

Pratique des Techniques Informatiques

BTI IG Développeur d'Application

Session 2010-2012

Romain BELLINA

FICHE DE SYNTHÈSE n°3

« Rencontre sportive »

OBJECTIF DE L'ACTIVITÉ

- ♦ Utilisation d'une API Google,
- ♦ Récupération d'une requête en AJAX,
- ♦ Événement en JavaScript.

SUPPORT DE L'ACTIVITÉ

Lieux de réalisation	Outils utilisés
<ul style="list-style-type: none"> ♦ Mac OS X 	<ul style="list-style-type: none"> ♦ PHP ♦ Javascript ♦ Ajax

COMPÉTENCES CONCERNÉES

- C32 ☒ Développer à l'aide d'un langage de programmation procédural.
 C33 ☒ Maquetter une application, la développer à l'aide d'un langage de programmation événementielle.
 C34 ☒ Développer à l'aide d'un langage de programmation objet.
 C35 ☒ Développer autour d'une base de données relationnelle.
 C36 ☒ Développer dans le cadre d'une architecture client-serveur.

Présentation de l'activité :

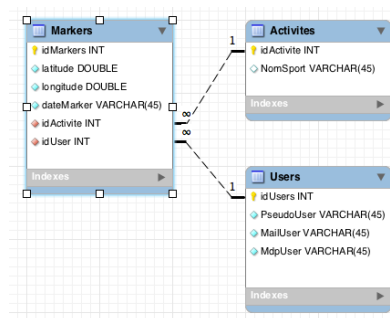
« Sport 3000 » est un site internet qui permet à un inscrit de mettre en ligne leurs position GPS afin de rencontrer de nouvelle personne qui pratique la même activité. Quand une personne se connecte sur le site, une carte « Google maps » apparait avec tous les points déjà enregistré.

Grace à un menu, il peut choisir le sport qu'il veut suivre en particulier.

Pour mettre un marker, l'utilisateur doit indiquer l'adresse où il pratique son activité. L'application récupère l'adresse GPS de ce lieu et affiche le marker sur la carte.

Avec les informations disponibles sur le marker, les autres utilisateurs peuvent prendre contact et ainsi partager leurs passions.

Modèle conceptuel des Données



API Google Maps :

L'API Google Maps est un script édité par Google qui permet aux développeurs web d'utiliser leur code afin d'améliorer nos site internet.

Pour utiliser une API Google, il faut avoir un compte Gmail.com. Avec ce compte, vous pouvez vous enregistrer sur la partie dédiée aux développeurs. Dans cette partie, toute une liste d'API est disponible. Nous allons donc juste activer l'API Google Maps et récupérer la clé de connexion qui nous est fournis.

Pour utiliser le code, il faut insérer une balise `<script></script>` ayant pour source

<http://maps.googleapis.com/maps/api/js?key=AlzaSyAL4PwzGKwFRnuuz3HKxGC1rCBb9cYRhrk&sensor=false®ion=FR>

Cette adresse a pour but de récupérer le script nécessaire au fonctionnement de l'API. Les paramètres passés sont ma clé de connexion, l'activation ou non du sensor qui permet d'avoir des statistiques dans mon espace développeur sur le site de Google et la région où je souhaite utiliser la carte.

Maintenant nous allons définir l'emplacement de notre carte en positionnant une `<div></div>` à l'endroit désiré. Ensuite, avec l'id de la div, le JavaScript va intégrer la carte

Pour cela j'utilise une fonction qui permet d'initialiser les paramètres de la carte et son emplacement.

```
function initialize(){  
    /* Instanciation du geocoder */  
    geocoder = new google.maps.Geocoder();  
    paris = new google.maps.LatLng(48.8566667, 2.3509871);  
    OptionsMap =  
    {  
        zoom: 8,  
        center: paris,  
        mapTypeId: google.maps.MapTypeId.HYBRID  
    }  
    /* Chargement de la carte */  
    map = new google.maps.Map(document.getElementById("map_canvas"), OptionsMap);  
    LoadMarker();  
}
```

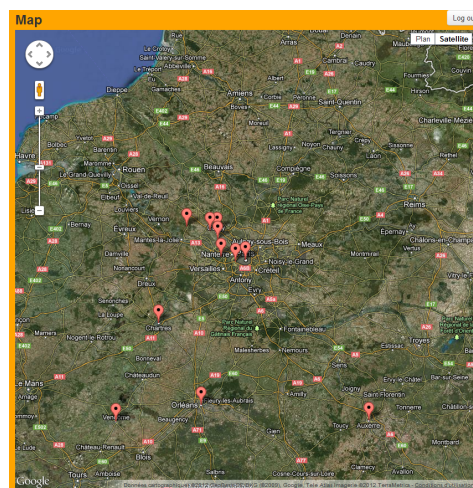
Cette fonction crée 3 objets :

- Un objet Geocoder
- Un objet paris
- Un objet map

L'objet geocoder permettra de transformer une adresse physique en adresse GPS, l'objet paris contient les coordonnées GPS de paris qui vont servir de point par défaut sur la carte et l'objet map représente la carte.

L'ajout d'un marker se fait par la création d'un nouvel objet qui s'appelle Marker. Lors de sa création on passe en paramètre la carte et les positions GPS du point à afficher.

Aspect fonctionnel et technique :

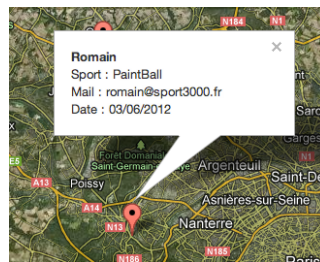


Quand l'utilisateur c'est enregistré sur le site, il arrive sur la page de la carte. Par défaut, tous les markers disponibles dans la base de données sont affichés. Nous avons une requête SQL qui récupère toutes les informations nécessaires a l'affichage du marker et de ses information.

```
SELECT latitude, longitude, idMarkers, NomSport, PseudoUser,  
MailUser,dateMarker FROM Markers m, Activites a, Users u WHERE  
m.idActivite = a.idActivite AND m.idUser = u.idUsers
```

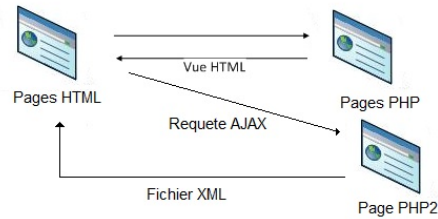
Les informations sont ensuite stockées dans un tableau. Comme l'API de Google ne peut accéder qu'aux informations stockées en JavaScript, une fonction PHP construit un tableau en JavaScript avec les données récupéré. Une fois la fonction terminée, une fonction JavaScript se déclenche pour ajouter les markers à la carte.

En plus de récupérer les coordonnées GPS, la requête SQL récupère le nom, l'adresse mail et la date de mise en place du marker. C'est information sont ainsi ajouté au marker dans une 'infoWindow' qui est rattaché a un marker.



Si l'utilisateur souhaite affiner sa recherche, une champ est disponible sous la carte et permet de sélectionner un sport. Sur le menu déroulant a été mis un 'listener' qui permet de déclencher une action a la modification du menu. Ainsi, en sélectionnant un nouveau sport dans ce menu, les markers s'effaceront et une requête envoyée en AJAX demandera les markers spécifiques au sport demandé.

Une requête AJAX est un appel via une fonction d'une page PHP pour récupérer ou envoyer des informations.



Une fois la réponse reçue, nous traitons le fichier pour récupérer les informations demandées et nous modifions la page avec des fonctions en JavaScript

Pour ajouter un marqueur, un autre menu est disponible. L'utilisateur y saisit une adresse postale pour définir l'emplacement de son marqueur et sélectionne le sport qu'il pratique. L'adresse sera transformée en coordonnées GPS par une classe spéciale nommée 'geocoder'. Une fois les coordonnées GPS récupérées, elles seront enregistrées dans la base de données avec toutes les informations relatives au sport sélectionné et à l'utilisateur qui a ajouté ce marqueur.

Si le sport que l'utilisateur n'apparaît pas dans la liste, alors il sélectionne une catégorie autre qui débloquera un champ texte permettant d'entrer une nouvelle discipline.

Conclusion :

Ce projet m'a permis de me familiariser avec le JavaScript et ses fonctionnalités. De découvrir les requêtes en AJAX et les traitements des données en JavaScript qui permettent d'avoir des sites internet plus conviviaux pour le visiteur et surtout une diminution du trafic et de génération de page sur le serveur.

Ce projet m'a permis également de découvrir les API de Google et leur fonctionnement qui permettent de faire des sites plus complets avec un minimum de code.