protocole d'exclusion des robots du site que les ressources depuis lesquelles je souhaitais extraire des informations n'étaient pas en accès non-autorisé. J'ai également vérifié les conditions d'utilisation du site, pour m'assurer de ne pas les enfreindre.

b. Extraction des différentes données

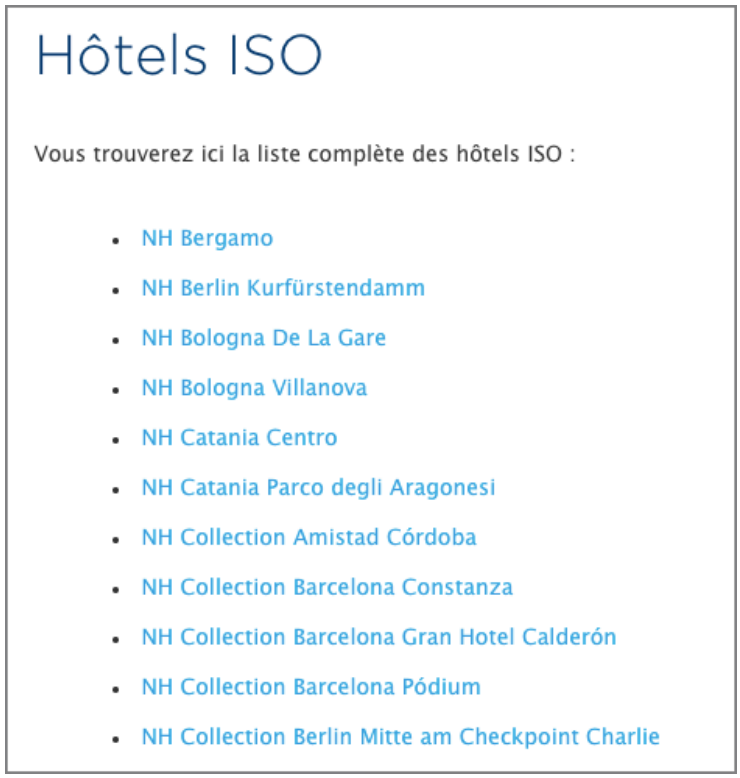
Le site web de NH Hotel Group est un MPA (Multi Page Application), j'ai effectué les extractions grâce aux outils request pour effectuer des requêtes GET, et BeautifulSoup pour parser les réponses.

J'ai exploité **4 URLs** comme points de départ pour scraper :

- https://www.nh-hotels.fr/hotels : contient des liens vers les listes d'hôtels par pays, groupés par continents.

Les trois listes suivantes ont été trouvées à partir de la section « Hôtels écologiques développement durable » du site.

- https://www.nh-hotels.fr/environnement/hotels-ecologiques-developpement-durable/iso-hotels : la liste des hôtels possédant un certificat d'énergie durable ISO.



Hôtels ISO

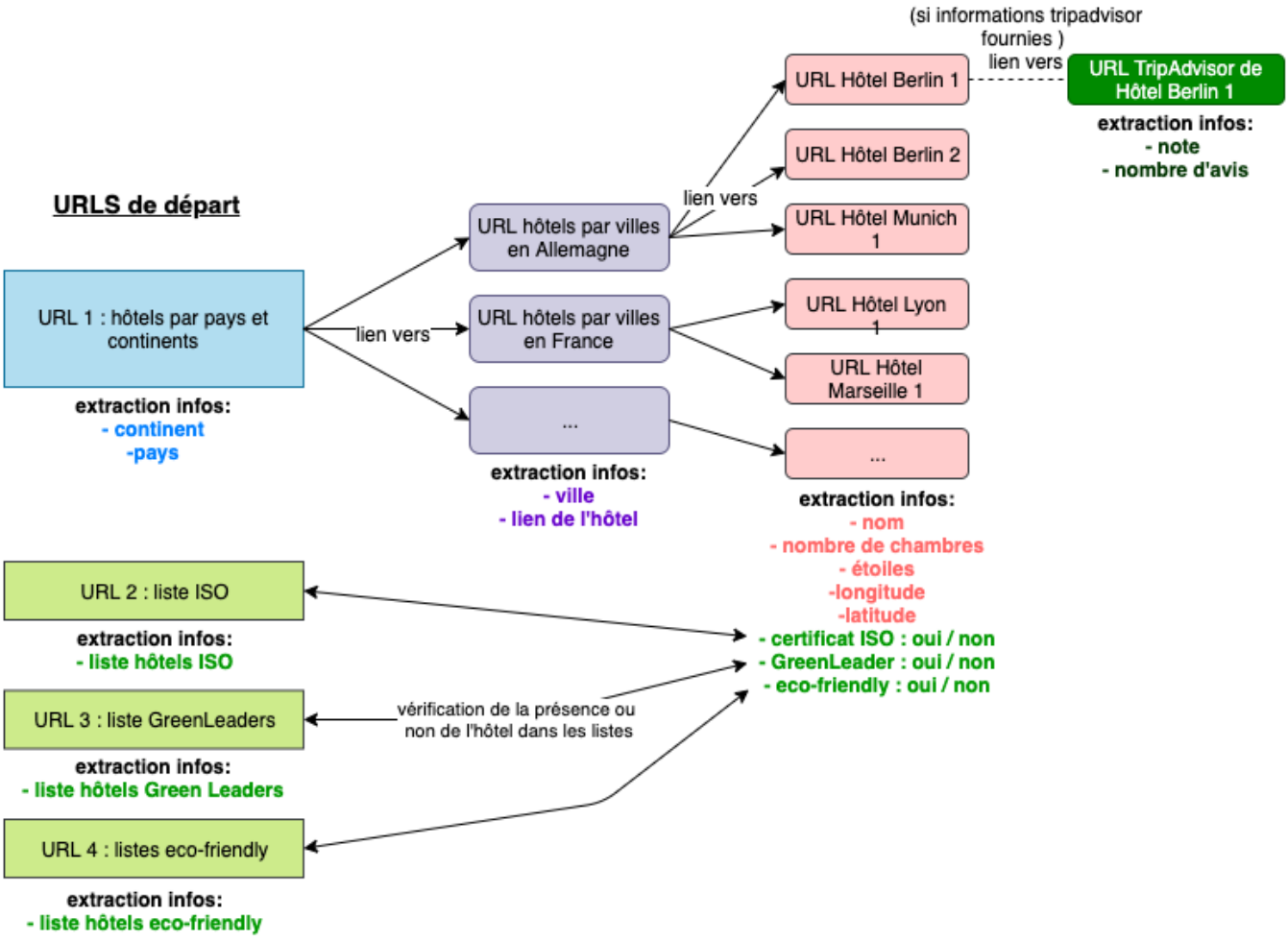
Vous trouverez ici la liste complète des hôtels ISO :

- NH Bergamo
- NH Berlin Kurfürstendamm
- NH Bologna De La Gare
- NH Bologna Villanova
- NH Catania Centro
- NH Catania Parco degli Aragonesi
- NH Collection Amistad Córdoba
- NH Collection Barcelona Constanza
- NH Collection Barcelona Gran Hotel Calderón
- NH Collection Barcelona Pódium
- NH Collection Berlin Mitte am Checkpoint Charlie

- https://www.nh-hotels.fr/environnement/hotels-ecologiques-developpement-durable/green-leaders-hotels : la liste des hôtels possédant la mention Green Leader décernée par TripAdvisor.

- https://www.nh-hotels.fr/environnement/hotels-ecologiques-developpement-durable/eco-friendly-hotels : la liste des hôtels possédant la mention eco-friendly.

A partir de la première URL, on peut accéder aux listes d'hôtels par ville, et aux URLS individuelles des hôtels, à partir desquelles on peut extraire les informations spécifiques à chaque hôtel. On rajoute ensuite les informations ISO/GreenLeader/EcoFriendly à partir des listes extraites des 3 URLs de listes présentées ci-dessus. La méthode peut être résumée par ce schéma :



Sortie obtenue (exemple) :

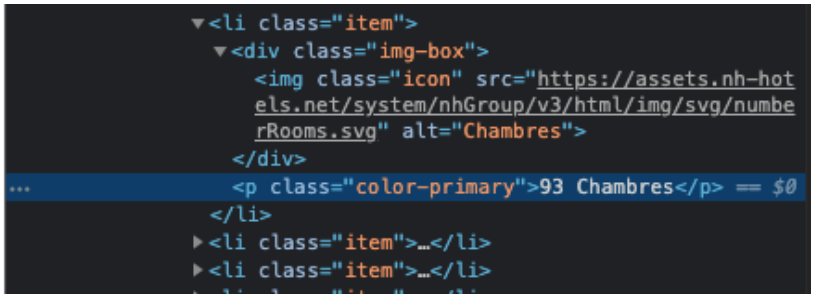
nom	nombre chambres	étoiles	longitude	latitude	note	nombre avis	ISO	greenLeader	EcoFriendly	lien	ville	pays	continent
Berlin 1	200	4	6666	6666	4.5	230	oui	non	oui	https://...	Berlin	Allemagne	Europe
Berlin 2	150	5	6665	6665	4	110	non	non	non	https://...	Berlin	Allemagne	Europe
...	...	...	...	...	...	...	...	...	...		...	...	...

Pour extraire les différentes informations dans chaque page, j'ai d'abord inspecté le code source avec l'inspecteur de Chrome afin d'appréhender la structure générale de la page et d'identifier les noeuds à extraire pour chaque information. Les extractions sont faites principalement grâce à la fonction .find(), en filtrant par nom de balise et attributs, parfois couplé à des expressions régulières pour extraire uniquement une partie du texte. Certaines données telles que le nombre de chambres où les avis ne sont pas nécessairement renseignées pour tous les hôtels : leur extraction est gérée dans des try/except. Ci-dessous quelques exemples d'extraction de données :

- extraction du nombre de chambres :

page :

code source :



extraction :

```
# extraction nombre de chambres
try :
    nb_rooms = parsed_content.find("img", {"alt": "Chambres"}).findNext('p').text
    nb_rooms = int(nb_rooms.replace(" Chambres", ""))
except :
    nb_rooms = ""
```

- extraction de la longitude et la latitude :

code source :

```
▸<div class="modal fade modal-hotel-map" id="modal-hotel-map-detail" tabindex="-1" role="dialog"
aria-labelledby="hotel map">…</div>
</section>
▾<script type="text/javascript">
  // SET MARKER ARRAY PARA MAPA
  markersArray = [];
  //Set HOTEL Data for Google Maps




      // JS ADD POI HOTEL
      markersArray.push({
          location : [parseFloat('33.896404'),parseFloat('8.079288')],
          icon : Pines.hotel['Anantara'],
          info : "<div class='thum-box-maps'><img width='136' src='https://img.nh-
hotels.net/ananatara_sahara_tozeur_resort_villas-033-views.jpg?output-
quality=70&resize=136:136&composite-to=center,center|136:136&background-color=white' ><div>
<b>Anantara Sahara Tozeur Resort & Villas</b></div></div>"
      });
```

extraction :

```
# extraction longitude et latitude
script_map = parsed_content.find("div",{"id" : "modal-hotel-map-detail"}).findNext("script").string
latitude = re.findall(r"location : \[parseFloat\('([^/']*)", script_map)[0]
longitude = re.findall(r",parseFloat\('([^/']*)", script_map)[0]
```

Les seules informations qui n'étaient pas directement accessibles depuis le code source de la page d'un hôtel étaient les informations de notes et le nombre d'avis fournis par le widget TripAdvisor. Pour les récupérer, j'ai extrait l'id TripAdvisor de l'hôtel depuis le code source de sa page, puis créé une fonction qui scrape la page TripAdvisor de cet id pour extraire la note et le nombre d'avis, après avoir vérifié le robot.txt de TripAdvisor. Cette fonction s'appelle *extract_reviews_info()* et sert uniquement d'intermédiaire pour récupérer les informations d'avis qui sont disponibles sur la page de l'hôtel mais pas dans le code source.

L'intégralité de la procédure d'extraction est consultable dans le script python d'extraction : *scripts_extraction/extraction_hotels.py* . Le fichier de sortie produit est généré par défaut dans le même répertoire que le script python, et est au format csv.

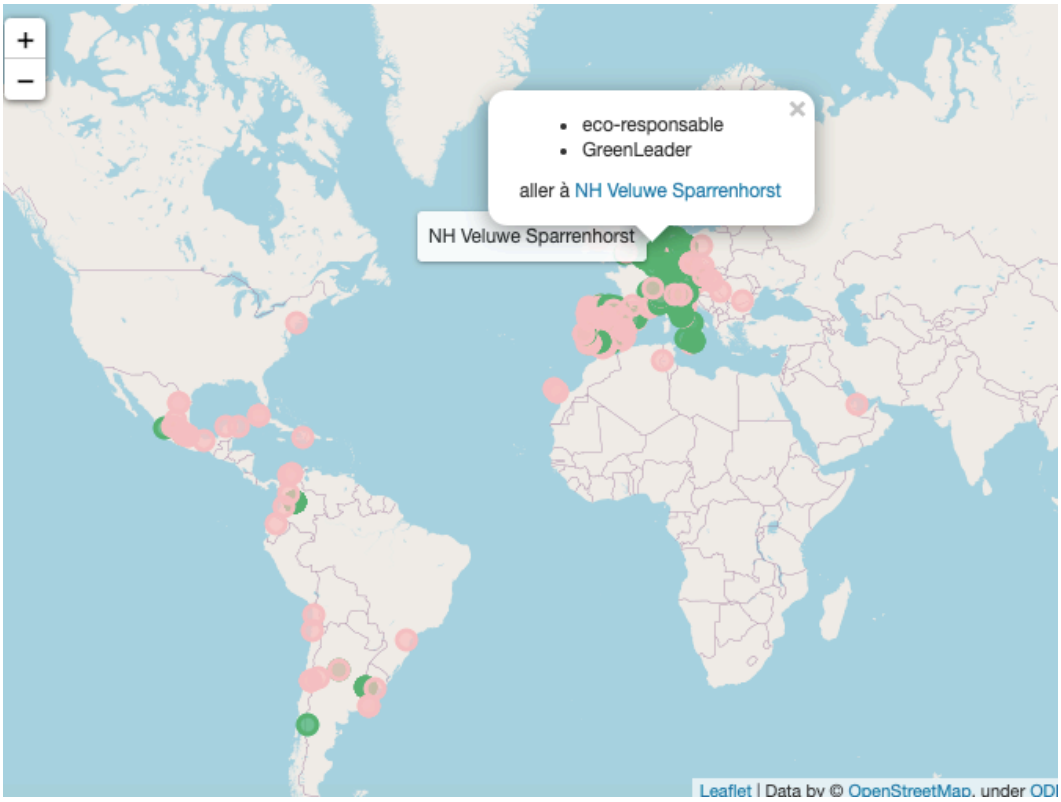
3. Visualisation des données

La présentation se fait grâce à l'outil Streamlit, qui permet de mettre facilement au point une application web interactive à partir de code python, et est compatible avec de nombreuses autres librairies. La visualisation des données pour les logements écologiques se fait dans la rubrique « Tourisme écologique ».

a. Présentation - carte interactive

Pour présenter la répartition des hôtels et montrer qu'une grande partie d'entre eux sont éco-responsables, j'ai décidé de les représenter sur une carte du monde interactive, avec des codes de couleurs, des informations complémentaires sur les mentions/certifications écologiques, et des liens vers les pages des différents hôtels. Pour faire cela, j'ai exploité les **informations de longitude et de latitude** extraite dans la partie précédente. J'ai d'abord utilisé l'outil st.map() intégré à streamlit, qui est très simple d'utilisation, mais également limité car très peu personnalisable. Résultat avec st.map :

J'ai donc préféré utiliser la librairie folium, compatible avec streamlit, qui offre davantage d'options. Résultat final avec folium.map() :



b. Diagrammes et statistiques

Les autres informations sur les hôtels extraites sont ensuite exploitées à travers diverses statistiques pour mettre en valeur le potentiel des logements écologiques :

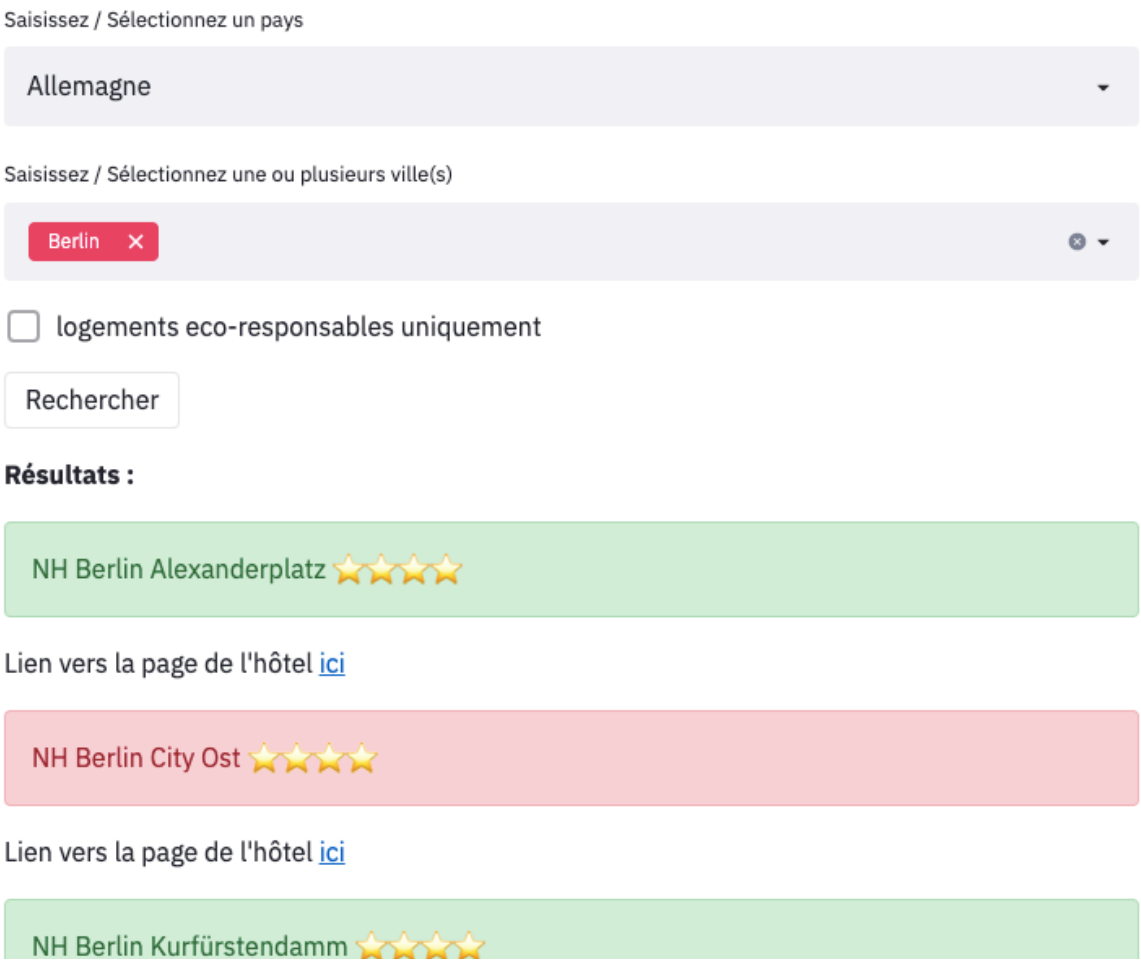
- proportions de mentions eco-responsables / GreenLeader / ISO
- proportions de logements éco-responsables / non éco-responsables par pays, par continent, par catégorie (nombre d'étoiles)
- nombre de chambres moyen en fonction des hôtels
- note moyenne en fonction des hôtels
- nombre d'avis moyen en fonction des hôtels

Ces diagrammes ont été générés grâce aux diverses options de la librairie Plotly. Les diagrammes obtenus et leur analyse détaillée sont consultables sur l'application web.

c. Petit moteur de recherche

Dans la sous-rubrique de conclusion, pour l'aspect ludique, j'ai présenté un embryon d'outil de recherche de logements écologiques sur la base des données NH Hotel extraites. L'utilisateur peut saisir chaque champ ou le sélectionner depuis un menu déroulant, les valeurs saisissables sont limitées par les valeurs existant dans la base de données. Un filtre par critère d'éco-responsabilité peut être appliqué. Les résultats sont ensuite affichés sous la forme d'une liste de noms d'hôtels, accompagnés du nombre d'étoiles et du lien vers la page web. Un code couleur vert/rouge permet de différencier les logements éco-responsables de ceux qui ne le sont pas (si le critère d'éco-responsabilité n'est pas appliqué comme filtre de recherche).

Exemple :



II. Dictionnaires collaboratifs - NTeaLan / Leo Dict

1. Choix des données

Pour présenter et mettre en valeur le potentiel de la plateforme de dictionnaires collaboratifs en ligne NTeALan, j'ai choisi d'extraire les 100 première entrées des 17 dictionnaires opérationnels sur le site (certains sont en maintenance). Pour chaque entrée, j'ai extrait les informations suivantes :

- entrée complète
- radical, préfixe, suffixe
- variants (nb + variants)
- type, forme
- catégorie grammaticale
- informations de classe
- informations de conjugaison
- traductions en français (nb + traductions)
- traductions en anglais (nb + traductions)
- exemples en français (nb + exemples)
- exemples en anglais (nb + exemples)
- présence ou non de fichier audio lié
- dictionnaire dont l'entrée est issue

J'ai voulu à l'origine extraire des données sur la base de filtres de recherche précis (comme une liste de mots clés courants pour voir si ils apparaissaient dans les traductions/exemples des entrées d'un dictionnaire), mais la fonction de recherche ne semble pas fonctionner (ou du moins pas sur ma machine) pour la plupart des dictionnaires, j'ai donc abandonné l'idée. Le nombre d'entrées extraites comme base pour les analyses est arbitrairement fixé à 100 car certains dictionnaires semblent présenter moins de 150 entrées (Eton-Français). J'ai enfin tenté d'extraire le **nombre d'entrées** pour chaque dictionnaire.

2. Extraction des données

a.Vérification des conditions

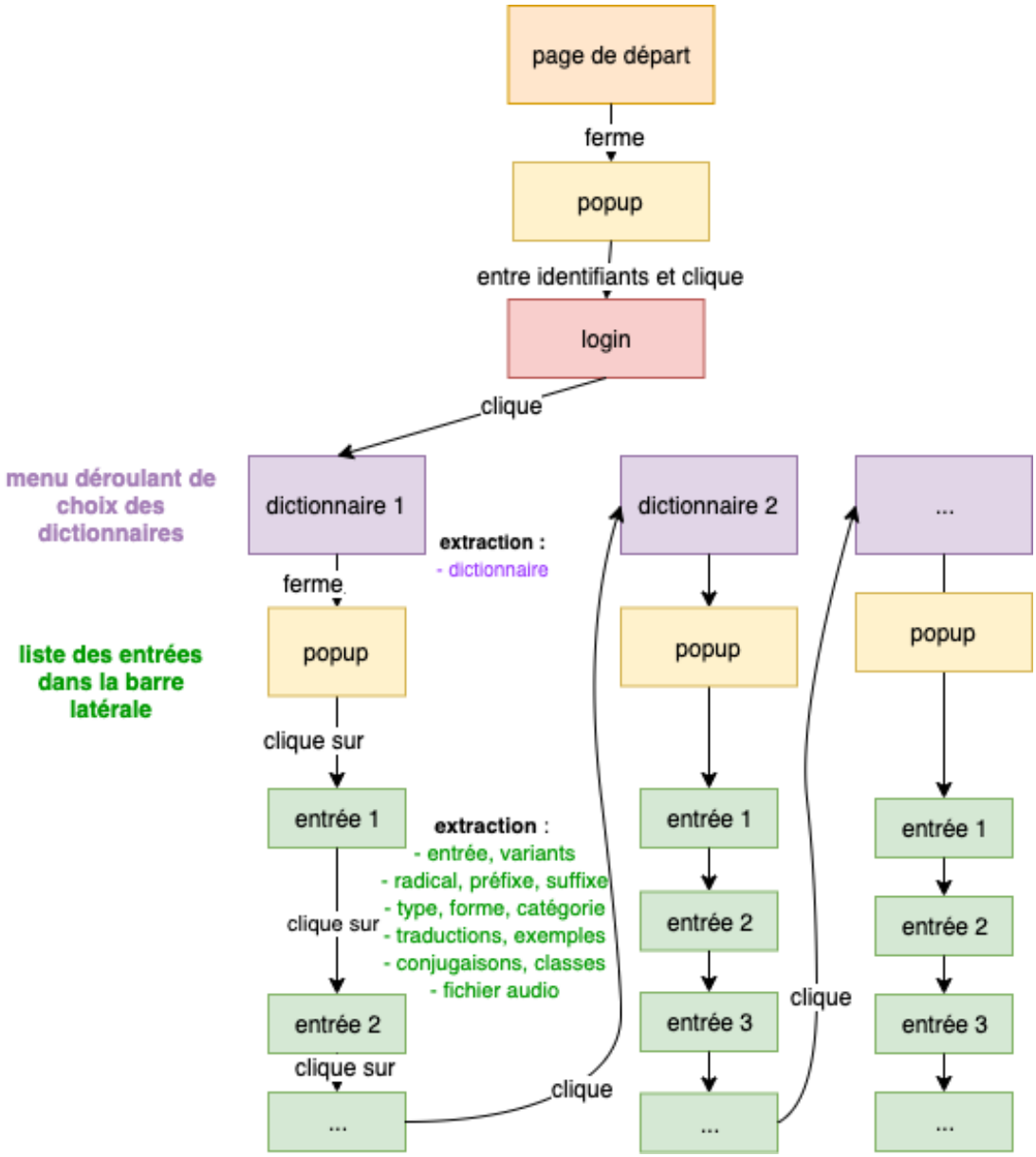
Avant de procéder à l'extraction des données, j'ai scrupuleusement vérifié que la plateforme autorisait l'extraction des informations qui m'intéressaient. Pour cela j'ai vérifié dans le protocole d'exclusion des robots de la plateforme que les ressources depuis lesquelles je souhaitais extraire des informations n'étaient pas en accès non-autorisé.

b. Extraction des entrées du dictionnaire

La plateforme de dictionnaires NTeALan est une SPA (Single Page Application). Il n'est donc pas possible d'extraire les informations directement à partir du code source de la page comme pour le site de NH Hotel Group. J'ai utilisé l'outil selenium pour ouvrir une session Chrome, et récupéré les données à extraire en cliquant sur différents éléments pour afficher les informations recherchées. Pour déterminer les clés (classes, identifiants, noms de balises etc…) permettant d'identifier les éléments à extraire/ sur lesquels cliquer, j'ai utilisé une fois de plus l'inspecteur de mon navigateur Google Chrome.

La plateforme NTeALan présente plusieurs challenges : il faut se connecter pour accéder à beaucoup de données, et il faut fermer une fenêtre pop-up d'informations sur le covid à chaque ouverture de dictionnaire.

La méthode utilisée pour extraire les informations des 100 premières entrées des 17 dictionnaires opérationnels de la plateforme peut être résumée par le schéma suivant :



La sortie générée est au format csv, pour pouvoir être facilement exploitable en tant que DataFrame Pandas avec Plotly. Pour les informations de type « liste » (les contenus de traductions, exemples, variants, conjugaisons), un problème se pose : une cellule ne peut contenir qu'une seule valeur. J'ai décidé de contourner cette limite en transformant les valeurs de type « liste » en une chaîne de caractère, où chaque valeur est séparée par un délimiteur défini : « @@@ ». De même, un délimiteur « <–> » est inséré pour séparer les exemples en langue cible et langue source. Extrait d'un rang en sortie :

entree	cat	nb trad fr	traductions fr	nb ex fr	exemples fr	nb conj	conjugaisons
búlá	Verbe	2	quitte@@@quittez	2	búlá: pìā: <--> quitte là-bas@@@búlágá pìā: <--> quittez là-bas	2	impératif présent : búlá@@@impératif présent : búláká

Le parcours de la plateforme avec selenium implique de prendre en compte les temps de chargement. Pour cela, j'ai utilisé la fonction implicitly_wait(), couplé à des time.sleep() quand nécessaire.

Toutes les entrées n'ont pas forcément le même format, certaines présentent des variants, des préfixes, suffixes, informations de conjugaison, de classe etc… d'autres non (il arrive également que des entrées n'aient pas de traduction associée). Pour gérer ces cas, j'ai utilisé des blocs try/except.

Enfin il arrive, de manière qui semble assez aléatoire (et différente à chaque lancement du script), que quelques entrées génèrent des erreurs StaleException. Pour gérer ces potentiels cas, j'ai également utilisé les blocs try/except.

Pour identifier les différents éléments, j'ai tâché de toujours privilégier la recherche par CSS Selector, apparemment plus rapide. J'ai parfois utilisé des recherches par Xpath.

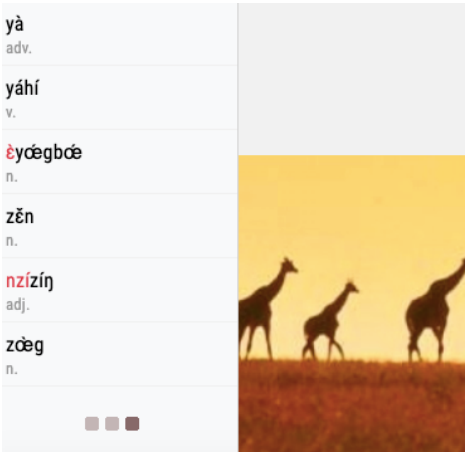
Les options du browser selenium sont réglées par défaut sur « headless », pour ne pas ouvrir de fenêtre, car j'ai remarqué des problèmes si on réduit la fenêtre ouverte par selenium au cours de l'exécution. Il semble donc plus sûr de ne pas ouvrir de fenêtre (cette option peut être modifiée au lancement en ligne de commande, cf partie « Détails techniques et installation/lancement »).

Le script d'extraction *scripts_extraction/extraction_ntealan.py* contient l'ensemble des opérations d'extraction des informations d'entrées de la plateforme NTeALan, organisées en multiples fonctions. La sortie est générée par défaut dans le même répertoire que le script.

c. Extraction du nombre d'entrées pour chaque dictionnaire

Pour extraire le nombre d'entrées pour chaque dictionnaire, j'ai d'abord essayé de scroller les listes d'entrées de la barre latérale avec selenium jusqu'à en arriver au bout. Malheureusement, après avoir essayé de multiples techniques, je n'ai pas réussi à déclencher le chargement des entrées au delà

des 200 premières avec selenium. J'obtenais uniquement l'icône de chargement :



J'ai donc finalement décidé d'interroger directement l'API NTeALan pour extraire le nombre d'entrées pour chaque dictionnaire. L'API propose une méthode « Show all articles from dictionaries », avec un paramètre de limite. Le problème est que le serveur renvoie une erreur 502 si ce paramètre de limite est trop élevé (au delà de 20 000). J'ai donc divisé la tâche d'extraction en utilisant la méthode « Show all articles » qui prend en paramètre l'id d'un dictionnaire et renvoie ses entrées, et ai ensuite compté ces entrées. J'ai fixé la limite à 15 000 entrées, ce qui était suffisant puisque le dictionnaire avec le plus grand nombre d'entrées en compte moins de 12 000.

La fonction qui s'occupe d'extraire le nombre d'entrées pour chaque dictionnaire opérationnel de la plateforme NTeALan est la fonction *count_entries_ntealan()* dans *scripts_extraction/extraction_nb_entries.py*

d. Plateforme de comparaison : LEO

Afin de comparer les chiffres de NTeALan avec ceux d'une plateforme plus grosse, et de présenter des perspectives de développement, j'ai extrait quelques informations statistiques depuis la plateforme LeoDict. J'ai bien entendu vérifié dans le robots.txt et dans les conditions d'utilisation de la plateforme que j'avais l'autorisation d'extraire ces données.

Depuis la page de chacun des 9 dictionnaires proposés par la plateforme LEO, j'ai extrait les informations de nombre d'entrées et de nombre de requêtes journalières. Ces nombres sont générés dynamiquement, j'ai donc du utiliser selenium pour les extraire.

La fonction qui s'occupe d'extraire le nombre d'entrées et de requêtes pour chaque dictionnaire de la plateforme LEO est la fonction *count_entries_leo()* dans scripts_extraction/extraction_nb_entries.py

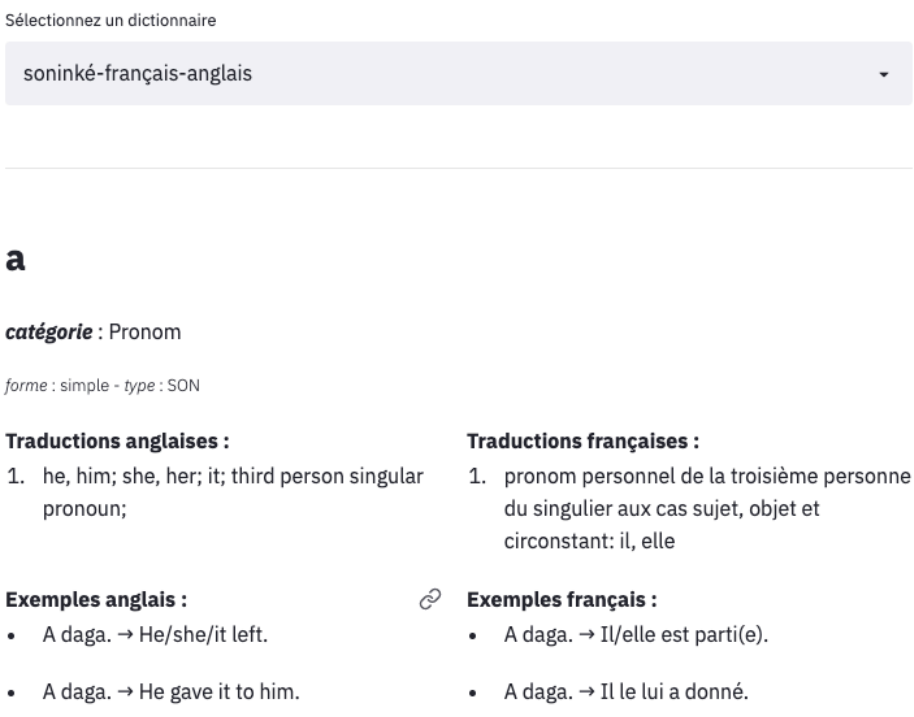
3. Visualisation des données

La visualisation des données se fait dans la rubrique « Dictionnaire collaboratif » de l'application web créée avec Streamlit.

a. Présentation d'un article pour chaque dictionnaire

Pour présenter la plateforme NTeALan, j'ai décidé de laisser l'utilisateur choisir un dictionnaire pour en afficher un article. Pour chaque dictionnaire, j'ai moi-même sélectionné une entrée jugée « optimale » pour la mise en valeur du dictionnaire parmi les entrées récupérées.

Exemple pour le dictionnaire soninké-français-anglais :



b. Statistiques sur les données des entrées

Afin de mettre en valeur la richesse de la plateforme NTeALan et des options proposées pour enrichir une entrée, j'ai présenté quelques statistiques à partir du nombre de traductions, d'exemples, des informations de conjugaisons ou de classes, et de la présence de fichiers audio. Ces statistiques sont présentées par des diagrammes générés avec plotly.

Les diagrammes et leurs analyses sont consultables sur l'application web.

c. Comparaison avec statistiques de LEO

Dans la section « perspectives d'évolution », je présente les nombres d'entrées par dictionnaire pour la plateforme NTeALan, et les compare à ceux de la plateforme LEO pour montrer les perspectives de développement d'une plateforme de dictionnaires collaboratifs. Je présente ensuite le nombre de requêtes journalières par dictionnaire pour la plateforme LEO, afin d'insister sur le caractère attractif des dictionnaires en ligne quand ils sont bien développés.

III. Détails techniques et installation/lancement

Ces étapes sont également détaillées dans le README.md, visualisable sur la page d'accueil du GitHub (de manière un peu plus lisible)

1. Installation

AVEC PIPENV (recommandé) :

L'installation requiert d'avoir installé pipenv sur son système.

Une fois pipenv installé, placez vous dans la racine du répertoire du projet, puis lancez le script d'installation :

```
sh setup.sh
```

Ce script créera un environnement virtuel et y installera les dépendances du fichier Pipfile. Il vous demandera également de renseigner le chemin de votre driver Chrome (assurez vous d'avoir téléchargé le driver compatible avec votre

version de Chrome), ainsi que vos identifiants d'authentification pour la plateforme NTeALan. Vous pouvez omettre de renseigner ces 3 informations en appuyant sur la touche ENTREE : vous pourrez alors lancer l'application web en local, mais vous ne pourrez pas lancer les scripts d'extraction.

SANS PIPENV :

Si vous ne souhaitez pas utilisez d'environnement virtuel, ou la commande pipenv, vous pouvez manuellement installer les dépendances manuellement à partir du requirements.txt :

```
pip install -r requirements.txt
```

Si vous souhaitez pouvoir lancer les scripts d'extraction, il vous faudra créer un fichier **.env** dans le répertoire *scripts_extraction/* et y écrire les informations nécessaires sous le format suivant :

DRIVER_PATH = chemin_vers_votre_driver_chrome

USERNAME = Votre_pseudo_Ntealan

PASSWORD = Votre_mot_de_passe_Ntealan

2. Lancement

a. Lancement de l'application Web

Après installation avec setup.sh :

Il suffit de lancer le fichier run.sh :

```
sh run.sh
```

Sans installation avec setup.sh :

Lancez le fichier app_visualisation.py avec streamlit :

```
streamlit run app_visualisation.py
```

L'application web sera alors lancée en local à l'adresse indiquée sur le terminal.

b. Lancement des scripts d'extraction

Après installation avec setup.sh :

Depuis le répertoire racine du projet, activez l'environnement virtuel :

```
pipenv shell
```

Sans installation avec setup.sh :

Assurez vous d'avoir bien installé préalablement les dépendances nécessaires dans votre environnement actuel, et créé le fichier .env dans le répertoire scripts_extraction/

Placez vous ensuite dans le répertoire scripts_extraction/ :

- extraction des informations des hôtels NH Hotel Group :

```
python extraction_hotels.py
```

La sortie « hotels.csv » sera automatiquement générée dans le répertoire courant.

- extraction des informations des d'entrées NTeALan :

```
python extraction_ntealan.py [-w true] [-e 150]
```

Des arguments optionnels peuvent être passés :

-w true : une fenêtre sera ouverte par selenium

-e un_entier_entre_1_et_200 : le nombre d'entrées à extraire par dictionnaire (100 par défaut)

La sortie « ntealan_entrees.csv » sera automatiquement générée dans le répertoire courant.

- extraction des nombres d'entrées/requêtes pour NTeALan et LEO :

```
python extraction_nb_entries.py [-w true]
```

-w (window) true : une fenêtre sera ouverte par selenium

La sortie « dic_nb_entries.csv » sera automatiquement générée dans le répertoire courant.