



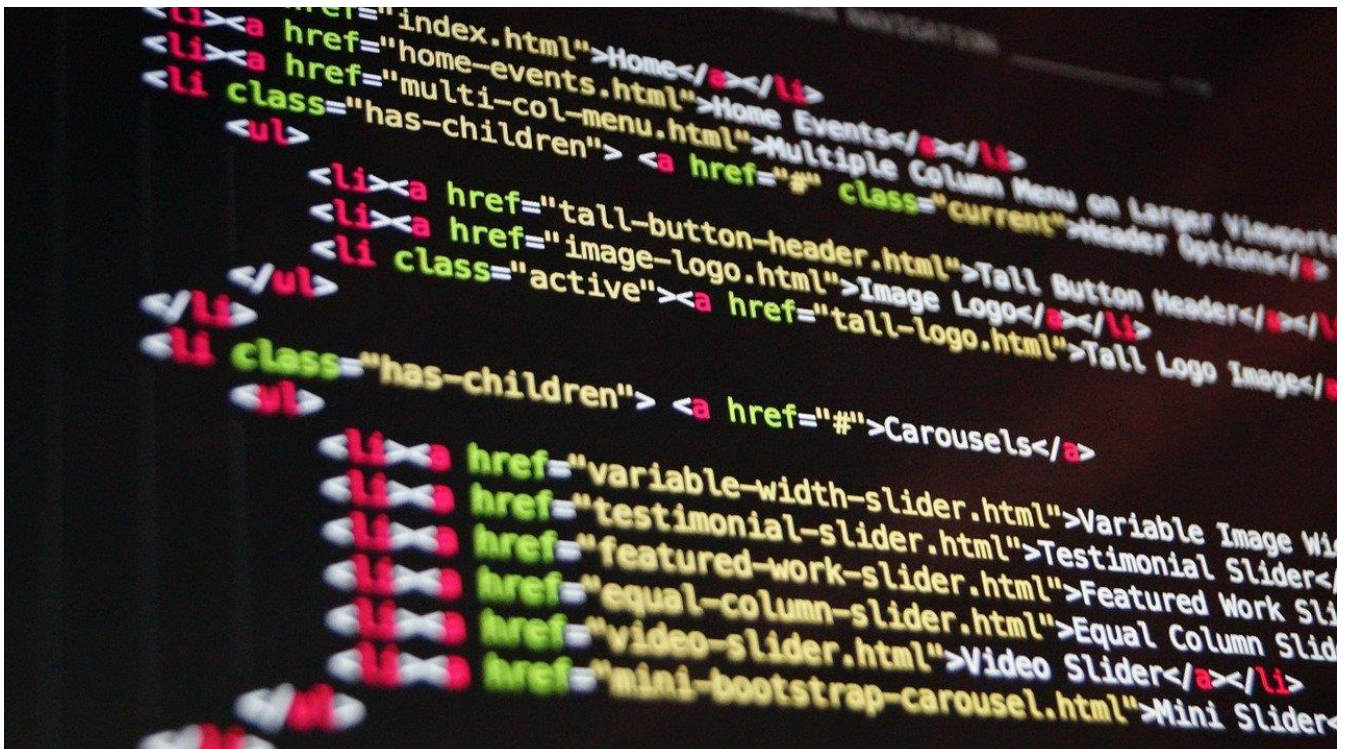
**POLYTECH**<sup>®</sup>  
NICE-SOPHIA



Université  
Nice  
Sophia Antipolis

## Rapport de projet

### Programmation Web Client Side



*Hoareau Grégory*

## Tâches effectuées

Au cours du développement de ce projet, j'ai pu travailler sur différentes tâches (voir tableau en annexe pour le temps de développement des tâches), aussi bien sur des tâches aidant au développement de l'interface utilisateur que sur des tâches permettant de faire évoluer le serveur.

La première tâche dont je me suis occupé est le développement du composant « *CovidCounter* ». Ce composant permet l'affichage de valeurs en lien avec la Covid-19, mise à jour toutes les heures. Les valeurs affichées par ce composant permettent d'avoir un aperçu rapide sur l'évolution de la crise sanitaire liée à la Covid-19 en France. En effet ce composant étant présent dès la page d'accueil du site, il n'est pas nécessaire de naviguer et de faire des recherches complexe pour obtenir ces informations.

Je me suis également occupé de développer la fonctionnalité permettant de récupérer la localisation de l'utilisateur. Ces données sont par la suite stockées dans un contexte afin qu'elles puissent être récupérer par tous les composants présents dans la page.

Enfin, j'ai pu développer un système de pagination pour les données renvoyées par le serveur. Bien que cela reste une fonctionnalité du serveur, cette dernière est très utile pour la partie client. La pagination permet de pouvoir récupérer les données du serveur par lots.

## Solutions choisies

Dans cette partie nous allons détailler un peu plus deux des fonctionnalités présentées dans la partie précédente est expliquer les choix de développement effectué.

Tout d'abords, nous allons détailler la mise en place de la récupération de la position de l'utilisateur. Nous avons voulu utiliser la position afin de pouvoir afficher des données sur graphique qui lui seront plus utile. En effet, les données cherchées par les utilisateurs sont généralement celles concernant son environnement proche. Pour récupérer la position on utilise l'API du navigateur. Il existe d'autre API cependant elles sont moins précises

que l'API du navigateur. Cette API permet donc de minimiser l'erreur sur la position de l'utilisateur

Enfin nous allons parler un peu plus en détail de la pagination. Nous avons choisi de mettre en œuvre ce système pour 2 raisons principal. La première est que le temps de réponse du serveur pour renvoyer toutes les données d'une catégorie peut être assez long (cela peut prendre environ une dizaine de secondes). Et la deuxième est qu'un grand nombre de données à traiter par l'interface utilisateur augmenterait les temps de latences de cette dernière et ainsi nuirait à l'expérience utilisateur.

Pour compenser cela, nous avons choisi de faire en sorte que le serveur renvoie les données par lot. De cette façon, le temps de réponse du serveur est ainsi grandement diminué mais aussi le temps de traitement des données est réduit puisque le nombre de données à traiter est réduit.

## Difficultés rencontrées

La première difficulté rencontrée est un problème d'ordre technique venant de l'API utilisée pour afficher les valeurs dans le compteur. Originellement, la valeur affichée dans le compteur était le nombre de cas confirmé en France. Cependant, cette valeur a été complètement retirée des données renvoyées par l'API. Pour compenser cela, nous avons choisi d'afficher d'autres valeurs et d'en afficher plusieurs comme ça si l'une d'entre elle venait aussi à être retiré des données renvoyées par l'API, il resterait encore d'autre valeurs à l'écran.

La deuxième difficulté principale rencontrée est que l'API de géolocalisation du navigateur n'est pas disponible sous toutes les versions de tous les navigateurs internet. De plus, cette API nécessite l'accord de l'utilisateur et peut donc être refuser. Il est donc possible que l'API ne soit pas disponible pour récupérer la position de l'utilisateur. Pour cette raison nous avons choisi de coupler l'API du navigateur avec une autre API qui permet d'approximer la localisation de l'utilisateur avec son adresse IP.

Néanmoins, cette deuxième API comporte 2 problèmes. Le premier est que cette API est moins précise. Cependant puisqu'on se sert de la localisation pour déterminer la région dans laquelle habite l'utilisateur, ce manque de précision peut être considéré comme négligeable.

Le second problème est que, de nos jours, l'adresse IP peut facilement être masqué, notamment par l'utilisation de VPN. Ce qui peut donc mener à afficher des données qui ne sont pas proche de l'utilisateur, voire inexistante puisque certains VPN passe par des pays étrangers. C'est pour cette raison que l'on privilégie l'utilisation de l'API du navigateur et dans le cas où cette API n'est pas accessible (soit à cause du navigateur soit à cause du refus de l'utilisateur) alors on passe par la seconde API afin de pouvoir de géolocaliser.

## Code

Dans cette partie, nous allons commenter un composant et une fonction que j'ai développés. Pour le composant, il s'agit de « *CovidCounter* » et pour la fonction, il s'agit de « *findUserLocation* » se trouvant dans le fichier *App.js*.

Pour le composant « *CovidCounter* », je considère que ce composant fait partie de ceux qui ont besoin d'être amélioré. En effet, ce composant va se rendre 3 fois lors du premier lancement du site. Puis à chaque fois qu'une des valeurs affichées dans le compteur va se mettre à jour, le compteur va se rendre à nouveau. Puisqu'il y a 2 valeurs affichées, le compteur va donc se rendre 2 fois à chaque mise à jour.

De plus, se compteur ayant pour rôle d'afficher les valeurs le plus à jour possible, il doit régulièrement faire des appels à l'API afin de se mettre à jour. Or plus le temps de rafraîchissement des valeurs est petit, plus le composant va se rendre souvent, cela pouvant entraîner un ralentissement de l'exécution du site. Et plus le temps de rafraîchissement est grand, moins les valeurs affichées seront à jour et donc le composant remplira moins bien le rôle qui lui a été assigné.

Quant à la fonction « *findUserLocation* », elle fait, selon moi, partie des bonnes fonctions de ce projet car elle couvre toutes les possibilités. En effet, comme vu dans la section précédente si l'API du navigateur n'est pas utilisable pour une raison quelconque alors on peut quand même avoir une estimation de la localisation de l'utilisateur. Et puisque l'on se sert essentiellement de la région de l'utilisateur et non de sa position précise.

Cependant, bien que cette seconde méthode marche correctement, elle possède un défaut majeur qui est de pouvoir être faussé très facilement car

il est assez commun que des personnes utilise de VPN pour masquer leur adresse IP. C'est pour cette raison que l'on privilégie l'utilisation de l'API géolocalisation d'HTML5.

## Annexe

Identifiant GitHub	Gregory-Hoareau
Temps de développement du compteur	7 h
Temps de développement de la géolocalisation	5 h
Temps de développement de la pagination	4 h