
Outils d'Analyse Sémantique

Manuel d'utilisation et de développement

Membres:

Killian MARCHAND
Romane BUON

Responsables :

Nathalie CAMELIN
Antoine LAURENT

Table des matières

I - Installation de l'application	2
Le projet	2
I.A - Nuage de mots	2
I.B - Le site	2
II - Identification	3
II.A - Connexion	3
III - Utilisation avec fichiers	3
IV - Utilisation en tant qu'administrateur	3
V - Développement	4
V.A - Nuage de mots	5
V.B - Application web	5

I - Installation de l'application

I.A - Le projet

```
# git clone  
git clone https://github.com/kilmhd/Outil_Analyse_Semantique.git
```

I.B - Le nuage de mots

I.A.1 - Flask

```
# install from pipi  
pip install Flask
```

I.A.2 - word_cloud

```
# install from pipi  
pip install wordcloud
```

I.A.3 - Exécution

```
# exec wordCloud service web in the myWordCloud folder  
python main.py
```

I.C - Le site

Il faut d'abord mettre le dossier "applicationWeb" du projet dans le dossier "var/www/html/" de votre serveur.

Exécution chmod:

```
# modification droit du fichier json  
chmod 644 applicationWeb/src/json/setting.json
```

Il suffit ensuite de se rendre sur: `votreNomDeDomaine/applicationWeb/index.html`.
Puis dans la zone de texte entrez votre Token pour accéder à vos fichiers.

II - Identification

Pour vous connecter, il suffit d'entrer votre token personnel pour accéder au service web de transcription. Ce dernier vous permettra d'accéder à vos fichiers sur le serveur via cette application.

III - Utilisation avec fichiers

L'utilisateur choisit le fichier audio qu'il souhaite transcrire. En cliquant sur le bouton "transférer" ce fichier est envoyé au serveur qui va le transcrire.

Une fois la transcription du fichier terminée, l'utilisateur a le choix entre deux modes de visualisation :

- le mode complet, où la totalité du texte est directement affichée à l'écran
- le mode suivi, où le texte apparaît en fonction de l'avancement du fichier audio

En parallèle, le nuage de mots est généré en fonction du mode choisi précédemment, sur le même modèle que la transcription.

Une fois la transcription effectuée, l'utilisateur peut corriger cette dernière (cf. bouton "correction"). Une fois corrigée et sauvegardée, il peut cocher la case pour sélectionner le fichier de correction et cliquer de nouveau sur le bouton "transcrire", ce qui chargera le fichier de transcription corrigé.

IV - Utilisation en tant qu'administrateur

Un utilisateur possédant le statut d'administrateur de l'application a la possibilité d'accéder aux paramètres. Ce qui lui permet de gérer les adresses des différents modules utilisés sur la page principale.

V - Développement

Le code source du projet se décompose en plusieurs parties, chacune illustrée par un dossier ou un sous-dossier.

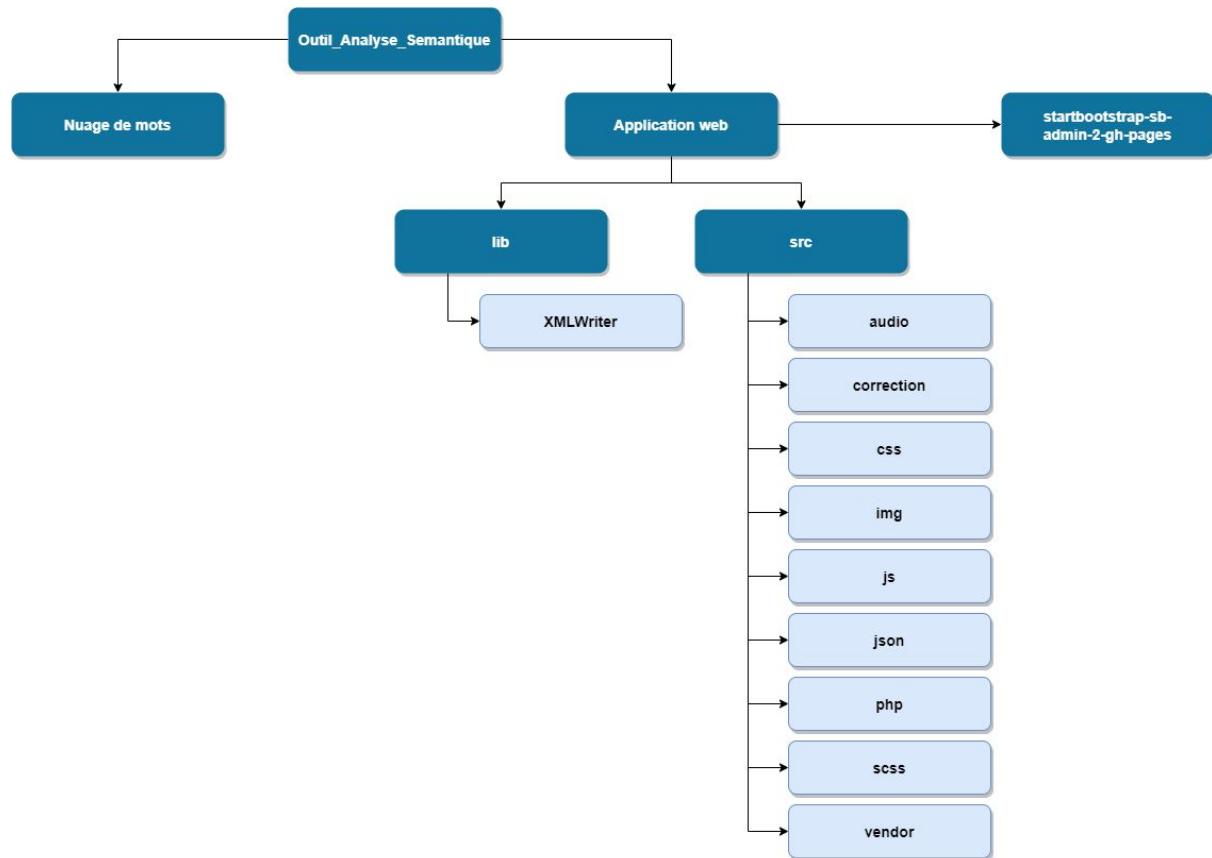


Figure 5.1 - Arborescence de fichiers du projet

Le dossier principal, *Outil_Analyse_Semantique*, contient trois sous-dossiers ainsi que plusieurs fichiers.

Le fichier *index.html* contient le code source HTML de la page web du projet.

Le dossier *startbootstrap-sb-admin-2-gh-pages* contient tous les fichiers CSS et JavaScript provenant du template Bootstrap et permettant la mise en page et la présentation de la page web.

V.A - Nuage de mots

MyWordCloud est la partie du projet concernant le nuage de mots en tant que service web. Cette application se décompose en trois fichiers sources.

- *Blacklist.py* permet de gérer une blacklist de mots pour enlever les mots perturbateurs du nuage de mots. Nous retrouvons cette blacklist dans un fichier texte (*blacklist.txt*), qui est construit avec un mot par ligne.
- *myCloud.py* est le fichier qui fait appel à l'API *word_cloud*¹. C'est ici que le nuage de mots est généré².

Nous sauvegardons une image du nuage de mots dans le dossier *myclouds*. Attention, cette image est écrasée à chaque génération du nuage de mots.

- *main.py* est le fichier utilisant Flask. C'est ici que se trouve la fonction qui gère l'appel au service web, envoie le traitement de la liste de mots, puis retourne l'image du nuage de mots.

V.B - Application web

Le dossier *lib* contient le code source de l'API *XMLWriter* qui permet d'écrire dans un fichier XML de façon automatique.

Le dossier *src* contient tout le code source du projet. Ce dossier est divisé en plusieurs sous-dossiers :

- *audio* contient le fichier audio à transcrire
- *correction* contient les fichiers de transcription corrigés
- *css* contient les fichiers CSS
- *img* contient les différentes images utilisées dans le service web
- *js* contient tous les fichiers JavaScript utilisés pour faire fonctionner le projet :
 - *APIFunction.js* gère les différentes fonctions des API utilisées
 - *button.js* gère les actions des différents boutons présents sur la page web
 - *fileTraitement.js* permet de traiter le fichier transcrit et de récupérer ses informations
 - *main.js* gère les fonctions principales ainsi que l'appel des outils pour les paramètres et la liste des fichiers sur le serveur
 - *sb-admin-2.js* (Fichier Bootstrap)
 - *sb-admin-2.min.js* (Fichier Bootstrap minimisé)
 - *settings.js* gère les différents paramètres de la page web
 - *toolsBox.js* contient plusieurs fonctions permettant de modifier le texte de la transcription du fichier audio
- *json* contient le fichier de sauvegarde des paramètres afin que l'utilisateur n'ait pas à les spécifier à chaque démarrage de la page web
- *php* gère les requêtes avec le serveur
 - *copyFile.php* copie le fichier audio choisi dans le dossier audio de l'application
 - *fileManager.php* permet de mettre à jour le fichier de correction correspondant à notre audio
- *scss* contient les fichiers gérant le template Bootstrap de départ
- *vendor* contient les fichiers gérant le template Bootstrap de départ

¹ cf. "[installation](#)" pour l'installation de l'API *word_cloud*

² cf. https://github.com/amueller/word_cloud pour modifier les paramètres de génération