



DES CARTES

RAPPORT DE PROJET

PWEBC

Logan
Tann

Ibrahime
Ahbib

Sebastien
Ramirez

Groupe 209
DUT Seconde Année

SOMMAIRE

Sommaire	2
Travail réalisé :	2
Services web développés :	3
Le Formulaire de connexion.	3
L'affichage de la carte interactive et ses éléments :	4
La fonctionnalité de recherche :	5
L'affichage des pistes cyclables sur la carte	6
Structure du projet React & dépendances :	7
Procédure d'installation :	7

TRAVAIL REALISE :

Le projet Pweb-C consistait à réaliser une carte interactive, les groupe étant libre d'implémenter leur idée de sujet.

Nous avons eu au départ l'idée de faire une carte des IOT vulnérables avec shodan, mais sa réalisation étant trop difficile, nous nous sommes tournés vers une google maps pour vélo, que nous avons appelé fièrement DesCartes.

Le vélo est de plus en plus utilisé sur Paris, et avec l'émergence des pistes cyclables, avoir une carte interactive dédiée aux cyclistes nous semblait une bonne idée. Non seulement cet outil est pratique pour préparer son itinéraire vélo, mais il pourrait aussi se rendre utile à la municipalité pour avoir une vue d'ensemble de la couverture vélo sur la ville.

Nous avons choisi React car faire ce projet est une très bonne opportunité pour apprendre ce Framework qui devient de plus en plus indispensable dans le monde professionnel. Choisir react apporte quelques difficultés supplémentaires Sébastien fera même un stage sur react.

SERVICES WEB DEVELOPPES :

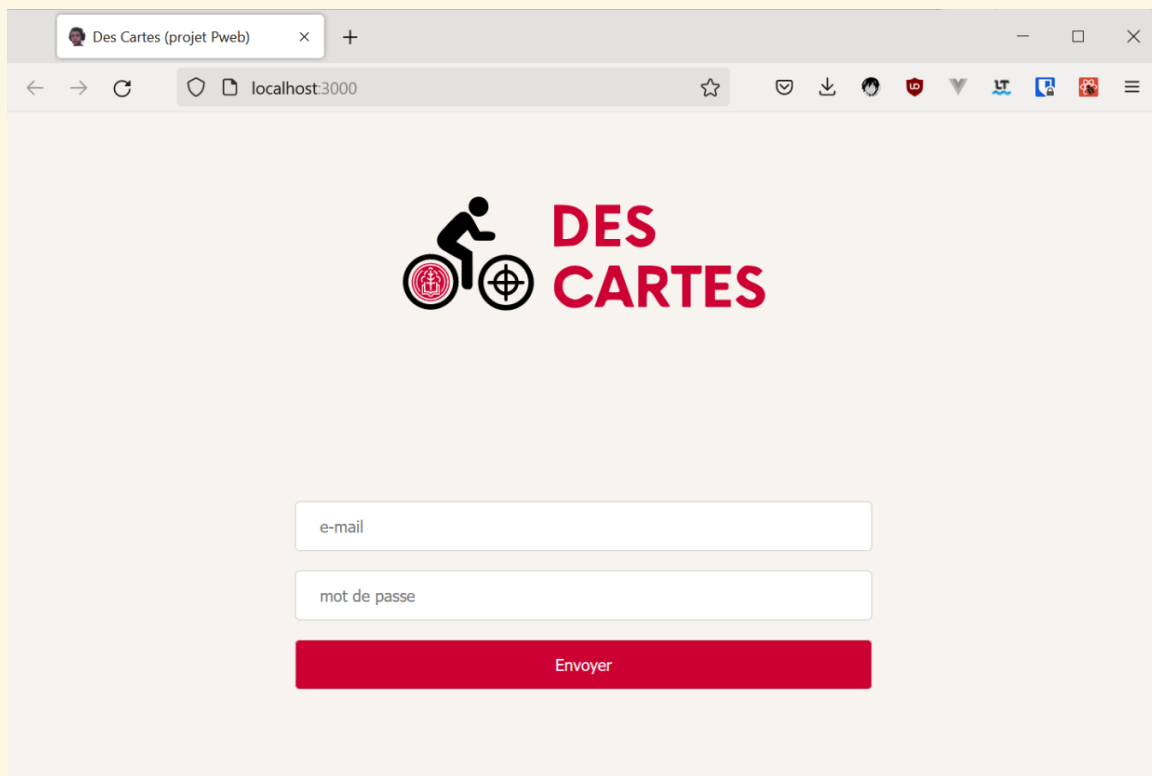
Le Formulaire de connexion.

Le formulaire de connexion s'occupe de récupérer l'e-mail et le mot de passe de notre application.

Le navigateur se charge de vérifier la validité des champs (required + type) ; bien que ces cas soient gérés par le backend en supplément.

À l'envoi du formulaire, l'évènement est récupéré pour faire une requête POST asynchrone sur un backend que nous avons créé. Si le résultat et code d'erreur http sont corrects, alors l'utilisateur est authentifié, sinon un message d'erreur apparaît.

Un identifiant de démo est : **a@a.fr** / mdp : **a**

A screenshot of a web browser displaying the login page for 'DES CARTES'. The browser's address bar shows 'localhost:3000'. The page features a logo with a stylized figure on a bicycle and the text 'DES CARTES' in red. Below the logo, there are two input fields: 'e-mail' and 'mot de passe'. A red button labeled 'Envoyer' is positioned below the password field.

L'affichage du formulaire de connection

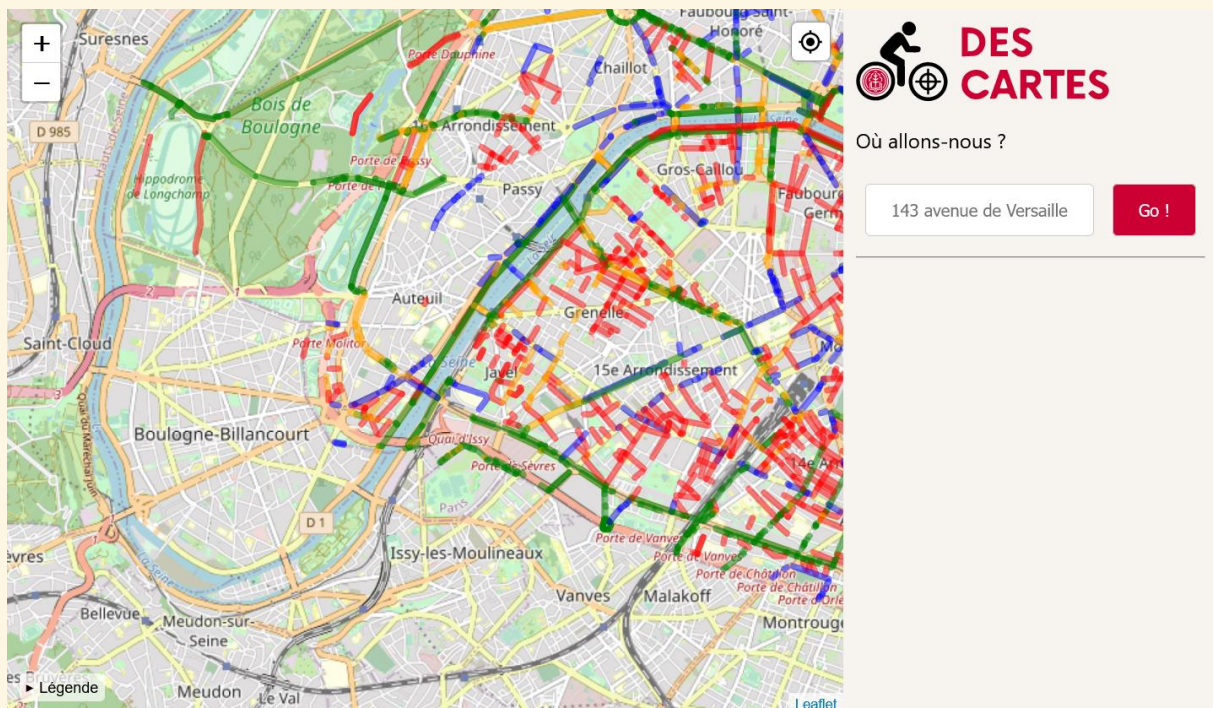
L'affichage de la carte interactive et ses éléments :

Pour afficher la carte, nous avons utilisé la librairie react-leaflet, qui est un binding de leaflet (librairie conseillée par le sujet) pour le framework react.

Celui-ci nous donne un couple de fonctionnalités d'interactions déjà existantes, tel que déplacer ou zoomer la carte.

L'apprentissage de react et la compréhension de la documentation de react-leaflet (assez légère en explications) a été une de nos difficultés.

Nous sommes tout de même parvenus à afficher divers éléments d'interaction sur la carte grâce à son API, tel qu'un bouton de géolocalisation, une légende, et bien évidemment, les lignes colorées représentant la piste cyclable ainsi que les marqueurs interactifs.



L'affichage de l'application après connexion

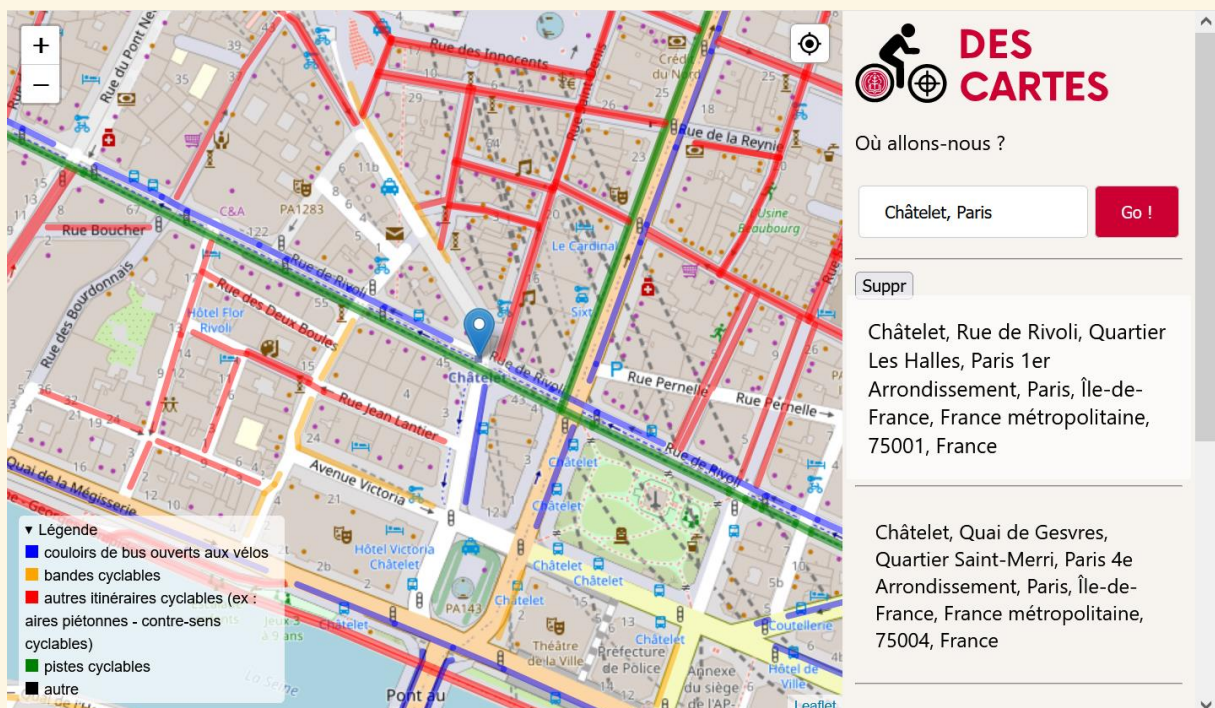
La fonctionnalité de recherche :

Nous avons créé une barre de recherche sur le côté. À l'envoi du formulaire, le résultat de la recherche s'affiche.

Pour ce faire, nous avons dû faire des appels ajax à l'API de **Nomati**, récupérer et effectuer un traitement sur les résultats pour définir les événements appropriés.

Au clic sur un résultat, un marqueur interactif s'occupe d'afficher l'emplacement de l'entrée cliquée et de centrer correctement la carte.

En capture d'écran est également affiché la légende par degré de dangerosité. Une piste cyclable propre sera affichée en vert, tandis qu'une bande cyclable à contresens sera affichée rouge.

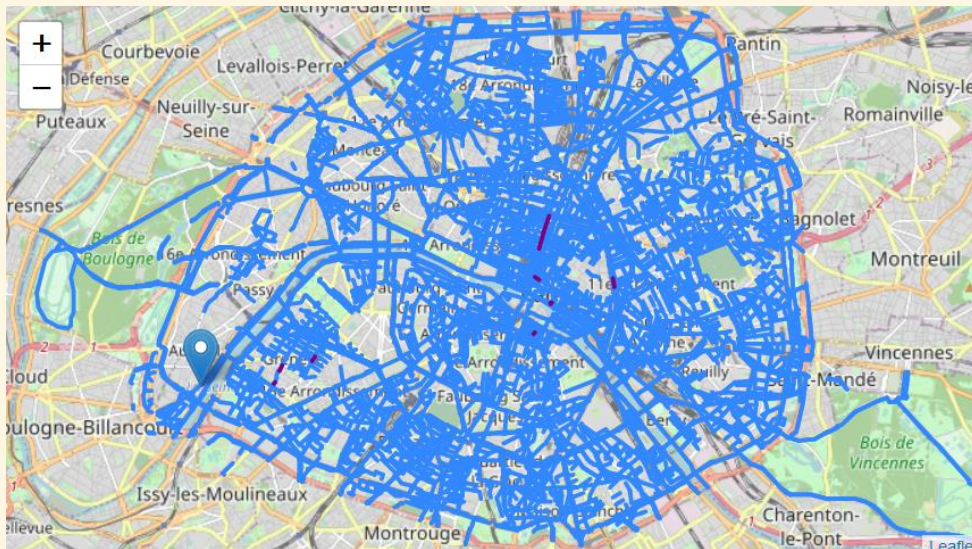


L'affichage du marqueur du résultat de la recherche.

L'affichage des pistes cyclables sur la carte

Probablement la partie la plus compliquée du projet : afficher des traits est une chose, mais devoir traiter de manière optimisée le résultat de l'API de paris (pour obtenir les positions GPS des pistes) en était une autre.

En particulier, il existe plus de 12 000 entrées de pistes cyclables à Paris. Ainsi, bien que récupérer l'ensemble des résultats sous forme de GeoJSON et afficher sur la carte est facile, nous rencontrons un bien vilain problème de performance et d'apparence sur notre carte :



L'affichage des 12 000 pistes ressemble à un gros pâté bleu.

Il fallait donc limiter les résultats à un nombre correct (par exemple, 1 500) tout en veillant que l'affichage soit correctement distribué sur toute la surface visible de la carte. Mettre en place un tel algorithme demande énormément de travail ainsi que de puissance de calcul. Heureusement, l'API open data de Paris nous propose une option qui permet de faire ce que nous recherchons exactement.

Voici les étapes du fonctionnement finale de l'affichage de notre carte :

- Au chargement de la carte ou lorsque celle-ci est déplacée/ redimensionnée
- Obtenir le centre de la carte et sa viewbox en positions GPS
- Envoyer une requête ajax à l'API de paris en spécifiant les paramètres corrects
- Traiter les résultats :
 - Nous avons dû normaliser les résultats (résultats sur le 2^e niveau hiérarchique, données inutiles ou mal ordonnancement entre latitude et longitude)
 - Nous avons dû également modifier l'objet afin d'assigner une couleur selon son type de piste cyclable
- Afficher sur la carte en tant que composant react `<PolyLine>` et veiller à adopter une épaisseur adaptée au niveau d'agrandissement de la carte.

STRUCTURE DU PROJET REACT & DEPENDANCES :

Dossier public : tous les fichiers statiques

Dossier src :

- Index.js : point d'entrée de l'application
- App.js : composant d'affichage principal
- Dossier Components/ : composants secondaires
 - AppMap.jsx : la carte leaflet
 - mapComponents/ : les composants affichés au sein de AppMap
 - Légende.jsx : la boîte de légende
 - Localisation.jsx : petit bouton pour la géolocalisation
 - PistesCyclables.jsx : affiche les pistes cyclables en faisant appel à ParisOpenData
 - Recherche.jsx : marqueur interactif du résultat de la recherche
 - LoginForm.jsx : le formulaire de connection
 - ResearchBar.jsx : le formulaire de la barre de recherche et ses résultats
- Dossier Images/ : contient les fichiers svg utilisés par l'application
- Dossier Styles/ : contient les styles CSS utilisés par l'application
- Utils/parisOpenData.js : Contient le code qui fait l'étape décrite précédemment pour afficher les pistes cyclables depuis l'API d'open data de paris. Le fichier le plus intéressant du projet après les composants.

Les dépendances utilisateur sont visibles dans le package.json :

- Par défaut : React (le framework) et react-script + react-dom (indispensables pour démarrer le site sur un serveur local)
- Leaflet et React-leaflet : le binding leaflet pour react dépends de la librairie originale
- Rien d'autre 😊

PROCEDURE D'INSTALLATION :

Pour installer, il faut avoir NodeJS (et NPM qui est installé par défaut). Notre projet react devrait fonctionner sur la version Long-Terme 14, même si react devrait être supporté sur n'importe quelle version encore maintenue de node.

Naviguer ensuite dans le dossier du projet et entrer **npm install** pour télécharger les dépendances.

Pour démarrer sur le navigateur, il faut faire **npm start**.

Pour compiler notre projet en HTML, il faut faire **npm run build**. Attention à bien configurer les paramètres corrects dans le package.json. Une démo compilée est actuellement en ligne et testable à tout moment à l'adresse

<https://logantann.github.io/pweb-pj2>