

Test technique recrutement Full Stack (Stage)

Contexte

“Nos clients sont des entreprises qui ont des projets avec de forts enjeux locaux sur l'acceptabilité par la population et par les élus. Par exemple, les entreprises implémentant des champs d'éoliennes ont envie de pouvoir répondre à des questions telles que: “Est-ce que la population est favorable aux énergies renouvelable ?” ou “Est-ce que le maire d'une ville s'est déjà prononcé en faveur de l'énergie éolien ?”.

Une des composante de notre logiciel consiste à permettre à l'utilisateur

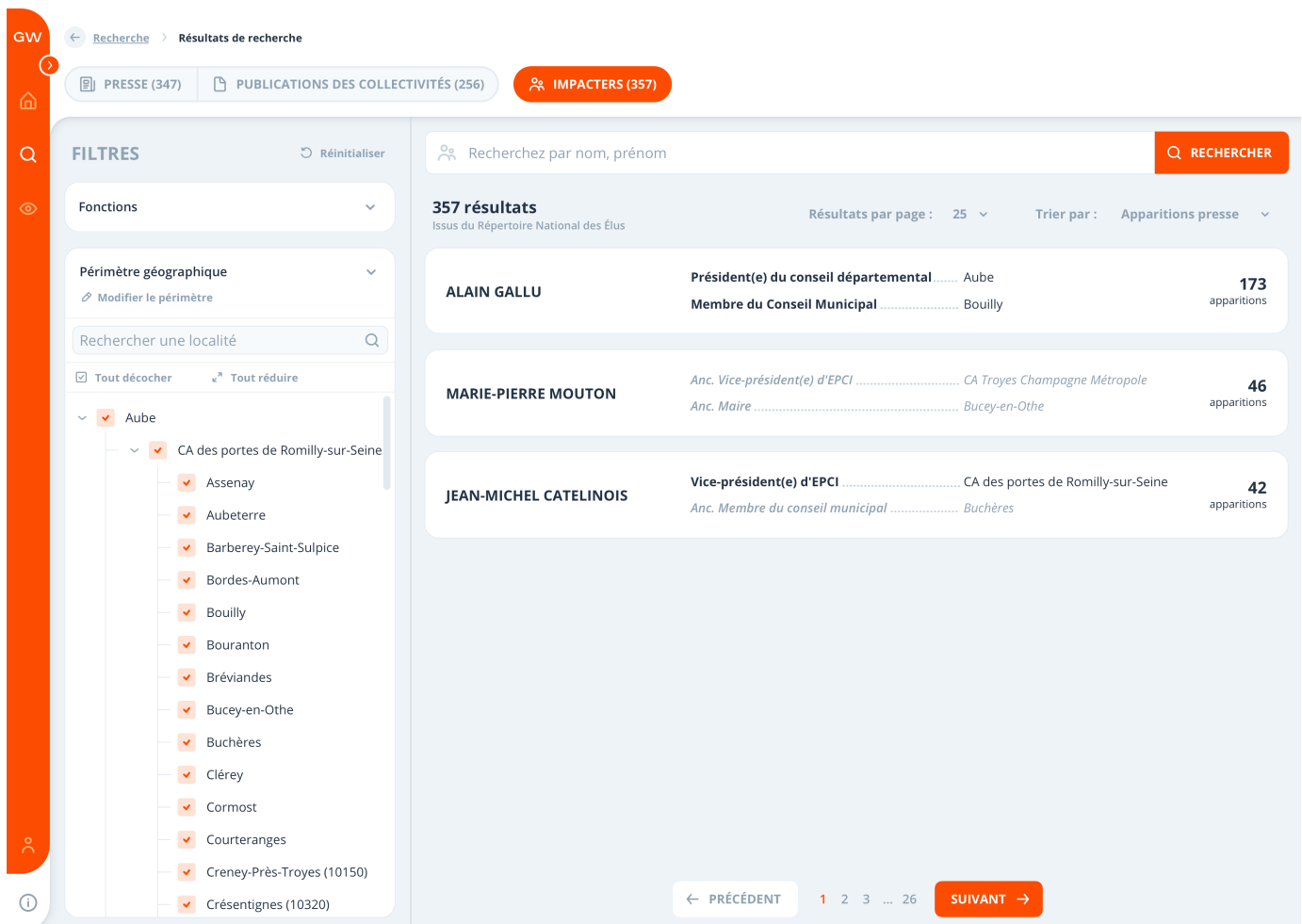
- de retrouver, parmi une base de données exhaustive des élus de France, les élus d'un territoire de son choix (Département, Communauté de commune, Commune...),
- effectuer une recherche par nom et prénom pour retrouver un élu pour sur lequel il souhaiterait avoir plus d'information,
- accéder pour un élu spécifique, à ses mandats (passés et en cours) et ses prises de position dans la presse.

Dans ce test, nous allons vous proposer de réfléchir à une version simplifiée de cette partie de notre application.”

Objectif

L'objectif de ce test est de construire **en partie** une page permettant de visualiser les élus sur un territoire et de faire une recherche sur ces élus par nom prénom.

Maquettes



Tâches à effectuer

- Créer la page de consultation des élus:

Créer la page html à partir de la maquette que nous avons fournie.

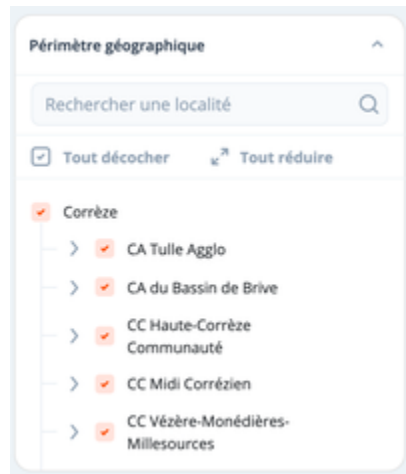
- **Créer une partie du filtre géographique:**

On se concentrera ici sur la création du filtre géographique (spécifications ci-dessous). Vous aurez pour tâche de:

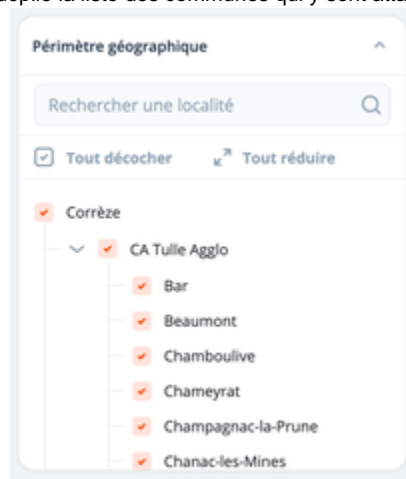
- Construire une API (de préférence en python) permettant de récupérer l'arborescence de territoires à partir du département (spécifié via l'url) pour nourrir le filtre.
 - Les données de hiérarchie des territoires se trouve dans:
 - le fichier `territories.csv` (qui contient la liste complète des territoires français avec leur id, code, nom et type (`id, code, name, kind`))
 - la table `territory_parents.csv` qui contient la hiérarchie (id du territoire parent, id du territoire enfant (`parent_id, child_id`)), chaque ligne étant une relation entre deux territoires (la description de la hiérarchie des territoires français se trouve ci-dessous).
- Implémenter la connexion avec la page html, pour qu'à l'arrivée sur la page, en fonction du département passé en paramètre d'url, le filtre contienne les bons territoires.

Spécifications du filtre géographique

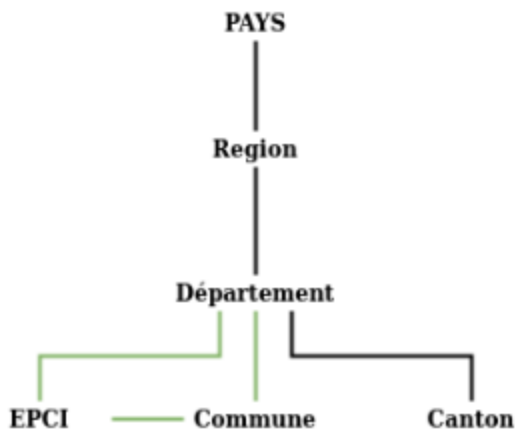
- Lorsque l'utilisateur arrive sur la page, il a sélectionné un département, qui se trouve dans l'url, par exemple: <http://beta.goodwill.fr/search/FRDEPA19/impacter> pour le Corrèze (département 19).
- Le filtre est initialement replié:



- Les niveaux sous le département sont les EPCI (territoires avec valeur `kind= 'FREPCI'` dans la base de donnée).
- Le clique sur la flèche à côté d'une EPCI dépile la liste des communes qui y sont attachées



- la hiérarchie des territoires en France est la suivante:



Livrable attendu

- Un repo Git contenant l'application.
- Idéalement, on voudrait pouvoir faire tourner en local l'application pour vérifier que tout est bien fonctionnel.

Données disponibles

- `territoires.csv` liste des territoires français.
 - Communes : `kind='FRCOMM'`
 - EPCI: `kind='FREPCI'`
 - Département: `kind='FRDEPA'`
- `territory_parents.csv` liens hiérarchiques entre territoires.
 - `child_id`: id de l'enfant
 - `parent_id`: id du parent