PROJET WEB DESIGN - AUDAS JESSICA

# Présentation du projet

L’application exploite une base de données en temps réel fournie par JCDECAUX. Les informations sur l’API sont disponibles via ce [lien](https://developer.jcdecaux.com/#/opendata/vls?page=dynamic) . Elle n’exploitera que des stations qui se trouvent à Paris.

Elle a été pensée afin de permettre une bonne expérience utilisateur en lui facilitant l’acces aux informations importantes : trouver une station et un itinéraire.

# Les sources

Se présentent de la façon suivante :

* App
  + Script : *scripts JS utilisés*
  + Views : *vues utilisées dans l’application*
* Assets
  + CSS
  + Fonts
  + Img
  + JS *sources JS*

# Detail

* 1 config utilisant $routeProvider (dans appModule.js)
* 1 filter personnalisé 'ago' (dans appModule.js)
* 4 controlleurs :
  + StationCtrl, qui dirige l'affichage de l'ensemble des stations est appelé également pour permettre l'auto completion
  + MapCtrl, qui gère le positionnement d'une station sur une carte
  + ItinCtrl devait permettre de tracer un itinéraire d'une station vers une autre, mais l'API google direction ne place pas les stations. Finalement, il permet de placer les deux stations sur une carte et de centrer la carte au milieu de celles ci
  + StatsCtrl génère des statistiques sur l'ensemble de la base de données
* 11 vues :
  + index.html : la vue principale qui contient le <head> et le <body>. Elle permet d'appeler les différentes vues, comme la navbar, le footer et le main (qui redirigera vers une autre vue)
  + main.html : page d'accueil du site, ne contient pas de directive angular mais dispose d'une div qui sera affichée uniquement en version mobile (largeur <480px)
  + itin.html : permet de choisir les stations d'arrivée et départ.

Attention, le champ de proposition ne founit que l'ID de la station. Je n'ai pas réussi à afficher uniquement le label

* itinMap.html : permet de visualiser les deux stations sur la carte. Il est parfois nécessaire de recharger la page pour afficher les markers Google maps correctement.
* stations.html : permet de trier les stations et d'afficher uniquement celles qui correspondent aux paramètres indiqués dans le formulaire
* details.html : affiche une station sur la carte et indique ses caractéristiques
* statistiques.html : affiche des statistiques sur l'ensemble de la base de données
* info.html : affiche des informations sur le site en lui-meme .
* 404.html : affiche un message d'erreur
* footer.html
* navbar.html

- directives :

- ng-view (utilisé dans index.html)

- ng-include (utilisé dans index.html)

- ng-model (utilisé dans itin.html puis dans stations.html)

- ng-true-value (dans stations.html)

- ng-false-value (dans stations.html)

- ng-repeat (utilisé dans itin.html puis dans stations.html)

- ng-href (utilisé dans itin.html puis dans itinMap.html)

- ng-show (utilisé dans itinMap.html puis dans stations.html, details.html)

- ng-class (utilisé dans itinMap.html puis dans stations.html, details.html)

- ng-if (utilisé dans itinMap.html)

- ng-click (utilisé dans stations.html)

# Notes et améliorations possible

* Le tracé de l’itinéraire ne fonctionne pas. En effet, l’API renvoie une erreur
* Pour tester les cas particuliers, il est possible d’utiliser la station 31705 qui est fermée et ne dispose pas de vélos.
* Afin de garantir les performances, j’ai parfois limité l’explotation de la base de données à 200 stations

Il est possible d’imaginer proposer à l’utilisateur de trouver les stations autour de lui pour deposer/prendre un vélo