

Création d'un front end permettant l'accès à des données via une API REST

BOUZARD ARTHUR & AUFFRET FABIENNE

***Institut National des Langues et Civilisations Orientales
(I.N.A.L.C.O)***

Master 2 TAL IM- Février 2021

Sommaire

CRÉATION D'UN FRONT END PERMETTANT L'ACCÈS À DES DONNÉES VIA UNE API REST.....	1
1. CRÉATION ET TESTS EN LOCAL.....	1
2. DÉPLOIEMENT ET TESTS SUR HEROKU.....	4
3. MODE D'EMPLOI.....	5
3.1 INSTALLATION EN LOCAL.....	5
3.2 UTILISATION.....	7
4. CONCLUSION & PERSPECTIVES.....	13

1. Création et tests en local

Nous avons tout d'abord développé notre application en local.

ARTTHUR

Nous avons utilisé essentiellement JQuery pour communiquer avec notre back-end et gérer une base de données sous PostgreSQL avec deux tables (Users et Data) qui ont été remplies en lisant les fichiers users.json et les données FR.tsv (voir partie back-end).

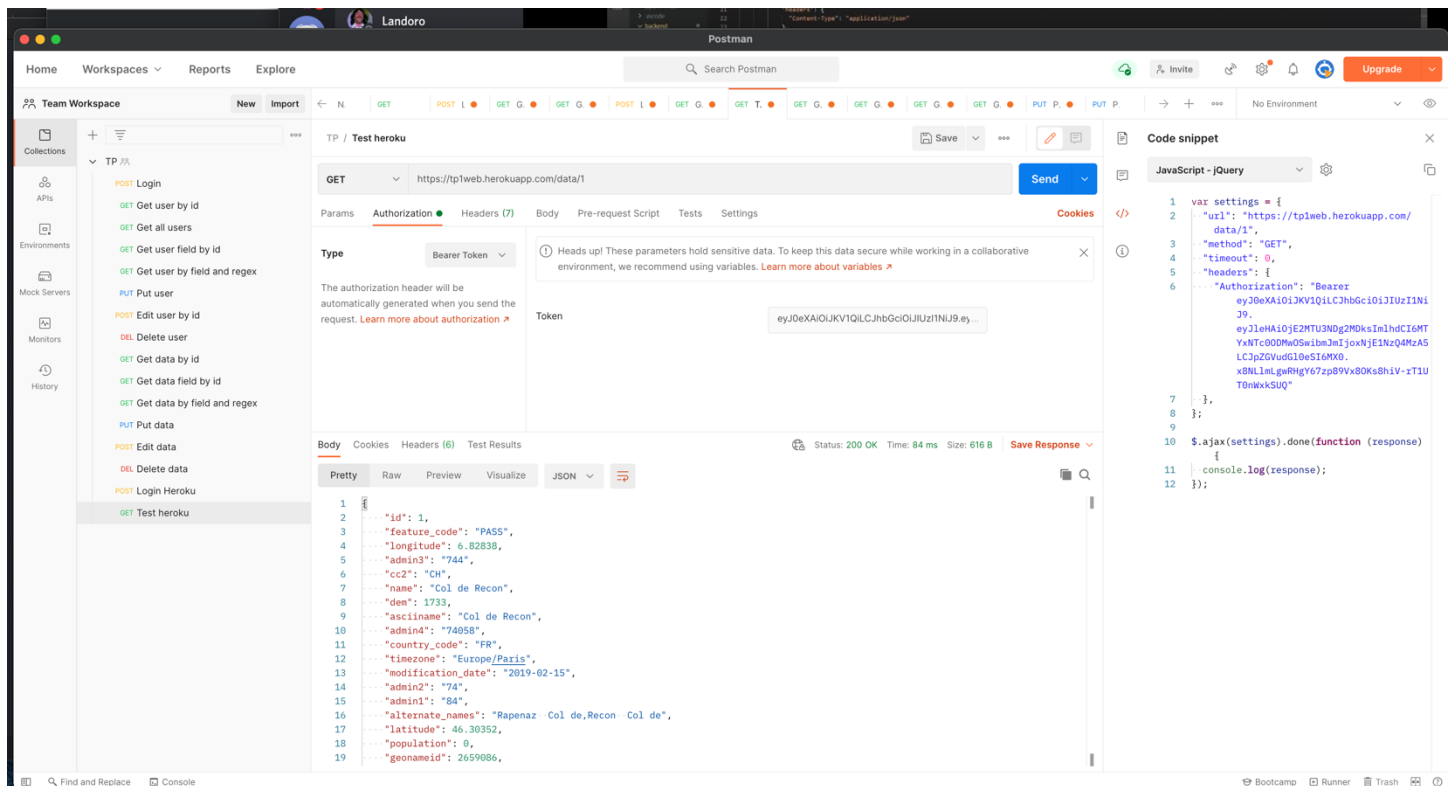
Nous avons ajouté les méthodes peu à peu, les testant au fur et à mesure (en contrôlant avec Postman).

Nous avons fait des requêtes et contrôlé leur affichage (apparition / disparition des boutons, création du bon code HTML).

Nous avons traité divers cas d'erreurs (données absentes, il n'y a jamais d'erreur de mauvais champ, puisque les requêtes sont formées par formulaire) et renvoyé un message à l'utilisateur, correspondant au statut de la requête traitée.)

Nous avons aussi proposé une requête avec regex (sachant qu'ici les `_` remplacent le point `.`)

Nous avons récupéré le code des diverses requêtes en javascript avec Postman en local pour l'inclure dans notre JQuery (get, put, post et delete) :



Ici une requête get.

Nous avons aussi utilisé la syntaxe de la requête avec le header bearer (token stocké et utilisable grâce à l'objet Sessionstorage d'ajax) pour que nos requêtes fonctionnent (si l'utilisateur s'est connecté ou non).

Si l'utilisateur n'est pas connecté et tente de gérer la base, il reçoit est redirigé sur la page de connexion. On peut se déconnecter en cliquant sur le bouton déconnexion en haut droite de l'écran et on sera redirigé sur la page de connexion. Si la page est quittée, la session se ferme donc le token stocké disparaît et il faudra alors se connecter de nouveau.

Si la connexion est réussie, il est redirigé vers la page de recherche sur les données géographiques.

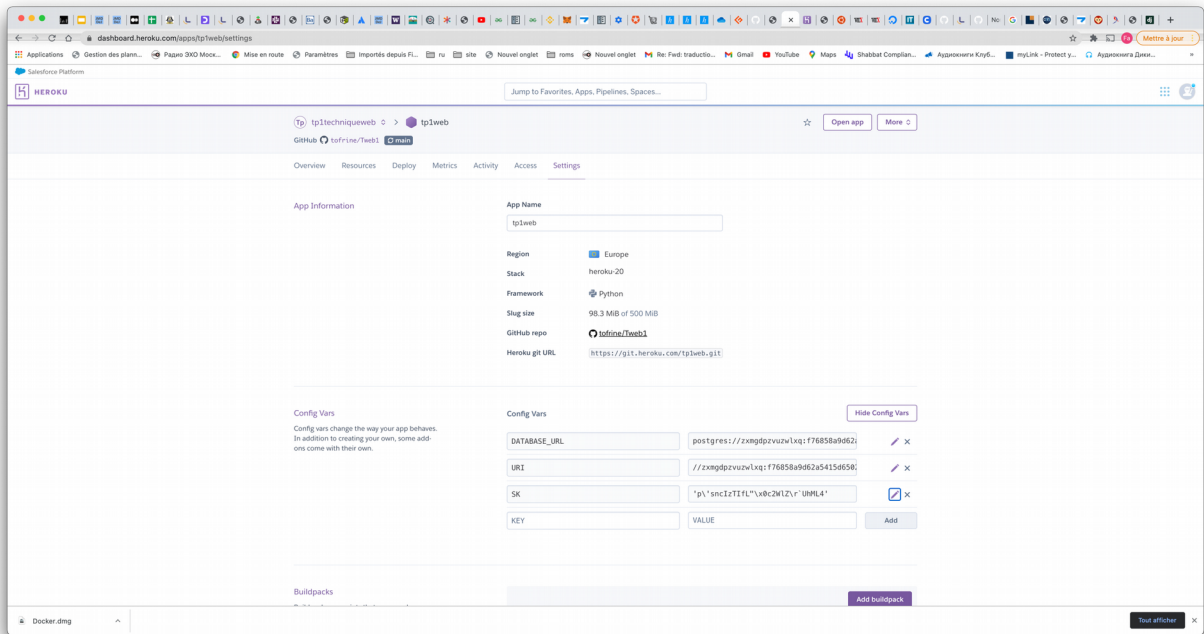
2. *Déploiement et tests sur Heroku*

Nous avons pushé les modifications faites sur le git connecté à heroku. Pour l'installation, voir la partie back-end.

Vous trouverez d'ailleurs notre application à l'adresse suivante : <https://webappinalco.herokuapp.com/>

Identifiant : admin , Mot de passe : elvis2021 , ne vous gênez pas pour essayer !

Rappel :
Nous avons stocké les identifiants de connexion dans une variable locale (utilisée dans le fichier config.py) :



Ce qui s'écrit dans le config.py sous la forme :

Nous avons procédé aux mêmes tests qu'en local (nous avons testé les cas limites, les requêtes classiques).

3. Mode d'emploi

3.1 Installation en local

On suppose que les utilisateurs ont déjà installé un back-end qui fonctionne (cf mode d'emploi du back-end). Ce backend est accessible sur

Il suffit d'installer Nginx, de créer un fichier api.conf dans le répertoire `usr\local\etc.\nginx\sites-enabled` avec le contenu suivant :

```
server {  
  
    listen      80 ;  
  
    listen      [::]:80;  
  
    location / {  
  
        proxy_pass http://localhost:5000/;  
  
    }  
  
}
```

Il faut ensuite modifier le fichier `etc/hosts` de linux et ajouter la ligne suivante : `digidata.api.localhost 127.0.0.1`

Le serveur sera alors accessible via le reverse proxy nginx à l'adresse `digidata.api.localhost`. Nous n'avons pas de serveur dédié à notre front end.

Puis : comme avec le back-end, il faut lancer avec `python run.py` en ligne de commande.

Vous deviez avoir l'affichage suivant : (Le remplissage de la base de données créée en local peut prendre entre 1 et 3 minutes selon la vitesse de votre machine et retarder l'affichage) :

```
WARNING:root:Database initialized!  
* Serving Flask app "myapp" (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
* Debug mode: on  
INFO: werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)  
INFO: werkzeug: * Restarting with fsevents reloader  
WARNING: root:Database initialized!  
WARNING:werkzeug: * Debugger is active!  
INFO:werkzeug: * Debugger PIN: 333-455-098
```

```

Last login: Fri Feb 26 18:13:27 on ttys003
(base) fabienneauffret@MSP-de-Fabienne webappinalco % python run.py
WARNING:root:Database initialized!
  * Serving Flask app "myapp" (lazy loading)
  * Environment: production
    WARNING: This is a development server. Do not use it in a production deployment.
    Use a production WSGI server instead.
  * Debug mode: on
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
INFO:werkzeug: * Restarting with fsevents reloader
WARNING:root:Database initialized!
WARNING:werkzeug: * Debugger is active!
INFO:werkzeug: * Debugger PIN: 333-455-898
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:02:34] "GET /data/9 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:02:43] "POST /users/1 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:10:19] "GET /users/1 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:16:43] "GET /data/admin4/3 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:16:57] "GET /data/admin4/Y HTTP/1.1" 400 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:17:11] "GET /data/admin4/c HTTP/1.1" 400 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:17:18] "GET /data/admin4/6 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:17:34] "GET /data/asciiName/Y HTTP/1.1" 400 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:17:38] "GET /data/asciiName/Y HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 21:18:04] "GET /data/asciiName/Y HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 22:14:10] "GET /data/9/name HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [26/Feb/2021 22:58:24] "GET /data/9/name HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [27/Feb/2021 12:04:39] "PUT /users HTTP/1.1" 200 -

```

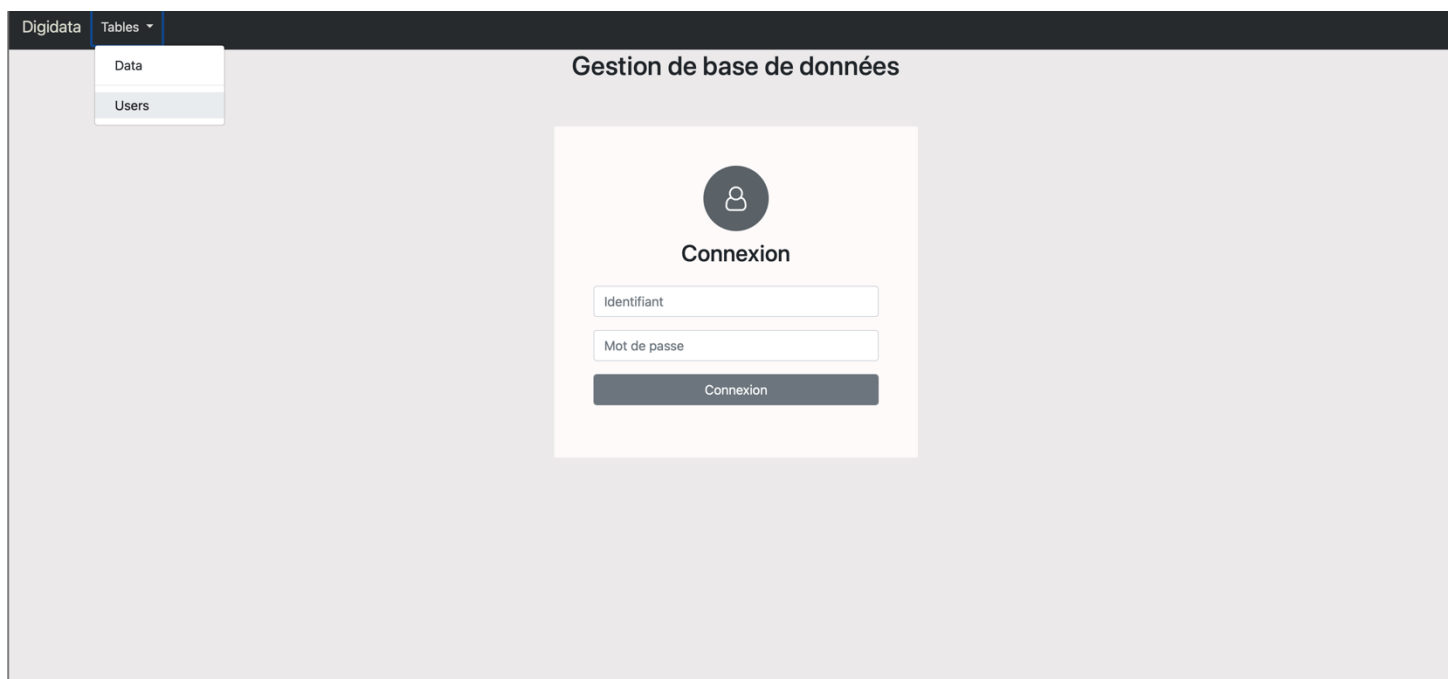
3.2 Utilisation

L'accès aux bases de données est protégé par un mot de passe (cf. Cahier des charges : « les utilisateurs doivent se connecter»). Par défaut l'identifiant est admin et le mot de passe admin.

Pour l'instant, il faut le changer dans l'api, puisqu'il n'y a pas de mot de passe individuel stocké dans la table utilisateurs (users) et il était demandé aux utilisateurs se connecter et non pas de s'authentifier.

Les utilisateurs doivent aller à l'adresse du serveur local et entrer le login et mot de passe nécessaires pour pouvoir faire des requêtes. S'ils ne sont pas connectés, ils peuvent voir les pages de recherche mais les boutons les redirigent vers l'accueil et la page de login.

Ensuite, en cliquant sur tables, ils peuvent choisir de gérer les tables users ou data /



The screenshot displays the 'Digidata' application interface. At the top, there is a dark navigation bar with 'Digidata' on the left and a 'Tables' dropdown menu on the right. The dropdown menu is open, showing 'Data' and 'Users' options. The main content area has a light gray background and is titled 'Gestion de base de données'. In the center, there is a white card titled 'Connexion' with a user icon. Below the title, there are two input fields: 'Identifiant' and 'Mot de passe'. At the bottom of the card is a dark button labeled 'Connexion'.

On arrive sur une page de gestion des utilisateurs ou des données géographiques :

tp1web.herokuapp.com/search_data

Applications Gestion des plan... Pappro 3XO Mock... Mise en route Paramètres Importés depuis FI... nu site Nouvel onglet roms Nouvel onglet Re: Fwd: traductio... Gmail YouTube Maps Shabbat Complian... Азбеченск Кныд... myLink - Protect y...

Digidata Tables Déconnexion

Recherche dans la base de données Digidata

Recherche dans les lieux

56 Recherche ou regex All Rechercher

Edition de la localisation 56

2967137	Zimmersheim	Zimmersheim	Zimmersheim
47.72079	7.38847	P	PPL
FR	null	44	68
684	68386	1080	null
263	Europe/Paris		

Valider

Ajout d'un lieu

geonameid	name	alternatenames	latitude
longitude	feature class	country code	cc2
admin1 code	admin2 code	admin3 code	admin4 code

search.html Tout afficher

On peut afficher toutes les colonnes de la base, faire une recherche (**sensible à la casse !**) et sur les résultats de cette recherche éditer ou supprimer des données.

On peut soit faire une recherche par id ou terme (ou regex) sur tous les champs (ici un exemple avec les utilisateurs) :

Digidata Tables ▾

Recherche dans la base de données
Digidata

Recherche dans les utilisateurs

id	nom	prenom	fonction	anciennete	conge	actif	actionnaire	missions	mise_a_jour	Editer	Supprimer
1	Mikolov	Thomas	Directeur des representations vectorielles	10	15	null	true	Bruxelle,Paris,Pakistan	2021-03-09	Editer	Supprimer
2	Bergier	Francine	Chef service matériel informatique	20	14	true	null	Paris	2021-03-09	Editer	Supprimer
3	Taken	Georges	Cuisinier en chef	3	5	true	null	Londre,Tunis	2021-03-09	Editer	Supprimer
4	Gate	Roland	Chef service juridique	9	10	true	null	Berlin,Paris,NewYork	2021-03-09	Editer	Supprimer
5	Fokwe	Yvette	Directrice normalisation lexicale	3	14	true	true	NewYork,Douala	2021-03-09	Editer	Supprimer
6	Samuel	Jackson	Chef service matériel informatique	20	14	true	null	Paris	2021-03-14	Editer	Supprimer

Rappel : les regex ici utilisent le _ (underscore) au lieu du point. Exemple :

Ici on cherche les noms qui commencent par « Par » .

Tables ▾

Recherche dans la base de données
Digidata

Recherche dans les lieux

id	geonameid	name	asciname	alternate_names
597	2967680	Parc Naturel Régional des Vosges du Nord	Parc Naturel Regional des Vosges du Nord	Parc Naturel Regional des Vosges du Nord,Parc Naturel Régional des Vosges du Nord,Parc des Vosges du Nord
1730	2968815	Paris	Paris	Departement de Paris,Département de Paris,Parigi,Paris,Paris Department
3390	2970479	Paris 15 Vaugirard	Paris 15 Vaugirard	15e Arrondissement,15eme Arr,15ème Arr,Arrondissement de Vaugirard,Paris 15e,Vaugirard,XVe
2909	2969998	Parc Régional du Vercors	Parc Regional du Vercors	null
18954	2986082	Paris 11 Popincourt	Paris 11 Popincourt	11e Arrondissement,11eme Arr,11ème Arr,Arrondissement de Popincourt,Paris 11e,Popincourt,XIe

Digidata

Tables

Recherche dans la base de données

Digidata

Recherche dans les utilisateurs

ID

Tho_*

prénom

Rechercher

id	nom	prenom	fonction	anciennete	conge	actif	actionnaire	missions	mise_a_jour	Editer	Supprimer
1	Mikolov	Thomas	Directeur des representations vectorielles	10	15	null	true	Bruxelle,Paris,Pakistan	2021-03-09	Editer	Supprimer

Ajout d'un utilisateur

Nom

Prénom

Fonction

Ancienneté

Congés

Actif

Actionnaire

Missions

Ajouter

Si vous ne spécifiez pas de champ sur lequel faire la recherche, vous avez un message d'erreur :

Digidata

Tables

Recherche dans la base de données

Digidata

Recherche dans les utilisateurs

ID

Tho_*

One

Rechercher

Veuillez sélectionner un champ valide sur lequel effectuer votre recherche.

Ajout d'un utilisateur

Nom

Prénom

Fonction

Ancienneté

Congés

Actif

Actionnaire

Missions

Ajouter

Le menu déroulant des champs vous permet de chercher la ligne par ID dans un des champs ou dans tous. La recherche par texte ou regex ne marche que si l'on spécifie le champs sur lequel on

effectue la recherche.
Que ce soit pour les données ou les utilisateurs :

Digidata Tables ▾

Recherche dans la base de données Digidata

Recherche dans les lieux

ID 🔍 Recherche ou regex 🔍

Rechercher

Ajout d'

geonameid	name	latitude
longitude	feature class	cc2
admin1 code	admin2 code	admin4 code
population	elevation	timezone
asciiname	feature code	

Ajouter

- ✓ All
- geonameid
- name
- alternatenames
- latitude
- longitude
- feature class
- country code
- cc2
- admin1 code
- admin2 code
- admin3 code
- admin4 code
- population
- elevation
- dem
- timezone

Note : Pour la table des utilisateurs, on a pour chaque champ le bon format (vrai ou faux pour actif et actionnaire) et nombre pour l'ancienneté et les congés.

Si vous cliquez sur le bouton « Supprimer » en face d'une ligne de données -, cette ligne sera supprimée..

Si vous cliquez sur « Éditer » un formulaire pré rempli apparaît. Ici un exemple sur la table utilisateurs :

Note : l'exemple ci-dessous est donné avec la table users, pour plus de lisibilité (il y a moins de champs et vous verrez tout de suite que sa fonction a changé) :

Digidata Tables ▾

Recherche dans la base de données
Digidata

Recherche dans les utilisateurs

ID 🔍

Recherche ou regex 🔍

All ▾

Rechercher

Edition de l'utilisateur 3

Taken

Georges

Cuisinier en chef

3 ▾

5 ▾

Inchangé ▾

Inchangé ▾

Londre,Tunis

Valider

Ajout d'un utilisateur

Nom 👤

Prénom 👤

Fonction 👤

Ancienneté 👤

Congés 👤

Actif ▾

Actionnaire ▾

Missions 👤

Ajouter

Il suffira de remplir les champs que vous souhaitez changer et de cliquer sur « Valider ».

Si on fait une recherche après avoir validé :

Digidata
Tables

Recherche dans la base de données
Digidata

Recherche dans les utilisateurs

ID

Taken

Nom

Rechercher

id	nom	prenom	fonction	anciennete	conge	actif	actionnaire	missions	mise_a_jour	Editer	Supprimer
3	Taken	Georges	Maitre cuisinier cadre	3	5	true	null	Londre,Tunis	2021-03-09	Editer	Supprimer

Ajout d'un utilisateur

Nom

Prénom

Fonction

Ancienneté

Congés

Actif

Actionnaire

Missions

Ajouter

Exemple avec la table data :

Recherche dans la base de données
Digidata

Recherche dans les lieux

5

Recherche ou regex

All

Rechercher

Edition de la localisation 5

2659973

Le Cheval Blanc

Le Cheval Blanc

Le Cheval Blanc

46.05193

6.87276

T

MT

FR

CH

84

74

742

74273

null

2831

2807

Europe/Paris

Valider

Ajout d'un lieu

geonameid

name

alternatenames

latitude

Pour l'ajout de données, la partie ajout d'un lieu permet d'entrer de nouvelles données dans la table de la base de données et vous renvoie un message de succès (en vert) ou d'erreur (en

rouge) le cas échéant avec éventuellement une alerte concernant les raisons précises (champ vide, mauvais format de string trop longue etc.).

4. *Conclusion & perspectives*

Voilà un front-end à l'interface simple, utilisable sur tous les supports grâce à l'utilisation de bootstrap dans sa dernière version, que les utilisateurs peuvent prendre en main tout de suite et qui vient compléter le back-end et donner aux utilisateurs une interface plus intuitive et plus facile à utiliser que Postman pour gérer leurs bases de données.