Démarrer avec ODAF Openturf

Solution Open Data Clé en main

Publication : Septembre 2014

Version: 2.2

Auteur : Philippe Beraud

Contributeurs/relecteurs : Sébastien Brasseur, Rémi Olivier, Frédéric Romanczuk, Alexandre Fournier, Nazer Aminou, Ludovic Panchaud

Pour les dernières informations sur la solution communautaire Open Data Clé en main, consultez

aka.ms/OpenDataFrance

Sommaire

[Licence d’ODAF Openturf 4](#_Toc398918475)

[Résumé 5](#_Toc398918476)

[Commentaires 5](#_Toc398918477)

[Prérequis 6](#_Toc398918478)

[Connaissances et attendu 6](#_Toc398918479)

[Présentation rapide d'ODAF Openturf 7](#_Toc398918480)

[Contenu d’ODAF Openturf 8](#_Toc398918481)

[Dorsale d’intégration 8](#_Toc398918482)

[Site Web ASP.NET MVC 8](#_Toc398918483)

[Base de données d’intégration 8](#_Toc398918484)

[Modèles d’application 9](#_Toc398918485)

[Application modèle Web 9](#_Toc398918486)

[Application modèle mobile pour Android 10](#_Toc398918487)

[Application modèle mobile pour iPhone 10](#_Toc398918488)

[Application modèle mobile pour Windows Phone 10](#_Toc398918489)

[Configuration requise pour ODAF Openturf 12](#_Toc398918490)

[Configuration système 12](#_Toc398918491)

[Comment installer les composants système nécessaires 14](#_Toc398918492)

[Obtention des sources d’ODAF Openturf 15](#_Toc398918493)

[Organisation du code source du Kit de démarrage 16](#_Toc398918494)

[Dossier Database 17](#_Toc398918495)

[Scripts SQL 17](#_Toc398918496)

[Organisation projet *ODAF.Data* 17](#_Toc398918497)

[Eléments de configuration 18](#_Toc398918498)

[Dossier *ODAF.Azure* 18](#_Toc398918499)

[Dossier *ODAF.Android* 19](#_Toc398918500)

[Eléments de configuration 19](#_Toc398918501)

[Dossier *ODAF.iPhoneApp* 19](#_Toc398918502)

[Eléments de configuration 19](#_Toc398918503)

[Dossier *ODAF.Website* 20](#_Toc398918504)

[Organisation projet 20](#_Toc398918505)

[Eléments de configuration 21](#_Toc398918506)

[Dossier *ODAF.WindowsPhone* 22](#_Toc398918507)

[Organisation projet 22](#_Toc398918508)

[Eléments de configuration 22](#_Toc398918509)

[Dépendances 23](#_Toc398918510)

[Dépendances de la solution Visual Studio 2013 23](#_Toc398918511)

[Dépendances des projets relatifs aux applications modèles mobiles 25](#_Toc398918512)

[Compilation des éléments d’ODAF Openturf 26](#_Toc398918513)

[Compilation des éléments en ligne de commande 26](#_Toc398918514)

[Compilation des éléments en utilisant Visual Studio 2013 26](#_Toc398918515)

[Mise en œuvre des éléments d’ODAF Openturf 27](#_Toc398918516)

# Licence d’ODAF Openturf



**ODAF** (Open Data Application Framework) **Openturf**, l’un des accélérateurs de la [solution Open Data Clé en main](http://aka.ms/OpenDataCleEnMain)[[1]](#footnote-1), est publiée 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

**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)[[2]](#footnote-2). Il constitue l’un des accélérateurs de la solution Open Data Clé en main.

ODAF Openturf utilise les services de la plateforme de publication de données ouvertes **OGDI** (Open Government Data Initiative) **DataLab**, un autre accélérateur de la solution. A l’instar d’OGDI DataLab, ODAF Openturf est hébergé dans l’environnement d’exécution et d’hébergement [Microsoft Azure](http://azure.microsoft.com/fr-fr/)[[3]](#footnote-3) dans le cloud pour faciliter 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.) et leur affichage sur une carte.

Après une courte présentation de cet accélérateur, de ses fondations et architecture, ce document décrit le contenu à proprement parler du projet, l’organisation du code source ainsi mis à disposition et ses modalités de compilation. D’autres documents accompagnant la solution Open Data Clé en main décrivent les modalités de mise en œuvre de cet accélérateur et de mise à disposition de données notamment dans Azure.

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

# Commentaires

Votre opinion nous intéresse. N’hésitez pas à nous transmettre vos commentaires via la boîte aux lettres Open Data France (<mailto:ogdifrance@live.fr>) afin que nous puissions améliorer la solution communautaire Open Data Clé en main.

Le [blog MSDN Open Data France](http://blogs.msdn.com/b/ogdifrance/)[[4]](#footnote-4) vous tient informé dans le même temps des dernières évolutions de la solution Open Data Clé en main, des différents accélérateurs proposés dans ce contexte, de l’actualité et des nouveautés relatives à ceux-ci.

# Prérequis

Le Framework ODAF Openturf à destination de l’environnement d’exécution Azure nécessite un ordinateur sous Windows Server (R2) 2008 ou ultérieur.

Le lecteur doit connaître les technologies des services Web RESTful, le Framework ASP.NET MVC, le langage C# et .NET Framework afin de pouvoir facilement comprendre les éléments mis ainsi à disposition.

# Connaissances et attendu

La mise en œuvre d’ODAF Openturf requiert une expérience préalable :

* De l’environnement d’exécution et d’hébergement Microsoft Azure,
* De l’environnement de développement Microsoft Visual Studio 2013,
* De la programmation avec le Microsoft Framework .NET,
* Du développement et de l’utilisation d’interfaces Web de style REST (Representational State Transfer),
* Des dernières technologies Web comme ASP.NET MVC ou JQuery.

Remarque Microsoft met à disposition des formations gratuites téléchargeables sur ces environnements et technologies. Nous vous invitons à considérer le cas échéant le kit de formation [Azure Readiness Content](http://www.microsoft.com/en-us/download/details.aspx?id=8396)[[5]](#footnote-5) couvrant l’environnement Azure et l’environnement de développement Microsoft Visual Studio 2013. Ce dernier comprend des présentations, des vidéos de formation, des démonstrations accompagnées de leur script ainsi que des ateliers didactiques complets.

Remarque Le [blog du groupe Produit Microsoft Azure](http://blogs.msdn.com/b/windowsazure/)[[6]](#footnote-6) propose également de nombreuses ressources en termes de formation sur cet environnement d’exécution et d’hébergement. Celui-ci constitue également un excellent moyen de se tenir informé des évolutions proposées par l’environnement.

* Du développement avec les interfaces programmatiques (API) Twitter et Facebook pour l’intégration avec les réseaux sociaux,

Remarque L’intégration programmatique avec les principaux réseaux sociaux est notamment couverte dans l’ouvrage [Developer's Guide to Social Programming: Building Social Context Using Facebook, Google Friend Connect, and the Twitter API](http://www.amazon.fr/Developers-Guide-Social-Programming-Building/dp/0321680774/ref=sr_1_1?ie=UTF8&qid=1313055760&sr=8-1)[[7]](#footnote-7) aux éditions Addison-Wesley Professional.

Remarque Vous pouvez également consulter l’ouvrage l’ouvrage [Professional Twitter Development: With Examples in .NET 3.5](http://www.wrox.com/WileyCDA/WroxTitle/Professional-Twitter-Development-With-Examples-in-NET-3-5.productCd-0470531320.html)[[8]](#footnote-8) aux éditions Wrox dont certains exemples de code sont repris dans le Framework ODAF.

* Du développement en environnements mobiles : Android, iOS, et Windows (Phone).

# Présentation rapide d'ODAF Openturf

Les informations publiques, ou données ouvertes (open data), sont aujourd’hui en termes de potentiel, d’opportunités et d’enjeu au cœur du développement de multiples innovations sociales et économiques grâce à leur mise en ligne par l’administration et les collectivités territoriales.

Portées à la connaissance des citoyens, des chercheurs, des associations, des entreprises, ces informations statistiques, géographiques, urbanistiques, etc. peuvent devenir des mines d’or dès lors qu’elles sont retravaillées, croisées entre elles (mash-up) et mises en scène et ainsi contribuer à la création de nouveaux services tout en favorisant la transparence de l’action publique.

Dans un tel contexte, la publication et la consommation d’informations publiques sur le Web sous une forme facilement accessible par les personnes et les applications constituent une dimension essentielle. Afin justement de faciliter la mise à disposition et l’utilisation d’informations publiques, Microsoft France souhaite contribuer à cette dynamique en marche au travers d’ODAF Openturf (Open Data Application Framework), un Framework applicatif sous licence libre qui permet aux développeurs d’applications, à l’administration et aux collectivités territoriales d’étendre leur écosystème applicatif en servant de base pour exposer des données publiques et civiques et aux citoyens d’accéder à ces données, d’apporter leurs suggestions, de les noter et de les partager via leur réseau social comme Twitter ou Facebook.

Les fonctionnalités ainsi proposées au travers de ce Framework permettent à la fois aux citoyens de bénéficier d’un accès plus facile aux données publiques et ainsi de bénéficier d’une meilleure visibilité sur les services mis à disposition par l’administration et les collectivités territoriales ainsi que de la plus grande transparence de ces dernières.

ODAF Openturf correspond à un ensemble d’évolutions du kit de développement logiciel éponyme disponible en téléchargement sous la même licence libre sur la [forge CodePlex](http://ogdi.codeplex.com/)[[9]](#footnote-9). Il s’inspire également librement du projet [VanGuide for Windows Phone 7](http://vanguidewp7.codeplex.com/)[[10]](#footnote-10) également disponible sur la forge CodePlex.

# Contenu d’ODAF Openturf

ODAF Openturf s’articule autour d’une dorsale d’intégration pour le Framework et de modèles d’application constituant en tant que tel le Framework.

## Dorsale d’intégration

La Dorsale d’intégration repose sur :

1. Un Site Web ASP.NET MVC.
2. Une Base de données SQL.

### Site Web ASP.NET MVC

Le Site Web ASP.NET MVC de la Dorsale d’intégration expose une couche de services RESTful à destination des modèles d’application proposés par le Framework ou d’autres applications de façon à lire les flux de données KML (Keyhole Markup Language), et interagir avec les données utilisateurs stockées dans la base de données (tags, commentaires, notes, profils, etc.).

Pour cela, le Site Web ASP.NET MVC communique avec une instance OGDI DataLab pour ré-utiliser les catalogues de jeux de données publiques ainsi hébergés par cette instance et récupérer les flux de données KML. L’instance de test OGDI DataLab France déjà disponible à l’adresse <http://ogdifrance.cloudapp.net/> peut être utilisée à cet effet.

Le Framework ODAF Openturf se présente donc comme un client naturel d’OGDI DataLab qui permet d’exposer des flux de données en KML. OGDI DataLab rend possible la mise à disposition facilitée par l’Administration et les Collectivités Territoriales de services de données performants qu’elles exploitent, les informations publiques étant hébergées dans Azure. Les citoyens et les développeurs d’applications peuvent accéder aux données ainsi publiées depuis toute plateforme ou navigateur.

Le Site Web ASP.NET MVC suit comme son nom et sa technologie l’indiquent l’architecture Modèle-Vue-Contrôleur (MVC). L’architecture MVC constitue une façon d'organiser une interface graphique, mais également et surtout une façon d’architecturer un code source en respectant le principe de responsabilité unique, le principe de séparation des préoccupations et répondre aux problématiques de testabilité du code. Ce paradigme divise l’interface de l’application en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface.

Une telle architecture s’adapte très bien à la création de tels services RESTful. En effet, une URL permet d’accéder à une ressource, c’est-à-dire, dans le cas présent, un contrôleur qui va invoquer une action côté serveur et retourner en guise de vue des données au format JSON aux consommateurs du service, en l’occurrence ici les modèles suivants d’application.

Dans la pratique, la couche de services RESTful est accessible par une très large variété de technologies utilisées aujourd’hui sur Internet, telles qu’ASP.NET, JavaScript, Flash, Flex, PHP, Python, Ruby, Silverlight, Objective-C (iPhone iOS), Android, etc. Ceci revêt une importance certaine dans le contexte d’un Framework applicatif comme ODAF Openturf.

### Base de données d’intégration

ODAF Openturf utilise une base de données Microsoft SQL Server ou Microsoft SQL Azure afin de stocker les données nécessaires à son exécution, à savoir les commentaires, les données et les comptes des utilisateurs. Les scripts de création de la base de données, de réinitialisation des données et de suppression de la base de données sont fournis avec l’accélérateur.

L’approche retenue permet d’envisager la possibilité de portage sur une base de données MySQL ou SQLite.

## Modèles d’application

Le Framework ODAF Openturf propose un ensemble de modèles d’application de cartographie, modèles déclinés pour différents environnements. Ces modèles affichent sur une carte Bing Map ou Google Maps les données publiques géo-localisées, tout en proposant une intégration avec les réseaux sociaux.

Pour afficher ces données géo-localisées, les modèles d’application consomment des flux de données au format KML ou JSON. Les modèles d’application qui utilisent Bing Map s’interfacent en effet avec le Site Web ASP.NET MVC pour récupérer et consommer les données géo-localisées au format [JSON (JavaScript Object Notation)](http://tools.ietf.org/html/rfc4627)[[11]](#footnote-11). Le site Web modifie alors la requête envoyé au service RESTFul pour obtenir les données au format JSON.

### Application modèle Web

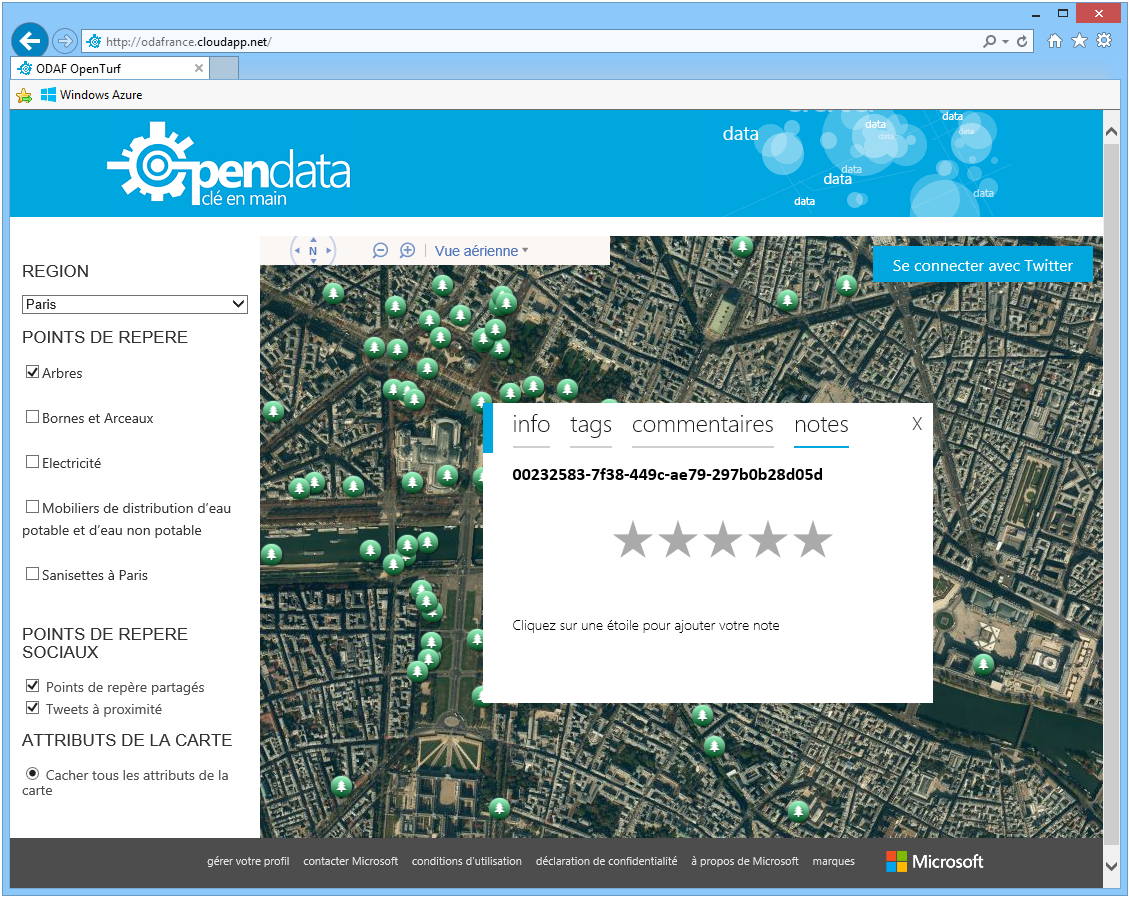
Le cœur de l’accélérateur est l’application modèle Web ASP.NET MVC en HTML5/CSS3/LESS centrée autour d’une carte Bing Cartes sur laquelle les points de données vont être affichés.

Tout comme le site Web permettant d’exposer la couche de services RESTful, l’application modèle Web suit elle aussi l’architecture Modèle-Vue-Contrôleur (MVC).

L’application modèle Web est composée d’une page publique pour les utilisateurs et d’une page d’administration, dont l’accès est limité, réservée aux administrateurs.

Chacune des pages est basée sur un Contrôleur qui dialogue avec le site Web ASP.NET MVC de la Dorsale d’intégration via les services RESTful précédents afin de lire les données des flux KML configurés et restituer les points correspondants sur la carte Bing Map au format JSON.

L’utilisateur a la possibilité de filtrer les sources de données (par région) qu’il souhaite afficher sur sa carte et de s’authentifier via son [compte Twitter](https://twitter.com/signup)[[12]](#footnote-12) dans le cas où il voudrait commenter, noter des données, voir même créer ses propres points de données.



Les administrateurs ont la possibilité d’administrer la solution ainsi que les informations altérées et/ou ajoutée par les utilisateurs en s’authentifiant via un compte associé au répertoire Azure Active Directory de la solution.

### Application modèle mobile pour Android

Le Framework ODAF Openturf propose également une application modèle mobile pour Android est également. Celle-ci dialogue également avec le site Web ASP.NET MVC de la Dorsale d’intégration via les services RESTful ainsi exposés pour consommer les données des flux de données KML.

L’application modèle utilise Google Maps comme technologie de cartographie et fournit la possibilité de se connecter à Twitter et de commenter les données.

### Application modèle mobile pour iPhone

Une application modèle mobile pour iPhone est également fournie. Celle-ci dialogue également avec le site Web ASP.NET MVC de la Dorsale d’intégration via les services RESTful ainsi exposés pour consommer les données des flux de données KML.

L’application utilise Bing Cartes comme technologie de cartographie et fournit la possibilité de se connecter à Twitter et de commenter les données.

### Application modèle mobile pour Windows Phone

Le Framework ODAF Openturf met enfin à disposition une application modèle mobile pour Windows Phone. A l’instar des applications modèle mobile pour Android et iPhone, cette application dialogue avec le site ASP.NET MVC de la Dorsale d’intégration pour consommer les données des flux KML.

Cette dernière utilise Bing Map comme technologie de cartographie et fournit la possibilité de se connecter à Twitter ou Facebook pour commenter un point de donnée. Cette version autorise également la prise de photos lorsque l’on se trouve à proximité d’un point de donnée au lieu de simples commentaires.

Ainsi, le contenu partagé est plus riche et plus concret aussi. Ces photos sont soient partagés dans son album Facebook, soit téléchargées sur TwitPic dans le cas où l’utilisateur choisit de partager via Twitter.

# Configuration requise pour ODAF Openturf

Cette section s’intéresse aux différents prérequis d’ODAF Openturf vis-à-vis de la mise en œuvre de son socle « back-end » et de l’application cliente Web riche.

Les prérequis spécifiques aux environnements mobiles Android, iPhone et Windows Phone sont abordés dans les documents Déployer ODAF Openturf pour Android, Déployer ODAF Openturf pour iPhone et Déployer le kit de démarrage ODAF Openturf pour Windows Phone qui accompagnent cet accélérateur. Vous pouvez vous reporter à ces trois documents pour une configuration complète de vos environnements de développement.

## Configuration système

L’accélérateur ODAF Openturf de la solution Open Data Clé en main a été conçu pour l’environnement Azure. Les prérequis exigés pour cet accélérateur sont donc ceux de cet environnement.

Remarque Ces prérequis avec les correctifs logiciels que cela suppose, peuvent bénéficier d’une installation automatisée de l’ensemble via le programme d’installation de la plateforme Web Microsoft 5.0. Celui-ci simplifie le téléchargement, l'installation et la tenue à jour des composants de la plate-forme Web Microsoft, notamment Internet Information Services (IIS) et ses extensions, SQL Server Express, .NET Framework, Visual Web Developer, etc. Il permet également d'installer des applications Web ASP.NET et PHP open source populaires.

Le programme d’installation de la plateforme Web Microsoft doit être installé préalablement depuis <http://www.microsoft.com/web/downloads/platform.aspx>.

En termes de systèmes d’exploitation, il s’agit en particulier de :

* Microsoft Windows 7 (32 bits ou 64 bits) ou Windows Server 2008 R2 (64 bits) avec le [correctif logiciel cumulatif KB977420 pour Windows Communication Foundation](http://support.microsoft.com/kb/977420)[[13]](#footnote-13).
* Microsoft Windows 8 (32 bits ou 64 bits) ou Windows Server 2012.

Les éléments suivants doivent être installés :

* Microsoft Internet Information Services (IIS) 7.0 ou ultérieur (avec ASP.NET, Activation WCF http, Contenu statique, Console de gestion IIS et éventuellement CGI),
* Microsoft .NET Framework 4.5,

Remarque Azure requiert Microsoft .NET Framework 3.5 (ou ultérieur).

* Microsoft Visual Studio 2013 (ou Microsoft Visual Studio 2013 Express).
* [Kit de développement logiciel Windows Azure 2.4 pour .NET (août 2014)](http://www.microsoft.com/en-us/download/details.aspx?id=43709)[[14]](#footnote-14).

Le Microsoft .NET Framework 4.5 est le modèle de programmation de code managé pour la plateforme Windows. Il combine la puissance du .NET Framework 2.0 avec de nouvelles techniques pour le développement d’applications qui communiquent très facilement au-delà des frontières technologiques, et qui sont capables de prendre en charge un large éventail de processus métiers. Parmi ces innovations se trouve notamment WCF (Windows Communication Foundation).

Auparavant nommé « Indigo », WCF constitue le cadre unifié Microsoft pour le développement d’applications distribuées sécurisées, fiables, transactionnelles et interopérables. Parmi beaucoup d’autres choses, WCF offre une implémentation complète des architectures de type REST au travers de WCF Data Services, technologie auparavant nommé « Astoria » puis ADO.NET Data Services. Il s’agit d’une implémentation du protocole OData par Microsoft.

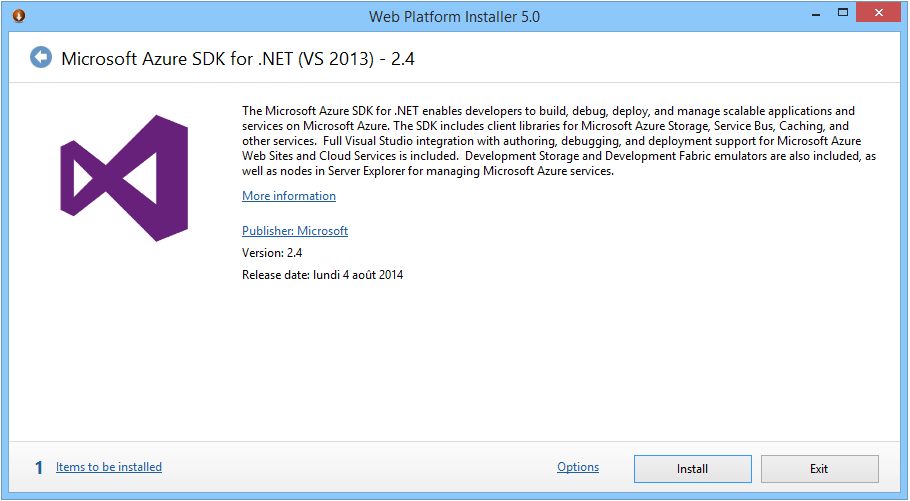
Pour des informations complémentaires, veuillez consulter le Centre de développement MSDN éponyme dédié à l’adresse <http://msdn.microsoft.com/fr-fr/netframework/aa663324.aspx>. Plus spécifiquement, vous pouvez consulter la page dédiée à WCF Data Services à l’adresse <http://msdn.microsoft.com/fr-fr/data/bb931106.aspx> pour la création et la consommation de services OData sur le Web.

ODAF Openturf repose sur la version 4.5 du Microsoft .NET Framework.

Le package redistribuable Microsoft .NET Framework 4.5 est disponible par téléchargement gratuit à l’adresse <http://www.microsoft.com/fr-fr/download/details.aspx?id=30653>.

Vous pouvez télécharger Microsoft Visual Studio Express 2013 for Web gratuitement sur <http://www.visualstudio.com/products/visual-studio-express-vs>.

Le Kit de développement logiciel (SDK) Azure 2.4 pour .NET (aôut 2014) est disponible en téléchargement à l’adresse <http://azure.microsoft.com/fr-fr/downloads/?fb=fr-fr>. cliquez sur le lien [Installation de VS 2013](http://go.microsoft.com/fwlink/p/?linkid=323510&clcid=0x40c) dans la rubrique .NET. Le programme d’installation de la plateforme Web Microsoft 5.0 prend alors en charge installation automatisée du kit de développement avec l’installation des prérequis et correctifs logiciels que cela suppose.



Remarque L’installation manuelle du Kit est disponible à partir de l’adresse <http://www.microsoft.com/en-us/download/details.aspx?id=43709>.

Le Kit de développement logiciel Azure comprend le package Azure Tools pour Microsoft Visual Studio 2013 qui étend Microsoft Visual Studio 2013 pour permettre la création, la configuration, la génération, le débogage, l'exécution, le packaging et le déploiement d'applications et de services Web évolutifs sur Azure.

Remarque Pour plus de détails, vous pouvez consulter la page [Azure pour les développeurs .NET](http://azure.microsoft.com/fr-fr/downloads/dotnet-sdk-24/?WT.mc_id=DOTNET_SDK_24)[[15]](#footnote-15). Cette page liste les paquets additionnels qui peuvent être installés à l’image de la [mise à jour 3](http://go.microsoft.com/fwlink/p/?LinkId=403040)[[16]](#footnote-16) pour Visual Studio 2013, les applets de commandes Azure PowerShell, etc.

## Comment installer les composants système nécessaires

Vous trouverez ci-dessous la procédure d’installation succincte pour la mise en place d’un environnement de développement et de test pour l’accélérateur ODAF Openturf sur une plateforme Windows 7 ou Windows Server 2008 R2 ou ultérieure :

1. Installer Internet Information Services (IIS) 8.0,
2. Installer le Microsoft Framework .NET 4.5,
3. Installer Microsoft Visual Studio 2013,
4. Enfin, utiliser le programme d’installation de la plateforme Web Microsoft 5.0 pour installer Kit de développement logiciel (SDK) Azure 2.4 pour .NET (aôut 2014), avec leurs prérequis et correctifs logiciels associés.

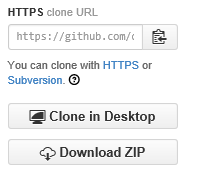
# Obtention des sources d’ODAF Openturf

L’accélérateur ODAF Openturf de la solution communautaire Open Data Clé en main est disponible sur la forge GitHub à l’adresse <https://github.com/openlab/ODAF->OpenTurf.

Les solutions communautaires évoluant avec l’investissement de la communauté, il est souhaitable de vous assurer que vous disposez de la dernière version, ne serait-ce que pour bénéficier, le cas échéant, de l’ensemble des correctifs.

[image](https://github.com/openlab/DataLab)

Pour télécharger les sources depuis la forge GitHub, le moyen le plus simple et rapide est de cliquer directement sur **Download ZIP**.



S’il constitue le moyen le plus simple, les autres options se révèlent plus adaptées pour mettre en place une gestion de versions.



A ce titre, nous vous invitons à consulter les billets [OGDI DataLab sur GitHub ! Impliquez-vous dans la communauté](http://blogs.msdn.com/b/ogdifrance/archive/2012/05/14/ogdi-datalab-sur-github-impliquez-vous-dans-la-communaut-233.aspx)[[17]](#footnote-17) et [GitHub for Windows](http://blogs.msdn.com/b/ogdifrance/archive/2012/06/14/github-for-windows.aspx)[[18]](#footnote-18), tous deux présents sur le blog MSDN [Open Data France](http://aka.ms/OpenDataFrance)[[19]](#footnote-19).

Par ailleurs, L’environnement Visual Studio 2013 propose une intégration directe avec GitHub.

Remarque La version précédente 2012 supposait d’installer avec les [outils Visual Studio pour Git](http://visualstudiogallery.msdn.microsoft.com/abafc7d6-dcaa-40f4-8a5e-d6724bdb980c)[[20]](#footnote-20). L’article [Integrating and Using Github in Visual Studio 2012](http://www.codeproject.com/Articles/581907/IntegratingplusandplusUsingplusGithubplusinplusVis)[[21]](#footnote-21) vous propose un guide pas-à-pas dans le contexte de cette version.

Quelle que soit l’approche retenue, et une fois, les sources téléchargées, nous vous invitons à passer à la section suivante.

# Organisation du code source du Kit de démarrage

L’intégralité du code source des sites Web, services et composants ODAF Openturf est située dans le sous-répertoire *Source* du répertoire d’extraction d’ODAF Openturf.

Compte tenu de ce qui précède, ce répertoire contient 8 sous-dossiers principaux :

1. Le dossier *Database* qui correspond à la couche de données de la solution ODAF Openturf. Il contient 2 sous-dossiers :
   1. Le sous-dossier *ODAF.Data* contient le projet éponyme qui est la couche d’accès aux données de la solution.

Remarque Ce projet utilise la bibliothèque [SubSonic](http://subsonicproject.com/)[[22]](#footnote-22), qui est un générateur de code basé sur les modèles (*templates*) [T4 (*Text Template Transformation Toolkit*)](http://www.olegsych.com/2007/12/text-template-transformation-toolkit/)[[23]](#footnote-23) pour l’interfaçage avec une base de données.

Les modèles T4 sont des fichiers qui mélangent des blocs de texte avec une logique de contrôle (en C# ou Visual Basic) afin de générer des fichiers de texte en sortie. Ils sont utilisés par SubSonic pour aller lire le schéma d’une base de données et générer les fichiers .cs qui contiennent les classes en C# pour se connecter et interagir avec celle-ci.

La bibliothèque SubSonic permet de s’interfacer avec des bases de données SQL Server, Oracle, MySql ou SQLite. Elle tire pleinement parti de la syntaxe [LINQ (Language-INtegrated Query)](http://msdn.microsoft.com/en-us/netframework/aa904594)[[24]](#footnote-24) et se fonde sur le patron de conception (*pattern*) « [Active Record](http://en.wikipedia.org/wiki/Active_record_pattern)[[25]](#footnote-25) ».

* 1. Le sous-dossier *scripts* contient quant à lui les scripts SQL de création de la base de données

1. Le dossier *ODAF.Azure* correspondant au projet de déploiement de la solution ODAF Openturf dans Azure, il y a un seul rôle Web qui correspond au projet *ODAF.Website*;
2. Le dossier *ODAF.Website* correspondant au projet de l’application modèle Web du Framework ODAF Openturf ;
3. Le dossier *ODAF.Android* correspondant au projet de l’application modèle mobile pour Android du Framework ODAF Openturf ;
4. Le dossier *ODAF.iPhoneApp* correspondant au projet de l’application modèle mobile pour iPhone du Framework ODAF Openturf ;
5. Le dossier *ODAF.WindowsPhone* correspondant au projet de l’application modèle mobile pour Windows Phone.

La suite de cette section décrit les projets/éléments de configuration constituants ces différents sous-dossiers de façon à mieux cerner la solution ainsi proposée par ODAF Openturf et ainsi vous l’approprier, nous l‘espérons, plus facilement en vue par exemple de sa personnalisation (légère ou avancée).

Tous les sites, services et composants afférents sont écrits en C#. Ils sont disponibles pour la version v4.0.30319 du Microsoft Framework .NET. Les fichiers de solution associés et présents dans ces dossiers sont destinés à l’environnement de développement Visual Studio (à l’exception des projets d’application modèle mobile pour Android, iPhone et Windows Phone) et sont regroupés dans la solution *ODAF\_Openturf.sln* (à l’exception des projets d’application modèle mobile pour Android, iPhone et Windows Phone).

## Dossier Database

Cette première composante de la solution correspond à la couche d’accès aux données du Framework ODAF Openturf. Comme évoqué précédemment, ce dossier est composé de 3 sous-dossiers correspondants aux scripts SQL de création de la base de données de la solution, à la bibliothèque *SubSonic* et au projet *ODAF.Data* qui contient le code d’accès à la base de données.

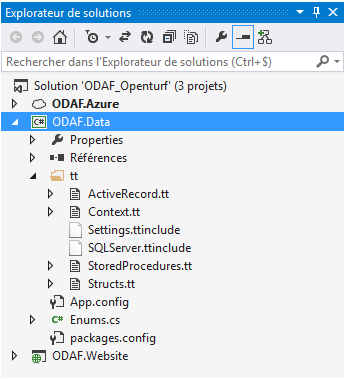
### Scripts SQL

Le dossier *scripts* contient les scripts SQL suivants :

| **fichier** | **Description** |
| --- | --- |
| odaf.sql | Ce script de création de la base de données permet de créer toutes les tables nécessaires et d’insérer des données par défaut (dans les tables *OAuthClientApp* et *PointDataLayer*) ainsi que des données qui n’ont pas vocation à changer à priori (tables *UserRole* et *UserAccess*) |
| odaf.wipe.tables.sql | Ce script permet d’effacer les données qui ont été partagées par les utilisateurs, à savoir les commentaires, les notes, les tags et les profils utilisateurs. |
| odaf.drop.tables.sql | Ce script supprime toutes les tables de la base de données. |

Le sous-dossier *tt* contient quant à lui les modèles (*templates*) T4 utilisés par le projet *ODAF.Data* pour générer le code de la couche d’accès aux données.

### Organisation projet *ODAF.Data*

* + - * 1. Le projet *ODAF.Data* est organisé de la façon suivante :
        2. 

| **Elément** | **Description** |
| --- | --- |
| Dossier tt | Ce dossier contient tous les modèles T4 qui vous permettent de générer le code d’accès à la base de données. Nous abordons plus spécifiquement la procédure de génération du code dans le document Déployer ODAF Openturf dans Azure. |
| App.config | Ce fichier de configuration du projet contient uniquement la chaine de connexion à la base de données de la solution ODAF Openturf. Il est nécessaire que cette chaine de connexion pointe vers une base de données valide afin de pouvoir générer le code d’accès à la base de données. |
| Enum.cs | Ce fichier constitue une énumération en C# correspondants aux rôles et aux permissions d’accès des utilisateurs. |

### Eléments de configuration

Le seul paramètre à configurer dans le projet *ODAF.Data* est la chaîne de connexion à la base de données qui est située dans le fichier *App.config*.

Remarque importante Tous les paramètres de la solution qui doivent être configurés sont précédés de la chaîne de caractères *CHANGEME.* Il devient alors très facile de rechercher toutes les occurrences de cette chaîne de caractères dans Visual Studio afin de compléter tous les paramètres.

| **Paramètre** | **Description** |
| --- | --- |
| ODAF | Chaine de connexion pointant vers la base de données de la solution ODAF Openturf.  Pour vous aider, il existe en ligne le site Web <http://www.connectionstrings.com/> qui vous propose les squelettes des différentes chaines de connexion ADO.NET. |

## Dossier *ODAF.Azure*

Ce projet standard Azure est configuré pour supporter le rôle *ODAF.Website* (Cf. ci-dessous). Ce projet contient la configuration de ce rôle unique.

Il inclut le fichier de définition de service *ServiceDefinition.csdef* qui contient les métadonnées nécessaires à Azure pour comprendre les exigences de l’application, tels que les deux rôles utilisés, leur niveau de confiance, les points de terminaison exposés par chaque rôle, les exigences de stockage local et les éventuels certificats utilisés par les rôles. La définition de service établit également les paramètres de configuration spécifiques à l'application.

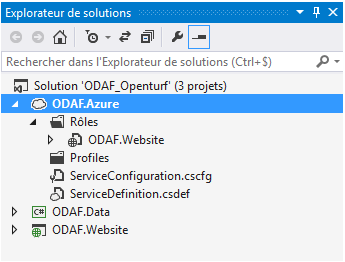
Le fichier de configuration de service *ServiceConfiguration.[Local|Cloud]cscfg* spécifie le nombre d'instances à exécuter pour le ou les rôles configurés dans le projet (ici un seul) et définit la valeur des paramètres de configuration définis dans le fichier de définition de service. Cette séparation entre la définition et la configuration de service vous permet de mettre à jour les paramètres d'une application en cours d'exécution en téléchargeant un nouveau fichier de configuration de service.

Le nœud **Rôles** dans le projet de service Cloud vous permet de configurer quels rôles le service inclut (web, travailleur (*worker*) ou les deux) ainsi que les projets à associer à ces rôles. L’ajout et la configuration des rôles au travers de ce nœud **Rôles** mettra à jour les fichiers *ServiceDefinition.csdef* et *ServiceConfiguration.[Local|Cloud]cscfg* précédents.

Ce projet sert directement pour publier dans Azure la solution complète ODAF Openturf.

Il n’y a pas de paramètres à configurer dans ce projet.

Ce projet est composé de la façon suivante :



## Dossier *ODAF.Android*

Ce dossier contient le projet de l’application modèle mobile pour Android. Il s’agit donc d’une application écrite en Java et reposant sur le kit de développement Android 12. Il y a différents paramètres situés dans différents fichiers à configurer dans ce projet.

### Eléments de configuration

| **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. |

## Dossier *ODAF.iPhoneApp*

Ce dossier contient le projet de l’application modèle mobile pour iPhone. Il s’agit donc d’une application écrite en Objective-C et reposant sur le kit de développement iPhone 3.x. Il y a différents paramètres situés dans différents fichiers à configurer dans ce projet.

### Eléments de configuration

| **Fichier** | **Paramètres** |
| --- | --- |
| Classes/Constants.h | * ***kTwitterAppConsumerKey***: (votre) clé de consommateur Twitter. * ***kTwitterAppConsumerSecret***: (votre) secret de consommateur Twitter. * ***BING\_MAPS\_DEV\_KEY***:(votre) clé de développeur Bing Cartes. |
| Config/Configuration.plist | * ***TintColourRGB***: couleur des barres d’outils et de navigation de l’application au format RRGGBB hexadécimal. * ***ButtonColourGradientHighRGB***: couleur du gradient haut des boutons de l’application au format RRGGBB hexadécimal. * ***ButtonColourGradientLowRGB***: couleur du gradient bas des boutons de l’application au format RRGGBB hexadécimal. * ***WebAppUrl***: lien vers votre application Web. * ***ClientAppId*** : valeur du champ Guid de votre application dans la table *OAuthClientApp*. * ***ReachableUrl***:nom de domaine (sans le schéma url) du serveur pour tester la disponibilité. * ***MapInitialLatitude***:latitude du point central de l’écran d’accueil. * ***MapInitialLongitude***: longitude du point central de l’écran d’accueil. * ***MapInitialZoom***: niveau de zoom initial du point central de l’écran d’accueil. * ***MapCurrentLocationlZoom***:niveau de zoom quand l’application zoom pour montrer votre localisation courante*.* * ***MapViewCommentZoom***:niveau de zoom quand l’application zoom pour vous montrer le marqueur d’un point de données dont vous regardez les commentaires. |
| ODAF.iPhoneApp-Info.plist | * ***Bundle Identifier***:remplacez « *CHANGEME* » par le nom de votre entreprise --  *com.CHANGEME.${PRODUCT\_NAME:rfc1034identifier}*. * ***URL types/URL identifier***: ajoutez votre identifiant unique sous forme de chaine de caractère, généralement de la forme « *com.entreprise.produit* ». * ***URL types/URL Schemes***:paramètre doit être égal à la valeur du paramètre *AppName* dans le fichier *Web.config* du projet *ODAF.Website.Mvc* mais tout en minuscules. |
| Default.png | Ecran de chargement. Il s’agit d’un fichier .png de dimension 320x480 pixels. |
| icon.png | Icône de l’application. Il s’agit d’un fichier .png de dimension 57x57 pixels. |

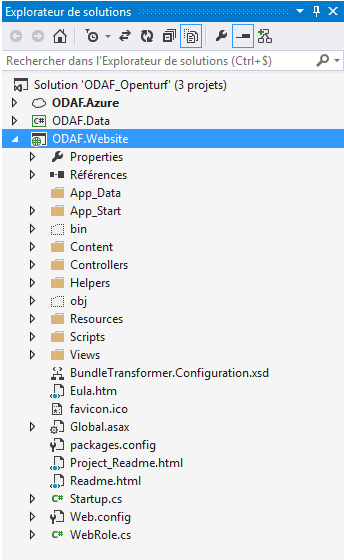
## Dossier *ODAF.Website*

Le dossier *ODAF.Website* correspond au projet *ODAF.Website* qui est l’application modèle Web ASP.NET MVC.

Le projet expose notamment un service permettant de consommer à la base des flux KML configurés dans le projet sous forme de données JSON pour les clients du service, un service pour publier des commentaires dans la base de données, un service pour poster des tweets sur Twitter, un service pour ajouter des notes, etc.

### Organisation projet

Le projet est organisé de la façon suivante :



| **Dossier** | **Description** |
| --- | --- |
| App\_Start | Ce dossier contient les fichiers de configuration de l’application Web. C’est notamment dans ces fichiers de config que des paramètres tel que le routage des URLs ou encore le type d’authentification peuvent être modifié. |
| Content | Ce dossier contient les fichiers de style CSS et LESS, les images, les polices d’écriture ainsi que les fichiers de script JavaScript, du site. |
| Controllers | Ce dossier contient les contrôleurs du site, l’un d’eux permet au utilisateur de tester l’application Web modèle du Framework ODAF Openturf. Un deuxième contrôleur permet aux administrateurs d’administrer les données de l’application Web. |
| Helpers | Ce dossier contient une classe contenant des champs constants représentants les noms les différents type d’authentifications qui peuvent être utilisés par l’application Web modèle. Cette classe est notamment utilisée dans les contrôleurs pour détecter le type d’authentification en cours. Elle est aussi utilisée dans le fichier *Startup.Auth.cs* du répertoire *App\_Start*, lors du paramétrage des types d’authentification à utiliser. |
| Resources | Ce dossier contient les fichiers ressources .resx de l’application Web modèle. Ces fichiers permettent de localiser l’application Web modèle pour proposer aux utilisateurs et aux administrateurs une interface disponible en plusieurs langues. Les langues actuellement supportées sont l’anglais comme langues par défaut et le français comme langue additionnelle. *Le fichier à utiliser est sélectionné en fonction de la langue définie dans le navigateur.* |
| Scripts | Ce dossier contient les scripts JavaScript utilisés par le site Web (JQuery et Ajax essentiellement). |
| Views | Ce dossier contient toutes les vues du site Web, notamment la vue */Home/Index* qui est la vue par défaut permettant aux utilisateurs de tester l’application Web modèle. La vue */Admin/Index* est la vue permettant aux administrateurs d’administrer les données. |

### Eléments de configuration

Ce projet contient différents éléments de configuration contenus pour la plupart dans le fichier de configuration *Web.config* à la racine du site Web.

| **Fichier** | **Paramètre** |
| --- | --- |
| Web.config | * **ODAF** (ligne 37) : chaine de connexion à la base de données de votre solution ODAF Openturf. * **TwitterConsumerKey** (ligne 47) : La clef d’application ‘ConsumerKey’ permettant d’utiliser l’API Twitter. * **TwitterConsumerSecret** (ligne 49) : La clef d’application ‘ConsumerSecret’ permettant d’utiliser l’API Twitter * **FederationMetadataAddress** (ligne 51) : L’adresse des metadata du repertoire Azure Active Directory utilisé pour l’authentification des administrateurs * **FederationWtrealm** (ligne 54) : L’url qui sera utilisé pour l’authentification des administrateurs via Azure Active Directory lorsque l’application Web modèle sera déployée. |
| App\_Start\Startup.Auth.cs | La fonction ConfigureAuth de cette classe permet d’activer/désactiver les diffèrent type d’authentification disponible, comme Azure Active Directory, Twitter, Facebook. |
| Resources | Les fichiers de ressources permettent de localiser l’application en différentes langues. Les traductions peuvent être ajoutées, modifiées ou supprimées. |

## Dossier *ODAF.WindowsPhone*

Le dossier *ODAF.WindowsPhone* correspond au projet d’application modèle mobile pour Windows Phone qui consomme (au même titre que l’application modèle Web, l’application modèle Android ou l’application modèle iPhone) les données exposées par la couche de services RESTful du site Web ODAF*.*

### Organisation projet

Le projet est organisé de la façon suivante :

| **Dossier** | **Description** |
| --- | --- |
| Commands | Le projet *ODAF.WindowsPhone* respectant l’architecture MVVM, le déclenchement d’un évènement par les interfaces utilisateurs est découplé de l’exécution de code métier associé par le biais de commandes. Une commande sert à faire le lien entre un évènement déclenché sur la vue et le code métier associé qui sera exécuté par le modèle logique de la vue. Le dossier *Commands* fournit la classe *DelegateCommand* qui est une implémentation de l’interface *ICommand* fournie par le Framework Silverlight. |
| Converters | Ce dossier contient des convertisseurs utilisés lors de liaisons de données entre le modèle logique de la vue et la vue. Ces convertisseurs servent par exemple à convertir un booléen en une valeur de l’énumération *Visibility* utilisée par les contrôles Silverlight. Ce qui encore une fois permet de découpler le modèle logique de la vue, de la technologie spécifique associé à la vue et augmente le potentiel de réutilisabilité du modèle logique de vue. |
| Helpers | Ce dossier sert à contenir des « helpers » divers et variés. Comme la classe *GeoMathHelper* par exemple. |
| Images | Ce dossier contient les icônes et images utilisées par le projet. |
| Models | Ce dossier contient les différents modèles de données représentant les données manipulées par l’application, par exemple les différents points sur la carte, les collections de points et les commentaires associés aux points. |
| Services | Ce dossier contient les différentes classes qui vont permettre de consommer les services utilisés par la solution, que ce soient les services exposés par le site ODAF Openturf notamment pour la gestion des commentaires ou bien le service Twitpic pour le chargement (*upload*) de photos depuis le smartphone. |
| ViewModels | Ce dossier contient les classes représentant les modèles logiques des vues associées. La logique applicative de ces classes manipule les modèles de données (dossier *Models*) et les restitue aux vues qui se chargeront de mettre en forme l’affichage. |
| Views | Ce dossier contient les vues de l’application (vue principale, vue Facebook et vue Twitter). Ces vues ne contiennent presque aucun code dans leur fichier « *code-behind* » associé, et pour cause, la logique applicative étant dans les classes contenant le modèle logique des vues (dossier *ViewModels*). A l’exception des contrôles de type « *ApplicationBar* » qui sont des contrôles un peu spécifiques, puisque ne dérivant pas de la classe *FrameworkElement* du Framework Silverlight. (et ne disposant donc pas de mécanisme de liaison de données par conséquent). |

### Eléments de configuration

Ce projet contient différents éléments de configuration contenus dans le fichier *App.xaml.cs* à la racine du projet.

| **Fichier** | **Paramètre** |
| --- | --- |
| App.xaml.cs | * ***AppId*** (ligne 23) : valeur du champ *Guid* de votre application dans la table *OAuthClientApp*. * ***BingMapsId*** (ligne 24) : (votre) clé de développeur Bing Cartes. * ***CenterLocation*** (ligne 25) : position initiale de votre carte. * ***FacebookAlbumName*** (ligne 26) : nom de l’album photo dans lequel seront uploadées les photos des utilisateurs lorsqu’ils les partageront sur Facebook. * ***FacebookAlbumDescription*** (ligne 27) : description de l’album photo dans lequel seront uploadées les photos des utilisateurs lorsqu’ils les partageront sur Facebook. * ***FacebookAppId*** (ligne 28) : identifiant de votre application sur votre compte développeur Facebook. * ***FacebookExtendedPermissions*** (ligne 29) : tableau contenant la liste des permissions d’accès demandées par votre application à l’utilisateur Facebook. * ***OdafWebsiteUrl*** (ligne 30) : URL de votre site Web ODAF Openturf. * ***OdafWebsiteUrl*** (ligne 30) : URL de votre site Web ODAF Openturf. * ***OdafWebsiteUrl*** (ligne 30) : URL de votre site Web ODAF Openturf. * ***TwitterConsumerKey*** (ligne 32) : (votre) clé de consommateur Twitter. * ***TwitterConsumerSecret*** (ligne 33) : (votre) secret de consommateur Twitter. |

## Dépendances

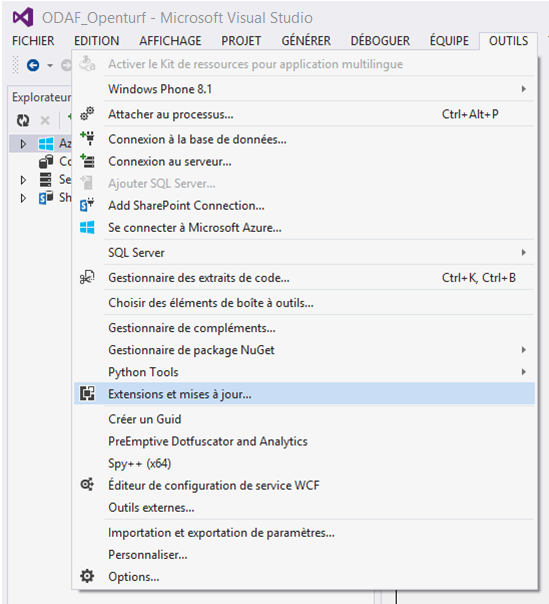
### Dépendances de la solution Visual Studio 2013

Au-delà des prérequis déjà mentionnés, les dépendances des projets présents dans la solution Visual Studio 2013 *ODAF\_Openturf.sln* sont résolues avec NuGet. NuGet est une extension qui vient s’ajouter à l’environnement Visual Studio 2013 et qui permet d’intégrer en quelques clics des bibliothèques et des outils sous licence libre à vos projets sous forme de paquets (package) installables.

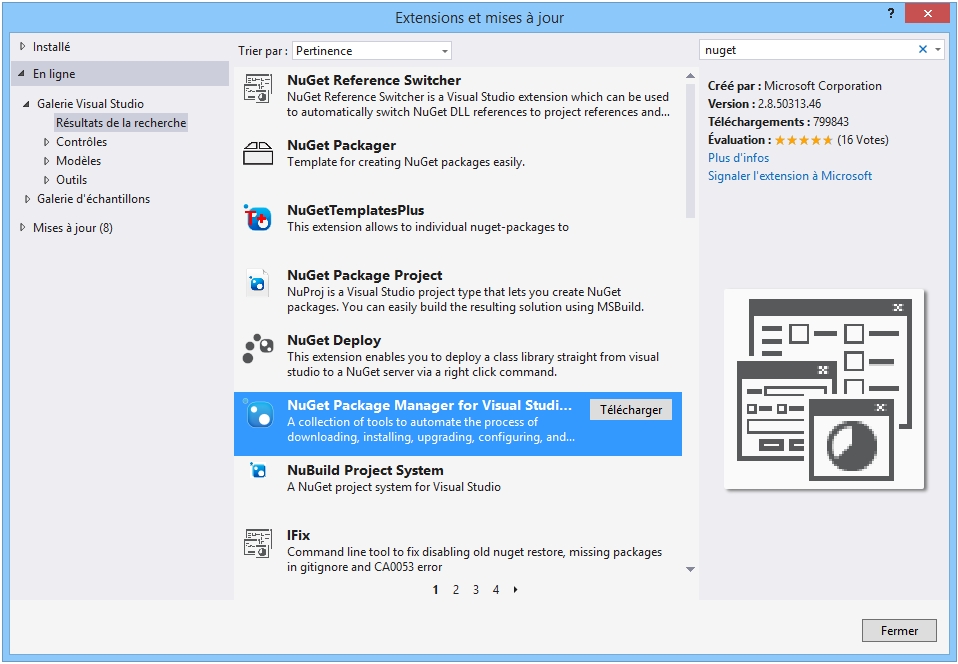
Les dépendances des projets de la solution sont disponibles sous forme de paquets depuis la [galerie en ligne](http://nuget.org/List/Packages)[[26]](#footnote-26) (online gallery) de NuGet.

Pour installer l’extension NuGet, procédez comme suit :

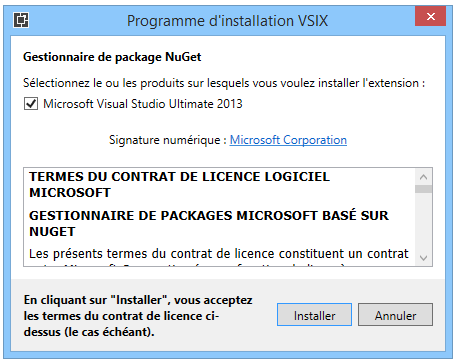
1. Lancez l’environnement de développement Visual Studio 2013.
2. Depuis la barre d’outils, cliquez sur **Outils**, puis **Extensions et mises à jour..**.



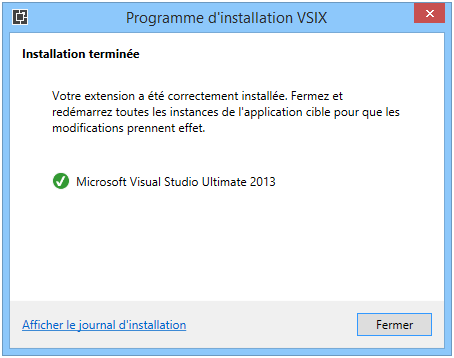
1. Une fois la fenêtre **Extensions et mises à jour**ouverte, sélectionnez l’onglet **En ligne**, puis, dans la barre de recherche, tapez simplement « *nuget* ».



1. Cliquez ensuite sur **NuGet Package Manager for Visual Studio 2013** puis sur le bouton **Télécharger**. Une fois le paquet télécharge, l’installation se lance alors.



1. Cliquer sur **Installer**. A l’issue de l’installation, un redémarrage de l’environnement est nécessaire.



### Dépendances des projets relatifs aux applications modèles mobiles

L’application modèle mobile pour Android référence la bibliothèque JTwitter. Cette dernière qui constitue un « wrapper » de l’API Twitter est disponible sous licence libre LGPL à l’adresse <http://www.winterwell.com/software/jtwitter.php>. Les deux archives Java  nécessaires sont :

* 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>.

Enfin, l’application modèle mobile pour Windows Phone référence les bibliothèques suivantes disponibles sous licence libre :

* La bibliothèque TweetSharp pour Windows Phone qui constitue un « wrapper » de l’API Twitter est disponible sous licence libre MIT à l’adresse <http://www.softpedia.com/dyn-postdownload.php?p=166939&t=0&i=1>. Le code source afférent est disponible à l’adresse <https://github.com/danielcrenna/tweetsharp>.

Cette bibliothèque utilise les modèles T4 de Visual Studio et référence :

* 1. La bibilothèque Hammock pour Windows Phone, une API client HTTP écrite en C# pour consommer des services RESful, et dont le code source est disponible à l’adresse <https://github.com/danielcrenna/hammock> ;
  2. La bibliothèque JSon.NET pour Windows Phone disponible sur la forge CodePlex à l’adresse <http://json.codeplex.com> ;
  3. La bibliothèque SharpZipLib pour Windows Phone dont le code source et l’assemblage sont disponibles à l’adresse <http://www.icsharpcode.net/OpenSource/SharpZipLib>.
* La bibliothèque Facebook C# SDK pour Windows Phone (*dossier sl3-wp*) disponible à l’adresse <http://facebooksdk.codeplex.com/> sur la forge CodePlex.

# Compilation des éléments d’ODAF Openturf

Les différentes composantes du Kit de démarrage ODAF Openturf (hormis les applications Android et iPhone) peuvent être compilées en utilisant Visual Studio ou la commande *msbuild* en ligne de commande. Les deux procédures sont décrites dans cette section.

## Compilation des éléments en ligne de commande

Pour compiler une solution particulière via la ligne de commande, veuillez suivre ces étapes :

1. Ouvrez l’invite de commande du SDK avec des privilèges administratifs et placez-vous dans le sous-répertoire *Source* du répertoire d’installation d’ODAF Openturf.
2. Saisissez *msbuild ODAF.Website.sln* sur la ligne de commande. Les éléments relatifs à la solution considérée sont compilés dans le répertoire */bin/Debug* relatif au répertoire de chaque projet composant la solution.

## Compilation des éléments en utilisant Visual Studio 2013

Pour compiler une solution particulière via Visual Studio 2013, veuillez suivre ces étapes :

1. Ouvrez Windows Explorer et placez-vous dans le sous-répertoire *Source* du répertoire d’installation d’ODAF Openturf.
2. Double-cliquez sur l’icône du fichier .*sln* situé dans l’un des dossiers du sous-répertoire *Source* pour ouvrir le fichier dans l’environnement Visual Studio.
3. Dans le menu **Build**, choisissez **Rebuild Solution**. Tous les éléments relatifs à la solution considérée sont compilés dans les répertoires */bin/Debug* relatifs aux répertoires de chaque projet composant la solution.

# Mise en œuvre des éléments d’ODAF Openturf

Le Kit de démarrage ODAF Openturf se présente sous la forme de composants logiciels qui permettent à toute entité publique :

1. De mettre à disposition de façon attrayante des données publiques sur une carte facilement consultable par les citoyens
2. De permettre au citoyen de jouer pleinement son rôle en apportant sa contribution et ses idées sur les données publiées par la collectivité territoriale dans le but d’améliorer la qualité des services délivrés.

La figure suivante illustre les éléments d’ODAF Openturf tels que décrits précédemment dans un contexte de mise en œuvre complète et opérationnelle dans Azure en consommant des flux de données KML exposés par une plateforme OGDI DataLab.



La plateforme OGDI DataLab est un autre accélérateur de la solution Open Data Clé en main sous licence libre disponible sur la [forge GitHub](https://github.org/openlab/OGDI-DataLab)[[27]](#footnote-27). Cette plateforme permet via ses composants d’exposer des données publiques sous différents formats standardisés (protocoles OData, KML, AtomPub, JSON et JSONP). Il convient de noter qu’une instance de démonstration est publiée à l’adresse <http://ogdifrance.cloudapp.net>.

ODAF Openturf constitue un client naturel d’OGDI DataLab et s’intègre donc particulièrement bien avec ce dernier. Comme le suggère la figure précédente, votre solution ODAF Openturf peut être déployée dans Azure sur le même compte que votre solution OGDI DataLab et il suffit de faire pointer celle-ci vers les flux de données KML exposés par la solution OGDI DataLab pour disposer rapidement d’une solution complètement fonctionnelle.

Le document Déployer ODAF Openturf dans Azure qui accompagne l’accélérateur ODAF Openturf décrit les modalités de mise en œuvre des composants ODAF Openturf dans Azure.

De façon similaire, les documents Déployer ODAF Openturf pour Android, Déployer ODAF Openturf pour iPhone et Déployer ODAF Openturf pour Windows Phone précisent les prérequis et les modalités de mise en œuvre des applications mobiles à destination des smartphones Android, iPhone et Windows Phone.

1. Solution Open Data clé en main : http://aka.ms/OpenDataCleEnMain [↑](#footnote-ref-1)
2. Site portail Web Microsoft Secteur Public : http://www.microsoft.com/france/entreprises/secteur-public/solutions-pour-secteur-public.aspx [↑](#footnote-ref-2)
3. Microsoft Azure : http://azure.microsoft.com/fr-fr/ [↑](#footnote-ref-3)
4. Blog MSDN Open Data France : http://aka.ms/OpenDataFrance/ [↑](#footnote-ref-4)
5. Azure Readiness Content : http://www.microsoft.com/en-us/download/details.aspx?id=8396 [↑](#footnote-ref-5)
6. Microsoft Azure Blog : http://azure.microsoft.com/blog/ [↑](#footnote-ref-6)
7. Developer's Guide to Social Programming: Building Social Context Using Facebook, Google Friend Connect, and the Twitter API : http://www.amazon.fr/Developers-Guide-Social-Programming-Building/dp/0321680774/ref=sr\_1\_1?ie=UTF8&qid=1313055760&sr=8-1 [↑](#footnote-ref-7)
8. Professional Twitter Development: With Examples in .NET 3.5 : http://www.wrox.com/WileyCDA/WroxTitle/Professional-Twitter-Development-With-Examples-in-NET-3-5.productCd-0470531320.html [↑](#footnote-ref-8)
9. ODAF : http://odaf.codeplex.com [↑](#footnote-ref-9)
10. VanGuide for Windows Phone 7 : http://vanguidewp7.codeplex.com/ [↑](#footnote-ref-10)
11. RFC 4627 The application/json Media Type for JavaScript Object Notation (JSON) : http://tools.ietf.org/html/rfc4627 [↑](#footnote-ref-11)
12. Compte Twitter : https://twitter.com/signup [↑](#footnote-ref-12)
13. Correctif logiciel cumulatif KB977420 pour Windows Communication Foundation : http://support.microsoft.com/kb/977420 [↑](#footnote-ref-13)
14. Kit de développement logiciel Windows Azure 2.4 pour .NET (août 2014) : http://www.microsoft.com/en-us/download/details.aspx?id=43709 [↑](#footnote-ref-14)
15. Azure pour les développeurs .NET : http://azure.microsoft.com/fr-fr/downloads/dotnet-sdk-24/?WT.mc\_id=DOTNET\_SDK\_24 [↑](#footnote-ref-15)
16. Mise à jour 3 pour Visual Studio 2013 : http://go.microsoft.com/fwlink/p/?LinkId=403040 [↑](#footnote-ref-16)
17. OGDI DataLab sur GitHub ! Impliquez-vous dans la communauté : http://blogs.msdn.com/b/ogdifrance/archive/2012/05/14/ogdi-datalab-sur-github-impliquez-vous-dans-la-communaut-233.aspx [↑](#footnote-ref-17)
18. GitHub for Windows: http://blogs.msdn.com/b/ogdifrance/archive/2012/06/14/github-for-windows.aspx [↑](#footnote-ref-18)
19. Blog MSDN Open Data France: http://aka.ms/OpenDataFrance/ [↑](#footnote-ref-19)
20. Visual Studio Tools for Git : http://visualstudiogallery.msdn.microsoft.com/abafc7d6-dcaa-40f4-8a5e-d6724bdb980c [↑](#footnote-ref-20)
21. Integrating and Using Github in Visual Studio 2012 : http://www.codeproject.com/Articles/581907/IntegratingplusandplusUsingplusGithubplusinplusVis [↑](#footnote-ref-21)
22. Projet SubSonic : http://subsonicproject.com/ [↑](#footnote-ref-22)
23. T4 Text Template Transformation Toolkit : http://www.olegsych.com/2007/12/text-template-transformation-toolkit/ [↑](#footnote-ref-23)
24. LINQ : http://msdn.microsoft.com/en-us/netframework/aa904594 [↑](#footnote-ref-24)
25. Patron de conception « Active Record » : http://en.wikipedia.org/wiki/Active\_record\_pattern [↑](#footnote-ref-25)
26. Galerie en ligne NuGet : http://nuget.org/List/Packages [↑](#footnote-ref-26)
27. OGDI DataLab : https://github.org/openlab/OGDI-DataLab [↑](#footnote-ref-27)