

Rendu Web-Client

Étudiante: Boutheina KAROUI

Filière: WIA 5

Identifiant Github: boutheinakaroui

Introduction

Ce cours d'introduction au framework React m'a appris à comprendre les bases d'un projet React: Architecture/ Composant/ Contextes.

Étant habituée à Angular/Ionic, l'architecture d'un projet React m'a été complètement différente. Grâce au cours/tp formateur réalisé au cours de ce module, j'ai pu apprendre les bases afin d'entamer ce projet.

Pendant la réalisation de ce projet, j'ai pu faire face à beaucoup de difficultés.

Tâches effectuées

Dans le groupe, nous nous sommes réparties 2 personnes sur le front et 2 autres sur le back. Ma tâche était de travailler sur le front.

1. Ma première tâche était de lier le back avec le front en utilisant les endpoints spécifiques: byCity / byDistance. Pour ce faire, j'ai utilisé la méthode fetch à la place d'axios car cela nous permet la gestion des promesses.
2. création des modèles Station afin de les pouvoir faire leur traitement et les filtrer par:
 - Ville
 - Type de gaz
 - Prix

=> Afin de faciliter ce traitement, pour chaque type de carburant nous stockons le nom, prix du carburant. Possibilité d'avoir le même nom de station pour un type de carburant différent.

et les afficher dans un map "leaflet" dans la librairie **react-leaflet**.

Ce choix a été fait avec le reste de notre équipe. Leaflet fournit beaucoup d'outils qui permettent l'affichage d'un point donné sur la map: Marker/PopUp/useMap pour les boutons flottants.

3. Parmi les avantages du framework React est la réutilisabilité de ses composants. Grâce à cela, j'ai créé des composants pour:
 - SearchBar qui va être utilisée dans le Home Page et aussi dans le Statics Page.
 - Map pour avoir un composant seul pour le MapContainer.
 - charts pour les graphes: le but c'est de pouvoir afficher plusieurs chart dans le futur.

=> Le passage des données dans les composants se fait avec les props. Le but c'est les données d'un haut niveau vers un bas niveau: Home Component vers Map Component.

4. Il nous a été demandé de pouvoir changer le thème. Pour ce faire, j'ai utilisé le useContext qui permet au composant/classe de communiquer entre eux.
5. Dans notre application, nous avons la possibilité d'afficher les stations à proximité grâce à ma position actuelle. Cela a été fait avec **navigator.geolocation.getCurrentPosition()**. Cela prend un peu de temps à charger (à améliorer dans le future).
6. En utilisant la library react-canvas, nous avons pu afficher notre graphe par ville afin d'avoir la moyenne des prix de chaque type de carburant. L'intégration de cette partie m'a bloquée quelques heures.

Cause: Les données prennent beaucoup de temps à charger pour calculer la moyenne des prix.

Solution: Ajouter un setTimeout pour attendre la récupération des données (stations).

Difficultés rencontrés

Comme précisé dans la partie précédente, la compréhension du business logic du framework react a été la tâche la plus difficile car en apprentissage/école, j'étais habituée à utiliser Angular. Une fois compris, je me suis lancée à implémenter les différentes fonctionnalités avec aisance.

Le plus grand blocage était lors de l'implémentation des charts car il m'a fallu choisir la librairie qui convient et aussi de savoir quoi afficher et comment. Pour la faire simple, j'ai opté pour une moyenne des prix par ville.

Comme expliqué précédemment, les données mettaient trop de temps à charger en général et ceci provoque des erreurs lors de la visualisation.

Un autre blocage est l'utilisation des librairies react-bootstrap. Grâce aux cours et à beaucoup de recherches, j'ai pu connaître des librairies qui m'ont beaucoup facilité la tâche.

Code

Pour la solution optimale, vous pouvez la trouver dans StationService:

```
// Solution optimale par Boutheina karoui

getClosestStations: async function(lat:any, long:any) {

    let closestStations:Station=[];

    const requestOptions = {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({

    "latitude":lat,

    "longitude":long,

    "radiusInMeter":1200

})

});

    let response = await fetch(

'http://localhost:8080/api/stations/current/byDistance/',requestOptions

);
```

```

    let responseJson = await response.json();

    let listOfStations=JSON.parse(responseJson);

    listOfStations.map((x:any)=>{

        let station: Station=new Station();

        station._latitude=x.latitude;

        station._id=x._id;

        station._longitude=x.longitude;

        station._adresse=x.adresse;

        station._ville=x.ville;

        if(x.listeDePrix!==undefined){

            x.listeDePrix.map((price:any)=> {

                station._valeur=price.valeur;

                station._nom=price.nom;

            } )}

        closestStations.push(station)

    })

    return closestStations;

}

```

=> Pour la solution à améliorer, notre application n'est pas encore responsive par manque de temps. Cela pourra se faire grâce à la librairie mediaQuery react.

Temps de développement

Appel endpoint: 4h

Map: 5h

Affichage sur Map: 5h

Filtre: 3h

Charts: 6h

Localisation: 2h

