

Programmation mobile avec Xamarin (2)

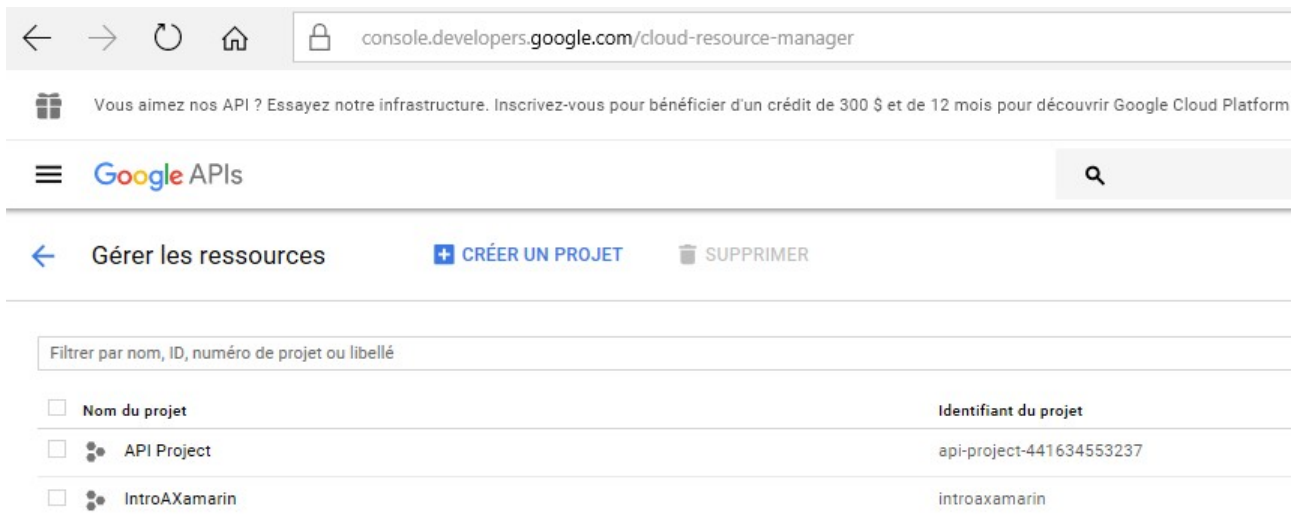
1. Ajout de cartes googleMaps au projet :

Nous allons enrichir notre projet en utilisant des cartes googleMaps afin de localiser nos contacts sur une carte. Ces cartes nous serviront à plusieurs niveaux :

- Pour choisir directement la position d'un contact lorsque nous le créons.
- Pour afficher tous nos contacts sur la carte et voir leur répartition géographique.

1.1. Préparation de l'API Google

Nous devons disposer d'un compte google. Si vous n'en avez pas, créez rapidement une adresse gmail. Ensuite, rendez vous dans les googleAPI pour y créer un nouveau projet (ici IntroAXamarin) :



Depuis le tableau de bord, ajoutez ensuite l'API suivante :

[Google Maps Android API](#)

Créez ensuite une clé sans restriction (ce n'est pas recommandé pour des questions de sécurité mais nous nous contenterons pour ce premier travail). Notez bien la clé générée qui va vous servir plus tard.

Une procédure plus complète peut être suivie ici en cas de problème : (en anglais)

https://developer.xamarin.com/guides/android/platform_features/maps_and_location/maps/obtaining_a_google_maps_api_key/

1.2. Préparation du projet sous Visual Studio

Il faut maintenant suivre la procédure décrite ici :

<https://developer.xamarin.com/guides/xamarin-forms/user-interface/map/>

Pour récapituler, vous devez tout d'abord :

- installer le package nuGet **Xamarin.Forms.Maps**. Dans l'activité MainActivity.cs du projet Android, il faut appeler la méthode init comme ceci :

```
namespace IntroAXamarin.Droid
{
    [Activity(Label = "IntroAXamarin", Icon = "@drawable/icon", Theme = "@style/MainTheme", MainLauncher =
true, ConfigurationChanges = ConfigChanges.ScreenSize | ConfigChanges.Orientation)]
    public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
    {
        protected override void OnCreate(Bundle bundle)
        {
            Xamarin.Forms.Maps.Init(this, bundle);

            TabLayoutResource = Resource.Layout.Tabbar;
            ToolbarResource = Resource.Layout.Toolbar;

            base.OnCreate(bundle);

            global::Xamarin.Forms.Forms.Init(this, bundle);
            LoadApplication(new App());
        }
    }
}
```

- Il va falloir ensuite activer des permissions Android pour accéder aux maps. Après ces activations, votre fichier Android.manifest.xml ressemblera à ceci :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:installLocation="auto">
    <uses-sdk android:minSdkVersion="15" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
    <uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application android:label="IntroAXamarin.Android">
        <meta-data android:name="com.google.android.geo.API_KEY" android:value="MACLEAMOI" />
        <meta-data android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
    </application>
</manifest>
```

- Remarquez que vous devrez entrer votre clé API_KEY générée chez google dans ce même fichier.

1.3. Modification de l'application : création du contact

Nous allons insérer une googleMaps pour positionner le lieu d'habitation d'un contact lorsqu'on le crée. Sur le formulaire de saisie, nous allons insérer un Label et un bouton qui ouvrira une carte sur laquelle nous choisirons un point sur la carte.

Voici comment procéder.

Créez une nouvelle page XAML avec le code suivant :

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:maps="clr-namespace:Xamarin.Forms.Maps;assembly=Xamarin.Forms.Maps"
    x:Class="IntroAXamarin.Carte">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Centrez l'habitation de votre ami sur la carte puis validez." />
            <Button x:Name="boutonValider" Text="Valider" Clicked="OnButtonValiderClicked"/>
            <maps:Map WidthRequest="320" HeightRequest="200"
                x:Name="MyMap"
                IsShowingUser="true"
                MapType="Hybrid"
                />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

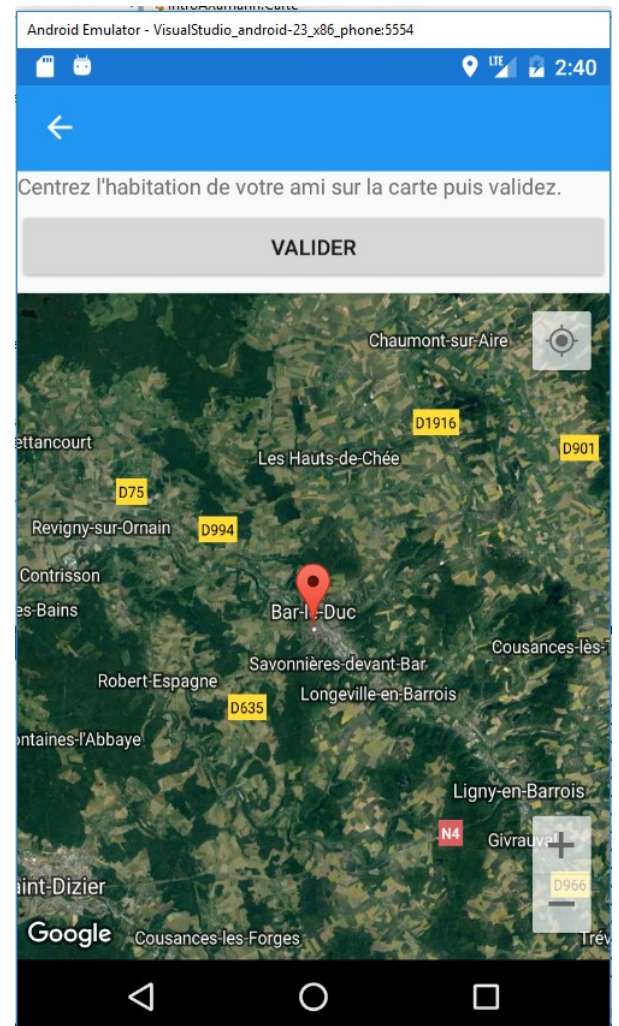
Nous allons demander à l'utilisateur de centrer la carte sur l'habitation de son contact et nous récupérerons les coordonnées de l'emplacement pour le mémoriser lorsqu'il cliquera sur le bouton valider. Le code associé au bouton est alors le suivant :

```
void OnButtonValiderClicked(object sender, EventArgs e)
{
    MyMap.Pins.Clear();

    Carte.latitude = MyMap.VisibleRegion.Center.Latitude;
    Carte.longitude =
    MyMap.VisibleRegion.Center.Longitude;

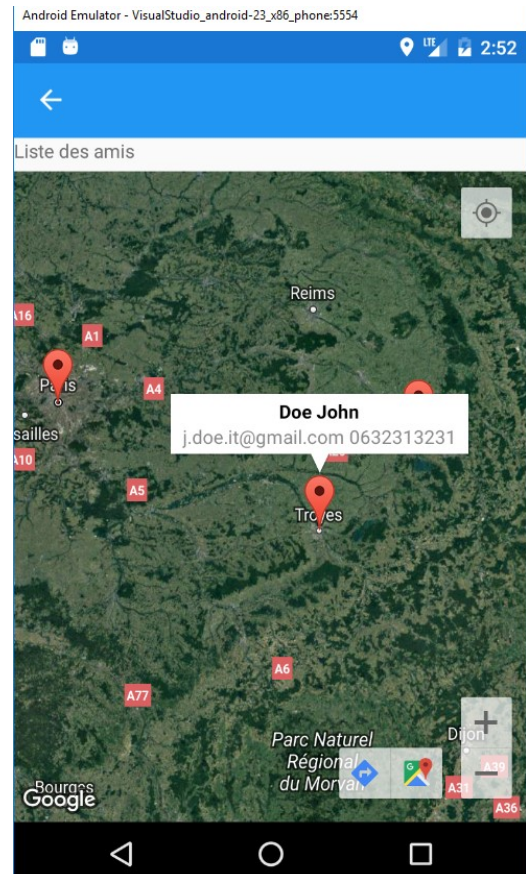
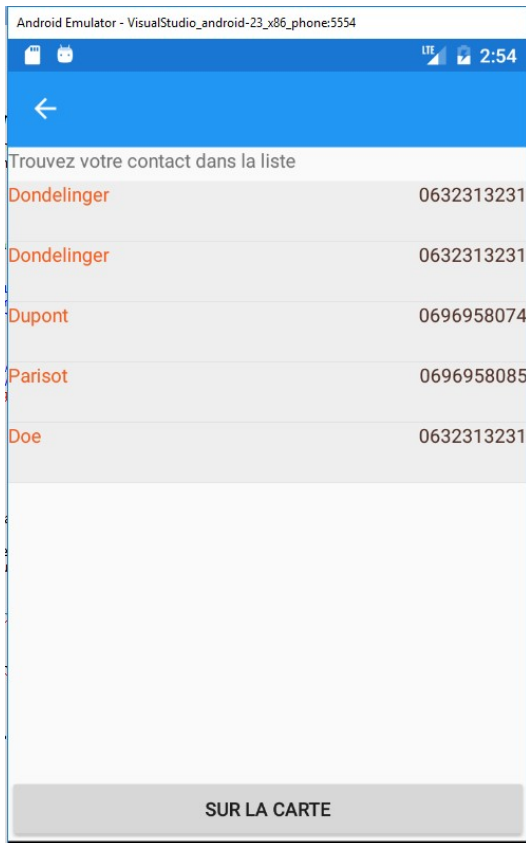
    var position = new Position(Carte.latitude,
    Carte.longitude); // Latitude, Longitude
    var pin = new Pin
    {
        Type = PinType.Place,
        Position = position,
        Label = "Ami",
        Address = "Habite ici !"
    };
    MyMap.Pins.Add(pin);
}
```

On pensera à ajouter les 2 attributs statiques sur la classe Carte pour stocker les coordonnées.



2. A vous de coder !

Fort de vos nouvelles compétences en géo-localisation, vous allez enrichir la page qui liste l'ensemble des contacts en lui ajoutant un bouton pour les visualiser sur une carte googleMaps. Le clic sur un marqueur affichera les informations du contact.



Essayez de modifier vos cartes pour placer la carte sur la zone où se trouve l'utilisateur. Le plugin GeoLocator devrait vous aider et sa documentation est disponible ici :

<https://blog.xamarin.com/geolocation-for-ios-android-and-windows-made-easy/>
<https://jamesmontemagno.github.io/GeolocatorPlugin/>

Quelques liens utiles :

<https://www.supinfo.com/articles/single/4167-xamarin-forms-map>
<https://developer.xamarin.com/guides/xamarin-forms/application-fundamentals/custom-renderer/map/customized-pin/>