

T-AIA-901

Groupe 12 :

Lyne ARMAND
Pierre BOBARD
Dylan HOCHBERGER
Richard LA
Quentin SOMMER

PRÉSENTATION D'ENSEMBLE DU PROJET

- 4** PRÉSENTATION DE L'ÉQUIPE
- 5** CONTEXTE DU PROJET
- 5** LANGAGE DE PROGRAMMATION

BACKEND

- 7** API
- 8** BACK.PY
- 9** EXTRACT_LOCATIONS.PY
- 10** INTENT_CLASSIFIER.PY
- 11** TRAVEL.PY

FRONTEND

- 13** LOGO
- 13** COULEURS
- 14** MICRO



Présentation de l'ensemble du projet



PRÉSENTATION DE L'ÉQUIPE

Nous sommes le groupe 12.



Lyne Armand

Développeur Frontend



Pierre BOBARD

Développeur Backend



Dylan HOCHBERGER

Développeur Frontend



Richard LA

Développeur Backend



Quentin SOMMER

Développeur Backend

CONTEXTE DU PROJET

Construire un programme qui traite les commandes vocales ainsi que textuelles pour émettre un itinéraire approprié.

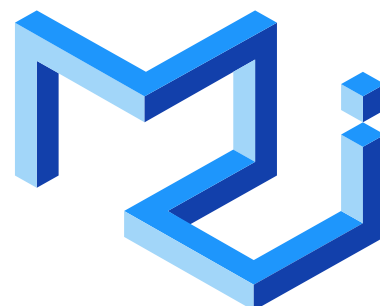
Plus précisément, le logiciel recevra les commandes de voyage sous forme d'un courriel ou d'un enregistrement vocal, et produira au moins une ligne de voyage approprié qui correspond aux attentes du client.

LANGAGE DE PROGRAMMATION

Pour le backend, on utilise :



Pour le frontend, on utilise :



Bibliothèque MUI



Backend





API

Route : POST <https://127.0.0.1:5000/travel>

Obtenir tous les trajets

Paramètre : la phrase enregistrée via le micro avec comme clé la langue sélectionnée, du style :

{ "fr" : "je voudrais aller de Paris à Strasbourg" }

Retour : une liste contenant tous les trajets du style :

*{ id : 'OCESN003300F030037323',
departure : 'Gare de Paris-Est',
destination1 : 'Gare de Sarrebourg',
destination2 : 'Gare de Strasbourg',
duration : '359' }*



BACK.PY

Tout d'abord, on initialise deux modèles : un pour le français et un pour l'anglais.

Puis, on récupère les données du dataset timetables.

Ensuite, on fait un appel du front avec en parametre la phrase enregistré via le micro avec comme clé la langue sélectionnée.

Fonction :

findTravel() :

- récupération de la phrase et de la langue
- que la langue choisit est français ou anglais, on fait une prédiction de l'intention avec le modèle fr/eng entraîné au lancement du serveur
- test l'intention:
 - si l'intention est un trajet alors on traite la phrase afin de définir le départ et la destination. Puis on récupère les trajets correspondant au départ et à l'arrivée.



EXTRACT_LOCATIONS.PY

Deux classes SetDirection et Word.

CCONJ_Score

NOUN_Score

ADP_Score

VERB_Score

Fonction :

analyseSentence(sentence, nlp) : Parcours tout le texte, si on trouve une location, elle est rajoutée dans le tableau sinon on ne fait rien.



INTENT_CLASSIFIER.PY

Fonctions :

data_prepare(lang) : préparation des données provenant du dataset de gestion des intentions.

naive_algo(lang) :

predict(question, clf, count_vect) :



TRAVEL.PY

Fonctions :

getData(csv="./dataset/timetables.csv") : Permet de récupérer le dataset et de formaliser les colonnes

getTravel(dataController, locations) : Permet d'obtenir le trajet direct ou indirects/

travelFunction(dataController, departure, destination) : Permet de renvoyer tous les trajets

getAllDestinationFromDeparture(dataController, departure) : Permet d'obtenir toutes les destinations à partir du départ



Frontend



LOGO



Mixage d'un train et d'un globe représentant les trajets possible en train.

COULEURS



Hexa : #1240ab

Signification : rêve, sagesse, sérénité, vérité, loyauté, fraîcheur

Couleur complémentaire du jaune.



Hexa : #ffaa00

Signification : fête, joie, chaleur, ego, puissance, connaissance, amitié

Le jaune est souvent associé aux voyages, aux transport et à la restauration.



MICRO

Utilisation de la bibliothèque SpeechRecognition.

useSpeechRecognition est un hook React qui permet à un composant d'accéder à une transcription de la parole captée par le microphone de l'utilisateur.

SpeechRecognition manages the global state of the Web Speech API, exposing functions to turn the microphone on and off.

Pour en savoir plus : <https://github.com/JamesBrill/react-speech-recognition>