

PROJET PWEBC

Une carte interactive avec géolocalisation

1. Présentation du projet

Le but est de réaliser une application WEB de type full web, dont l'IHM est munie d'un fond de carte interactive, permettant à un utilisateur d'interagir dynamiquement au regard des informations et objets présentés spatialement sur la carte, à la manière de « Google Map ».

2. Spécifications fonctionnelles (synoptique)

L'application à développer sera fondée sur différentes capacités en liens avec la présentation à l'écran d'une carte géographique interactive :

- . Déplacer ou zoomer le fond de carte (comme afficher/recentrer la carte sur point ou un objet)
- . Etablir et afficher des connexions entre des points (par exemple pour définir des chemins)
- . Positionner et afficher des objets variés et/ou des avatars représentant des utilisateurs
- . Leur associer des informations, par exemple de type GPS.
- . Définir les interactions pour l'utilisateur de votre application.
- . Sauvegarder les données de l'application, le contexte d'utilisation de la carte par l'utilisateur.

Il vous appartient de définir le thème de votre application WEB et son développement dès la première semaine. Par exemple, il pourra s'agir d'un jeu, d'un éditeur de cartes géographiques, d'une application partagée intégrant des notions geo-spatiales s'affichant sur une carte, ou encore d'une application de services en lien avec des données publiques numérisées. Ce dernier point peut vous aider à définir le thème de votre application, c'est pourquoi, nous vous demandons de consulter rapidement les données rendues disponibles au format JSON par les services de la mairie de paris et du gouvernement.

L'application sera développée pour une personne en interaction avec l'application. Elle devra pouvoir s'étendre à des groupes de personnes. Des données devront être échangées entre l'application tournant sur le navigateur et sa partie métier tournant sur votre serveur WEB et vice versa.

La réalisation de ce projet implique l'apprentissage et l'utilisation des différentes technologies du Web 2.0 vues en TDs :

- JACAScript et JQUERY pour l'interactivité et la manipulation des cartes,
- AJAX pour le requêtage de services WEB distants (récupérer les informations géographiques).
- Il faudra exploiter le potentiel du HTML 5, par exemple pour la géolocalisation, ainsi que les styles améliorés du CSS 3, par exemple pour les formulaires.

L'affichage du fond de carte et son interaction seront développés sur la base de codes préconstruits, comme nous le ferons en TPs :

- Bibliothèque de fonctions javascript prédéfinies, comme `LEAFLET` ou `MAPBOX`
- SERVICES WEB : accès à des services web distants avec AJAX, comme ceux de `NOMINATIM` pour obtenir des positionnements géographiques en latitude et longitude, pays, ville, ...

3. Développement

Le développement se fera en 3 étapes qui se dérouleront durant les séances de TP ainsi que sous la forme d'un travail personnel.

Etape 1

- + Formation des équipes (les trinômes), Découverte du sujet,
- + Recherche de données publiques par exemple sur data.gouv.fr. De telles données sont au format JSON et accessibles sur le net ou dans une version dégradée dans des fichiers.
- + Conception de votre application sur la base d'une carte numérique interactive
- + Découvertes des fonctionnalités de `leaflet/routing` ou `MapBox` via les documentations en ligne et applications démos, comme ajout de points d'intérêts sur fond de carte, utilisation de calques pour différencier des types de points d'intérêts, définition d'écouteurs d'événements sur la carte, élaboration d'une fonction de calcul d'un chemin
- + Définition des écrans et interfaces utilisateurs de votre application, à l'instar des démos de `jquery-ui`.

Etape 2

- + Prise en main et test de fonctionnalités de la bibliothèque de gestion de cartes numérisées interactives.
- + Prise en main de données publiques, test de réception et exploitation sur votre navigateur.
- + Développement du frontal de votre application incluant le fond de carte : écrans web et interactions avec l'utilisateur de votre application sur son navigateur (html, css, jquery et spécialisation jquery-ui).

Etape 3

- + Prise en main des outils de gestion de bases de données offrant une vision graphique des bases. Développement d'une base de données simple de 2 à 3 tables (pas moins de 2)
- + Réaliser les scripts de services avec accès à votre base de données.
- + Intégrer vos scripts de services via les procédures AJAX.

3. Aspects technologiques

Exploitation de services web distants à interroger

Il vous est demandé d'exploiter des données publiques et des services web de votre propre application. La récupération des données géographiques sur le serveur se fera par le biais de **requêtes AJAX** qui permettront de récupérer dynamiquement **au format JSON** les informations utiles à la personnalisation de la carte. Exemple de requête AJAX :

```
$.ajax({
  type: 'GET',
  url: "http://nominatim.openstreetmap.org/search",
  dataType: 'json',
  jsonpCallback: 'data',
  data: {format:"json", limit:1, country:france, json_callback: 'data' },
  error: function (xhr, status, error) { alert("ERROR "+error); },
  success: function(data){ //Données JSON reçues accessibles
                           // via le tableau data  }
});
```

Cas des données géographiques avec Leaflet

Le service web principal que vous devrez interroger pour récupérer les informations géographiques est <http://nominatim.openstreetmap.org>. Nominatim (du Latin, 'par le nom') est un webservice

permettant de chercher des données Open Street Map (OSM) par leur nom et leur adresse (Exemple : recherche pays France). Les différents paramètres sont disponibles à l'adresse suivante

<http://wiki.openstreetmap.org/wiki/FR:Nominatim>

Exemple d'utilisation du service pour récupérer les coordonnées géographiques (latitude, longitude) de la France :

<http://nominatim.openstreetmap.org/search?country=france&limit=1&format=json&addressdetails=1>

Exemple de résultat, retourné au client au format JSON :

```
[{"place_id":"9157173744",
"licence":"Data \u00a9 OpenStreetMap contributors, ODbL
1.0.http://www.openstreetmap.org/copyright",
"osm_type":"relation","osm_id":"2202162",
"boundingbox":["-85.0500030517578","51.2683181762695","-
178.38737487793","172.305725097656"],
"lat":"46.603354","lon":"1.8883335",
"display_name":"France"}]]
```

Une autre possibilité offerte par Nominatim est la fonction 'reverse' qui permet de retrouver des informations géographiques (ville, pays) à partir de coordonnées géographiques (latitude, longitude).

Exemple d'utilisation de la fonction reverse (pour un point (lat,long) situé en Angleterre) :

<http://nominatim.openstreetmap.org/reverse?format=json&lat=52.5487429714954&lon=-1.81602098644987&limit=1&zoom=4&addressdetails=1>

Exemple de résultat (retourné au client au format JSON) :

```
{"place_id":"97282697","licence":"Data \u00a9 OpenStreetMap
contributors, ODbL 1.0.
http://www.openstreetmap.org/copyright","osm_type":"relation","osm
_id":"62149","lat":"54.7023545","lon":"-3.2765753",
"display_name":"Royaume-Uni",
"address":{"country":"Royaume-Uni","country_code":"gb"}}
```

Utilisation de JavaScript pour gérer le fond de carte

La librairie JavaScript utilisée pour interagir avec le fond de carte sera une librairie tel que Leaflet ou MapBox qui permet de créer/gérer/manipuler des fonds de carte.

Un exemple d'utilisation de la librairie Leaflet est donné à l'adresse suivante :

<http://leafletjs.com/examples/quick-start.html>

La première étape sera alors d'importer cette librairie depuis votre page web :

```
<script src="http://cdn.leafletjs.com/leaflet-0.7.1/leaflet.js"></script>
```

Vous pourrez ensuite utiliser depuis votre code javascript des fonctions définies dans l'API Leaflet .

Voici un exemple de création d'un fond de carte en JS (en supposant l'existence dans la page html d'un bloc DIV avec l'identifiant "map") :

```
//Chargement initial de la carte (MAP) dans le bloc DIV map et affichage
var map = L.map('map').setView([14,-14.8883335],4);
L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {attribution:
'PING'}) .addTo(map);
```

Utilisation de la bibliothèque JQUERY-UI pour les interactions homme-machines

Dans ce projet, vous devez concevoir votre site suivant des critères ergonomiques. Nous vous demandons d'utiliser des objets d'interface de haut niveau comme les menus, dialogues-formulaires, boutons, icônes Présentés sur le site de [JQUERY-UI demos](#) vus en TP. Vous pourrez aussi exploiter les capacités d'animation de jquery-ui pour dynamiser vos pages.

Utilisation de la bibliothèque PDO de PHP pour l'accès à une base de données SQL

L'accès à une base de données et la présentation de données dans les pages web se réalisera sous la forme de scripts PHP exploitant la bibliothèque PDO. Vous pourrez ainsi archiver les données de votre application et répondre à ses services. La bibliothèque PDO de PHP a l'avantage d'être orientée objet et de pouvoir fournir l'accès à des serveurs de données SQL variés comme MYSQL, ORACLE... Il suffit donc de quelques paramètres à préciser à l'instanciation pour vous connecter à votre base et exploiter les fonctionnalités d'un objet \$pdo :

```
$pdo = new PDO ("mysql:server= $hostname; dbname=$base", "$loginBD", "$passBD");
```

De plus son mécanisme de Binding des paramètres permet de protéger la base de possibles injections de code-script malveillant à destination de votre base ou du serveur de bases de données

```
$commande = $pdo->prepare($sql); // $sql requête paramétrée avec :nom  
$commande->bindParam(':nom', $nom); // paramètre instancié avec $nom  
$bool = $commande->execute() ;
```

Pour aller plus loin,

Vous pourriez ajouter les fonctionnalités suivantes :

- CSS alternative pour mobiles (attention au faible support du javascript sur ces périphériques)
- Authentification externalisée : OpenID / facebook/google connect.
- Interaction client-client, avec rafraichissement sur timer javascript ou sur écoute événementielle du serveur de type push.