**TP4**

Salma ABOUMZRAG

**07/02/2024**

**input\_handler.py**

**Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement**

La fonction get\_user\_input() est conçue pour interagir avec l'utilisateur et collecter deux pièces d'information essentielles pour le fonctionnement du programme : une adresse spécifique (composée d'un numéro et d'une rue) et le nombre maximal de villes à rechercher qui correspondent à cette adresse. Voici une explication détaillée de son fonctionnement :

**Boucle While Infinie**

* But de la boucle : La fonction utilise une boucle while infinie (while True:) pour répéter la demande d'entrée jusqu'à ce que l'utilisateur fournisse une adresse valide, c'est-à-dire une adresse qui commence par un numéro. Cette méthode garantit que le programme reçoit des données d'entrée exploitables avant de continuer.

**Vérification de l'Adresse**

* Validation de l'adresse : Pour chaque itération de la boucle, la fonction demande à l'utilisateur de saisir une adresse. Elle vérifie ensuite si la première partie de l'adresse saisie (obtenue par adresse.split()[0]) est un nombre, ce qui est fait avec la méthode .isdigit(). Cela assure que l'adresse suit le format requis, commençant par un numéro de rue, ce qui est essentiel pour la précision de la recherche d'adresse dans l'API de géolocalisation.

**Saisie du Nombre de Villes Maximum**

* Saisie du nombre de résultats souhaités : Une fois qu'une adresse valide est saisie, la fonction demande ensuite à l'utilisateur de spécifier le nombre maximal de villes à rechercher (limit). Cette valeur permet de contrôler le volume de résultats retournés par l'API, offrant à l'utilisateur la possibilité de limiter la recherche selon ses besoins.

**Retour des Informations**

* Retour des données saisies : Après avoir validé l'adresse et recueilli le nombre de villes maximum recherchées, la fonction retourne ces informations pour qu'elles puissent être utilisées dans la suite du programme. Cela permet d'effectuer la requête à l'API de géolocalisation avec les paramètres spécifiés par l'utilisateur.

**Gestion des Entrées Invalides**

* Demande répétée en cas d'entrée invalide : Si l'adresse saisie par l'utilisateur ne commence pas par un numéro, la fonction affiche un message d'erreur ("L'adresse doit commencer par un numéro. Veuillez réessayer.") et réitère la demande d'adresse. Cela garantit que le programme ne procède pas avec des données d'entrée incorrectes, minimisant ainsi le risque d'erreurs ou de résultats inattendus lors de l'interrogation de l'API.

**api\_handler.py**

**Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement**

La fonction query\_api joue un rôle central dans le programme en gérant la communication avec l'API externe de géolocalisation pour rechercher des adresses basées sur les entrées utilisateur. Voici une explication détaillée de son fonctionnement :

**Importation du Module Requests**

* Utilisation de Requests : La fonction commence par importer le module requests, qui est une bibliothèque HTTP pour Python. Cette bibliothèque simplifie les requêtes HTTP, rendant le code plus lisible et la gestion des requêtes plus accessible.

**Fonction query\_api**

* Paramètres : La fonction query\_api prend deux paramètres : adresse (la chaîne de texte que l'utilisateur souhaite rechercher) et limit (le nombre maximum de résultats que l'utilisateur souhaite obtenir).

**Construction et Envoi de la Requête**

* URL de base et Paramètres : La fonction construit l'URL de la requête en utilisant l'URL de base de l'API (https://api-adresse.data.gouv.fr/search/) et ajoute les paramètres de la requête (q pour la chaîne de recherche et limit pour le nombre de résultats souhaités) en utilisant un dictionnaire params.
* Envoi de la Requête : La requête est envoyée à l'API en utilisant requests.get, en passant l'URL de base et le dictionnaire des paramètres. La réponse de l'API est stockée dans la variable response.

**Traitement de la Réponse**

* Vérification du Statut de la Réponse : La fonction vérifie d'abord le code de statut HTTP de la réponse pour s'assurer qu'elle est 200, indiquant une réponse réussie.
* Extraction et Affichage des Données : Si la réponse est réussie, les données JSON sont extraites en utilisant response.json(). La fonction parcourt ensuite les éléments de data['features'], qui contiennent les adresses correspondantes à la requête, et affiche pour chaque adresse son libellé complet, le code postal et la ville.
* Gestion des Réponses Sans Adresses : Si aucune adresse n'est trouvée (data['features'] est vide), la fonction informe l'utilisateur qu'aucune adresse correspondante n'a été trouvée et suggère d'essayer avec des termes de recherche différents.

**Gestion des Erreurs**

* Erreurs de Requête : Si la requête à l'API échoue (par exemple, en raison d'un problème de connexion ou si l'API retourne un code de statut différent de 200), la fonction affiche un message d'erreur général indiquant que la requête à l'API a échoué.

**Valeur de Retour**

* Retour des Données : Si la requête est réussie et que des adresses sont trouvées, la fonction retourne les données JSON extraites pour une utilisation ultérieure dans le programme (par exemple, pour permettre à l'utilisateur de choisir une adresse à visualiser sur une carte).
* Retour de None en Cas d'Erreur ou d'Absence de Résultats : Si aucune adresse n'est trouvée ou si une erreur se produit, la fonction retourne None, ce qui peut être utilisé par le reste du programme pour gérer ces cas.

**map\_visualizer.py**

**Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement**

La fonction visualize\_on\_map est conçue pour permettre à l'utilisateur de visualiser une adresse spécifique sur une carte, en utilisant Google Maps. Cette fonction fait partie du processus d'interaction utilisateur final dans le programme, après que les adresses ont été récupérées via l'API. Voici une explication détaillée de son fonctionnement :

**Importation du Module Webbrowser**

* Utilisation de Webbrowser : La fonction utilise le module webbrowser, une interface de haut niveau pour afficher du contenu Web à l'utilisateur. Ce module permet d'ouvrir le navigateur Web par défaut sur une page spécifique, ici utilisé pour afficher une adresse sur Google Maps.

**Vérification des Données**

* Validation des Données d'Entrée : Avant de procéder, la fonction vérifie si elle a reçu des données valides et exploitables (data). Elle s'assure que data n'est pas vide, contient l'élément features, et que cet élément n'est pas vide non plus. Si ces conditions ne sont pas remplies, elle informe l'utilisateur qu'aucune donnée n'est disponible pour la visualisation et termine son exécution.

**Interaction avec l'Utilisateur**

* Demande de Confirmation : La fonction demande ensuite à l'utilisateur s'il souhaite visualiser une adresse sur une carte. Elle accepte plusieurs variantes de réponse affirmative (oui, o, yes, y) pour améliorer la convivialité et la flexibilité de l'interaction.

**Sélection de l'Adresse**

* Choix de l'Adresse : Si l'utilisateur souhaite procéder, il est invité à choisir le numéro de l'adresse qu'il souhaite visualiser, basé sur la liste précédemment affichée par le programme. Ce numéro est ensuite ajusté (num\_adresse - 1) pour correspondre à l'indexation basée sur zéro des listes en Python.

**Ouverture de Google Maps**

* Récupération des Coordonnées et Ouverture de la Carte : La fonction récupère les coordonnées géographiques (latitude et longitude) de l'adresse sélectionnée et construit une URL pour Google Maps qui pointe vers ces coordonnées. Elle utilise ensuite webbrowser.open\_new\_tab pour ouvrir un nouvel onglet dans le navigateur par défaut de l'utilisateur, affichant la localisation souhaitée sur Google Maps.

**Gestion des Erreurs et Annulation**

* Validation du Numéro d'Adresse : Si le numéro choisi par l'utilisateur n'est pas valide (c'est-à-dire, en dehors de la plage des adresses disponibles), la fonction affiche un message d'erreur approprié.
* Annulation par l'Utilisateur : Si l'utilisateur choisit de ne pas visualiser une adresse sur la carte, la fonction informe simplement de l'annulation de la visualisation.

**app.py**

**Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement**

Ce script principal coordonne les différentes composantes du programme pour offrir une expérience utilisateur fluide et intégrée, depuis la saisie initiale des données jusqu'à la possibilité de visualiser une adresse sur une carte. Chaque étape du processus est gérée par une fonction spécialisée, ce qui rend le code modulaire, facile à comprendre, à maintenir et à étendre.

**Résultat :**

**Une image contenant texte, logiciel, Logiciel multimédia, ordinateur

Description générée automatiquement**

**Une image contenant texte, carte, capture d’écran, logiciel

Description générée automatiquement**

**Fin 🙌**