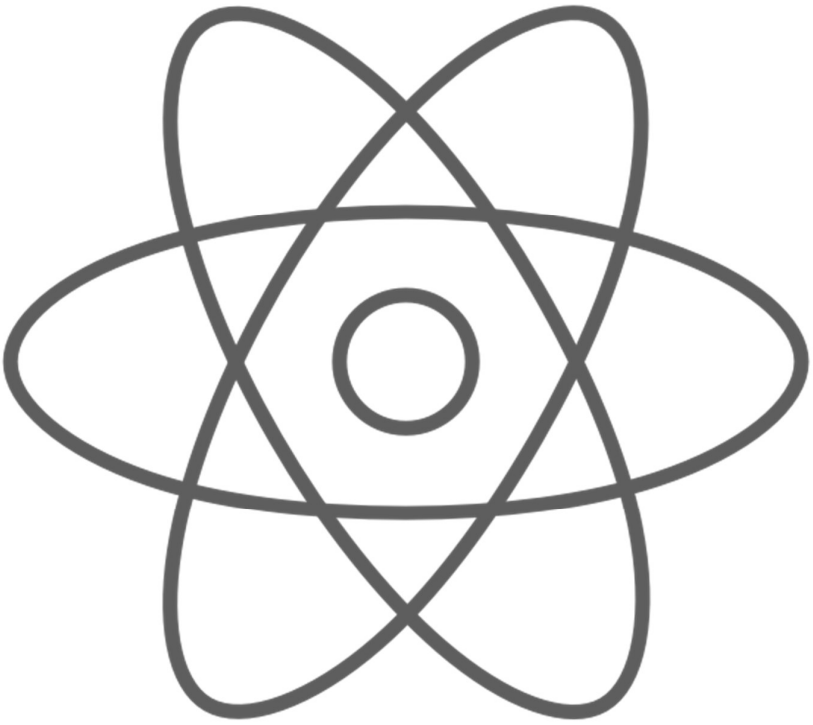
Rapport projet React



Etudiant : Aldric Ducreux

Identifiant github : Aldric-Ducreux

Tâches

Front-end

- Initialisation du projet avec un template : 10 min
- Nettoyage du Template et refonte de partie pour notre projet : 2h
    - Suppression de composant inutile
    - Refonte du routing
    - Déplacement d'élément pour le visuel

- Mise en place de la visualisation : 12h
    - Mise en place du composant de la carte vectoriel interactive : 4h
        - Création de la carte interactive à partir de amCharts
        - Ajout des éléments sur la carte pour chaque département
        - Mise en place de la communication avec le back
        - Traitement et injection des données du back
    - Mise en place des composant des graphes : 4h
        - Création des graphiques à partir de amCharts
        - Mise en place de la communication avec le back
        - Traitement et injection des données du back
    - Essaie de résolution de bug sur les states :4h

Back-End

- Mise en place des routes pour la visualisation : 4h
    - Création des model et contrôleur pour la visualisation
    - Création des routes pour get les données

Utilisation de Github

Nous avons utilisé Github pour avoir accès au versionning de notre projet. Pour son utilisation, nous avons mis en place des branches en fonction des tâches effectuées.

Après nous avons utilisé principalement la branche dev pour le maintien du projet, sur laquelle nous venions merge nos autres branches. Pour le merge, nous avons effectué des pull request pour faire valider notre code avant de l'ajout à dev ou encore à main.

Dans mon cas, j'ai utilisé la branche Visualisation pour réaliser le plus gros de mes tâches.

Solution Choisie

Pour le front-end, j'ai décidé d'utiliser un Template pour nous faciliter le travail et avoir une bonne base sur laquelle avancer. Pour cela, j'ai pris un Template avec ReactStrap ce qui permet d'avoir une base de components fournie par la librairie en plus de ceux du Template.

De plus, l'utilisation d'un template nous assurais d'avoir une base de site déjà responsive et donc d'un gain de temps.

Mise en place de la carte

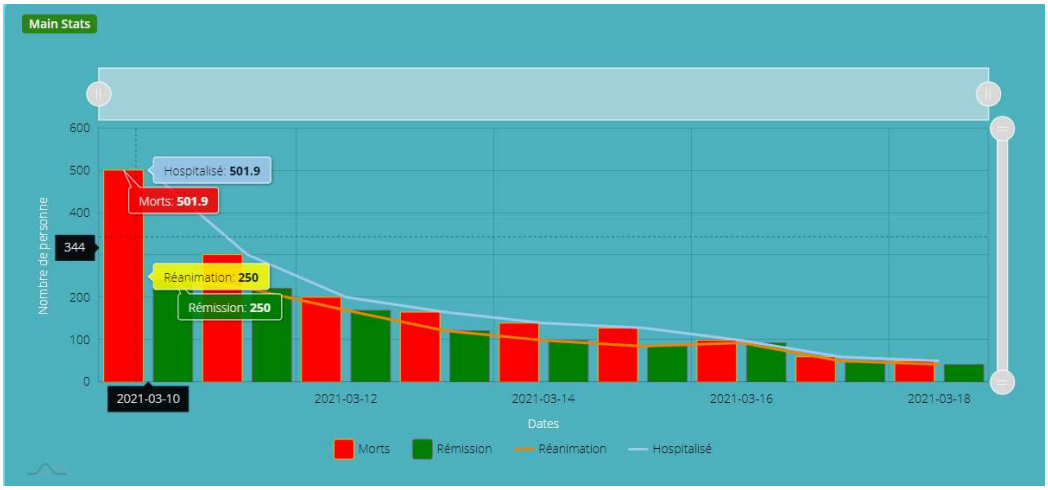
Pour afficher des données sur une carte, nous avions plusieurs solutions en passant par une google map (ou tout autre carte du style), utiliser une image ou encore une carte vectorielle.

Dans notre cas j'ai décidé d'utiliser la bibliothèque amCharts qui permet de réaliser des graphiques ou des cartes vectorielles. Nous avons utilisé la version 4 (am4Charts) qui comprend nativement une image svg (carte vectoriel) de la France avec ses départements.

De plus, cela nous permet d'afficher rapidement des informations dessus comme les noms des départements en survolant la carte, ou des informations dans chaque département comme le nombre de cas.

Mise en place des graphes

Ayant utilisé amCharts pour la carte, j'ai donc aussi utilisé celui-ci pour afficher des graphiques de données.
Nous aurions aussi pu implémenter cela avec d'autre bibliothèque react comme les charts de ReactStrap.



Difficulté rencontrée

Au niveau du back-end, j'ai rencontré des difficultés pour la réalisation de certaine requête où il fallait réaliser des tris sur les données pour en former de nouvelle.

Dans notre cas, il fallait faire la somme du nombre de personnes contaminées en fonction de la date et de la classe d'Age commune et des départements. Cela permet de récupérer des données sur les cas dans toute la France pour avoir une évolution des cas dans les hôpitaux.

Ses filtres m'ont posé un problème pour la mise en place et j'ai perdu énormément de temps dessus.

De plus ayant l'impossibilité de lancer le back sur mon ordinateur comme un autre membre du groupe, et n'ayant pas la possibilité de changer totalement le back pour tout le monde, je ne pouvais dons pas tester mes routes et mettre en place les datas sur le front.

Au niveau du front-end, la principale difficulté rencontrée a été la compréhension de la bibliothèque amCharts pour l'utilisation de la map. J'ai eu du mal pour trouver comment afficher des éléments en fonction des coordonnées vectoriel des polygons sur la carte.

Les polygons représentent nos départements et je voulais afficher le nombre de cas confirmé dans chaque département en statique. Pour cela, j'avais dans un premier temps voulu afficher le tooltips (information affichée durant le hover) en permanence, mais le rendu visuel était illisible. J'ai alors cherché dans la documentation un autre moyen que je n'ai pas trouvé.

La solution a été de récupérer les coordonnées centrales des polygones calculés en amont par une fonction de la bibliothèque pour afficher des informations avec la longitude et la latitude calculée.

De plus n'ayant pas la possibilité de tester comme voulu les données, nous avons rencontrer un problème avec l'implémentation des données dans les graphiques et dans la maps, les state ne s'actualisant pas correctement à la récupération des données.
Il y a donc la logique dans les components mais les données ne sont pas traitées. Aucun membre du groupe n'a réussi à résoudre le problème.

Fâcheusement, n'ayant aucune possibilité encore d'avoir un back fonctionnel je ne pouvais pas tester les données et développer comme je le voulais et suis extrêmement déçus de ce projet.

Code

```
let newData =  this.state.listCasDepByDate? this.state.listCasDepByDate: ["0"];
// Set up label series to populate
polygonSeries.events.on("inited", function () {
    newData.map(incident =>{
      let polygon = polygonSeries.getPolygonById("FR-"+incident.dep);
      if(polygon){
        let label = labelSeries.mapImages.create();
        label.latitude = polygon.visualLatitude;
        label.longitude = polygon.visualLongitude;
        label.children.getIndex(0).text = incident.P;
      }
    })
});
```

J'aime bien cette fonction elle me permet d'afficher tous les éléments sur la carte, ce qu'y m'avais posé un problème avec l'utilisation. Elle est loin d'être parfaite, on pourrait changer l'égalité du newData.

```
<Col lg={3}>
  <Widget
    className="bg-transparent"
    title={<h5 className="fw-semi-bold">Your statistics</h5>}
  >
    <p>
      Location : <strong>{this.state.localisation}</strong>
    </p>
    <p>
      <span className="circle bg-default text-white"><FontAwesomeIcon icon={faVirus} /></span>
      {this.state.CasDepByDate ? this.state.CasDepByDate.P: "x"} case
    </p>
    <div className="row progress-stats">
      <div className="col-md-9 col-12">
        <h6 className="name fw-semi-bold">Number of hospitalized cases</h6>
        <Progress
          color="info"
          value={this.state.CasDepByDate && this.state.HospitalByDepAndDate ?(this.state.HospitalByDepAndDate.hosp/this.state.CasDepByDate.P *10
          className="bg-custom-dark progress-xs"
        />
      </div>
      <div className="col-md-3 col-12 text-center">
        <span className="status rounded rounded-lg bg-default text-light">
          <small>
            <AnimateNumber value={this.state.HospitalByDepAndDate ? this.state.HospitalByDepAndDate.hosp: 0} />
          </small>
        </span>
      </div>
    </div>
    <div className="row progress-stats">
      <div className="col-md-9 col-12">
        <h6 className="name fw-semi-bold">Number of cases in resuscitation</h6>
        <Progress
          color="warning"
          value={this.state.CasDepByDate && this.state.HospitalByDepAndDate ?(this.state.HospitalByDepAndDate.rea/this.state.CasDepByDate.P *100
          className="bg-custom-dark progress-xs"
```

Pour un élément à changer, je dirais cette partie de l'affichage du Dashboard, il aurait fallu créer un component et le rappeler avec les données à afficher plutôt que d'avoir une répétition de code comme c'est actuellement le cas.