

Utilisation de Google Maps

Soumis par Nicolas PIED
05-09-2006
Dernière mise à jour : 06-09-2006

Google Maps est un service gratuit de cartographie en ligne. Lancé depuis 2004, celui-ci est désormais accessible pour de nombreux pays. Ce service propose deux types de plans :

- Le plan routier avec les noms des rues, des quartiers, etc.
- Le plan satellitaire qui couvre à l'heure actuelle le monde entier. Google permet aux développeurs qui le souhaitent et pour un domaine donné, de pouvoir utiliser ce service qui propose les fonctionnalités suivantes :

- Compatible avec la plupart des navigateurs graphiques.
- Contrôles disponibles par défaut permettant de dé/zoomer, de se déplacer sur la carte.
- Possibilité d'ajouter des contrôles personnalisés pour modifier l'affichage de la carte.
- Création d'icônes personnalisées pour localiser un point sur une carte.
- Gestion des événements JavaScript.
- Extraction des données « Géocodes » à partir d'une adresse.
- Support de HTTP Request (Ajax).

Obtenir une clef d'utilisation de l'API Google Maps

L'utilisation de Google Maps est basée sur une API JavaScript utilisable gratuitement à partir de l'instant où on possède une clef. Il vous est possible de récupérer cette clef gratuitement pour un domaine donné à l'adresse ci-dessous :

Obtenir une clef pour l'utilisation de l'API Google Maps.

Cette demande est réalisée en une minute environ en complétant un formulaire d'inscription à l'issue duquel vous obtiendrez une clef textuelle qui sera disponible uniquement pour le domaine Internet précisé. En effet, les clefs sont affichées en clair dans les applications Google Maps, il est donc possible à une autre personne de récupérer celle-ci en affichant le code source HTML. La contrainte du nom de domaine permet donc d'éviter des utilisations non autorisées par vous et Google. {mospagebreak title=Votre première application Google Maps}Créer votre première application Google Maps

La première application qui va être présentée ci-dessous va permettre de localiser un point géographique sur Google Maps à l'aide d'une coordonnée sous la forme latitude / longitude. La navigation sera ensuite possible via certains composants permettant de réaliser les actions suivantes :

- Zoom / Dé zoom
- Déplacement à l'aide de boutons
- Déplacement à l'aide de la souris (draggable)

La première ligne dans le code source HTML est la suivante :

```
<script src="http://maps.google.com/maps?file=api&v=2.x&key=VOTRE_CLEF"
  type="text/javascript"></script>
```

Celle-ci permet de faire appel au script conçu par Google pour manipuler les fonctionnalités de Google Maps. Cette ligne précise également l'utilisation de l'API dans sa version 2.x qui présente certaines nouveautés comme GeoCoder, qui permet de localiser un point sur une carte à partir d'une adresse postale, et quelques corrections de bugs présents dans la version 1.

Pour rappel, il est possible d'obtenir une clef d'utilisation de l'API Google Maps à l'adresse suivante :

Obtenir une clef d'utilisation Google Maps.

```
<div id="map" style="width: 400px; height: 300px"></div>
```

La division utilisée ici, va par la suite contenir la carte routière ou satellitaire. Il est possible, comme dans ce cas, de limiter les dimensions de la carte en limitant celles de la division.

```
<script type="text/javascript">
```

```
//<![CDATA[
/* Variable qui va correspondre à l'affichage de la carte dans la "div" */
var map = new GMap2(document.getElementById("map"));
/* Centre la carte aux coordonnées indiquées et réalise un zoom de niveau 14 */
map.setCenter(new GLatLng(49.41483, 2.817895), 14);

/* Cette ligne permet de bloquer le déplacement sur la carte à l'aide de la souris
```

```

map.disableDragging(); */

map.addControl(new GSmallZoomControl());
//]]>

</script>

```

Le niveau de zoom est ici de 14, il faut savoir que celui-ci accepte pour valeur les nombres entiers compris entre 1 et 17 inclus et que plus la valeur indiquée est grande et plus le niveau de zoom est élevé.

Au niveau des dernières lignes de cet exemple, il est possible de remplacer GLargeMapControl par les valeurs ci-dessous afin d'afficher différents types de contrôles :

- GMapTypeControl : Permet de passer de la vue routière à la vue satellitaire et réciproquement. Il est aussi possible de mixer les deux modes.
- GLargeMapControl : Ajoute les contrôles permettant le zoom et le déplacement sur la carte.
- GSmallMapControl : Identique à GLargeMapControl mais la barre de zoom est remplacée par deux boutons permettant de zoomer et de dézoomer.
- GSmallZoomControl : Présente uniquement les deux boutons permettant de zoomer et de dézoomer.

Il est aussi possible d'ajouter plusieurs contrôles pour une même carte ; ceux-ci se complètent comme dans l'exemple ci-dessous :
`map.addControl(new GSmallMapControl());`
`map.addControl(new GMapTypeControl());`
 Visualiser ce premier exemple.
 Télécharger les sources de ce premier exemple. {mospagebreak title=Les « info-bulles » Google Maps}
 Ouvrir une « info-bulle » Google Maps

L'API Google Maps possède des fonctions permettant l'affichage « d'info-bulles » ce qui permet de positionner des bâtiments, des annotations, etc. L'exemple ci-dessous permet, par rapport à l'exemple précédent, d'afficher une « info-bulle » au centre de la carte indiquant le nom du point géographique donné.

Le premier paramètre correspond aux coordonnées géographiques de type « GLatLng » auxquelles attacher « l'info-bulle » et le second correspond au contenu affiché dans cette « info-bulle ».

```

map.openInfoWindow(point,
    document.createTextNode("Université Technologique de Compiègne"));

```

L'API Google Maps met à disposition deux types d'info-bulles, la première ne permet d'afficher que du texte (comme celle utilisée dans l'exemple) et la seconde permet d'afficher du contenu HTML (donc images, etc.). Cette dernière est disponible en utilisant « openInfoWindowHtml » au lieu de « openInfoWindow ».

Visualiser l'exemple permettant l'affichage d'une « info-bulle ».

Télécharger les sources de cet exemple. {mospagebreak title=Afficher des marqueurs}

Afficher un marqueur sur la carte
 Les marqueurs sont associés à une coordonnée géographique et permettent de situer celle-ci sur la carte issue de l'utilisation de l'API Google Maps. Par défaut, ceux-ci ressemblent à une poire renversée, mais il vous est possible de les personnaliser comme le montre Google sur cet article :

Personnaliser les icônes de Google Maps.

```

map.addOverlay(new GMarker(point));

```

En ajoutant cette ligne à la fin du code source issu du premier exemple, on obtient ainsi un marqueur attaché aux coordonnées nommées « point » permettant de localiser l'Université Technologique de Compiègne. La fonction « addOverlay », comme son nom l'indique, permet de superposer ce marqueur sur la carte.

Il est également possible d'attacher un « listener » d'événements sur ce marqueur, permettant ainsi des interactions avec l'utilisateur ; possibilité que nous verrons dans la suite de cet article. Exemple permettant d'afficher un marqueur.

Exemple affichant deux marqueurs.

Exemple affichant des marqueurs aléatoires (utilisation d'un tableau JavaScript).

Télécharger les sources de cet exemple. {mospagebreak title=Attacher des événements JavaScript} Attacher un événement à la carte Google Maps

L'exemple ci-dessous présente une interaction réalisée entre la carte et un utilisateur. Ce dernier, à l'aide de la souris, va cliquer à un endroit de la carte ; si aucun marqueur ne se trouve sous le curseur un marqueur est alors affiché, sinon il est supprimé.

Ces quelques lignes précisent qu'un « listener » (entité chargée d'écouter les événements) est ajouté aux événements généraux « GEvent ». Ce « listener » a pour but d'exécuter la fonction passée en 3ème argument lors d'un « clic » sur le conteneur « map » soit la division HTML correspondante.

```
GEvent.addListener(map, "click", function(marker, point) {
  /* Si un marqueur est présent, alors il est supprimé */
  if (marker) map.removeOverlay(marker);
  /* Sinon, on ajoute un marqueur au point demandé */
  else map.addOverlay(new GMarker(point));
});
```

Lorsqu'un événement correspondant survient, la fonction en argument, va alors recevoir deux paramètres :

- Le premier correspond au marqueur qui a reçu le clic si celui-ci existe.
- Le second correspond aux coordonnées, de type « GLatLng », issues d'un clic de souris sur la carte.

On pourrait très bien imaginer une application demandant à l'utilisateur d'indiquer son adresse ou d'apporter des précisions sur un lieu. Les résultats pourraient ensuite être enregistrés dans une base de données en vue de futures utilisations.

Visualiser l'exemple sur les événements Google Maps.

Exemple : Apparition d'une « info-bulle » après un « clic » sur un marqueur.

Télécharger les sources de cet exemple. {mospagebreak title=Des onglets dans les « info-bulles »}Afficher des onglets dans les « info-bulles »

L'exemple ci-dessous montre comment afficher des onglets dans les info-bulles HTML. Le principe en soit est assez simple, puisqu'il s'agit de définir un tableau de type « GInfoWindowTab » qui prend comme premier argument le nom de l'onglet et comme second le contenu de l'onglet ; contenu qui peut être sous format texte ou HTML.

Ensuite, il suffit de lier ce tableau au marqueur en passant ce premier comme argument à la fonction « openInfoWindowTabsHtml » comme indiqué dans les lignes ci-dessous : /* Tableau de GInfoWindowTab qui va contenir les onglets de l'info-bulle */

```
var infoTabs = [
  new GInfoWindowTab("Onglet #1", "Le contenu de l'onglet 1"),
  new GInfoWindowTab("Onglet #2", "Voici un texte en gras")
];
```

```
/* Création du marqueur attaché au point */
var marker = new GMarker(point);
/* Enregistrer un listener lors d'un clic sur le marqueur */
GEvent.addListener(marker, "click", function() {
  /* Lors d'un clic, afficher les onglets */
  marker.openInfoWindowTabsHtml(infoTabs);
});
```

```
/* Afficher le marqueur et l'info-bulle */
```

```
map.addOverlay(marker);
marker.openInfoWindowTabsHtml(infoTabs);
```

Visualiser l'exemple sur les onglets dans les « info-bulles ».

Télécharger les sources de cet exemple. {mospagebreak title=« Drag and Drop » des marqueurs}

Déplacer les marqueurs Google Maps à la souris l'exemple ci-dessous propose de déplacer un marqueur à l'aide de la souris en utilisant la technique dites « drag & drop ». Pour cela, il est nécessaire d'indiquer au marqueur dans un premier temps que celui-ci à la possibilité d'être déplacé : où le « {draggable: true} » en second paramètres du constructeur « GMarker ».

Ensuite, mais cela reste facultatif, il est possible d'attacher des « listeners » qui vont intervenir lors des différents événements engendrés par le déplacement du marqueur. Dans notre cas, seuls les événements « dragstart » et « dragend » seront pris en compte.

```
/* Création du marqueur attaché au point
```

```
* Le marqueur est ici déplaçable à l'aide de la souris
*/
var marker = new GMarker(point, {draggable: true});
```

```
/* Lorsque son déplacement commence, on ferme l'info-bulle */
```

```
GEvent.addListener(marker, "dragstart", function() {
  map.closeInfoWindow();
});
```

```
/* Lorsque son déplacement se termine, on affiche l'info-bulle */
```

```
GEvent.addListener(marker, "dragend", function() {
  marker.openInfoWindowHtml("Bien joué !");
});
```

La liste complète des évènements Google Maps pour l'API 2 est disponible à l'adresse ci-dessous au niveau de la partie « Events » :

Consulter la liste des évènements Google Maps.

Visualiser l'exemple du marqueur « drag & drop ».

Télécharger les sources de cet exemple. {mospagebreak title=Rechercher une adresse postale}

Obtenir les coordonnées géographiques d'une adresse postale donnée

L'API Google Maps, dans sa version 2, permet de déterminer les coordonnées géographiques d'une adresse postale donnée. Ceci est réalisé via l'utilisation de l'objet « GClientGeocoder » qui utilise les méthodes HTTP pour interroger les serveurs concernés. Cette fonctionnalité est donc bien plus simple pour l'utilisateur que de saisir les latitudes et longitudes.

A partir d'un champ de saisie, l'utilisateur va indiquer l'adresse du lieu qu'il souhaite localiser. Suite à la validation du formulaire, la fonction « showAddress » ci-dessous lui indiquera le lieu d'un marqueur si celui-ci a pu être localisé. Sinon un message d'alerte avertira l'utilisateur que le lieu n'a pu être trouvé.

La carte est automatiquement recentrée sur un lieu localisé et une « info-bulle » affiche son adresse./* Création de l'objet GClientGeocoder */

```
var geocoder = new GClientGeocoder();
```

```
/* Fonction qui à partir d'une adresse va déterminer le point géographique */
```

```
function showAddress(address) {
  if (geocoder) {
    geocoder.getLatLng(address, function(point) {
      if (!point) { /* Si les coordonnées n'ont pas été trouvés */
        alert("Impossible de localiser l'adresse :\n\n" + address);
      } else { /* Les coordonnées ont été trouvés */
        /* Centrer la carte sur le point */
        map.setCenter(point, 14);
        /* Création d'un marqueur */
        var marker = new GMarker(point);
        /* Afficher le marqueur */
        map.addOverlay(marker);
        /* Associer une info-bulle au marqueur */
        marker.openInfoWindowHtml(address);
      }
    });
  }
}
```

Visualiser l'exemple utilisant « Geocoder ».

Télécharger les sources de cet exemple. {mospagebreak title=Conclusion}

Conclusion

Google Maps présente des fonctionnalités avancées qui peuvent fournir des utilisations intéressantes au sein d'applications informatiques. L'API étant en développement continue, des nouveautés verront sûrement le jour d'ici peu, ce qui promet des emplois répondant de plus en plus aux attentes des utilisateurs et développeurs.

Cet article ne présente pas toutes les spécificités de l'API Google Maps mais seulement les principales. Si vous souhaitez obtenir plus d'informations sur son utilisation, le site de Google Code, propose la référence et la documentation de toutes les fonctions disponibles.

Bonne utilisation !

API Google Maps.

Référence des fonctions Google Maps.

{moscc}6{/moscc}

