

Projet SAE 05 - Traitement de données

TP 2.2
SCHER Florian
DAIRIN Côme

- PokéZone -

Fonctionnement en front-end :

L'utilisateur rentre un nom de zone dans un jeu Pokémon.

Le programme retourne une liste de Pokémon pouvant être rencontrés ainsi que la répartition en pourcentage de leur types.

L'utilisateur obtient alors une fiche sur la zone recherchée, présentée sous la forme d'une page HTML.

Fonctionnement en back-end :

Bibliothèques utilisées :

- Requests – Pour les requêtes vers l'API PokéAPI
- Markdown – Pour la présentation des données en Markdown
- RapidFuzz – Pour la recherche approximative
- Matplotlib – Pour les diagrammes
- Json – Pour la conversion de XML à Json
- OS – Pour vérifier si le cache est présent

Implémentation du cache :

Le programme commence par faire une lourde requête de 1036 éléments vers <https://pokeapi.co/api/v2/location/> puis crée un fichier "root.json". Celui-ci est alors utilisé pour toute autre zone recherchée afin d'optimiser les requêtes.

Recherche approximative :

L'utilisateur rentre le nom anglophone d'une zone et le programme sélectionne le meilleur match possible grâce au module RapidFuzz.

Si les données contiennent la clé "area" le programme parcourt chaque sous-zone et combine les données trouvées sous le nom de la zone mère.

Si aucune donnée n'est trouvée, le programme notifie l'utilisateur de l'absence de résultats.

Le nom doit être en anglais en raison du nombre **extensif** de requêtes nécessaires pour parcourir l'architecture des données.

Récupération des données :

- Requête cache : <https://pokeapi.co/api/v2/location/?limit=1036>
- Nom français zone (si présent) : data['names'][2]['name']
- Pokémons : data['pokemon-encounters']
- Types Pokémon : data['types']
- Nom français Pokémon :
 - Première requête vers data['species']['url']
 - Seconde requête vers data['names'][4]['name']

Mise en forme :