Déployer ODAF (Open Data Application Framework) Openturf pour Android

# Détails du document

Version : 2.0

Publication : juin 2012

Auteurs : Philippe Beraud (Microsoft France), Sébastien Brasseur (Microsoft France), Rémi Olivier (Microsoft France), Frédéric Romanczuk (Microsoft France)

Sommaire

[Détails du document 1](#_Toc327455229)

[Avertissement 3](#_Toc327455230)

[Licence 3](#_Toc327455231)

[Résumé 4](#_Toc327455232)

[Commentaires 4](#_Toc327455233)

[Connaissances et attendu 5](#_Toc327455234)

[Prérequis du Kit de démarrage ODAF Openturf pour la partie Android 5](#_Toc327455235)

[Création d’un compte applicatif Twitter 6](#_Toc327455236)

[Etape 1 – Création d’un compte développeur Twitter 6](#_Toc327455237)

[Etape 2 – Mise à jour de la base SQL Azure ODAF Openturf 9](#_Toc327455238)

[Création d’un compte applicatif Google Maps 13](#_Toc327455239)

[Obtenir l’empreinte MD5 de votre environnement de développement/débogage 13](#_Toc327455240)

[Obtenir l’empreinte MD5 de production 14](#_Toc327455241)

[Installation de la bibliothèque Google Maps 14](#_Toc327455242)

[Résolution des dépendances du projet ODAF.Android 15](#_Toc327455243)

[Modification du paramétrage du projet ODAF Openturf pour Android 16](#_Toc327455244)

[Exécution sur un émulateur local 17](#_Toc327455245)

[Restons en contact 21](#_Toc327455246)

# Avertissement

Ce document est fourni uniquement à titre indicatif. MICROSOFT N’APPORTE AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, À CE DOCUMENT. Les informations figurant dans ce document, notamment les URL et les références aux sites Internet, peuvent être modifiées sans préavis. Les risques d’utiliser ce document ou ses résultats sont entièrement à la charge de l'utilisateur.

# Licence

 Le Kit de démarrage ODAF (*Open Data Application Framework*) Openturf est publié sous le contrat de licence de logiciel libre Microsoft Public License (Ms-PL) tel que décrit à l’adresse Internet <http://opensource.org/licenses/ms-pl.html>.

# Résumé

Le kit de démarrage ODAF (*Open Data Application Framework*) Openturf sous license libre est une initiative menée par les équipes Microsoft en charge du [Secteur Public](http://www.microsoft.com/france/entreprises/secteur-public/solutions-pour-secteur-public.aspx)[[1]](#footnote-1). ODAF utilise la plateforme [OGDI(Open Government Data Initiative) DataLab](http://ogdifrance.cloudapp.net/)[[2]](#footnote-2) hébergée sur [Windows Azure](http://www.microsoft.com/france/windows-azure/)[[3]](#footnote-3) dans le Cloud pour faciliter l’affichage sur carte et l'utilisation d’informations publiques, ou données ouvertes (*open data*) géo-localisées variées provenant d'entités publiques (gouvernement, ministères, collectivités locales et territoriales, mairies, services déconcentrés de l'Etat, etc.).

Ce document a pour vocation de traiter plus en détail la partie Android de ce kit. Après un bref rappel des connaissances, prérequis et attendus, nous abordons la mise en œuvre de l’application client mobile ainsi proposée et plus particumièrement les procédures relativesà la création du compte applicatif Twitter associé et à celle d’un compte Google Maps. Nous revenons ensuite sur le paramétrage du projet avant de nous intéresser à l’exécutions dans l’émulateur Android.

Ce document s’adresse aux architectes, aux développeurs et à tous ceux qui souhaitent consommer des informations publiques géolocalisées via la plateforme OGDI DataLab.

Il est fortement conseillé d’avoir lu au préalable les deux documents Démarrer avec le kit de démarrage ODAF Openturf et Déployer le kit de démarrage ODAF Openturf dans Windows Azure puisqu’ils traitent du déploiement de la partie serveur d’ODAF Openturf dans Windows Azure, du paramétrage des projets qui composent le kit ainsi que de la résolution des dépendances des projets.

# Commentaires

Votre opinion nous intéresse. N’hésitez pas à nous transmettre vos commentaires via la boîte aux lettres OGDI France (<mailto:ogdifrance@live.fr>) afin que nous puissions améliorer le kit de démarrage ODAF (*Open Data Application Framework*) Openturf avec la communauté GitHub.

Le [blog MSDN éponyme OGDI France](http://blogs.msdn.com/b/ogdifrance/)[[4]](#footnote-4) vous tient informé dans le même temps des dernières évolutions du kit de démarrage ou de l’actualité et des nouveautés relatives à ce celui-ci.

# Connaissances et attendu

La mise en œuvre de la partie Android du Kit de démarrage ODAF (*Open Data Application Framework*) Openturf requiert une expérience préalable :

* De l’environnement de développement Eclipse.
* Du langage Java,
* Du kit de développement (SDK) Android.

# Prérequis du Kit de démarrage ODAF Openturf pour la partie Android



Pour développer une application sur le système [Android](http://www.android.com/)[[5]](#footnote-5), vous aurez besoin d’une copie du SDK Android (dans sa version 12 ou ultérieure) ainsi que du SDK Java dans sa version 6.

Vous aurez également besoin d’un environnement de développement (IDE) Java, Eclipse étant particulièrement recommandé pour s’intégrer avec le SDK Android.

Toute la procédure d’installation et de préparation du SDK et de l’IDE Eclipse est détaillée sur la [page Installing the SDK](http://developer.android.com/sdk/installing.html)[[6]](#footnote-6)du site Web dédié aux développeurs Android : developer.android.com.

Il convient de noter que la version fournie d’ODAF pour Android s’exécute de base sur la [version 2.2 d’Android (Froyo)](http://developer.android.com/sdk/android-2.2.html)[[7]](#footnote-7).

[](http://twitter.com) Enfin, vous avez aussi besoin de disposer d’un compte Twitter existant. En effet, la version Android du Framework ODAF propose une dimension sociale forte en permettant aux utilisateurs de partager leurs ressentis, impressions ou critiques via Twitter. Vous avez donc besoin d’un compte d’un compte développeur sur Twitter. Nous évoquons dans la suite de ce document la démarche pour créer ce compte.

# Création d’un compte applicatif Twitter

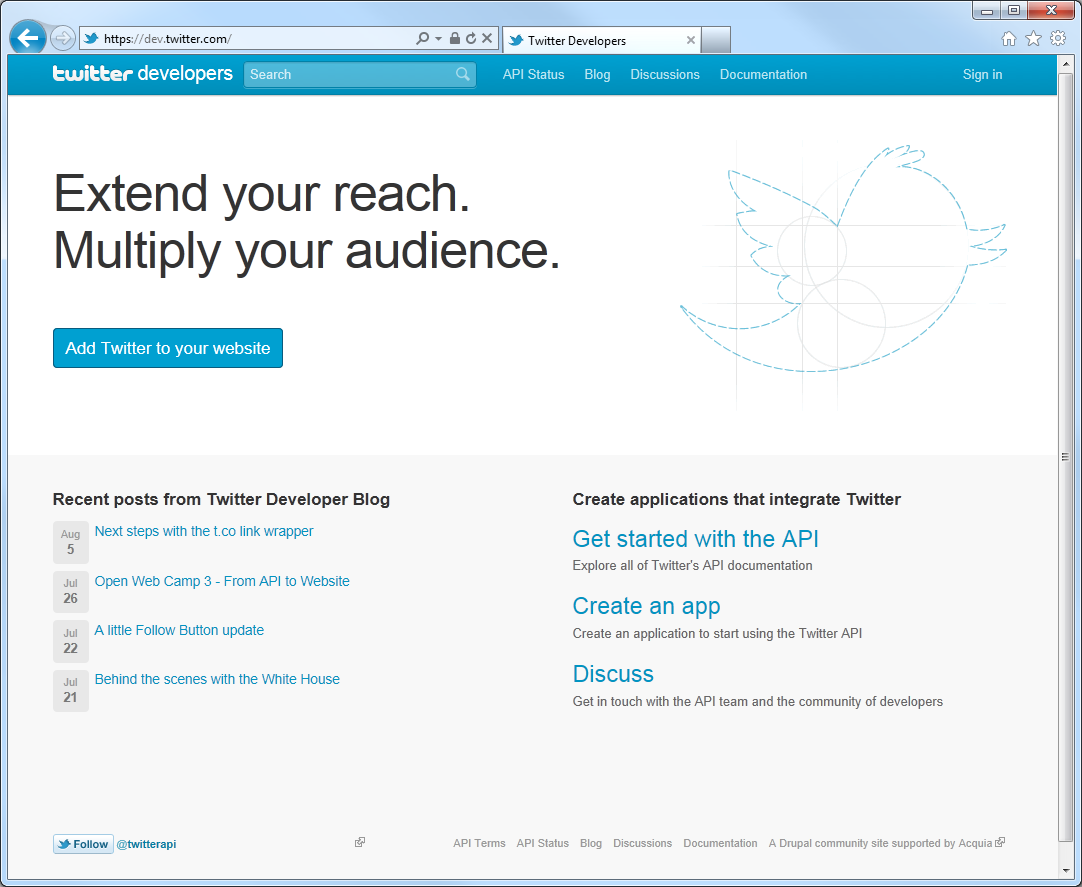
Dans le Framework ODAF Openturf, chaque composante applicative (Web riche, iPhone, Windows Phone et désormais Android) dispose de son propre enregistrement pour l’API Twitter. Vous allez donc voir dans cette section comment créer un compte Twitter pour la composante Android puis vous allez mettre ensuite à jour la base de données SQL Azure de la solution ODAF Openturf pour qu’elle enregistre bien l’application considérée.

## Etape 1 – Création d’un compte développeur Twitter

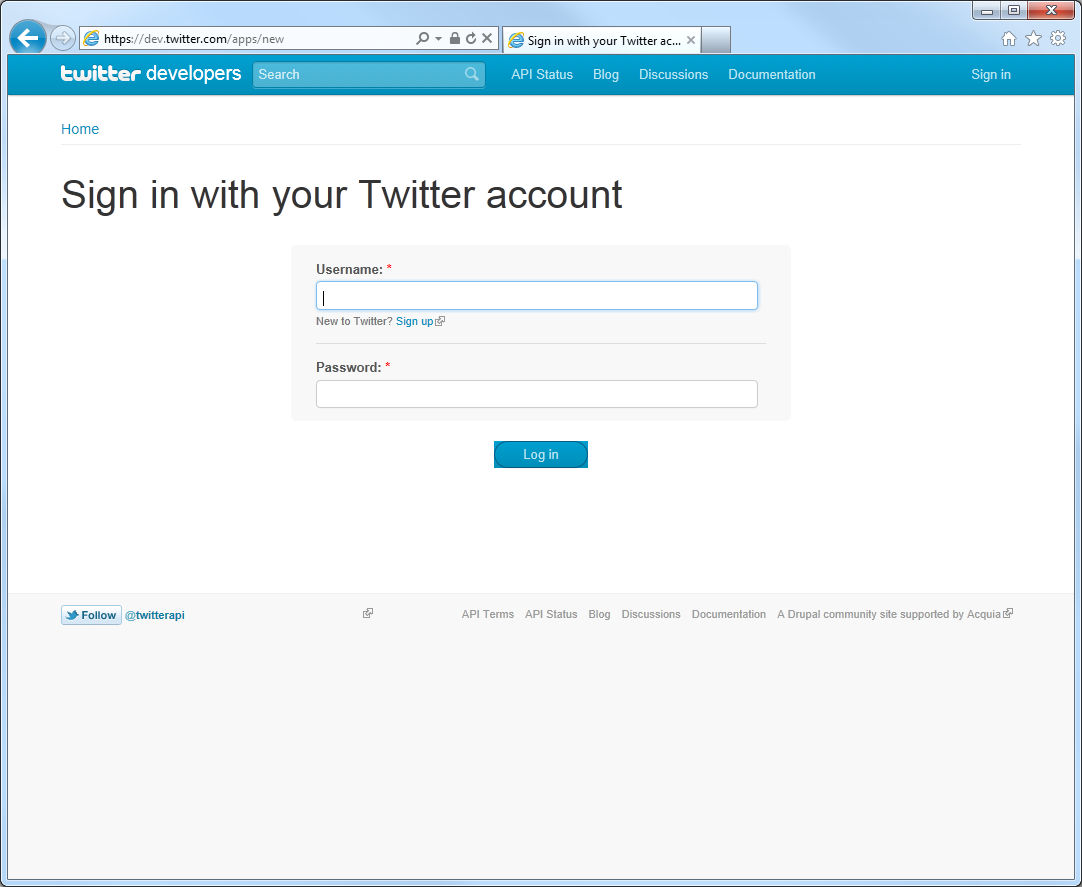
Procédez comme suit :



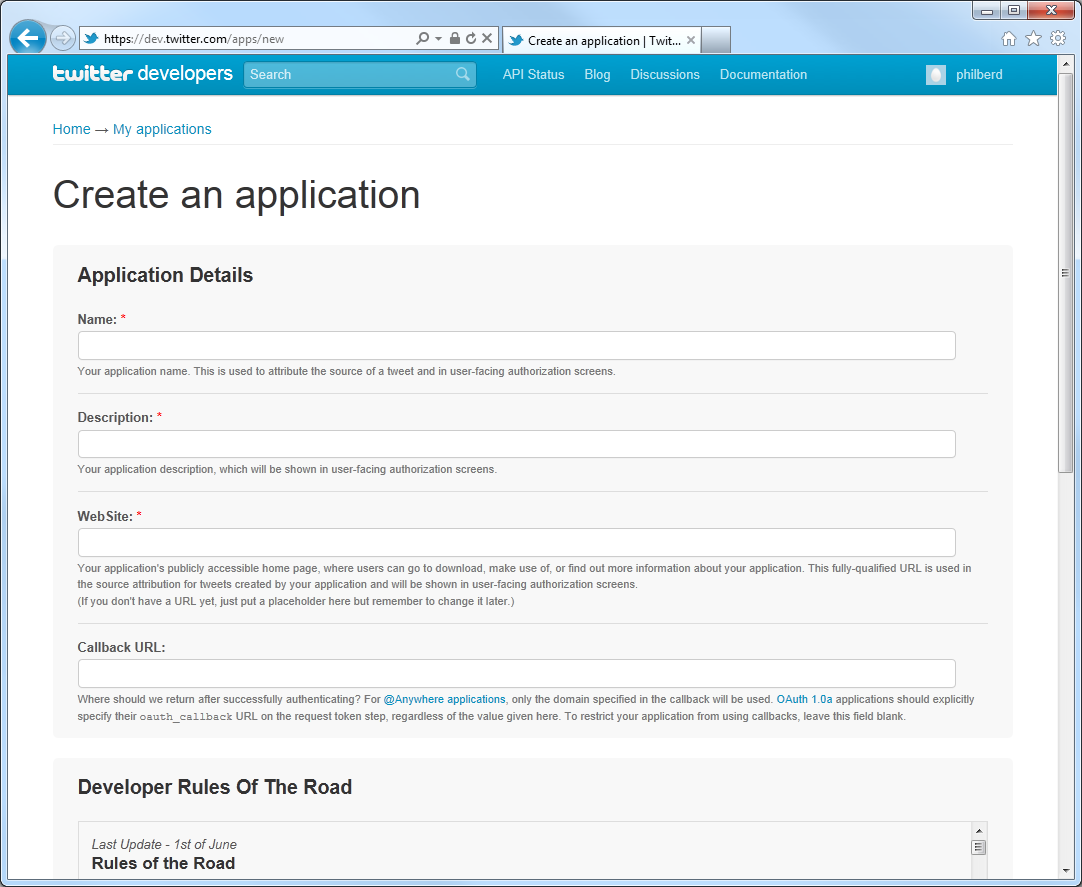
1. Connectez-vous sur le portail développeur de Twitter à l’adresse <https://dev.twitter.com/> et cliquez ensuite sur **Create an app** pour commencer la procédure de création de compte.



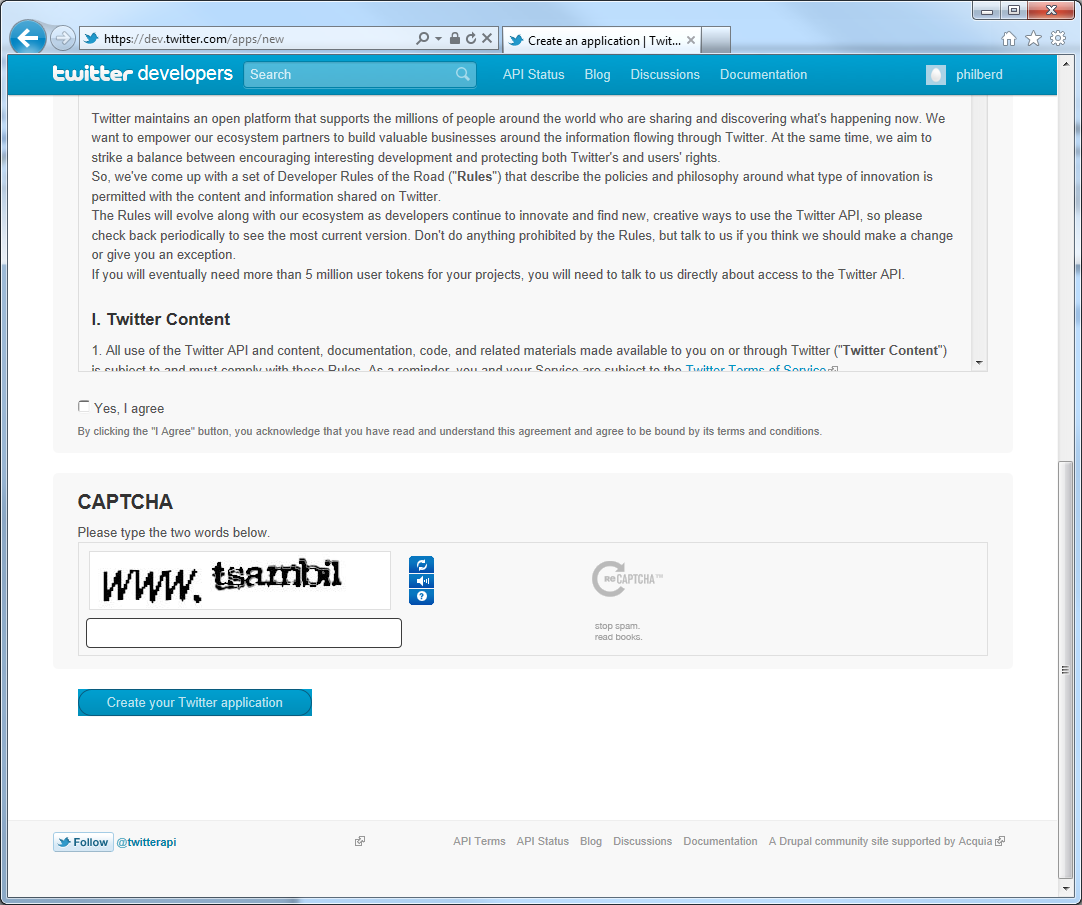
1. Connectez-vous avec les identifiants de votre compte Twitter. Si vous n’avez pas de compte Twitter, la procédure de création est très simple et très rapide, il suffit de cliquer sur **Sign up** et de se laisser guider.



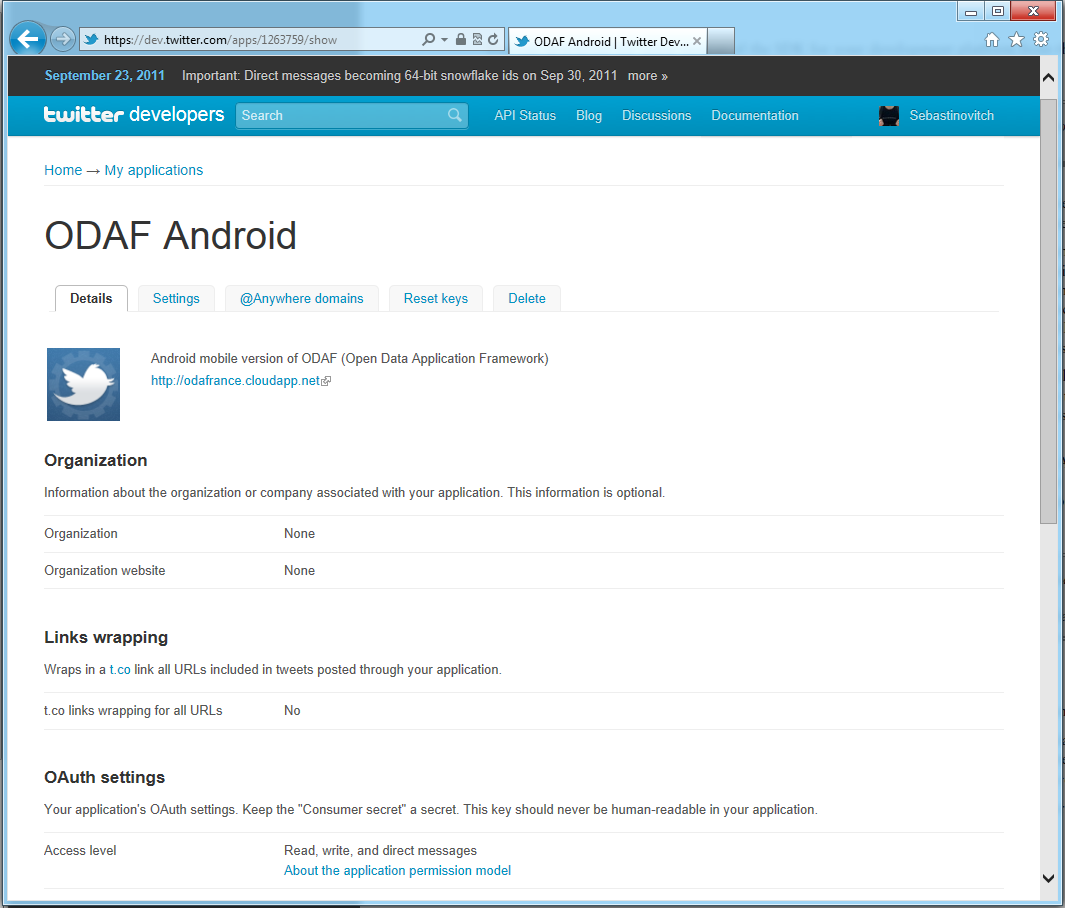
1. Une fois connecté, vous arrivez sur la page d’enregistrement de votre application mobile. Remplissez les champs habituels, à savoir **Name**, **Description** et **WebSite**. Mais cette fois-ci et contrairement à la version Silverlight Web du Framework ODAF Openturf, renseignez le champ **Callback URL** avec l’URL de votre site Web avec le suffixe */user/authorizereturn* ajouté à la suite ; par exemple <http://odafrance.cloudapp.net/user/authorizereturn>.



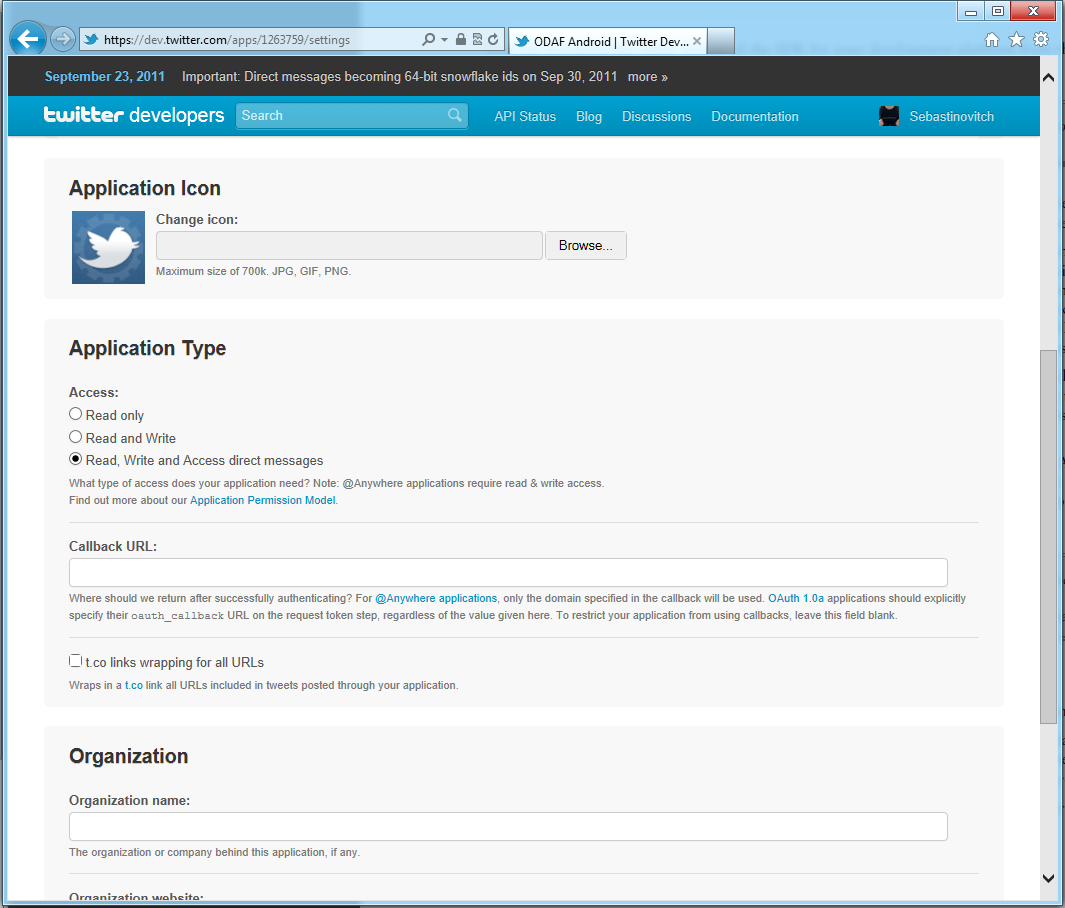
1. Défilez la fenêtre vers le bas, cochez **Yes, I agree**, saisissez le Captcha et cliquez sur **Create Your Twitter application**.



1. Votre application est maintenant créée. Conservez vos identifiants **Consumer key** et **Consumer secret**. Vous en aurez besoin pour la 2ème étape de la procédure d’enregistrement.



1. Sélectionnez l’onglet **Settings** pour changer votre changer votre paramètre **Application Type** à la valeur **Read, Write and Access direct messages**.



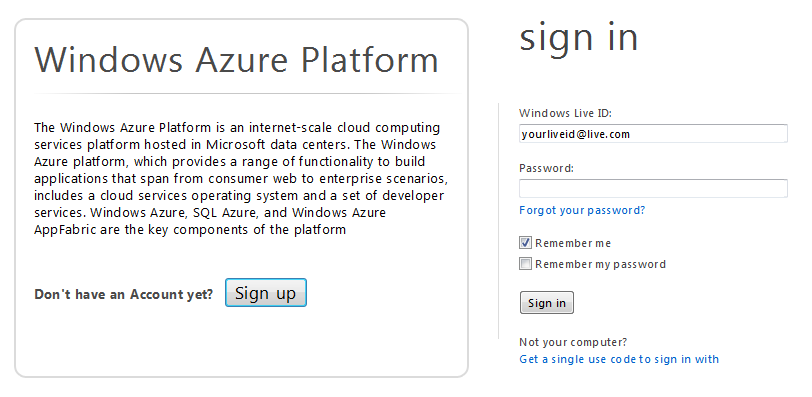
## Etape 2 – Mise à jour de la base SQL Azure ODAF Openturf

Comme décrit dans le document Déployer le kit de démarrage ODAF Openturf dans Windows Azure, ODAF Openturf dispose pour son « *Back-End* » d’une base de données SQL Azure qui contient notamment la table *OAuthClientApp* dont le rôle est de lister les applications pouvant consommer les services RESTful exposés par votre site web ODAF Openturf. Il est donc nécessaire de rajouter une entrée dans cette table pour enregistrer votre application Android.

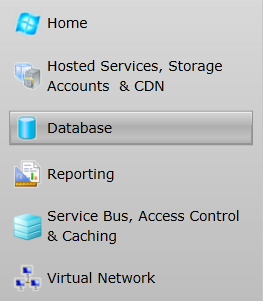
Procédez comme suit :



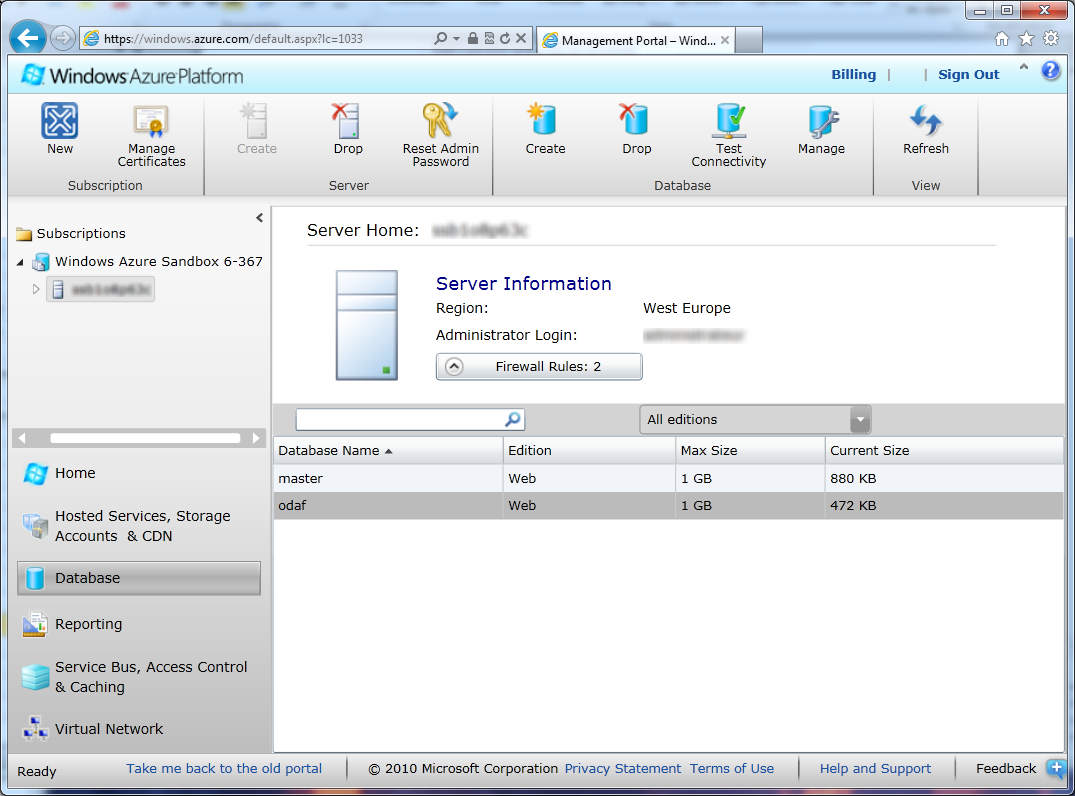
1. Naviguez à l’adresse <http://windows.azure.com/> et connectez-vous avec votre Windows Live ID associé à votre compte Windows Azure.



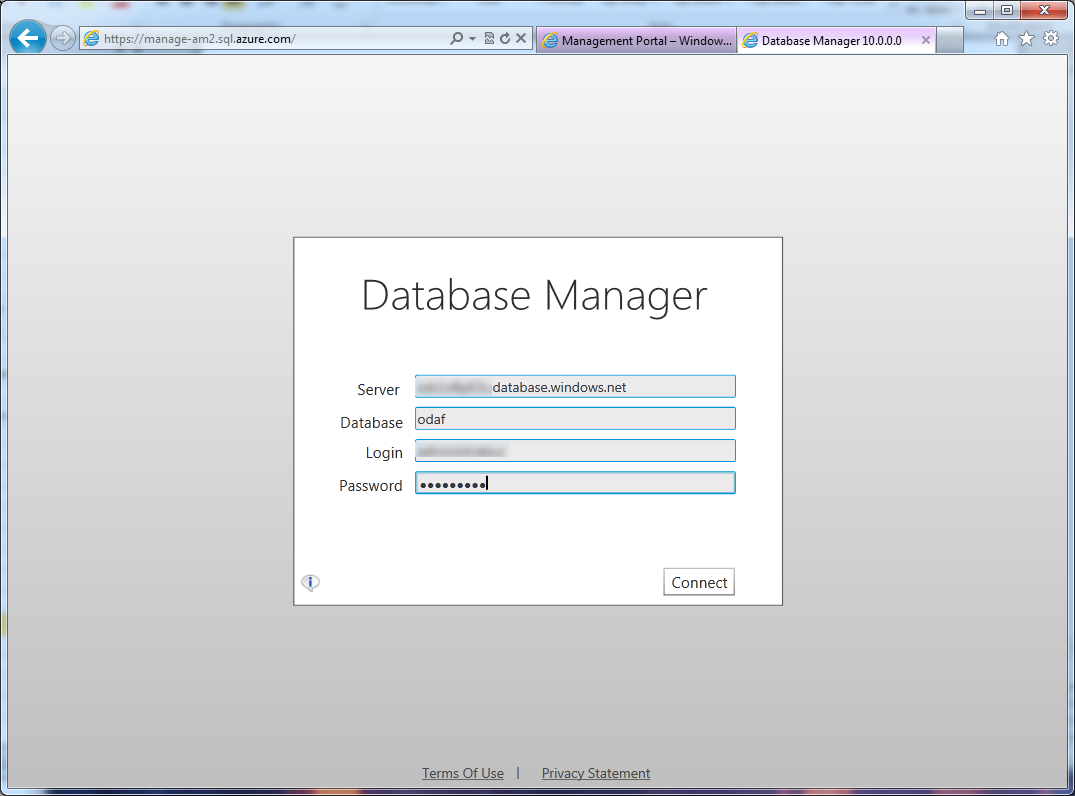
1. Cliquez sur l’onglet **Database** dans le panneau en bas à gauche.



1. Sélectionnez ensuite votre serveur de base de données dans le panneau de gauche, puis votre base de données ODAF Openturf que vous avez créé préalablement (Cf. section § Mise en place de la configuration du compte Windows Azure du document Déployer le kit de démarrage ODAF Openturf dans Windows Azure). Enfin, cliquez sur le bouton **Manage** dans le ruban Windows Azure.



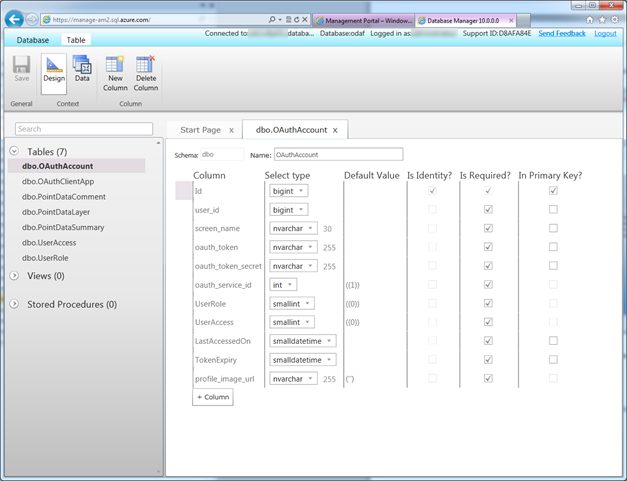
1. Une nouvelle page s’ouvre en vous demandant de vous loguer avec votre compte administrateur du serveur de base de données. Saisissez vos identifiants puis cliquez sur **Connect**.



1. Une fois connecté, il vous suffit tout simplement de rajouter une ligne dans la table *OAuthClientApp*. Pour cela, il vous faut un **Guid** qui sert d’identifiant pour votre application.

Les autres champs importants sont ***ConsumerKey*** et **ConsumerSecret**qui correspondent à votre clé et votre secret de votre compte applicatif Twitter précédemment créé. Vous pouvez en générer via le site <http://www.guidgenerator.com/> si besoin.

Pour le champ **CallbackUrl**, saisissez la valeur de votre URL de rappel que vous avez créé sur Facebook (URL se terminant par */user/authorizereturn*). A noter que toute cette procédure peut être aussi réalisée à travers Microsoft SQL Server Management Studio si vous préférez.



# Création d’un compte applicatif Google Maps



La technologie de cartographie utilisée par le Framework ODAF Openturf pour Android est Google Maps. Pour pouvoir utiliser une vue carte (*MapView*) dans votre application Android, vous devez premièrement obtenir une clé d’API depuis l’adresse suivante : <http://code.google.com/android/maps-api-signup.html>.

Sans cette clé d’API, le contrôle *MapView* ne téléchargera pas les tuiles qui composent la carte.

Pour obtenir une telle clé, vous devez spécifier l’empreinte MD5 du certificat utilisé pour signer vos applications Android.

Généralement, vous signerez votre application en utilisant deux certificats :

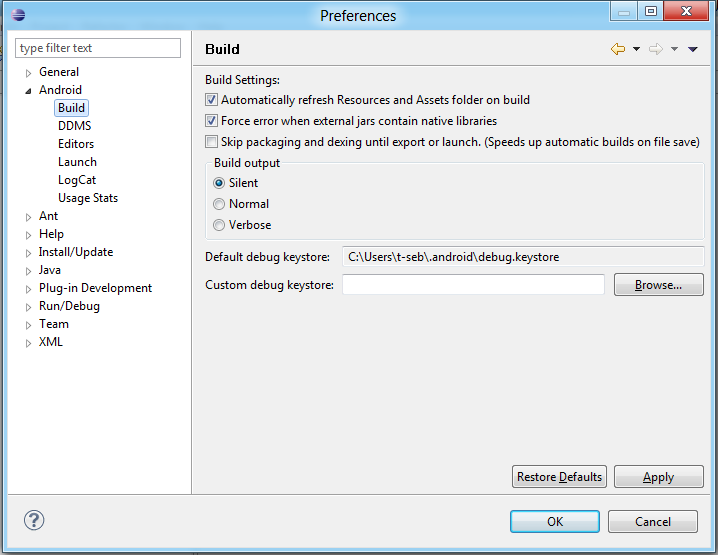
1. un certificat pour votre environnement de « développement et de débogage »,
2. puis un certificat de production.

Les sections suivantes précisent comment obtenir l’empreinte MD5 de chacun de ces certificats.

## Obtenir l’empreinte MD5 de votre environnement de développement/débogage

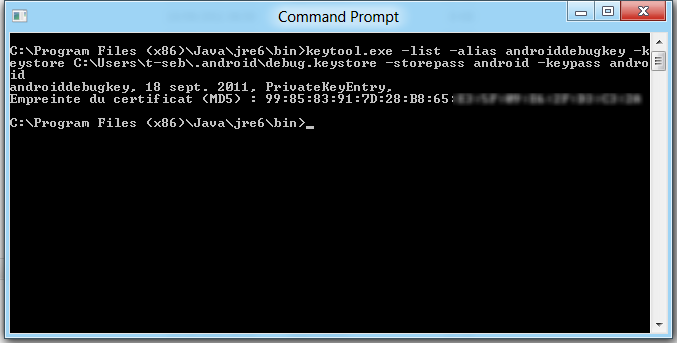
Si vous utilisez Eclipse avec le [plug-in ADT (Android Development Tools)](http://developer.android.com/sdk/eclipse-adt.html)[[8]](#footnote-8) pour développer vos applications Android, ces dernières seront signées avec le certificat de débogage par défaut. Pour voir les tuiles de la carte Google Maps pendant la phase de développement, vous aurez besoin d’obtenir la clé d’API Google Maps enregistrée via l’empreinte MD5 du certificat de débogage.

Vous pouvez trouver l’emplacement de votre magasin de certificats ou « keystore » via l’environnement de développement Eclipse en sélectionnant l’onglet **Windows**, puis **Preferences**, **Android**, **Build**. L’emplacement par défaut sera spécifié dans la zone de texte **Default debug keystore**.



Pour trouver l’empreinte MD5 de votre certificat de débogage, vous devez utiliser l’outil *keytool* en ligne de commande depuis votre installation Java comme suivant :

keytool -list -alias androiddebugkey -keystore <keystore\_location>.keystore -storepass android -keypass android



Assurez-vous de bien disposer de **la version 6 du JRE (Java Runtime Environment)**. En effet, si vous avez la version 7, vous obtiendrez une empreinte SHA1 au lieu de l’empreinte MD5, empreinte qui ne sera pas valide pour obtenir votre clé d’API.

## Obtenir l’empreinte MD5 de production

Juste avant la compilation et la signature de votre application pour la livraison, vous aurez besoin d’obtenir la clé d’API Google Maps utilisant l’empreinte MD5 de votre certificat de production. La ligne de commande est en fait la même mais paramétrée avec votre alias et votre magasin de clés.

keytool -list -alias my-android-alias -keystore my-android-keystore

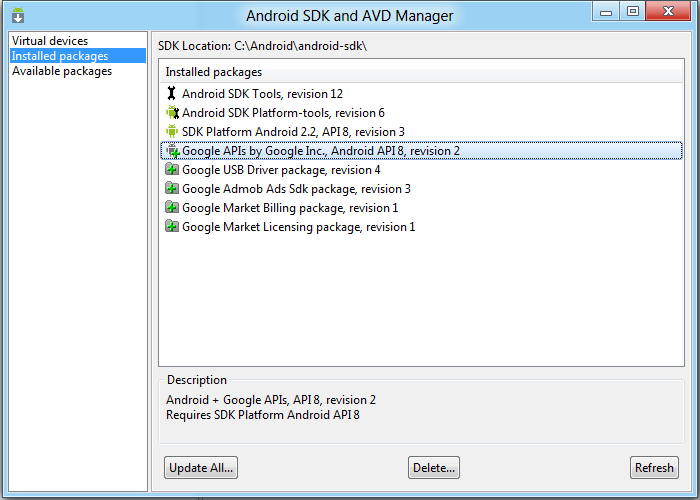
Vous trouverez plus d’informations sur la signature de vos applications Android sur la [page Signing Your Applications](http://developer.android.com/guide/publishing/app-signing.html)[[9]](#footnote-9) du guide de développement en ligne sur le site Web dédié aux développeurs Android.

## Installation de la bibliothèque Google Maps

La composante *ODAF.Android* du kit de démarrage ODAF Openturf requiert la bibliothèque *Google Maps* qui constitue un élément externe à l’environnement du SDK Android.

La bibliothèque est incluse dans les add-on Google API que vous pouvez installer depuis l’outil **Android SDK and AVD Manager**.

La procédure d’ajout de composant est décrite au niveau du guide de développement en ligne sur le site Web développeurs Android à la [page Adding SDK Components](http://developer.android.com/sdk/adding-components.html)[[10]](#footnote-10) du guide de développement en ligne. Vous aurez besoin d’installer la bibliothèque qui cible au moins la version 8 du SDK Android.



# Résolution des dépendances du projet ODAF.Android

Pour utiliser l’API Twitter, le projet *ODAF.Android* s’appuie, par ailleurs, sur la bibliothèque sous licence libre JTwitter.

Pour utiliser cette bibliothèque, deux archives Java vous seront nécessaires :

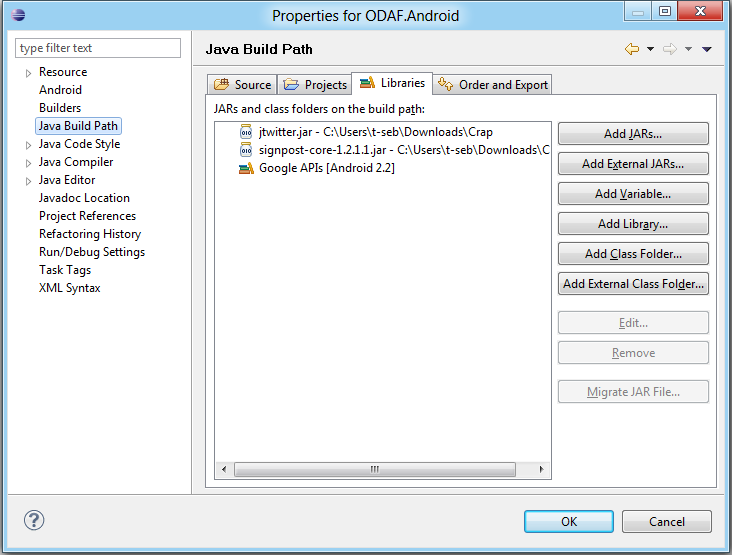
* jtwitter.rar : <http://www.winterwell.com/software/jtwitter/jtwitter.jar> .
* signpost.rar (Simple OAuth message signing for Java) : <http://oauth-signpost.googlecode.com/files/signpost-core-1.2.1.1.jar>.

Une fois accepté les licences associées et téléchargé ces archives java, vous devez les ajouter à votre projet.

Pour cela, procédez comme suit :



1. Ouvrez le projet *ODAF.Android* dans Eclipse.
2. Faites un clic-droit sur votre projet puis sélectionnez **Properties**.



1. Cliquez ensuite sur **Add External JARs…**  puis ajoutez les 2 bibliothèques susmentionnées.

# Modification du paramétrage du projet ODAF Openturf pour Android

Avant de rentrer dans l’exécution de l’application client mobile pour Android au niveau de l’émulateur local, nous rappelons ci-après brièvement les différents paramètres du projet comme précédemment abordé dans le document Démarrer avec le kit de démarrage ODAF Openturf.

Tous les paramètres sont localisés dans le sous-répertoire *res*de la solution mise à disposition.

Il s‘agit en substance des paramètres suivants :

| **Fichier** | **Paramètres** |
| --- | --- |
| ***res/values/strings.xml*** | * ***app\_name***: nom de l’application affiché sur l’écran d’accueil. * ***odafWebsitebaseUrl***: URL de base de votre serveur ODAF Openturf, n’oubliez pas le « / » à la fin. * ***odafAppId*** : GUID de votre application enregistré dans la base de données SQL Azure d’ODAF Openturf. * ***latitude***: latitude initiale du point central de l’écran d’accueil. * ***longitude*** : longitude initiale du point central de l’écran d’accueil. * ***zoomLevel***:niveau de zoom initial du point central de l’écran d’accueil. * ***consumerkey***: (votre) clé de consommateur Twitter. * ***consumerSecret***: (votre) secret de consommateur Twitter. * ***twitterCommentSent*** : message qui est affiché à l’écran après l’envoi d’un tweet. |
| ***res/drawable- [hdpi|ldpi|mdpi]/icon.png*** | Icône de l’application. |

# Exécution sur un émulateur local

Une fois tous vos comptes Twitter et Google Maps créés, vos dépendances résolues et vos paramètres configurés, vous pouvez exécuter le projet sur l’émulateur Android.

Pour cela, procédez comme suit :

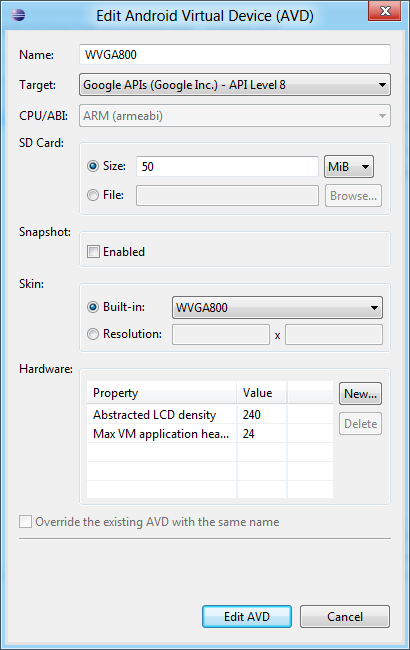


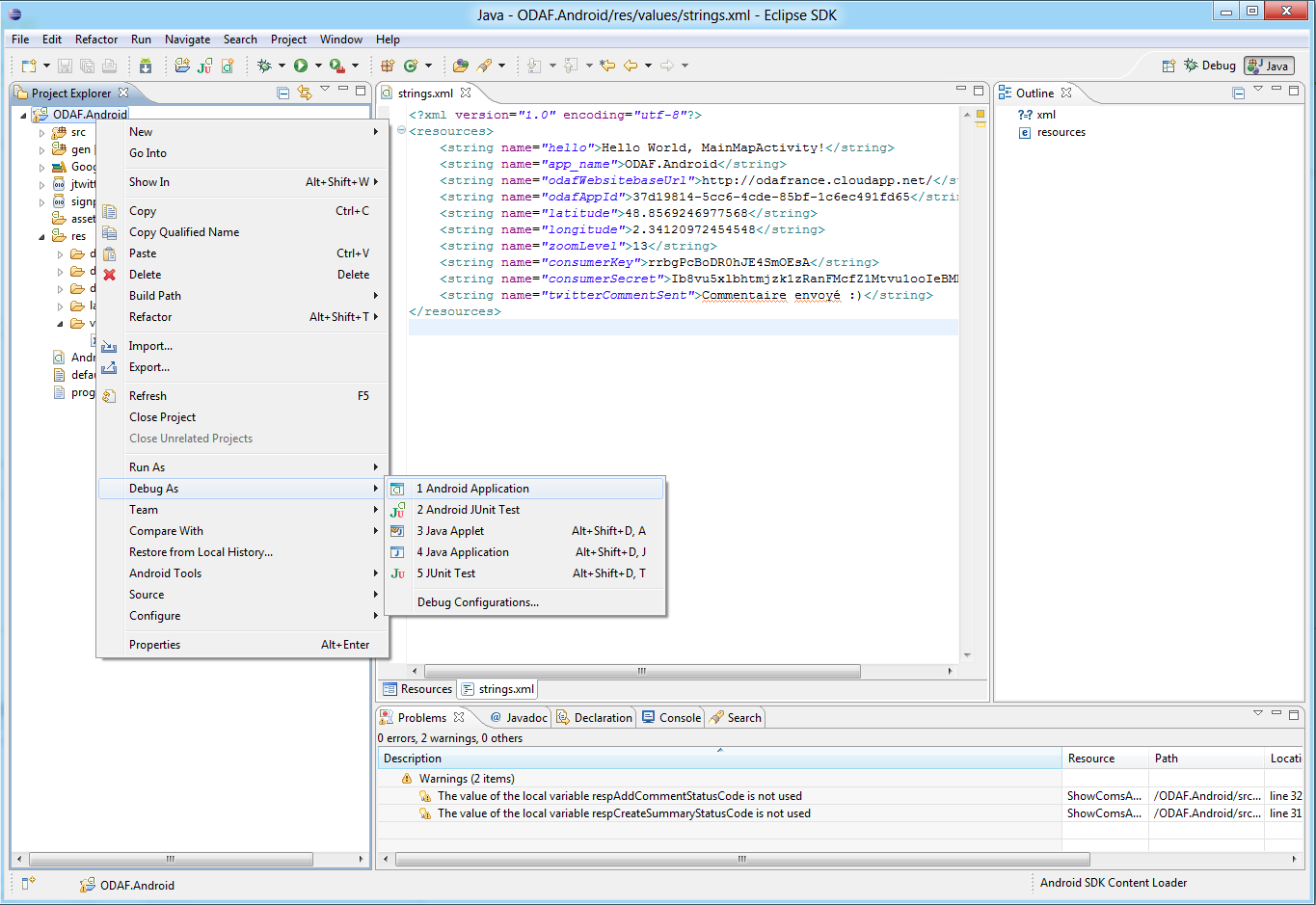
1. Faites un clic-droit sur votre projet et sélectionnez **Debug As** puis **Android Application** pour lancer l’émulateur Android.

Si vous n’avez pas de système Android virtuel préparé, vous allez devoir en créer un ciblant l’API Google Maps pour ajouter les références nécessaires vers les bibliothèques de l’API.

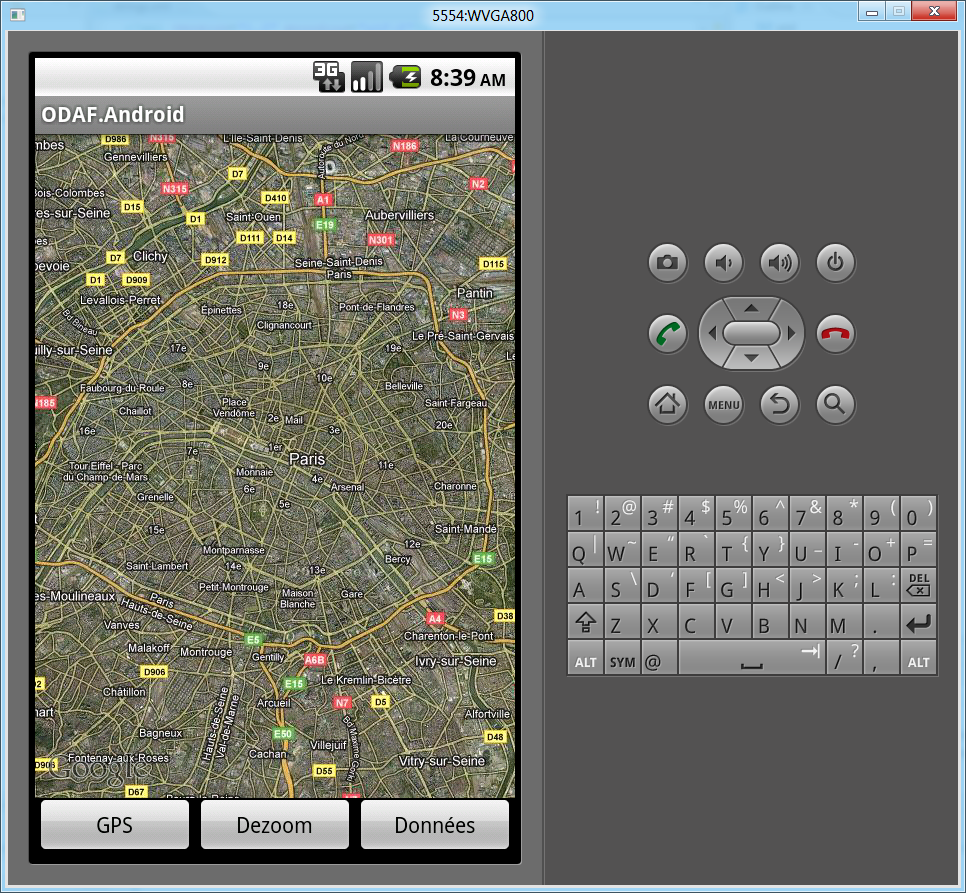
note_ddRemarque :

La procédure complète de création d’un système virtuel est décrite sur la [page Managing AVDs with AVD Manager](http://developer.android.com/guide/developing/devices/managing-avds.html)[[11]](#footnote-11) du guide de développement disponible sur le site Web à destination développeurs Android.

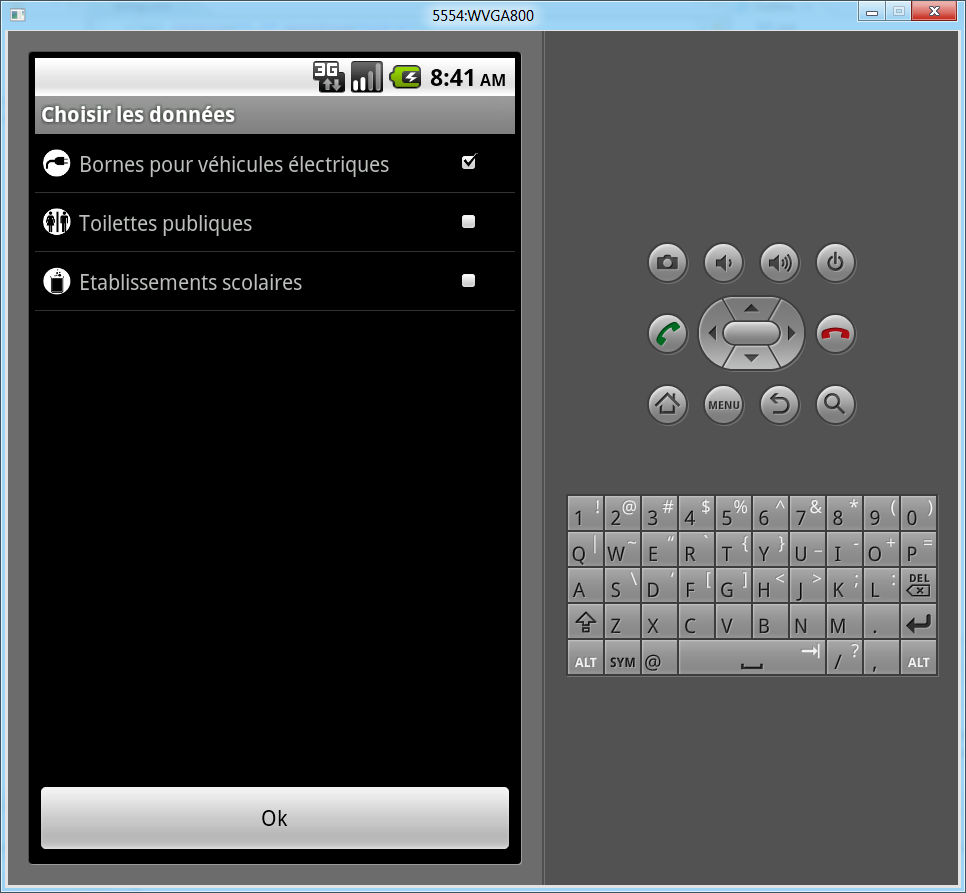


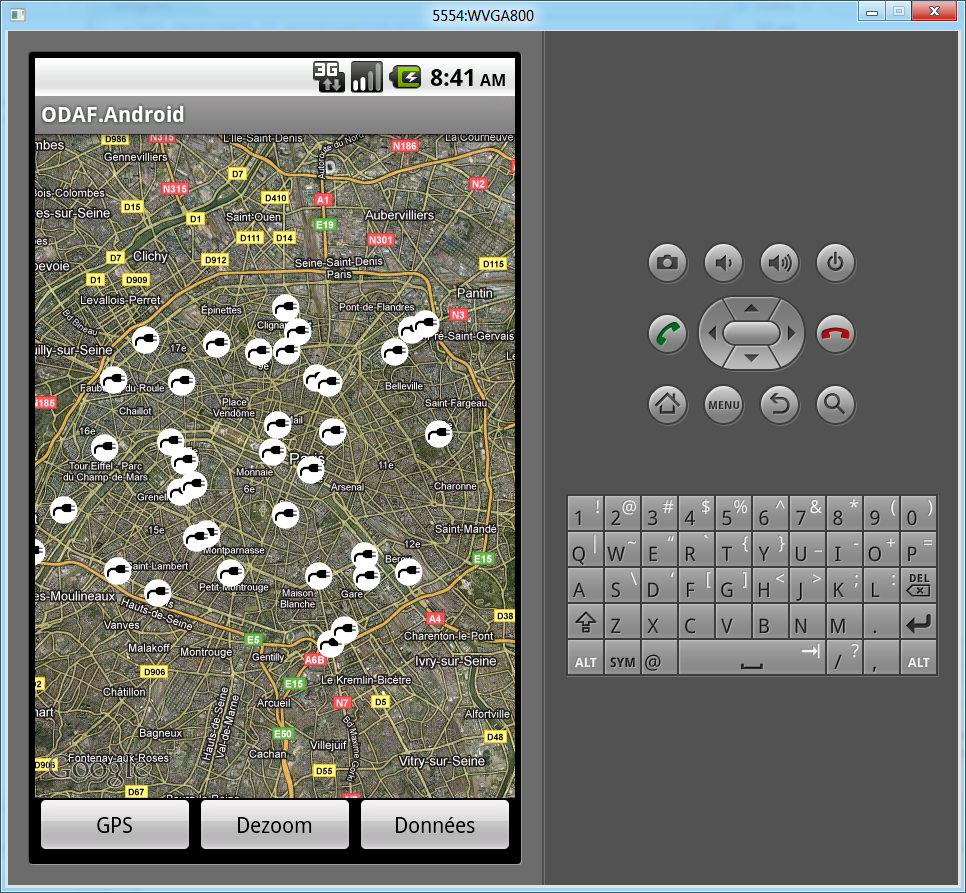


1. Au bout de quelques secondes, l’émulateur doit se lancer et l’application s’exécuter dessus. Vous devez alors arriver sur une carte vierge.

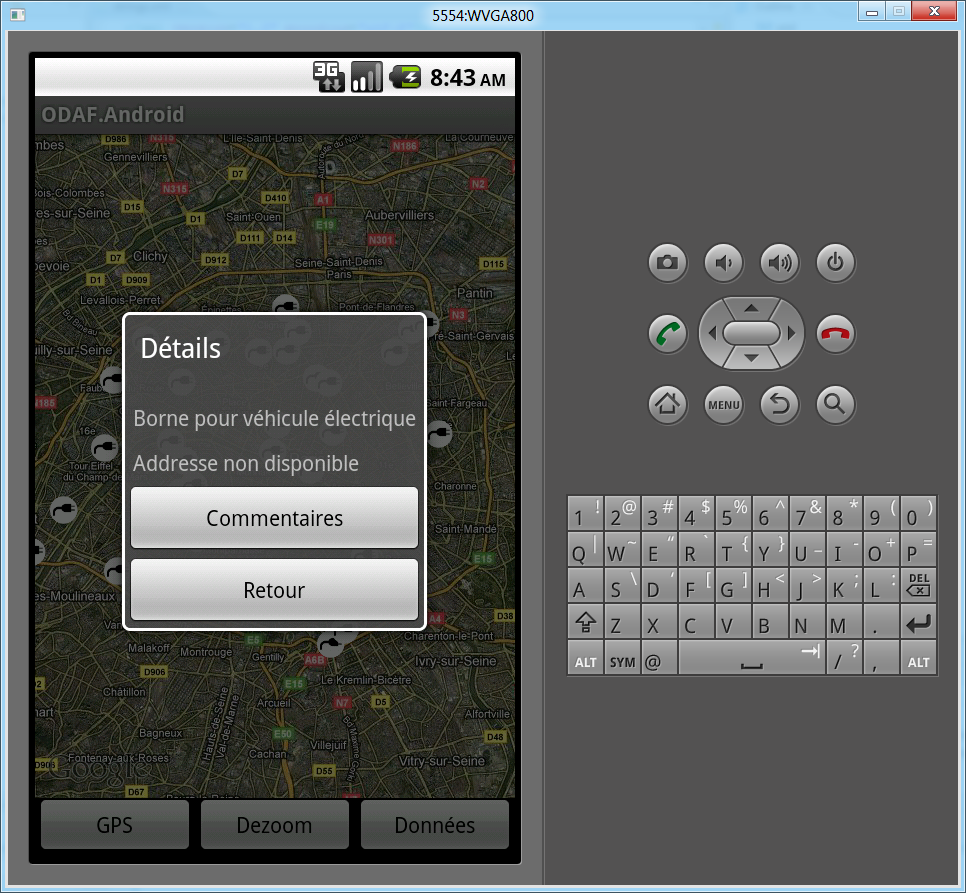


1. Si vous cliquez sur le bouton **Données** en bas à droite, vous arrivez sur l’écran de filtrage des données. Ainsi, vous pouvez choisir les sources de données que vous voulez afficher à l’écran. Cliquez sur le bouton **Ok**pour revenir sur la carte.

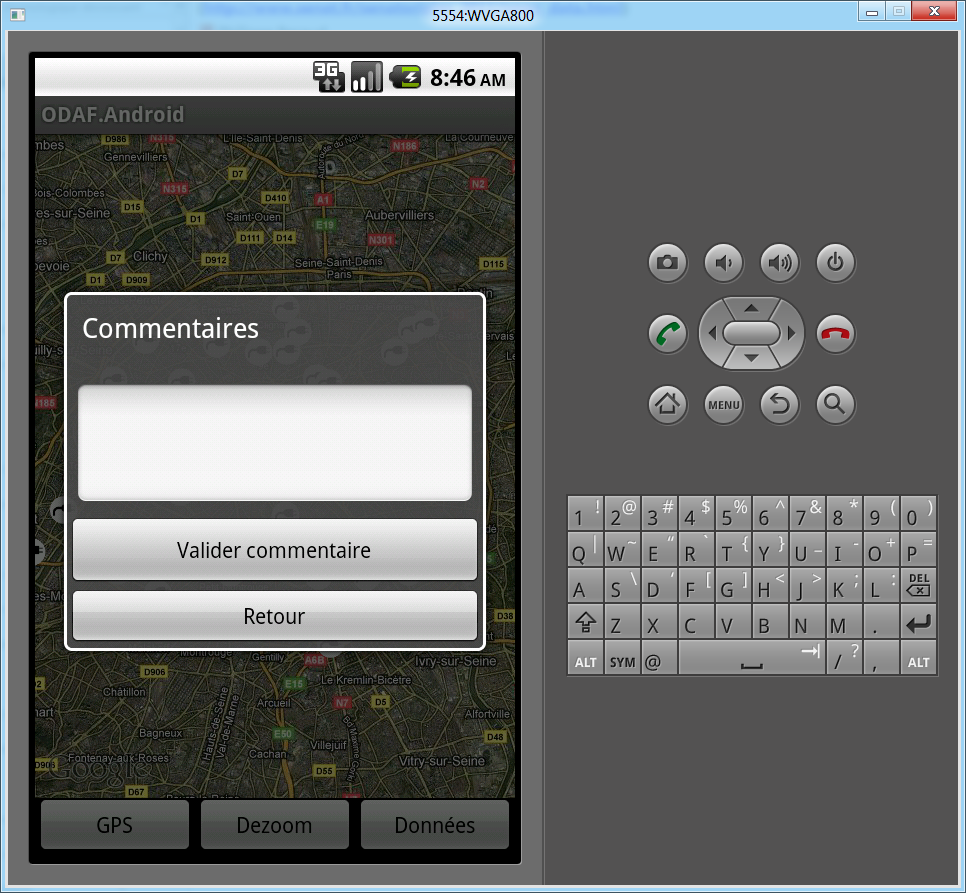




1. Si vous « cliquez » sur un point de donnée, vous avez alors une boite de dialogue qui s’ouvre et vous donne les différents détails du point de données.



1. Depuis cette page de détail, vous pouvez accéder à la liste des commentaires, puis poster un commentaire qui sera ensuite publié sur votre instance d’ODAF Openturf et sur Twitter.

A ce stade, le modèle d’application pour Android étant opérationnel, il ne reste plus qu’à l’adapter à vos besoins.

# Restons en contact

 Le groupe LinkedIn [Forum des Architectures Applicatives Microsoft](http://linkd.in/archiappms)[[12]](#footnote-12) regroupe des architectes en informatique qui ont des choix de technologies à faire dans les projets pour lesquels ils travaillent.

L’architecte applicatif, en situation de projet, travaille typiquement aux côtés de la direction de projet pour choisir et assumer des choix techniques en fonction des contraintes du projet (fonctionnalités, délais, ressources). Pour effectuer ces choix à bon escient, il doit connaître ce que le marché offre en termes de technologies. Cela peut prend typiquement deux formes : veille technologique continue, recherches dans le cadre du projet.

L’architecte applicatif a aussi pour rôle de faire le lien entre les équipes de développement et les équipes d’infrastructure et d’exploitation de la future application. Il doit également veiller à ce que ses choix soient bien mis en œuvre pendant le développement.

Ce forum, à l’initiative de Microsoft France, a pour but d’aider les architectes applicatifs dans la dynamique de ce Kit de démarrage, à :

* Faciliter la connaissance de l’offre de Microsoft pour les projets en entreprise (envoi de liens vers des présentations, documents ou livre blanc comme celui-ci, webcasts, conférences, etc.) ;
* Echanger sur des problématiques d’architecture ayant un rapport, même partiel, avec la plateforme.

Cet espace est le vôtre, faites le vivre, nous sommes aussi et surtout là pour vous lire.

1. Site portail Web Microsoft Secteur Public : http://www.microsoft.com/france/entreprises/secteur-public/solutions-pour-secteur-public.aspx [↑](#footnote-ref-1)
2. OGDI France : http://ogdifrance.cloudapp.net [↑](#footnote-ref-2)
3. Microsoft Windows Azure : http://www.microsoft.com/france/windows-azure/ [↑](#footnote-ref-3)
4. Blog MSDN OGDI France : http://blogs.msdn.com/b/ogdifrance/ [↑](#footnote-ref-4)
5. Site Web Android : http://www.android.com/ [↑](#footnote-ref-5)
6. Installing the SDK : http://developer.android.com/sdk/installing.html [↑](#footnote-ref-6)
7. Plateforme Android 2.2 : http://developer.android.com/sdk/android-2.2.html [↑](#footnote-ref-7)
8. Plug-in ADT pour l’environnement Eclipse : http://developer.android.com/sdk/eclipse-adt.html [↑](#footnote-ref-8)
9. Signing Your Applications : http://developer.android.com/guide/publishing/app-signing.html [↑](#footnote-ref-9)
10. Adding SDK Components : http://developer.android.com/sdk/adding-components.html [↑](#footnote-ref-10)
11. Managing AVDs with AVD Manager : http://developer.android.com/guide/developing/devices/managing-avds.html [↑](#footnote-ref-11)
12. Forum des Architectures Applicatives Microsoft: http://linkd.in/archiappms [↑](#footnote-ref-12)