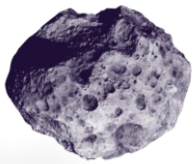
MAKINA CORPUS

GEO TREK



Synthèse 2j d'étude
sur les développements réalisés et à réaliser
Février 2022

Paris
Nantes
Toulouse
Bruxelles

Makina Corpus

Siège social
11 rue du Marchix
44000 Nantes

Établissement de Toulouse
52 Rue Jacques Babinet
31100 Toulouse

+33 (0)9 70 33 21 50

MAKINA CORPUS

Table des matières

1 Contexte

Un groupement de commande a été initié en 2019 pour réaliser différentes évolutions sur l'outil Geotrek-admin. Un des sujets majeurs concernait les performances de l'outil. Ces évolutions sont suivies sur le projet GitHub https://github.com/GeotrekCE/Geotrek-admin/projects/2.

À ce jour, différentes évolutions ont déjà été réalisées, notamment l'amélioration des calculs altimétriques qui se comportaient de manière empirique en fonction du nombre de tronçons https://github.com/GeotrekCE/Geotrek-admin/issues/1493

Différents travaux ont été initiés et non terminés, ce que nous allons détailler ici

2 Rappel des problématiques

- Volume des données
- Affichage et utilisation

Suite à cette étude de deux jours, nous avons repris le travail en cours et listé tous les points qui devraient être abordés dans cette session de développement :

2.1 Les listes de données des modules

Actuellement les listes de données ne sont pas dynamiques, et bien qu'elles semblent paginées visuellement elles ne sont pas paginées côté serveur : elles sont toute envoyées en un bloc au navigateur.

Sur les instances contenant plusieurs dizaines de milliers d'éléments (surtout les tronçons) on a un comportement très lourd à l'affichage, et qui nécessite d'avoir un ordinateur et un client web très (trop) performant.

Solution :

- Mettre en place la pagination côté serveur qui sera interrogé page par page
- Travail déjà initié par Gaël à reprendre
- Conséquence : Il ne sera plus possible d'afficher en surbrillance l'élément de la liste lors d'un survol sur la carte
- Contres propositions :
  - afficher une « popup Leaflet » sur la carte lors d'un survol / clic d'élément pour proposer un lien d'accès à l'élément
  - ajouter dynamiquement l'objet à la liste pour l'afficher

- Déterminer la page de l'élément pour l'afficher directement et ainsi conserver la fonctionnalité de survol

- Gestion du cache à améliorer :
  - Les listes étant mises en cache sur la base de la dernière date de modification et une expiration de huit jours, il serait intéressant d'améliorer cette gestion pour être plus fin et de ne pas expirer les cache

2.2 Les données cartographiques affichées (GeoJSON)

Actuellement les GeoJSON sont affichés intégralement sur la carte. Le volume de données a un impact sur :

- la durée de téléchargement
- la durée d'affichage

Solution :

- Tuiler les données à afficher
- Travail initié par Gaël à reprendre
- Gestion du cache à améliorer (voir point 2.1)
- Étudier la possibilité d'utiliser des tuiles MVT (Mapbox Vector Tiles)
  - Affichage plus rapide
  - Sur les éléments aux styles complexe (statuts), cela applique un style à la volée, sans rajouter des éléments dans le DOM qui alourdi le chargement de la carte

2.3 Les tracés linéaires topologiques

Le tracé d'itinéraires utilise une version filtrée des tronçons n'incluant pas les brouillons. Cela nécessite de recharger les données encore une fois après une modification de la base tronçons.

- Exemple : Base de données de 100 000 tronçons.
  - Processus pour dessiner un itinéraire :
    - ajouter un ou plusieurs tronçons
    - le cache est invalidé, l'affichage de la liste prend environ 10minutes
    - se rendre sur la page 'ajouter un itinéraire'
    - le cache ne peut pas être utilisé, car on affiche la liste en excluant les tronçons 'brouillons', et nécessite environ 10 min de plus pour tracer un itinéraire.

Solution : Déplacer les tronçons 'brouillons' dans une nouvelle couche de données, en permettant bien sûr d'afficher la totalité sur les cartes. On pourrait les mettre dans une 'sous-vue' du module tronçons, comme les statuts ou la signalétique.

- le graph.json

C'est un fichier mis à jour et téléchargé par l'outil pour le tracé de linéaires topologiques et permettant au client web (navigateur) de réaliser le routage et le tracé.
Ce fichier bien que mis en cache est lui aussi de nouveau généré à chaque modification de tronçon. Son temps de génération et téléchargement est lié au volume des tronçons.

Solution :

Déplacer le routage de l'outil côté serveur, en utilisant des outils comme PGRouting
On pourrait toujours afficher le réseau de tronçon de manière tuilée et aimanter dessus pour le tracé.

On perdrait légèrement en réactivité, mais on gagnerait au passage la résolution du problème du tracé qui change lors d'une modification lorsque qu'un plus court chemin est trouvé (cf. problèmes topologiques). De plus, il ne serait plus nécessaire d'enregistrer les 'points de passage obligatoires'.

2.4 La mise à jour des bibliothèques cartographiques

Les bibliothèques cartographiques Leaflet sont tous en 0.7 sur Geotrek. Pour utiliser le tuilage, GeoJSON ou MVT, il est nécessaire de toutes les mettre à jour en version 1.x

Solution :

- Reprise du travail initié par Gaël
- Makina Corpus impliqué sur le sujet pour remettre à jour ces bibliothèques dans les règles de l'art (CI / test E2E / documentation / publications sur le registre NPMJS)

3 Conclusion et plan d'action

D'après les développements déjà réalisés et à reprendre, et dans le but de ne pas repenser ce qui avait déjà été acté et décidé, voici le plan d'action que nous proposons, par étapes :

1. Mettre en place la pagination côté serveur sur les listes de données
2. Utiliser Django-RestFramework-GIS pour générer les GeoJSON
   - premier pas vers le tuilage qui améliorera les performances, car la génération du GeoJSON se fera en un coup côté Base de données PostGIS, et pas élément par élément dans le code Python
3. Séparer les tronçons 'brouillon' dans une nouvelle sous-liste du module tronçon (nouveau modèle de données)

4. Mettre à jour les bibliothèques Leaflet (cette étape est déjà initiée et peut se dérouler en parallèle des premières étapes, du moins sur la partie des bibliothèques en elle-même et pas de leur utilisation dans Geotrek)
5. Mettre en place le tuilage des données cartographiques (en fonction de la faisabilité rapide, GeoJSON ou MVT)

Et après ?

- Étudier et optimiser la gestion du cache de Geotrek pour s'affranchir de l'expiration
- Lancer une réflexion globale sur le tracé d'itinéraire et du déplacement du routage côté serveur. (PGRouting, gestion topologique)
- Faire une passe générale sur les différentes vues pour optimiser les requêtes de base de données, notamment en fonction des champs à afficher configurés dans les réglages. (cf derniers développements de Célia)

4 Quelques alertes

- Les développements réalisés pour pouvoir choisir les champs à afficher dans les listes et export risquent de compliquer la reprise du code déjà écrit.

5 Suite de l'étude

Nous allons attendre vos retours sur le plan d'action proposé, puis débuter les étapes dans l'ordre, dans le cadre des vingt-cinq jours de commande.

Nous souhaitons lancer les discussions sur les avantages / inconvénients à utiliser :

- les tuiles MVT
- le routage côté serveur

Ce sont des sujets qui ne pourront certainement pas être étudiés et réalisés dans le cadre de ces vingt-cinq jours mais qui mériteraient d'être tracés et discutés.