Application utilisateur

AlpesTransport

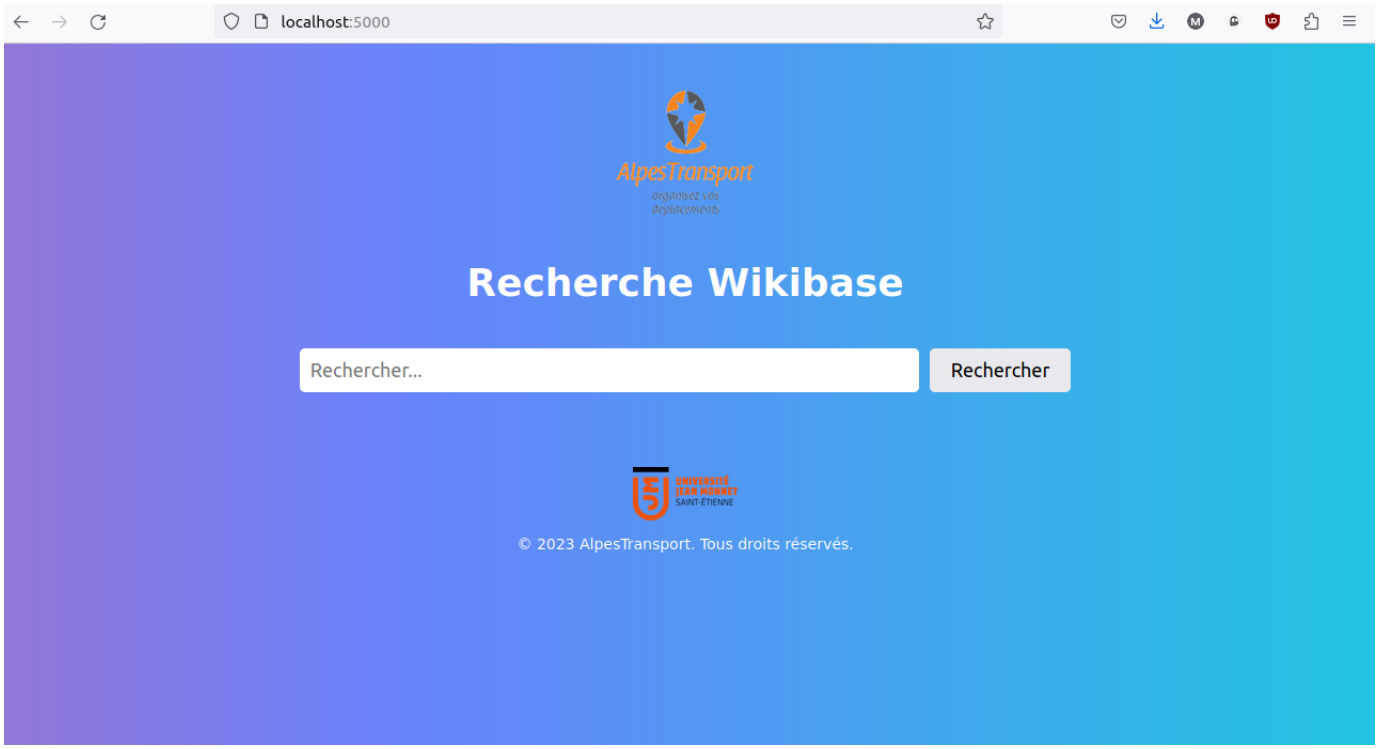
Mohammed ROUABAH
Ilyes ZEGHDALLOU
William MAILLARD

01/03/2023

1 l'objectif de l'application AlpesTransport

L'application utilisateur va permetre d'interoger la wikibase remplis au préalable à l'aide de nos parseurs. Notre application va permettre à l'utilisateur de rechercher 3 types d'informations, qui feront chacune l'objet d'un format de requête :

— demande d'informations sur un arrêts
— demande d'information sur les horraires d'un itinéraire
— demande de calcul d'un itinéraire entre un point A et B (suivant le temps disponible)



2 fonctionnement d'envois de requêtes

FIGURE 1 – Paramètres de la requête envoyé à la wikibase

Pour envoyer une requête à l'API WIKIBASE, nous devions spécifier l'URL de l'API et les paramètres de requête nécessaires pour obtenir les données souhaitées. Les paramètres de requête peuvent inclure le type d'entité recherché, les propriétés à récupérer, les langues à utiliser, etc.

Dans notre application, nous avons utilisé Flask, un framework web Python, pour créer une application web qui permet à l'utilisateur de rechercher des entités dans la base de connaissances Wikibase. Nous avons aussi utilisé la bibliothèque Requests pour envoyer une requête GET à l'API Wikibase et récupérer les résultats de la recherche.

Nous avons par ailleurs, utilisé le langage de requête SPARQL pour récupérer les entités

correspondant à la requête de recherche de l'utilisateur(toujours en cours). Nous avons également spécifié les propriétés que nous voulions récupérer pour chaque entité (par exemple, l'ID, le nom et la description de l'entité).

Enfin, nous avons utilisé Flask pour afficher les résultats de la recherche dans un template HTML, en utilisant des boucles et des conditions pour afficher les informations de chaque entité.

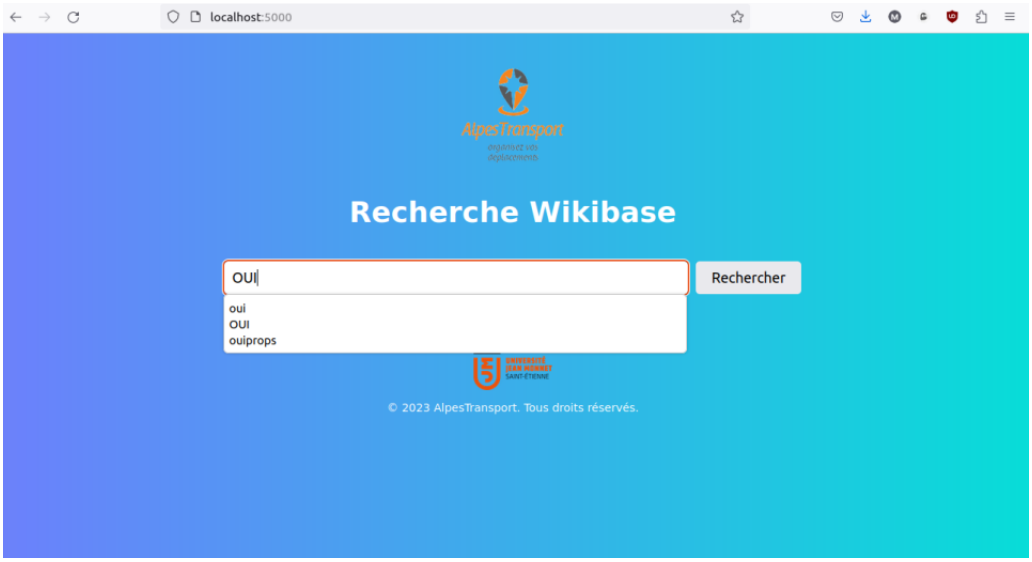


FIGURE 2 – Recherche d'un item par itemLabel

FIGURE 3 – Affichage du résultat de la réponse

3 fonctionnement de la réception des requêtes

Flask a été utilisé pour recevoir des requêtes HTTP entrantes. La fonction @app.route est utilisée pour définir des routes qui seront utilisées pour traiter différentes requêtes HTTP.

— Dans notre cas, deux routes ont été définies :

* La route de base ("/") qui affiche simplement la page d'accueil de notre application,
* La route "/search" qui est utilisée pour effectuer une recherche dans la base de données Wikibase Cloud.

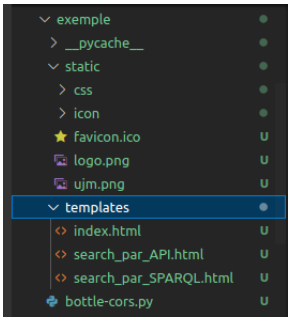


FIGURE 4 – Architecture de l'application

Lorsqu'un utilisateur accède à la page d'accueil, la fonction home() est appelée. Cette fonction utilise la méthode rendertemplate pour renvoyer le contenu du fichier HTML "index.html" en réponse à la requête.

Lorsqu'un utilisateur effectue une recherche, la fonction search() est appelée. Cette fonction récupère la chaîne de recherche saisie par l'utilisateur à partir de la requête HTTP à l'aide de la méthode request.args.get(). Ensuite, une requête GET est envoyée à l'API de Wikibase Cloud à l'aide de la bibliothèque requests.

La réponse de l'API est retournée sous forme de JSON, qui est ensuite utilisé pour extraire les informations pertinentes, telles que les entités et leurs propriétés correspondantes. Ces informations sont ensuite transmises à la page HTML "search.html" à l'aide de la méthode rendertemplate, qui est utilisée pour générer une réponse HTTP contenant le contenu HTML généré.