

TP programmation web

Réalisation d'une interface de visualisation de tracés GPS

Le but de ce TP est de créer une interface de visualisation de tracés GPS (type itinéraire de randonnée). L'objectif est de charger des tracés GPS sur une carte et de visualiser leurs profils altimétriques. Le fait de déplacer la souris sur le profil altimétrique montre l'emplacement exact sur le tracé.

Le design de l'application est totalement libre. Seules les indications du besoin technique sont définies.



Exemple de rendu

Données

Voici les données disponibles :

- Deux fichiers KML (des tracés GPS qui contiennent des points x, y, z)

Note : attention au copiés-collés ! Bien que ce soit un travail très cadré, et que vous pouvez travailler à plusieurs, cela ne signifie pas que vos fichiers doivent être identiques entre vous. Il est même ~~difficile~~ improbable que ça puisse être le cas. Si jamais vous êtes bloqués, ou si vous avez des questions, n'hésitez pas à passer en L308 ou à m'envoyer un mail.

Création de la base de données

Créez une base de données pour stocker les informations des tracés GPS. A minima, il doit y avoir une table et chaque enregistrement doit avoir un titre et le nom du fichier associé. Les données des points ne seront pas stockées en base.

Création de la page d'accueil (PHP)

Votre page d'accueil doit contenir au minimum

- Le nom de votre application (trouvez-en un !)
- Un formulaire contenant :
 - une liste des randonnées de votre base (deux éléments). Chaque option doit afficher le titre de la randonnée, mais sa valeur doit être l'identifiant correspondant
 - un bouton de validation

- Une carte

Création du webservice (PHP)

- Votre service web doit lire la base de données, extraire les informations d'une randonnée, et afficher une liste des points de cette randonnée, au format JSON. Pour cela :
 - Connectez-vous à votre base
 - Faites une requête SQL pour extraire les infos d'une randonnée (testez avec la 1^e rando)
 - Avec ces infos, ouvrez et lisez le fichier correspondant (avec la fonction `file_get_contents` en PHP)
 - Le fichier KML est au format XML, il faut donc convertir en tableau PHP (en passant par une conversion JSON). En PHP :

```
$xml = simplexml_load_string($content);
$json = json_encode($xml);
$array = json_decode($json, TRUE);
```

- Traitez ce tableau, pour ne garder que ce qui est nécessaire
- Affichez en JSON le résultat (avec `json_encode()` en PHP) :
 - Une clé `titre`, avec le titre
 - Une clé `points`, avec un tableau de points (chaque point doit être un tableau avec x, y, z)

Création de l'interaction (JS)

- Détectez la validation du formulaire, et empêchez l'action par défaut de votre navigateur
- A la place, faites une requête AJAX vers votre service web
 - Passez à votre service une variable `rando`, égale à la valeur de l'option choisie dans votre formulaire
 - Modifiez alors votre service web pour qu'il utilise cette variable (en lieu et place de la 1^e rando utilisée pour vos tests)
- Lors du retour de cette requête AJAX
 - Créez et ajoutez une polyline dans votre appli carto
 - Adaptez l'emprise de votre carte à cette polyline
 - Créez un graphique pour le profil altimétrique
 - Utilisation de la librairie Chart.js (<https://www.chartjs.org/>) ou autre (d3.js)
 - Suivez le tutoriel (<https://www.chartjs.org/docs/latest/getting-started/>)
- Conseils pour la création du graphique et l'interaction
 - Lors de la création du graphique, le second paramètre est un objet
 - Les seules options à modifier sont : `data.labels` (un tableau contenant les labels de chaque donnée, que vous pouvez simuler) et `data.datasets[0].data` (un tableau contenant les altitudes)
 - Cet objet contient également une clé `options` qui contient des clés `hover` et `tooltip` pour définir le mode d'interaction avec la souris (ne pas modifier)
 - Enfin, la clé `onHover` permet d'exécuter une fonction qui sera appelée à chaque interaction avec le graphique. Cette fonction prends deux paramètres: l'événement (`evt`) et un tableau d'objets survolés (`array`), comme expliqué ici: <https://www.chartjs.org/docs/latest/general/interactions/events.html?h=onhover>
 - Utilisez cette fonction pour déplacer un marker sur la carte en récupérant la position du point survolé

- Pour cela, vous aurez sûrement besoin de conserver les points de la polyline créée dans un tableau de points
- Chart.js fournit l'indice de l'élément survolé sous la clé `_index` (commence par underscore)
- Voici un exemple de code de création du graphique (non fonctionnel car des manques)

```
let chart = new Chart(ctx, {
  // type de graphique
  type: 'line',
  // les données
  data: {
    labels: /* labels */,
    datasets: [{
      label: 'Rando',
      backgroundColor: 'rgb(255, 99, 132)',
      borderColor: 'rgb(255, 99, 132)',
      data: /* les donnees */
    }]
  },
  // Les options du graphique
  options: {
    hover: {
      mode: 'index',
      intersect: false
    },
    tooltips: {
      mode: 'index',
      intersect: false
    },
    onHover: (evt, array) => {
      /* lors du survol
      evt = l'événement
      array = un tableau d'objet survolé (un seul objet dans notre cas)
      array[0]._index = indice de l'objet survolé (sa position dans le tableau des
points)
      */
    }
  }
});
```