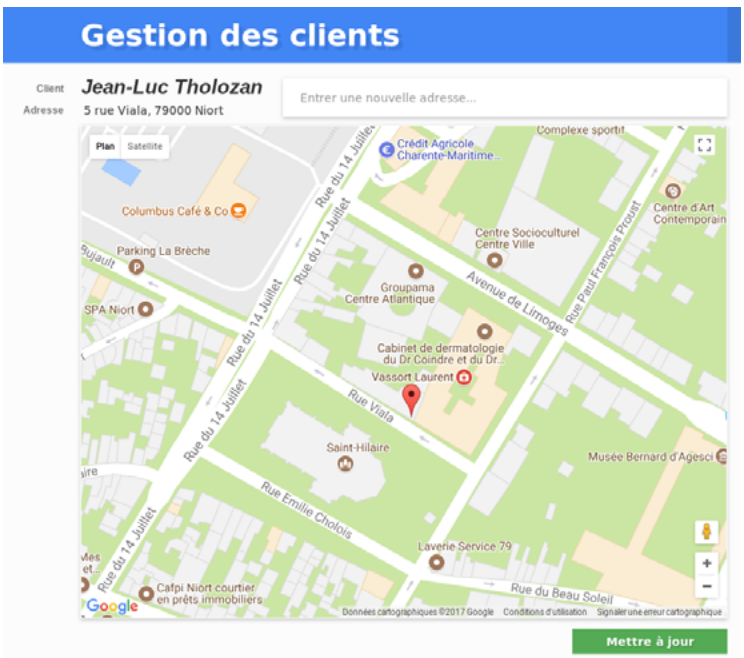
Fiche Technique N°5
NatJet & Google Maps

L'intégration de cartes est de plus en plus courante dans les applications web : elles permettent la **localisation rapide d'un site ou d'un ensemble d'établissements.** La solution la plus répandue aujourd'hui repose sur l'intégration de **Google Maps** grâce à ses API publiques en javascript.



NatJet est un outil qui, bien que s'appuyant sur deux technologies, Java et Javascript, masque complètement aux développeurs la seconde, permettant à ceux-ci de n'avoir à acquérir qu'une seule compétence : **Java**.

Le sentiment de **simplicité** et de **confort** qu'offre NatJet ne l'empêche cependant pas de faire preuve de **souplesse** afin de s'adapter à vos besoins d'architecture.

L'intégration de nouvelles briques dans NatJet est simple mais requiert de l'expertise sur la nouvelle brique. Dans le cadre de Google Maps, cela va nécessiter un **savoir faire Javascript et une connaissance des API Google Maps**.

Pour assurer l'intégration de Google Maps dans votre application, il faut d'abord se **familiariser avec la documentation du service** : il demande en effet l'obtention d'une clé et le respect de certaines règles.

Pour obtenir votre **clé Google Maps**, connectez vous à la page *https://developers.google.com/maps/documentation/javascript/get-api-key?hl=Fr,* et suivez les indications fournies.

Une fois la clé obtenue, revenons dans Natjet pour **créer une page HTML qui intégrera la carte Google**. La page sera créée dans un **sous-répertoire «map»** dans le **répertoire «WebContent»** de l'application NatJet.

Pour ce faire, créer un **nouveau répertoire «Map»** dans le **«WebContent»** de l'application NatJet.

Dans ce dossier , créer un **fichier «map.html»** avec le contenu suivant :

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Google map</title>
        <style>
            #map {
                height: 400px;
                width: 100%;
            }
        </style>
    </head>
    <body>
        <div id="map"></div>
        <script src="map.js"></script>
        <script async defer src="https://maps.googleapis.com/maps/
api/js?key=USER-KEY&callback=API.init">
        </script>
    </body>
</html>
```

Il s'agit d'une **page simple** avec un élément **div#map** auquel on donne 400px de hauteur.
C'est cet élément qui sera utilisé pour **afficher la carte Google**, à l'aide du javascript chargé par les **2 appels de script** : un fichier **«api.js»** que nous allons créer et un **fichier Google** chargé par la seconde déclaration. Il faut y **remplacer «USER-KEY»** par la clé récupérée plus haut.

C'est ce fichier qui fera tout le travail de création de la carte et qui appellera notre code javascript.

Créer ensuite le fichier **javascript «api.js» dans le dossier «Map»** avec le code suivant :

Fiche TEchnique N°5 - NatJet & Google Maps

```
var API = {
    /**
     * @type {google.maps.Geocoder}
     */
    geocoder: null,
    /**
     * @type {google.maps.Map}
     */
    map: null,
    /**
     *  Initialisation de la carte Google. Méthode de callback appelée
par
     *  les Apis de google map.
     */
    init: function () {
        console.log("API.init()");
        API.map = new google.maps.Map(document.getElementById("map"), {
            zoom: 9,
            center: {lat: 46.52863469527167, lng: 2.43896484375}
        });
        API.geocoder = new google.maps.Geocoder();
    },

    /**
     * Afficher  dans la carte les adresses demandées.
     * @param {String[]} address Adresse à afficher
     */
    geocodeAddress: function (listeAdresse) {
        console.log("API.geocodeAddress(): " + listeAdresse);
        listeAdresse.forEach(function (address) {
            API.geocoder.geocode({"address": address}, function (re-
sults, status) {
                if (status === "OK") {
                    API.map.setCenter(results[0].geometry.location);
                    API.marker = new google.maps.Marker({
                        map: API.map,
                        position: results[0].geometry.location
                    });
                } else {
                    console.warn("Geocode was not successful for " +
                            "the following reason: " + status);
                }
            });
        });
    }
};
```

Nous créons un **objet API avec 2 variables** :

- **map :** objet Google qui représente la carte
- **geocoder :** objet Google permettant de transformer une adresse postale en position GPS

et **2 méthodes** :

- **init :** Appelée par le javascript de Google quand il est chargé dans la page, elle va créer la carte dans le div#map en définissant un facteur de zoom de 9 et en positionnant la carte au centre de la France (coordonnée GPS fixe). Pour finir, elle initialise l'objet geocoder.
- **geocodeAddress :** Méthode qui reçoit une liste d'adresses postales en paramètres pour créer un marqueur sur la carte Google. La méthode sera appelée plus tard dans le code java de l'application NatJet.

Une fois les deux fichiers enregistrés, map.html et api.js dans le dossier «WebContent/map», il est possible de **tester leur fonctionnement** en **relançant** l'application NatJet.
Si votre projet NatJet s'appelle njMap, vous affichez la page principale de votre application NatJet avec l'url *localhost:8080/njMap/app*.
Pour **afficher la page map.html**, tapez l'url : *localhost:8080/njMap/map/map.html*.

Si vous avez un accès à internet et une clé Google Maps valide, vous devriez voir s'afficher une **carte Google Maps**.

Une fois que vous avez compris comment créer une carte Google, vous avez fait le plus compliqué, **l'intégration dans une application NatJet** va être très simple.
Ouvrez le panneau qui doit accueillir la carte. Donnez une **hauteur minimale de 400px** pour la carte. **Glissez&déposez** un composant **WebBrowser** dans votre panneau. **Définissez sa hauteur** (height) à 400px et la largeur que vous souhaitez.
Vous pouvez définir un **X Anchor à BeginFixedAndEndFixed** si vous souhaitez que la **carte s'étende en largeur** en fonction de l'espace disponible : Le X Anchor va agrandir l'espace disponible en largeur pour la carte, et comme le style de la carte est pour width: 100%, la carte va elle-même afficher plus d'informations.

Le composant **WebBrowser** sert à **afficher des pages** dans une application NatJet ; ici on va afficher notre page map.html créée plus haut grâce à la propriété Uri en lui donnant la valeur */map/map.html*.

À ce niveau, si vous relancez votre application NatJet et que vous affichez votre panneau, vous verrez la page afficher une **carte centrée sur la Vallon-en-Sully**, comme lorsque l'on a affiché la page map.html directement via l'url :
localhost:8080/njMap/map/map.html.

Votre objectif étant de **positionner un marqueur** provenant de votre application NatJet, nous allons ajouter un **bouton Positionner (positionPB)** et un **champs de saisi d'adresse (streetNameTF)** dans le panneau. Le code de l'événement Executed du bouton est très simple :

```
@Override
public void positionPB_ExecutedEvent(NsExecutedEvent event) {
    String jsCommand = "API.geocodeAddress(['" + streetnameTF.getText() +
"'])";
    mapWB.performJavascript(jsCommand);
}
```

On récupère la valeur de l'adresse saisie dans le champ texte et on l'envoie dans une commande javascript *«API.geocodeAddress(['cours des juilliottes Maisons-Alfort'])»*. La commande javascript appelle la méthode **geocodeAddress** écrite plus haut en lui passant la **valeur tapée** dans le champ texte en **paramètre.**

Vous venez d'intégrer en quelques étapes votre carte Google Maps à votre application NatJet.
Retrouvez toutes nos fiches techniques sur notre site **http://www.natsystem.fr/newsletter**