

## Les API de cartographie en ligne (2ème partie)

[Retour au sommaire de la lettre](#)

*Les services de cartographie en ligne sont aujourd'hui nombreux et ils proposent pour la plupart des API ou interfaces de programmation, permettant d'insérer de façon très simple des cartes dans vos pages web ou pages d'accueil personnalisées. Dans cet article en deux parties, nous allons vous expliquer comment utiliser ces API pour créer des cartographies simples et/ou sophistiquées, y ajouter des données dynamiques, diffuser ces informations géolocalisées et, d'une manière générale, mieux maîtriser les outils que nous fournissent Google, Yahoo!, Microsoft et les autres. Vous allez devenir un as des cartes... Ce mois-ci, nous explorons plus particulièrement le monde des mapplets, des API Google Earth et Google Maps au travers de nombreux exemples. Attachez vos ceintures cartographiques...*

Nous allons examiner dans cet article - en deux parties - les multiples services de cartographie en ligne permettant de créer des cartes personnalisées, développer la notoriété de sa société sur Internet, accroître la popularité de son site web et mettre en place des outils permettant d'échanger des informations et des données géolocalisées. Nous verrons donc comment :

- Créer des cartographies à vocation informative ou commerciale ;
- Ajouter des données générées dynamiquement ;
- Maîtriser les outils permettant de diffuser ces informations géolocalisées ;
- Construire un système d'information utilisant de la cartographie.

Nous allons donc apprendre à créer des cartes statiques, des Mapplets, utiliser l'API de Google Maps ainsi que celle proposée par Virtual Earth et Google Earth.

Le mois dernier, nous avons appris à utiliser l'option "mes cartes" de Google, l'API Google Maps et Yahoo! Pipes, et nous avons créé nos premières cartes. Ce mois-ci, nous allons un pas plus loin en traitant des mapplets, des API Google Earth et Google Maps grâce à de nombreux exemples de codes les utilisant. C'est parti pour un voyage dans le monde fabuleux de la cartographie...

### Les Mapplets

Les Mapplets sont une sorte de mini-page Web dont les informations ou les fonctionnalités vont s'intégrer à Google Maps. Un Mapplet est un gadget généré à partir d'un fichier XML et qui contient des métadonnées. Le contenu HTML est affiché aux internautes en utilisant les informations contenues entre les balises `<Content>`. En plus d'un code HTML, vous pouvez aussi y insérer du JavaScript ou du Flash. Un Mapplet permet de :

- Faire connaître votre marque ou votre société auprès des utilisateurs de Google Maps ;
- Faire un lien vers votre site Web au travers de votre Mapplet ;
- Bénéficier d'un hébergement gratuit (sur [maps.google.com](http://maps.google.com)) si votre Mapplet figure dans l'annuaire dédié ;

Signalons enfin qu'aucun type d'installation n'est nécessaire... La documentation sur cette branche de l'API Google Maps est visible à cette adresse :

<http://code.google.com/apis/maps/documentation/mapplets>.

#### Installer les outils pour développeur

Tout d'abord, il faut installer deux outils :

1. Connectez-vous sur votre compte Google.

2. Accédez à cette page :

<http://maps.google.com/ig/directory?synd=mpl&pid=mpl&features=sharedmap%2Cgeofeed&at=devtools&backlink=http%3A%2F%2Fmaps.google.com%2Fmaps%2Fmm>.

3. Ajoutez ces deux Mapplets : **Mapplet Scratch Pad** et **Developer Mapplet**.

Ils seront visibles dans la rubrique *Created by others*.

Par la suite, votre Mapplet devra être envoyé sur un serveur public comme un site web ou un service d'hébergement de fichiers.

Afin de partager votre Mapplet, vous devez utiliser cette syntaxe :

<http://maps.google.com/ig/add?synd=mpl&pid=mpl&moduleurl=<URL de votre mapplet>>.

Avant de pouvoir utiliser votre Mapplet, vous devez l'installer en accédant à l'annuaire Google Maps

(<http://maps.google.com/ig/directory?synd=mpl&pid=mpl&features=sharedmap,geofeed&backlink=http%3A%2F%2Fmaps.google.com%2Fmaps%2Fmm>) et en cliquant sur le lien *Add by URL*. Il apparaîtra ensuite dans l'onglet *Mes cartes*.

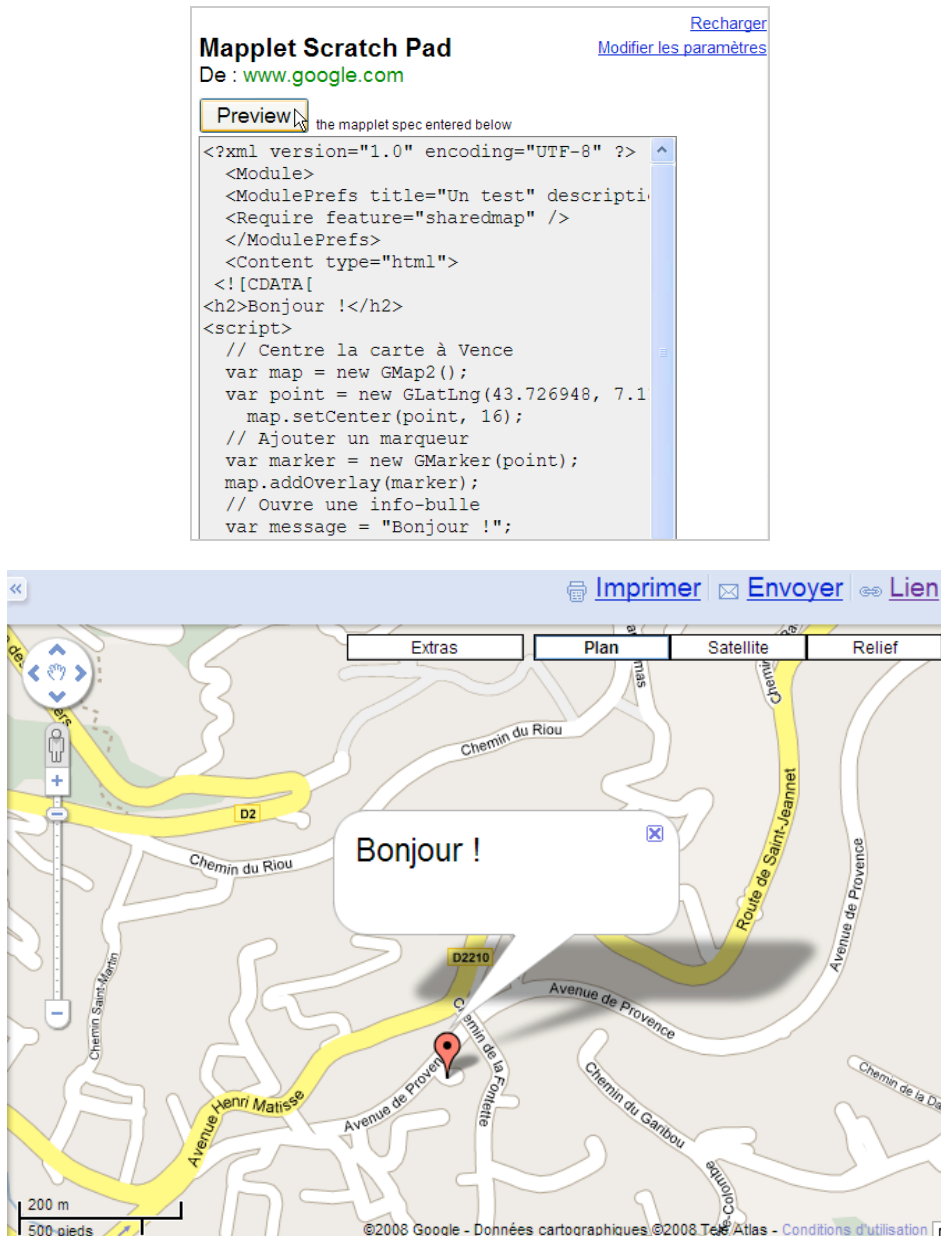
Il est possible d'ajouter votre Mapplet à l'annuaire Google en utilisant le formulaire visible à cette adresse : <http://maps.google.com/ig/submit?synd=mpl&pid=mpl>.

#### *Créer votre premier Mapplet*

1. Dans Google Maps, sélectionnez le Mapplet appelé **Mapplet Scratch Pad**.
2. Dans la zone de texte visible en-dessous, copiez ce contenu :

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
  <ModulePrefs title="Un test" description="Un simple test" author="Moi"
author_email="e-mail@gmail.com" height="150">
    <Require feature="sharedmap" />
  </ModulePrefs>
  <Content type="html">
    <![CDATA[
<h2>Bonjour !</h2>
<script>
  // Centre la carte à Vence
  var map = new GMap2();
  var point = new GLatLng(43.726948, 7.113304);
  map.setCenter(point, 16);
  // Ajouter un marqueur
  var marker = new GMarker(point);
  map.addOverlay(marker);
  // Ouvre une info-bulle
  var message = "Bonjour !";
  marker.openInfoWindowHtml(message);
</script>
]]>
</Content>
</Module>
```

3. Cliquez sur le bouton *Preview*.



4. Afin de forcer Google Maps à effacer le cache, cliquez sur le bouton *Recharger* puis collez le texte modifié.

Notez que la référence de l'API Maplet est accessible à cette adresse :

<http://code.google.com/apis/maps/documentation/maplets/reference.html>.

Les commentaires doivent être précédés de deux barres obliques suivies d'un espace : `//`.

La section *Module* contient les informations principales de votre module, comme ses paramètres, son titre, sa version, etc.

La déclaration `<Require feature="sharedmap"/>` procède automatiquement à l'initialisation de l'API sans qu'il soit besoin d'utiliser une clé pour faire appel à la librairie JavaScript.

La section *CDATA* indique à l'analyseur XML qu'il n'existe pas de balisage dans les caractères contenus dans ces sections. Il rapporte donc le contenu suivant comme des caractères simples, sans tenter de les interpréter comme des balises d'éléments ou d'entités. Lorsqu'il rencontre le `]]>` final, l'analyseur cesse de rapporter le contenu et reprend l'analyse normale.

`var map = new GMap2();` : instancie la classe Gmap2 afin de créer la carte ;

`var point = new GLatLng(43.726948, 7.113304);` : crée un point aux longitudes et latitudes indiquées ;

```
map.setCenter(point, 16); : positionne le point de vue de la carte au niveau de zoom indiqué (16) ;  
var marker = new GMarker(point); : définit un marqueur au centre de la carte ;  
map.addOverlay(marker) : ajoute le marqueur au centre de la carte ;  
var message = "Bonjour !"; : définit un message en tant que variable ;  
marker.openInfoWindowHtml(message); : ajoute le message dans une info-bulle attachée au marqueur.
```

Afin de modifier l'apparence de votre carte, utilisez cette syntaxe :

```
var map = new GMap2();  
map.setMapType(G_NORMAL_MAP);
```

Les valeurs possibles sont :

G\_NORMAL\_MAP : la vue par défaut en mode Plan ;

G\_SATELLITE\_MAP : la vue en mode Satellite ;

G\_HYBRID\_MAP : un condensé des deux précédentes vues ;

G\_DEFAULT\_MAP\_TYPES : une vue des trois précédentes vues si vous définissez une séquence d'instructions destinée à être exécutée plusieurs fois.

Afin d'ajouter une icône personnalisée, utilisez cette syntaxe :

```
<script>  
var map = new GMap2();  
var point = new GLatLng(43.726948, 7.113304)  
map.setCenter(point, 16);  
var blueIcon = new GIcon(G_DEFAULT_ICON);  
blueIcon.image = "http://www.google.com/intl/en_us/mapfiles/ms/micons/blue-dot.png";  
markerOptions = { icon:blueIcon };  
var marker = new GMarker(point);  
map.addOverlay(new GMarker(point, markerOptions));  
</script>
```

Afin d'ajouter une ligne, utilisez cette syntaxe :

```
<script>  
var polyOptions = {geodesic:true};  
var polyline = new GPolyline([  
  new GLatLng(43.726948, 7.113304),  
  new GLatLng(44.736948, 7.443304)  
], "#ff0000", 5, 1, polyOptions);  
map.addOverlay(polyline);  
</script>
```

Afin de superposer une carte ou une image, utilisez cette syntaxe :

```
<script>  
var map = new GMap2();  
map.setMapType(G_NORMAL_MAP);  
map.setCenter(new GLatLng(40.740, -74.18), 12);  
var boundaries = new GLatLngBounds(new GLatLng(40.716216,-74.213393), new  
GLatLng(40.765641,-74.139235));  
var oldmap = new GGroundOverlay("image.jpg", boundaries);  
map.addOverlay(oldmap);  
</script>
```

L'ajout d'un polygone est passablement complexe puisque vous devez définir manuellement chacun des points qui forment la figure. Aussi, une méthode plus simple consiste à créer un fichier KML dans Google Earth puis de l'appeler en utilisant cette syntaxe :

```
var map = new GMap2();  
var geoXml = new GGeoXml("http://exemple.fr/fichier.kml");  
map.setCenter(new GLatLng(40, 7),10);
```

```
map.addOverlay(geoXml);  
</script>
```

Le principe est identique en ce qui concerne un fichier GeoRSS...

Cela nous amène directement à parler de l'extraction de données en utilisant un Maplet... Il existe deux méthodes :

- `IG_FetchContent(url, callback)` : retourne le contenu de l'URL en tant que texte. Cette fonction permet de récupérer du contenu HTML ou du contenu texte.
- `IG_FetchXmlContent(url, callback)` : retourne les informations d'un fichier XML comme un objet DOM.

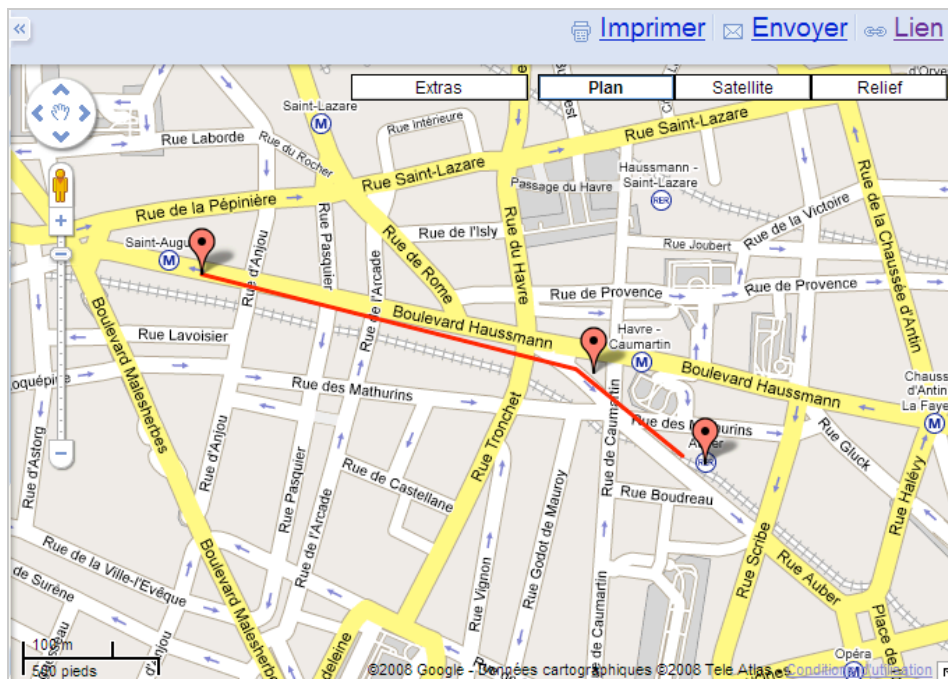
**Document Object Model (ou DOM)** : cette recommandation du W3C décrit une interface permettant à des programmes informatiques et à des scripts d'accéder ou de mettre à jour le contenu, la structure ou le style de documents.

En imaginant que votre fichier XML contienne la localisation de ces trois stations de métro :

```
<?xml version="1.0" encoding="UTF-8" ?>  
<points>  
  <point lng="2.321956" lat="48.874425">Saint-Augustin</point>  
  <point lng="2.327986" lat="48.873423">Havre-Caumartin</point>  
  <point lng="2.329702" lat="48.872506">Auber</point>  
</points>
```

Nous pouvons imaginer ce code :

```
<script>  
var map = new GMap2();  
map.setCenter(new GLatLng(48.874425, 2.321956), 16);  
var url = "http://jeannoel.anderruthy.googlepages.com/metro.xml";  
_IG_FetchXmlContent(url, function(response) {  
  var trainline = [];  
  var points = response.getElementsByTagName("point");  
  for (var i = 0; i < points.length; i++) {  
    var point = points.item(i);  
    var lat = point.getAttribute("lat");  
    var lng = point.getAttribute("lng");  
    var latlng = new GLatLng(lat, lng);  
    trainline.push(latlng);  
    if (point.firstChild) {  
      var station = point.firstChild.nodeValue;  
      var marker = createMarker(latlng, station);  
      map.addOverlay(marker);  
    }  
  }  
  var polyline = new GPolyline(trainline, "#ff0000", 3, 1);  
  map.addOverlay(polyline);  
});  
function createMarker(point, name) {  
  var marker = new GMarker(point);  
  GEvent.addListener(marker, "click", function() {  
    marker.openInfoWindowHtml(name);  
  });  
  return marker;  
}  
</script>
```




Afin de définir un itinéraire, utilisez cette syntaxe :

```
<script>
var map;
var directions;function initialize() { map = new GMap2();
map.setCenter(new GLatLng(43. 679294, 7.134247), 15);
directions = new GDirections(map);
directions.load ("from: cannes to: Nice");}
</script>
```

De manière générale, il est préférable de placer les données "source" sur un fichier externe;  
En cas de modification, vous n'aurez simplement qu'à modifier le fichier XML ou KML.

Nous avons déjà vu qu'il existait une galerie des Mapplets (

<http://maps.google.com/ig/directory?synd=mpl&pid=mpl&features=sharedmap,geofeed&backlink=http:%2F%2Fmaps.google.com%2Fmaps%2Fmm>). Vous pouvez aussi ouvrir la page dédiée d'un Mapplet qui vous intéresse et cliquer sur le lien *View source*.



### Circle Filter

Layer circles to view the markers that would match a proximity query.

Jean-Francois Noel (jfno) -  
Quebec, QC, Canada

Add it to Maps

[Contact the developer - View source »](#)

Vous n'avez plus ensuite qu'à adapter le code visible à vos propres besoins...

Dernier point : cette page vous explique comment convertir un Mashup utilisant l'API Google Maps en un Mapplet :

<http://code.google.com/support/bin/answer.py?answer=69786&topic=11364>.

L'API Google Maps ? C'est justement l'objet du paragraphe suivant...

### Utiliser l'API Google Maps

Utiliser cette API suppose que vous ayez obtenu une clé API Google Maps... Vous devez faire attention à la version de l'API que vous utilisez : certaines méthodes ne sont pas supportées dans la version 2 alors qu'elles le sont dans la version 2.130d (et vice versa !).

Pour certains exemples qui suivent, nous nous sommes inspirés de cette source : <http://econym.googlepages.com/index.htm>. Vous trouverez également quelques didacticiels en français sur ce site : <http://www.touraineverte.com/Google-Maps>.

Afin d'avoir une idée de départ, testez les résultats renvoyés par ce code :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-
com:vml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
  <title>Google Maps</title>
  <script
src="http://maps.google.com/maps?file=api&v=2.130d&key=ABQIAAAA
mMep7oeFvSfyUCfA296tmRSkNtJc7ujDIIs-
sF8EjCfdjro3stxR3P3rctu_ka9oPssawFrw1AQPfcw"
type="text/javascript"></script>
</head>
<body onload="GUnload()">
  <div id="map" style="width: 450px; height: 450px"></div>
  <script type="text/javascript">
    function createMarker(point,html) {
      var marker = new GMarker(point);
      GEvent.addListener(marker, "click", function() {
        marker.openInfoWindowHtml(html);
      });
      return marker;
    }
    var map = new GMap2(document.getElementById("map"));
    map.addControl(new GLargeMapControl());
    map.addControl(new GMapTypeControl());
    map.setCenter(new GLatLng(43.675818,7.202911),12);
    var point = new GLatLng(43.709579,7.267456);
    var marker = createMarker(point,'<div style="width:240px">Le contenu de la
fenêtre avec un <a href="http://www.exemple.fr">lien</a> vers votre
site</div>')
    map.addOverlay(marker);
    var point = new GLatLng(43.656447,7.145576);
    var marker = createMarker(point,'Le contenu de votre seconde fenêtre')
    map.addOverlay(marker);
  </script>
</body>
</html>
```





Le lien direct vers cette carte est celui-ci :

<http://jeannoel.anderruthy.googlepages.com/google-maps-api01.html>.

Au chargement de la page, la fonction `load()` est exécutée. Dès sa fermeture, la fonction `GUnload()` sera exécutée.

La fonction `createMarker` permet de définir un ou plusieurs repères qui vont se superposer à votre carte.

Nous associons un événement (un clic) à chacun des marqueurs qui seront ajoutés.

Les déclarations `map.addControl` contrôlent l'apparence de la barre de contrôle et des boutons de *Layers* (Plan, Satellite, etc.)

Les paramètres pour `GLatLng()` sont, dans l'ordre, la latitude, la longitude et le niveau de zoom.

La méthode `map.setCenter()` vous permet de centrer votre carte.

Si vous souhaitez ajouter du contenu HTML utilisant des guillemets, vous pouvez utiliser des guillemets simples dans la chaîne JavaScript et des guillemets doubles pour le contenu HTML.

Si vous souhaitez ajouter une image ou un tableau dans le contenu HTML, définissez précisément la largeur et la hauteur des éléments.

*Ajouter un volet de menu*

Le code est accessible à cette adresse : <http://jeannoel.anderruthy.googlepages.com/google-maps-api02.html>.





Il vous suffit d'éditer le code source... Le seul point vraiment important est que la fonction `monclit(i)` ajoute un événement cliquable à chacun des marqueurs définis. La méthode `document.getElementById` retourne un objet HTML à partir de son ID, défini dans la propriété ID de la balise de l'objet.

#### Définir le type de Layers

Le code est accessible à cette adresse : <http://jeannoel.anderruthy.googlepages.com/google-maps-api03.html>.

Il existe plusieurs options :

- `G_NORMAL_MAP` : affiche la carte en mode "Plan" ;
- `G_SATELLITE_MAP` : affiche la carte en mode Satellite ;
- `G_HYBRID_MAP` : affiche une carte satellite sans les noms des lieux ;
- `G_PHYSICAL_MAP` : ajoute un bouton "Mixte" ("Satellite" avec les noms de lieux).

#### Ajouter une ligne simple

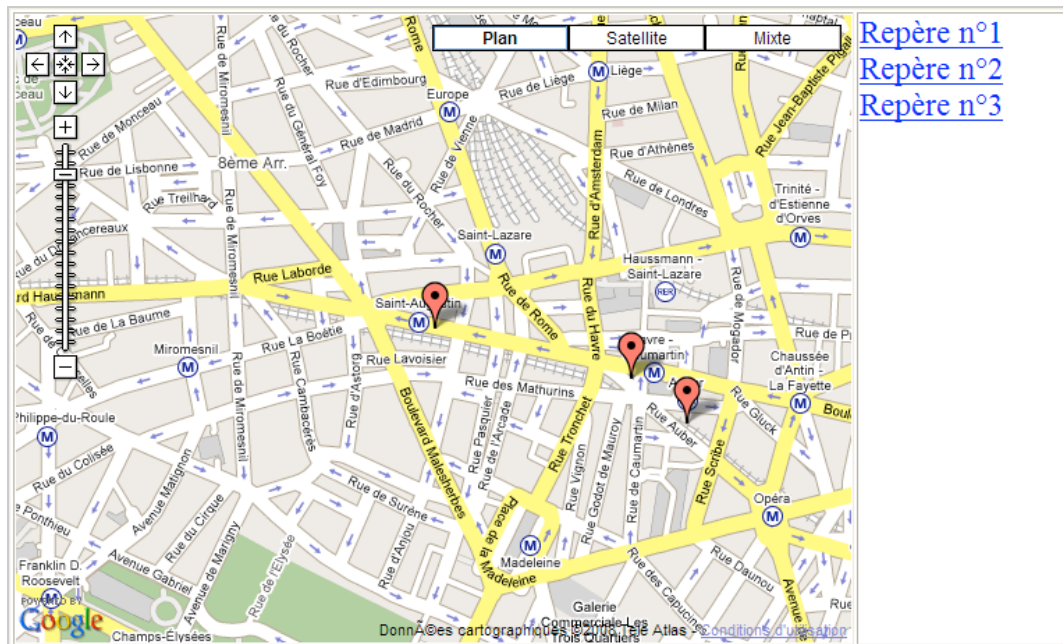
Le code est accessible à cette adresse : <http://jeannoel.anderruthy.googlepages.com/google-maps-api04.html>.

Nous utilisons une méthode appelée `GPolyline`(latitude et longitude, couleur, épaisseur, opacité, zoom).

#### Intégrer les données d'un fichier XML

Voici un exemple d'un fichier XML :

```
<?xml version="1.0" encoding="UTF-8" ?>
<markers>
  <marker lat="48.874425" lng="2.321956" html="Saint-Augustin"
label="Repère n°1" />
  <marker lat="48.873423" lng="2.327986" html="Havre-Caumartin"
label="Repère n°2" />
  <marker lat="48.872506" lng="2.329702" html="Auber" label="Repère n°3" />
</markers>
```



Le code correspondant est à cette adresse :

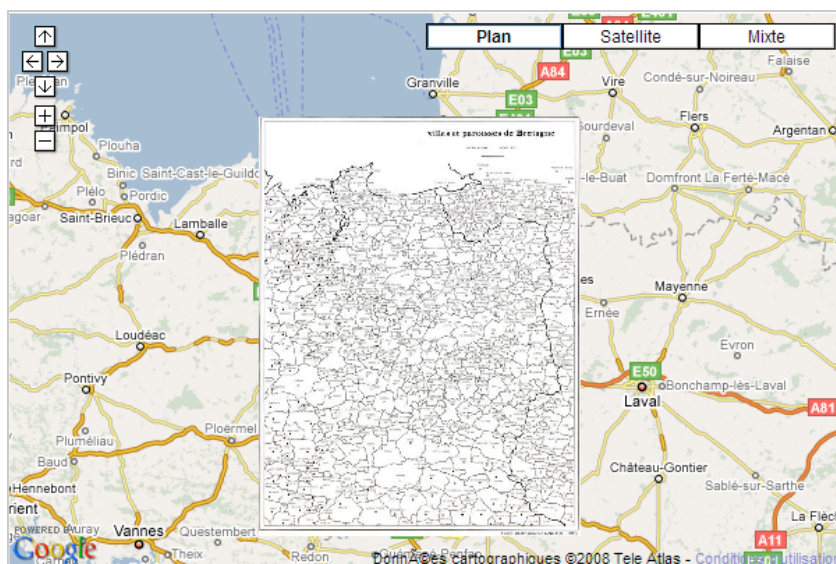
<http://jeannoel.anderruthy.googlepages.com/google-maps-api05.html>.

La fonction `GdownloadUrl` offre une manière simple de récupérer en mode asynchrone l'adresse URL spécifiée.

*Ajouter une superposition d'image*

Le code correspondant est visible à cette adresse :

<http://jeannoel.anderruthy.googlepages.com/google-maps-api06.html>.



Nous définissons une variable appelée "Boundaries" ("limites"). L'objet `GgroundOverlay` permet de créer la forme rectangulaire dont les limites sont définies par l'instance `GlatLngBounds` avec deux positions : sud-ouest et nord-est. Ce sera l'emplacement de notre image.

### Ajouter un flux GeoRSS

Le code correspondant est visible à cette adresse :

<http://jeannoel.anderruthy.googlepages.com/google-maps-api07.html>.

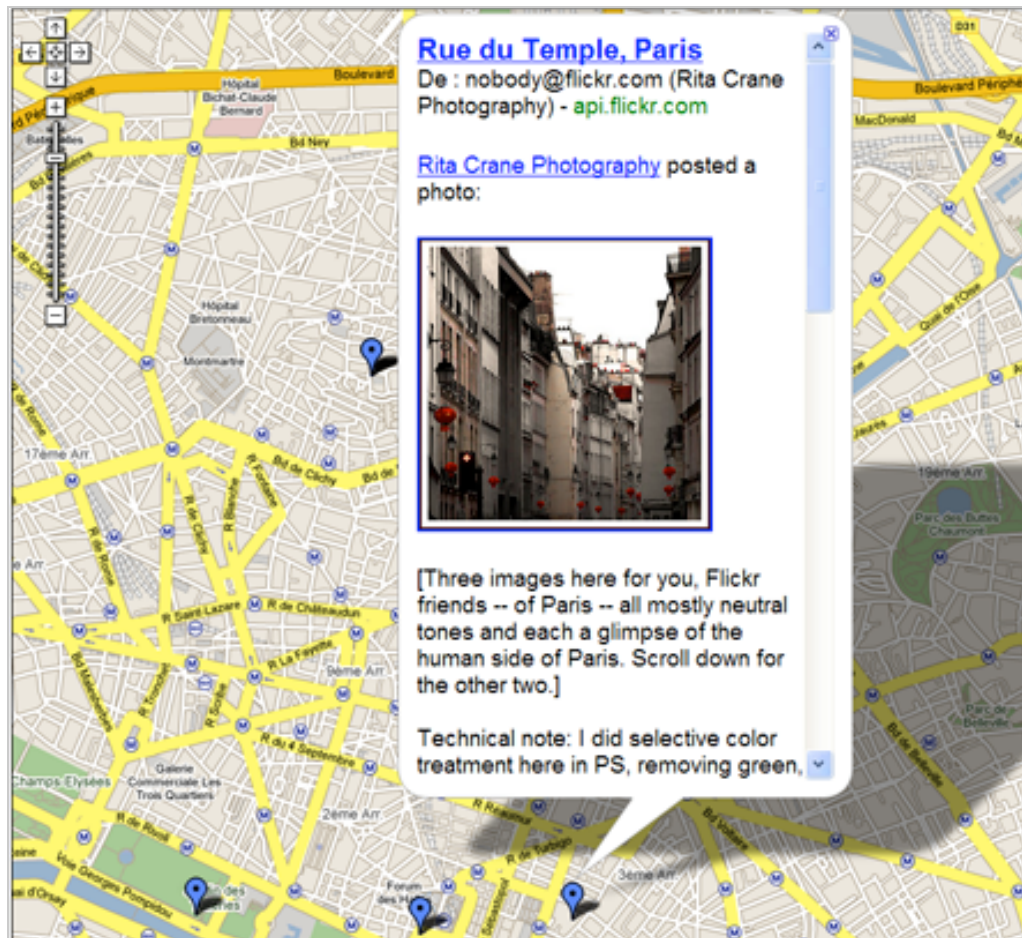
L'objet `GGeoXml` ajoute du contenu géolocalisé à partir d'un fichier XML. Il implémente l'interface `GOverlay` en utilisant la méthode `GMap2.addOverlay()`.

### Ajouter un fichier KML

Le code correspondant est visible à cette adresse :

<http://jeannoel.anderruthy.googlepages.com/google-maps-api08.html>.

Le fonctionnement de cette page est, en tout point, identique à l'exemple précédent.




### Ajouter un itinéraire

Le code correspondant est visible à cette adresse :

<http://jeannoel.anderruthy.googlepages.com/google-maps-api09.html>.





Cannes, France		
33,6 km (environ 35 minutes)		
1.	Prendre la direction <b>ouest</b> sur <b>Rue de la Castre</b> vers <b>Rue Louis Perrissol</b>	0,2 km
2.	Tourner complètement à <b>droite</b> sur <b>Rue Louis Perrissol</b>	0,1 km
3.	<b>Rue Louis Perrissol</b> tourne légèrement à <b>gauche</b> et devient <b>Rue Hibert</b>	0,1 km
4.	Tourner à <b>droite</b> sur <b>Rue Georges Clemenceau</b>	0,1 km
5.	Tourner à <b>droite</b> sur <b>Avenue des Anciens Combattants d'Afrique du N</b>	0,5 km
6.	Continuer sur <b>Avenue Bachaga Boualam</b> (panneaux vers <b>A8/Antibes</b> )	0,4 km
7.	Tourner à <b>gauche</b> sur <b>Place du 18 Juin</b> (panneaux vers <b>Grasse/Le Cannet</b> )	14 m
8.	Continuer sur <b>Boulevard Carnot</b>	1,2 km
9.	Continuer sur <b>Boulevard Sadi Carnot</b>	0,6 km
10.	Continuer sur <b>D6285/Rue Georges Guynemer</b>	1,9 km

Gdirections permet de créer une nouvelle instance d'un itinéraire en interrogeant puis en stockant les résultats correspondants. La méthode `GDirections.load` est utilisée afin de définir l'itinéraire qui sera affiché.

#### Ajouter un Layer Wikipédia

Le code correspondant est visible à cette adresse :

<http://jeannoel.anderruthy.googlepages.com/google-maps-api10.html>.

Les valeurs disponibles sont répertoriées sur cette feuille de calcul :

<http://spreadsheets.google.com/pub?key=p9pdwsai2hDN-cAocTLhnag>.



**10e arrondissement de Paris**

Le 10e arrondissement de Paris est un des vingt arrondissements de Paris, situé sur la rive droite de la Seine. Aux termes de l'article R2512-1 du Code général des collectivités territoriales (partie réglementaire), il porte également le nom d', mais cette appellation est rarement employée dans la vie courante.

La classe `GLayer` initialisée un calque de données en utilisant l'interface `GOverlay`. Les données peuvent aussi être ajoutées à la carte grâce à la méthode `GMap2.addOverlay()`.

#### Ajouter des données Street View

Le code correspondant est visible à cette adresse :

<http://jeannoel.anderruthy.googlepages.com/google-maps-api11.html>.



Il est aussi possible d'utiliser une méthode appelée `setLocationAndPOV()` dans laquelle vous allez définir à la fois la localisation géographique et le point de vue choisi :

```
var monPanoramique = new
GStreetviewPanorama(document.getElementById("panoramique"));TEST = new
GLatLng(42.345573,-71.098326);monPOV = {yaw:370.64659986187695,pitch:-
20};monPanoramique.setLocationAndPOV(TEST, monPOV);
```

Afin de déterminer si un lieu supporte la fonctionnalité Street View, il faut utiliser un autre objet appelé `GStreetviewOverlay` :

```
var map = new GMap2(document.getElementById("map_canvas"));map.setCenter(new
GLatLng(37.4419, -122.1419), 13);svOverlay = new
GStreetviewOverlay();map.addOverlay(svOverlay);
```

Il ne vous reste plus ensuite qu'à ajouter une réponse en image dès qu'un internaute va cliquer sur un des liserés de couleur bleue. L'objet utilisé s'appelle `GStreetviewPanorama`.

```
var monPanoramique = new
GStreetviewPanorama(document.getElementById("panoramique"));var map = new
GMap2(document.getElementById("map_canvas"));map.setCenter(new GLatLng(42.345573,-
71.098326), 14);svOverlay = new
GStreetviewOverlay();map.addOverlay(svOverlay);GEvent.addListener(map,"click",
function(overlay,latlng) { monPanoramique.setLocationAndPOV(latlng);});
```

Dernière étape : il existe des méthodes avancées d'utilisation de cette classe d'objets qui vous permettent de répondre automatiquement à des requêtes de la part des internautes et dont un exemple intéressant est accessible à partir de cette page :

<http://code.google.com/apis/maps/documentation/examples/streetview-data.html>.

D'autres illustrations de cette fonctionnalité sont proposées sur le blog de Google Maps API et dans cette galerie dédiée :

<http://code.google.com/apis/maps/documentation/demogallery.html?searchquery=street>.

Cette utilisation est particulièrement intéressante puisqu'elle vous permet de visualiser votre itinéraire à la fois sur une carte et en bénéficiant de la fonction StreetView : [http://gmaps-samples.googlecode.com/svn/trunk/streetview/streetview\\_directions.html](http://gmaps-samples.googlecode.com/svn/trunk/streetview/streetview_directions.html).

Cliquez simplement sur le bouton *Route*.

Comme vous avez pu le voir, l'utilisation de cette API n'est pas très simple d'utilisation mais elle permet de faire réellement un nombre très important de choses. Afin de mieux vous en rendre compte, visitez la galerie dédiée à cette API qui est accessible à cette adresse :

<http://code.google.com/apis/maps/documentation/demogallery.html>.

Vous pouvez effectuer une recherche par thématique ou en fonction de la classe employée.

#### *Google Maps API pour Flash*

Cette API permet aux développeurs Flex de publier des cartes Google Maps dans un environnement en Flash.

Vous devrez :

- Signer pour une clé Google Maps API ;
- Installer le SDK Adobe Flex ;
- Installer le SDK pour Google Maps API.

Ce SDK se télécharge à partir de cette adresse :

<http://maps.googleapis.com/maps/flash/release/sdk.zip>.

Décompressez l'archive ZIP à la racine de votre disque dur ou dans un dossier dédié. Bien entendu, il est plus simple qu'il fasse partie des variables d'environnement définies sur votre système.

Une fois l'archive Zip décompressée, vous allez obtenir deux répertoires : *docs* et *lib*.

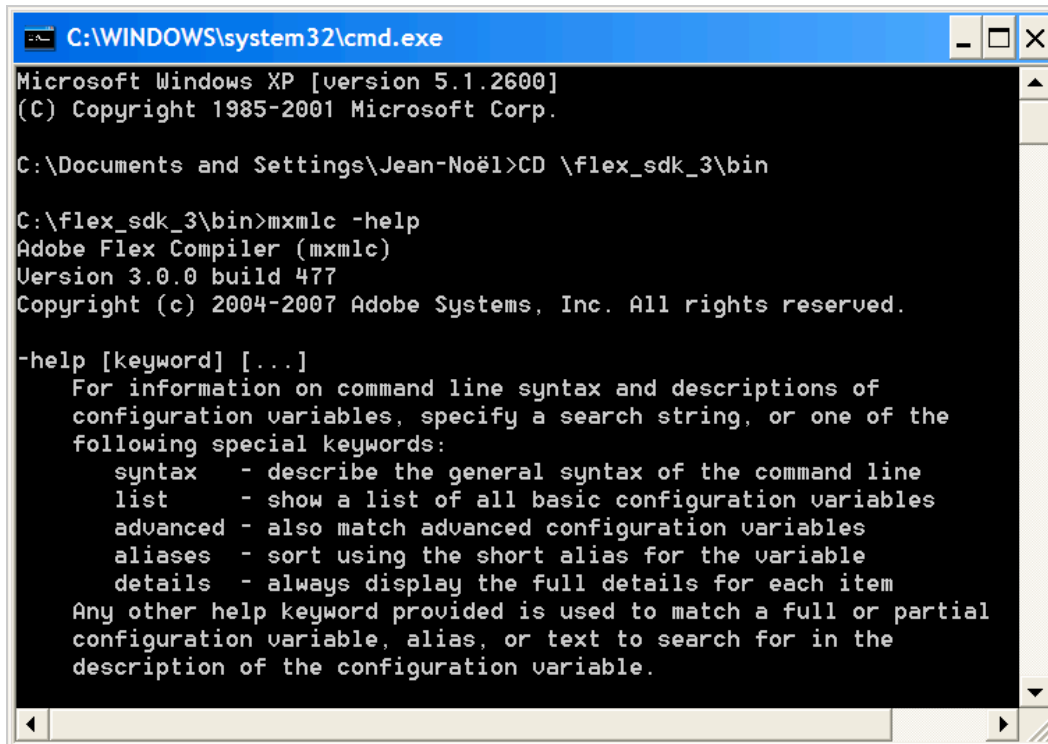
Il faut maintenant télécharger le SDK Adobe Flex :

<http://www.adobe.com/products/flex/overview/#section-3>.

1. Cliquez sur le lien *Get the Free Adobe Flex 3 SDK*.



2. En bas de page, cochez la case *I have read the Adobe Flex 3.0 SDK License*.
  3. Cliquez sur le lien *Download the Flex 3.0 SDK for all Platforms*.
  4. Décompressez l'archive ZIP.
  5. Ouvrez une fenêtre d'Invite en exécutant cette commande : `cmd`.
  6. Placez-vous dans cette arborescence en utilisant la commande `CD \flex_sdk_3\bin`.
- Afin de vérifier que tout fonctionne correctement, exécutez ceci : `mxmlc -help`.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Jean-Noël>CD \flex_sdk_3\bin

C:\flex_sdk_3\bin>mxmlc -help
Adobe Flex Compiler (mxmlc)
Version 3.0.0 build 477
Copyright (c) 2004-2007 Adobe Systems, Inc. All rights reserved.

-help [keyword] [...]
  For information on command line syntax and descriptions of
  configuration variables, specify a search string, or one of the
  following special keywords:
    syntax - describe the general syntax of the command line
    list   - show a list of all basic configuration variables
    advanced - also match advanced configuration variables
    aliases - sort using the short alias for the variable
    details - always display the full details for each item
  Any other help keyword provided is used to match a full or partial
  configuration variable, alias, or text to search for in the
  description of the configuration variable.
```

Mxmlc est le compileur Flex...

Il faut maintenant ajouter l'ensemble des méthodes offertes par l'API Google maps en Flash :

1. Ouvrez le répertoire `\sdk\lib`.
  2. Copiez le fichier `map_flex_1_1.swc` dans `\flex_sdk_3\frameworks\libs\air`.
- Dans le cas contraire, vous aurez ce message d'erreur quand vous compilerez le fichier XML :
- "Accès à la propriété non définie".*

#### Coder en MXML

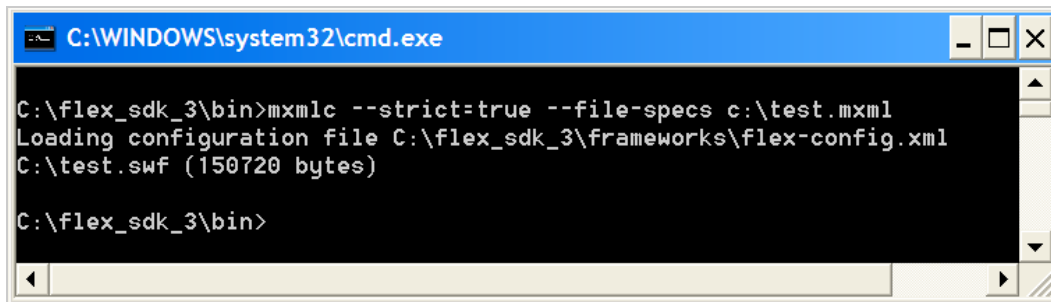
Le MXML est le langage de description développé par Macromedia, puis repris par Adobe Systems pour la plateforme Adobe Flex. Voyons comment créer votre première mini-application :

1. Ouvrez un nouveau document Bloc-notes ou, mieux, servez-vous d'un éditeur comme Notepad++ (<http://notepad-plus.sourceforge.net/fr/site.htm>).
2. Copiez le code suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
xmlns:mx="http://www.adobe.com/2006/mxml"
horizontalAlign="center" verticalAlign="center">
  <mx:Button id="monbouton" label="Je suis un bouton !" />
</mx:Application>
```

3. Enregistrez le fichier en ayant soin de lui affecter une extension `.mxml`.
4. Ouvrez une fenêtre d'Invite de commandes.
5. De la même façon que précédemment, placez-vous dans le répertoire bin puis saisissez ce type de commande : `mxmlc --strict=true --file-specs c:\test.mxml`.

Un fichier SWF va être créé dans le même répertoire.



```
C:\WINDOWS\system32\cmd.exe
C:\flex_sdk_3\bin>mxm1c --strict=true --file-specs c:\test.xml
Loading configuration file C:\flex_sdk_3\frameworks\flex-config.xml
C:\test.swf (150720 bytes)
C:\flex_sdk_3\bin>
```

Vous pouvez le lire en téléchargeant un Player gratuit comme GeoVid Flash Player :

[http://www.geovid.com/Flash\\_Player](http://www.geovid.com/Flash_Player).

Nous allons maintenant voir comment créer rapidement votre première application Adobe Flex.

Notez que vous pouvez aussi consulter la documentation en français :

[http://www.adobe.com/fr/devnet/flex/quickstart/handling\\_events](http://www.adobe.com/fr/devnet/flex/quickstart/handling_events).

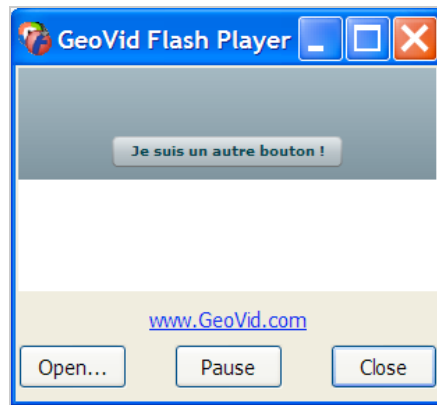
### Utiliser ActionScript

Les balises MXML correspondent aux classes (ou propriétés de classe) ActionScript. Flex va parcourir les balises MXML et générer les classes ActionScript correspondantes.

De la même manière que précédemment copiez cette fois-ci ce contenu :

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
xmlns:mx="http://www.adobe.com/2006/mxml"
viewSourceURL="src/GettingStartedActionScript/index.html"
creationComplete="creationCompleteHandler();"
width="300" height="80">
<mx:Script>
<![CDATA[
import mx.controls.Button;
import mx.events.FlexEvent;
private var monBouton:Button;
private function creationCompleteHandler():void
{
monBouton = new Button();
monBouton.label = "Je suis un autre bouton !";
monBouton.addEventListener (FlexEvent.CREATION_COMPLETE,
buttonCreationCompleteHandler);
addChild (monBouton);
}
private function buttonCreationCompleteHandler ( evt:FlexEvent ):void
{
monBouton.x = parent.width/2 - monBouton.width/2;
monBouton.y = parent.height/2 - monBouton.height/2;
}
}]>
</mx:Script>
</mx:Application>
```

Ouvrez le fichier SWF dans un Player quelconque afin de vérifier que tout fonctionne correctement...



### Créer votre première carte en Flash

Afin de créer un fichier XML, rendez-vous à cette adresse : <http://code.google.com/p/gmaps-samples-flash/source/browse/trunk/samplecode/MapSimple.mxml>.

Cliquez sur le lien *View Raw File* puis copiez le code généré dans un fichier MXML.

Nous copions le contenu d'un fichier nommé **MapSimple.mxml**.

Compilez le fichier comme expliqué précédemment.

Vous allez avoir ce message d'erreur : *"Content is not allowed in prolog"*.

Il suffit de supprimer ces lignes (ainsi que les tirets en début de certaines déclarations) :

- *<!-- Copyright 2007 Google Inc.*

*-->*

- *<!-- All Rights Reserved.*

*-->*

De manière un peu similaire vous pouvez avoir ce message d'erreur *"Whitespace is not allowed before an XML Processing Instruction"*. Vous devez supprimer les espaces vides qui peuvent précéder une de vos déclarations.

Si vous voulez afficher le fichier SWF contenant du code utilisant l'API Google Maps, ce message d'erreur va apparaître : *"Initialization failed: please specify an API key"*.

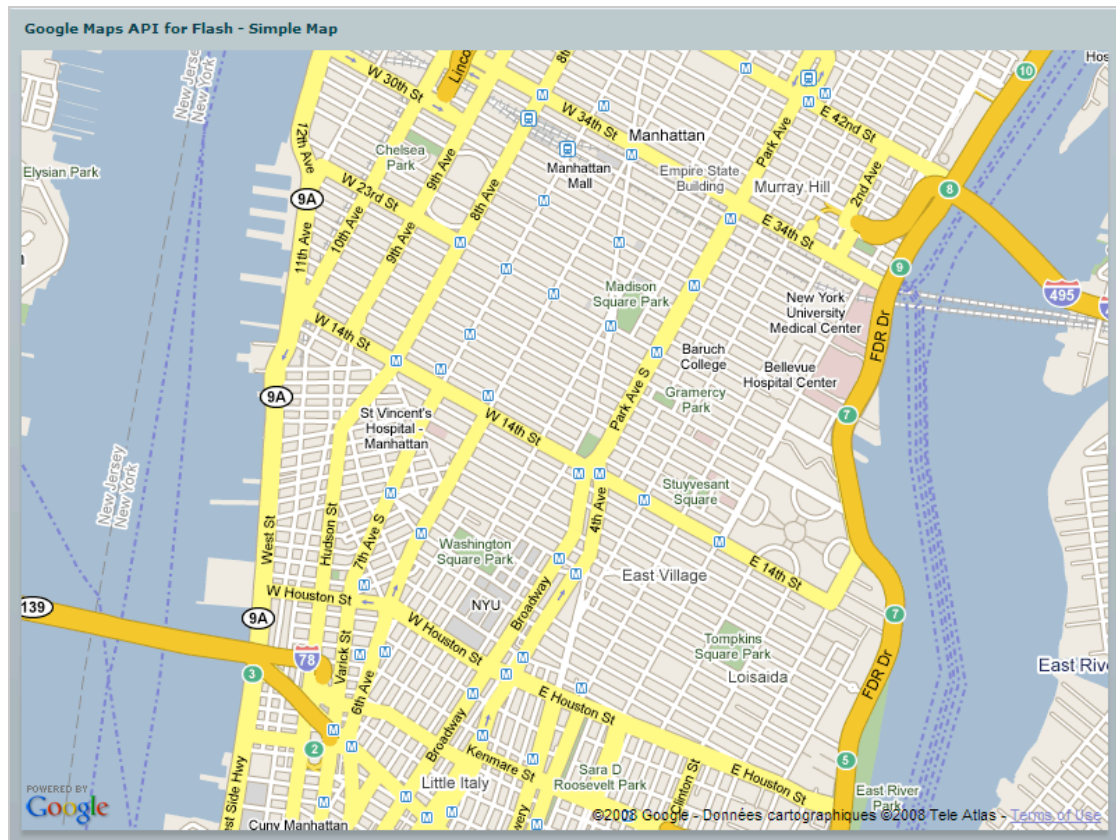
Vous devez donc utiliser votre propre clé API, héberger le fichier SWF sur un serveur et créer une page HTML qui fera un appel vers ce fichier (via un code ActionScript).

Le code à utiliser dans une page HTML peut ressembler à celui-ci :

```
<div id="map_canvas" name="map_canvas">
  <object
    classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
    ab#version=6,0,29,0"
    width="800px"
    height="600px">
    <param name="movie" value="Emplacement du fichier SWF">
    <param name="quality" value="high">
    <param name="flashVars" value="key=Votre clé API">
    <embed
      width="800px"
      height="600px"
      src="Emplacement du fichier SWF"
      quality="high"
      flashVars="key=Votre clé API&client=google-flashmaps"
      pluginspage="http://www.macromedia.com/go/getflashplayer"
      type="application/x-shockwave-flash">
    </embed>
  </object>
</div>
```

Vous en avez un exemple à cette adresse :

<http://jeannoel.anderruthy.googlepages.com/test.html>.



Afin d'avoir une idée des multiples possibilités qui vous sont offertes, visitez la galerie dédiée aux applications utilisant l'API Google Maps pour Flash :

<http://code.google.com/apis/maps/documentation/flash/demogallery.html>.

### Créer une carte au format Flash avec UMapper

UMapper (<http://www.umapper.com>) est une application en ligne vous permettant, sans aucune connaissance en programmation, de créer des cartes en Flash. Elle est basée sur un composant appelé Umap (*Universal Mapping API for Adobe Flash™*). Il est possible d'utiliser les services de cartographie proposés par Google Maps, Yahoo: Maps, Virtual Earth ou OpenStreetMap.

Afin d'utiliser simplement ce service, il vaut mieux créer un compte sur UMapper.

1. Cliquez ensuite sur le bouton *Create Map* puis indiquez un titre, une description et un ou plusieurs libellés (tags).
2. Cochez ensuite l'option de votre choix :
  - *Who can view this map?* : audience de votre carte ;
  - *Who can edit this map?* : définition d'une carte collaborative sur laquelle vous travaillerez à plusieurs ;
  - *Which map provider would you like to use?* : nom du service de cartographie que vous souhaitez utiliser (Virtual Earth, Google Maps ou OpenStreet).

Cliquez sur le lien *Search by location* afin de localiser votre carte. Il vous suffit ensuite de cliquer sur le bouton placé à droite pour centrer rapidement la carte sur l'objet de votre recherche ou de vous servir de la commande *Zoom the Map* afin d'adopter rapidement le point de vue voulu.

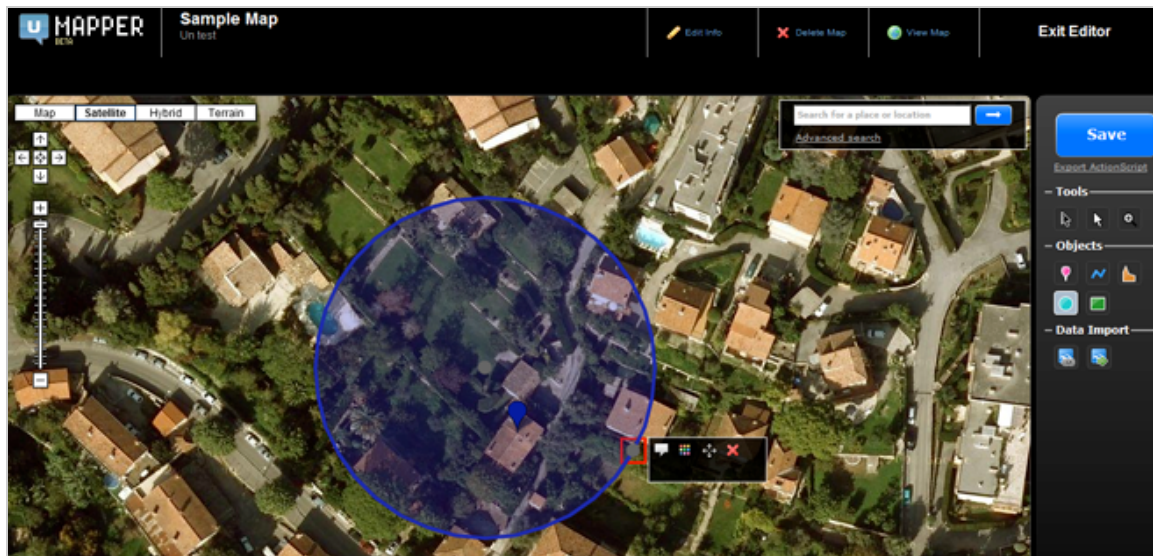
Afin de la positionner de manière interactive, cliquez sur le lien *Drag Map to Navigate*.

Notez que vous pouvez aussi lancer une recherche par sujet en cliquant sur le lien *Search Wikipedia*. C'est une fonctionnalité intelligente qui vous permet de dresser des cartes historiques...

N'oubliez pas de cliquer sur le bouton *Save* afin de sauvegarder au fur et à mesure les changements que vous avez apportés à votre carte...

Vous pouvez ajouter des marqueurs, des lignes ainsi que des polygones en vous servant des liens visibles dans la rubrique *Objects*.

Par ailleurs, il est possible de tracer des cercles...



Cliquez sur le marqueur afin de définir le contenu de la bulle d'information, sa couleur d'arrière-plan, l'épaisseur de la bordure, les effets de transparence et le numéro d'index. Afin d'ajouter une icône personnalisée, cliquez sur le marqueur puis sur l'icône nommée *Marker Appearance*.

Cliquez sur l'onglet *Custom* afin de sélectionner un fichier image présent sur votre disque dur ou indiquez l'adresse URL de votre icône personnalisée.

Dans la fenêtre *Info Windows Content*, cliquez sur le bouton *HTML* afin de définir un code personnalisé et d'insérer, par exemple, une image.

## Google Earth et Virtual Earth

Voici les adresses indispensables à l'utilisation de l'API Google Earth :

- La page d'accueil de cette API : <http://code.google.com/apis/earth> ;
- La documentation : <http://code.google.com/apis/earth/documentation/index.html> ;
- La galerie montrant un éventail des possibilités : <http://code.google.com/apis/earth/documentation/examples.html>.

Les avantages sont les suivants :

- Vous pouvez transformer un Mashup Google Maps en un Mashup Google Earth avec une ligne de code supplémentaire (une déclaration `G_SATELLITE_3D_MAP`) ;
- Il est possible d'afficher des modèles 3D ;
- Vous pouvez créer vos propres Mashups "Google Sky" ;
- Il est possible d'utiliser toute sorte de fichiers KML.

### Installer le plugin Google Earth

Le plugin Google Earth et l'API JavaScript correspondante vous permet d'utiliser les fonctionnalités Google Earth dans vos pages web.



1. Accédez à la page d'accueil de Google Earth API puis cliquez sur le bouton *Get the Google Earth Plugin now*.

Notez que ce plugin ne peut s'installer que sur des systèmes Windows XP et Vista. Le fait d'installer ce plugin pour un navigateur l'installe automatiquement sur les autres qui sont déclarés sur votre machine.

Le plugin Google Earth va s'installer puis il va vous être demandé de redémarrer votre navigateur. La barre d'informations va apparaître et vous demander si vous souhaitez autoriser l'exécution du module complémentaire **GEPlugin**.

2. Cliquez sur la barre d'informations pour autoriser l'exécution de ce contrôle ActiveX.

#### *Créer votre première application utilisant l'API Google Earth*

Vous devez posséder une clé Google Maps API. Affichez ce très bon exemple de ce qu'il est possible de faire sur cette page de démonstration :

<http://www.google.com/earth/plugin/examples/samples/index.html>.

Notez qu'il vous faut créer une exception dans votre pare-feu de connexion Internet pour cette application : `\Program Files\Google\Google Earth Plugin\geplugin.exe`.

Avec l'API Google Earth, vous pouvez :

- Superposer une image fixe à une carte ;
- Ajouter un calque de données (Relief, Bâtiments 3D, Frontières et légendes) ;
- Créer une animation quand l'internaute clique sur un repère ;
- Afficher les données d'un fichier KML ;
- Afficher des modèles comme un fichier Collada ;
- Basculer en mode Sky.

La galerie propose une mini-application appelée *Options* qui vous explique comment retrouver l'interface complète de Google Earth sur une page web.

Vous avez un exemple de l'intégration à Google Maps (ajout d'un bouton *Google Earth*) à partir de cette adresse : <http://jeannoel.anderruthy.googlepages.com/google-api-earth-12.html>.

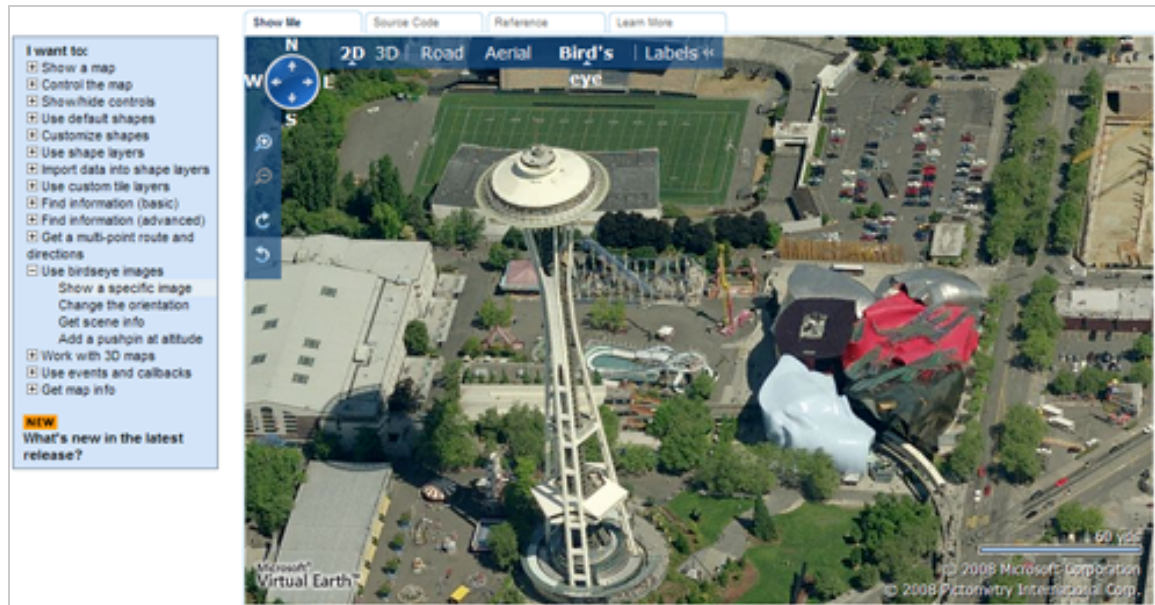




### Utiliser le SDK Virtual Earth

Cet assistant vous permet de créer facilement une carte utilisant l'API Virtual Earth. Cette API ne nécessite pas de clé !

1. Accédez à cette adresse : <http://dev.live.com/virtualearth/sdk>.



2. Sélectionnez une des actions visibles dans le tableau de gauche. Le résultat va s'afficher sur la droite.

3. Cliquez sur l'onglet *Source Code* afin d'afficher le code source nécessaire.

4. Cliquez sur le bouton *Copy* afin de récupérer le code et, ultérieurement, le modifier.

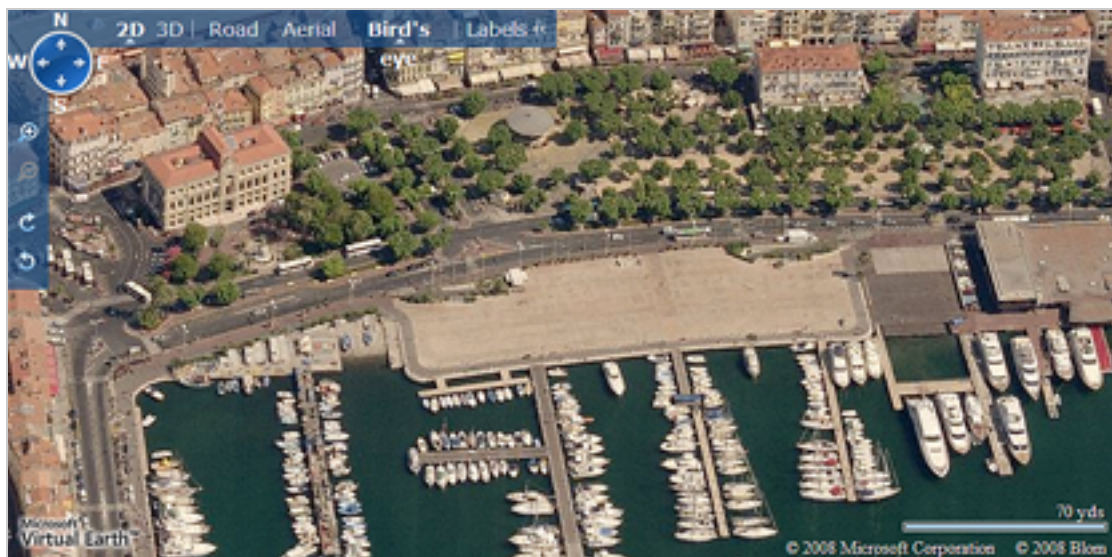
Voici un code très simple :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script type="text/javascript"
src="http://dev.virtualearth.net/mapcontrol/mapcontrol.ashx?v=6.2"></script>
<script type="text/javascript">
var map = null;
function GetMap()
{
map = new VEMap('myMap');
map.LoadMap(new VELatLong(43.55141,7.009792), 10, 'h', false);
}
</script>
</head>
<body onload="GetMap();">
<div id='myMap' style="position:relative; width:400px; height:400px;"></div>
</body>
</html>
```

Il est disponible à cette adresse : <http://jeannoel.anderruthy.googlepages.com/virtual-earth01.html>.

Ajoutons la même carte mais avec la vue Bird's Eye directement paramétrée :

<http://jeannoel.anderruthy.googlepages.com/virtual-earth02.html>.



### Le GeoWeb

L'avènement de la géolocalisation et, plus généralement, du GeoWeb, marque l'irruption de la "vraie vie" sur la Toile. C'est une chose que de tenir un blog sur la vie nocturne à Ibiza, c'en est une autre de publier des cartes des principaux lieux festifs, de proposer un service de "Push Mail" des événements musicaux, de géolocaliser précisément chaque membre de sa communauté, de créer des itinéraires en mode *Street View*, de publier en temps réel du contenu multimédia et de rendre l'ensemble de ces informations immédiatement accessibles sur tout type de terminal (ordinateur, téléphone mobile, console de jeu, etc.).

Le monde est maintenant à la portée de chacun et semble pouvoir tenir dans un mouchoir de poche. En ce sens, la géolocalisation joue un rôle déterminant dans le façonnement d'un Web ancré dans notre existence quotidienne. Et si les services diffusant de la réalité augmentée forment comme autant de prolongement de nous-mêmes, ils offrent également une approche à la fois ludique et informative de la connaissance.

**Jean-Noël Anderruthy**, *webmaster spécialisé dans les technologies Google.*

Réagissez à cet article sur le blog des abonnés d'Abondance :

<http://abonnes.abondance.com/blogpro/2008/12/les-api-de-cartographie-en-ligne-2me.html>