Démarrer avec le Kit de démarrage OGDI DataLab

Solution Open Data Clé en main

Publication : Juin 2013

Version: 6.0

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

Pour les dernières informations, consultez

aka.ms/OpenDataFrance

Sommaire

[Licence du Kit de démarrage OGDI DataLab 4](#_Toc358204877)

[Résumé 5](#_Toc358204878)

[Commentaires 6](#_Toc358204879)

[Prérequis 7](#_Toc358204880)

[Connaissances et attendu 7](#_Toc358204881)

[Présentation rapide du Kit de démarrage OGDI DataLab 8](#_Toc358204882)

[Présentation rapide du protocole OData 12](#_Toc358204883)

[Présentation rapide de Windows Azure 15](#_Toc358204884)

[Configuration requise pour le Kit de démarrage OGDI DataLab 18](#_Toc358204885)

[Configuration système 18](#_Toc358204886)

[Comment installer les composants système nécessaires 20](#_Toc358204887)

[Obtention des sources du Kit de démarrage OGDI DataLab 21](#_Toc358204888)

[Contenu du Kit de démarrage OGDI DataLab 22](#_Toc358204889)

[Kit de développement logiciel interactif : DataBrowser 22](#_Toc358204890)

[Service de données 23](#_Toc358204891)

[Utilitaires de chargement de données 25](#_Toc358204892)

[Organisation du code source du Kit de démarrage 26](#_Toc358204893)

[Dossier DataBrowser (Kit de développement logiciel interactif) 26](#_Toc358204894)

[Organisation projet 27](#_Toc358204895)

[Eléments de configuration 29](#_Toc358204896)

[A propos des vues proposées et de la localisation 33](#_Toc358204897)

[Dossier DataService (Service de données) 36](#_Toc358204898)

[Organisation projet 37](#_Toc358204899)

[Eléments de configuration 37](#_Toc358204900)

[Dossier DataLoader (Utilitaires de chargement de données) 38](#_Toc358204901)

[Organisation projet 38](#_Toc358204902)

[Eléments de configuration 38](#_Toc358204903)

[Dossier ConfigTool 39](#_Toc358204904)

[Dossier DAISY 40](#_Toc358204905)

[Dossier Shared 41](#_Toc358204906)

[Organisation projet 41](#_Toc358204907)

[Dépendances 42](#_Toc358204908)

[Résolution des dépendances avec NuGet 42](#_Toc358204909)

[Compilation des éléments du Kit de démarrage OGDI DataLab 47](#_Toc358204910)

[Compilation des éléments en ligne de commande 47](#_Toc358204911)

[Compilation des éléments en utilisant Visual Studio 2012 47](#_Toc358204912)

[Compilation des éléments via le script d’automatisation 47](#_Toc358204913)

[Mise en œuvre des éléments du Kit de démarrage OGDI DataLab dans Windows Azure 48](#_Toc358204914)

[Pour aller plus loin 51](#_Toc358204915)

# Licence du Kit de démarrage OGDI DataLab



Le Kit de démarrage **OGDI** (Open Government Data Initiative) **DataLab** 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 **OGDI** (Open Government Data Initiative) **DataLab** sous licence 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). Il constitue l’un des accélérateurs de la [solution Open Data Clé en main](https://mspartner.microsoft.com/fr/fr/Pages/Solutions/Open-Data-Cle-en-main.aspx)[[2]](#footnote-2) disponible sur le site Microsoft Partner Network.

OGDI DataLab utilise la [plateforme Windows Azure](http://www.microsoft.com/france/windows-azure/)[[3]](#footnote-3) dans le Cloud pour faciliter la publication et l'utilisation d’informations publiques ou données ouvertes (*open data*) variées provenant d'entités publiques (gouvernement, ministères, collectivités locales et territoriales, mairies, services déconcentrés de l'Etat, etc.).

Pour se faire, la plateforme de publication des données ouvertes OGDI DataLab propose une gestion multi-catalogues de (jeux de) données ouvertes et un service de données RESTful [OData (*Open Data Protocol*)](http://www.odata.org/)[[4]](#footnote-4) pour la consommation des données ainsi mises à disposition. Cette solution communautaire disponible sur la [forge GitHub](https://github.com/openlab/DataLab)[[5]](#footnote-5) s’adapte très facilement à la stratégie de l’acteur public notamment en termes de portail usager.

Après une courte présentation du kit de démarrage, de ses fondations et de son architecture, ce document décrit le contenu à proprement parler du kit, l’organisation du code source ainsi mis à disposition et ses modalités de compilation. D’autres documents accompagnant le kit décrivent les modalités de mise en œuvre du kit et de mise à disposition de données dans Windows Azure.

Ce document s’adresse aux architectes, aux développeurs et à tous ceux qui souhaitent exposer et/ou consommer des informations publiques au sens du protocole OData. Les site Web, services, et utilitaires et le code source afférent fournis dans le Kit de démarrage OGDI DataLab sont conçus pour interagir avec une très large variété de technologies utilisées aujourd’hui sur Internet, telles que ASP.NET, JavaScript, Flash, Flex, PHP, Python, Ruby, Silverlight, etc., ou encore les sites de cartographie comme Géoportail, le portail des territoires et des citoyens, Bing Maps, Google Maps, etc.

# 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 le kit de démarrage OGDI DataLab, l’un des accélérateurs de la [solution Open Data Clé en main](https://mspartner.microsoft.com/fr/fr/Pages/Solutions/Open-Data-Cle-en-main.aspx)[[6]](#footnote-6) disponible sur le site Microsoft Partner Network.

Le [blog MSDN Open Data France](http://blogs.msdn.com/b/ogdifrance/)[[7]](#footnote-7) 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

Ce Kit de démarrage OGDI DataLab à destination de l’environnement d’exécution Windows Azure nécessite un ordinateur sous Windows Server 2003 (R2) SP1, Windows 7, Windows Server (R2) 2008 ou ultérieur.

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

# Connaissances et attendu

La mise en œuvre du Kit de démarrage OGDI DataLab requiert une expérience préalable :

* De la plateforme Microsoft Windows Azure,
* De l’environnement de développement Microsoft Visual Studio 2012,
* 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 encore jQuery.

Microsoft met à disposition des formations gratuites téléchargeables sur ces environnements et technologies. Nous vous invitons à considérer le cas échéant les kits de formation suivants Microsoft Web Camps Training Kit

Remarque : Microsoft met à disposition des formations gratuites téléchargeables sur ces environnements et technologies. Nous vous invitons à considérer le cas échéant les kits de formation suivants qui comprennent des présentations, des vidéos de formation, des démonstrations accompagnées de leur script ainsi que des ateliers didactiques complets :

* [Microsoft Windows Azure Training Kit (mise à jour de mai 2013)](http://www.microsoft.com/downloads/en/details.aspx?FamilyID=413E88F8-5966-4A83-B309-53B7B77EDF78&displaylang=en)[[8]](#footnote-8) couvrant la plateforme Windows Azure et l’environnement de développement Microsoft Visual Studio 2012.
* [Microsoft Web Camps Training Kit](http://trainingkit.webcamps.ms/Default.htm)[[9]](#footnote-9) couvrant notamment dans le contexte de ce kit de démarrage les technologies ASP.NET MVC, OData et jQuery.

Remarque : Le [blog du groupe Produit de la plateforme Windows Azure](http://blogs.msdn.com/b/windowsazure/)[[10]](#footnote-10) 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 la plateforme.

# Présentation rapide du Kit de démarrage OGDI DataLab

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.



L’[édition du premier trimestre 2011](http://www.rslnmag.fr/Content/Files/ff/UERGL1YyUkN6ZHdFaTZNSFJOQS5wZGY=)[[11]](#footnote-11) du magazine RSLN - Regards sur le numérique propose une enquête complète sur les informations publiques :

*« L’information détenue par le gouvernement fédéral est un bien national ! » Dès le lendemain de son investiture, le 20 janvier 2009, Barack Obama hissait l’Open data au premier rang de son mandat. Quatre mois plus tard, le site data.gov était lancé, qui rendait accessibles des informations publiques de tout ordre : consommation d’énergie dans les foyers, rapports d’inspection des centrales nucléaires, dépenses de l’Etat, etc. Une petite révolution, montrant qu’un état peut, sans perdre son pouvoir, jouer la carte de la transparence avec ses concitoyens.*

*Aujourd’hui personne ne mesure ni l’étendue ni les conséquences de cette philosophie, qui parie sur l’intelligence collective, la créativité fertilisée par le croisement d’autant de données et d’usages à inventer.*

*Nul ne sait, mais le rêve est permis. Car des possibilités immenses se dessinent pour créer les services et les applications qui vont faciliter la vie quotidienne des citoyens, injecter une forte dose de transparence dans la vie démocratique, entraîner de spectaculaires avancées de la connaissance…*

*À bien y regarder, l’Open data n’est ni plus ni moins que l’opportunité, pour nous tous, de produire du sens, du bien-être et de l’intelligence à partir des tonnes d’informations amassées, classées, archivées et produites par nous-mêmes. Ne manquent plus que les idées pour leur donner vie… une seconde fois !*

Remarque : Le site Web du magazine RSLN propose un [dossier complet](http://www.rslnmag.fr/blog/dossier/l-open-data_et-nous_et-nous_et-nous_/2011/4/19/quel-sens-donner-au-monde-des-donnees-ouvertes_/)[[12]](#footnote-12) sur le sens à donner au monde des données ouvertes avec des éclairages complémentaires comme notamment au travers de la publication des vidéos tirées de la conférence [L’Open Data, et nous, et nous et nous ?](http://www.rslnmag.fr/blog/theme/les-rencontres-rsln/2011/2/10/l-open-data_et-nous_et-nous_et-nous_venez-imaginer-le-monde-de-l-abondance-des-donnees_/)[[13]](#footnote-13) organisée le 17 mars dernier avec le [World e.gov Forum](http://wegf.org/fr/)[[14]](#footnote-14) au campus Microsoft.

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 constitue 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 du kit de démarrage OGDI DataLab, une solution sous licence libre qui permet à l’administration et aux collectivités territoriales d’accélérer leur projet de publication de données, de publier ainsi les informations ouvertes de leur choix plus rapidement et efficacement, et de les rendre navigables et interrogeables par les citoyens et les applications.

La technologie ainsi mise à disposition sous forme de kit de démarrage gratuit et sous licence libre 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 le Cloud Windows Azure. Les citoyens et les développeurs d’application peuvent accéder aux données ainsi publiées dans une solution fondée sur le kit de démarrage OGDI DataLab depuis toute plateforme ou navigateur.

Pour cela, l’architecture de la solution prend en charge le style d’architecture REST et les services Web RESTful. [Wikipedia](http://en.wikipedia.org/wiki/Representational_State_Transfer)[[15]](#footnote-15) définit REST (*Representational State Transfer*) et les services Web RESTful comme suit:

« Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. The term Representational State Transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation. Fielding is one of the principal authors of the Hypertext Transfer Protocol (HTTP) specification versions 1.0 and 1.1.

Conforming to the REST constraints is referred to as being ‘RESTful’. …

A RESTful web service (also called a RESTful web API) is a simple web service implemented using HTTP and the principles of REST. It is a collection of resources, with three defined aspects:

* The base URI for the web service, such as http://example.com/resources/
* The MIME type of the data supported by the web service. This is often JSON, XML or YAML but can be any other valid MIME type.
* The set of operations supported by the web service using HTTP methods (e.g., POST, GET, PUT or DELETE). »

Les services Web RESTful sont accessibles par une très large variété de technologies utilisées aujourd’hui sur Internet, telles que ASP.NET, JavaScript, Flash, Flex, PHP, Python, Ruby, Silverlight, Objective-C (iPhone iOS), Android, etc., ou encore les sites de cartographie comme Géoportail, le portail des territoires et des citoyens, Bing Maps, Google Maps, etc.

Que vous soyez une Administration, une Collectivité Territoriale, une entreprise en contrat avec une telle entité, une entreprise souhaitant utiliser les informations publiques, ou un développeur « citoyen », ces interfaces de programmation (*API*) ouvertes vous permettent de construire des applications innovantes, des représentations graphiques avancées et des agrégations (*mash-up*) de données dont pourront bénéficier les citoyens et usagers.

Aujourd’hui, la plupart des initiatives se concentrent uniquement sur la publication des données uniquement au travers de fichiers dits « plats », fichiers Excel par exemple, qu’il est nécessaire de télécharger intégralement. L’écriture d'une application signifie alors téléchargement, hébergement et maintenance des données.

Le kit de démarrage OGDI DataLab procure de nombreux avantages. En premier lieu, il s’agit d’une plate-forme :

* Les personnes peuvent écrire des programmes pour accéder aux données sans téléchargement ni hébergement ; les créateurs d’applications ont seulement besoin d'accéder à des interfaces de programmation fondées sur le Web de style REST ;
* Les Administration et Collectivités Territoriales peuvent conserver les données à jour afin que les créateurs d’applications n'aient pas à s'en inquiéter ;
* Les Administration et Collectivités Territoriales n’ont pas la nécessité d'investir et de maintenir des plateformes et logiciels système et ne paieront que pour les ressources qui ont été effectivement consommées ;
* Les applications clientes peuvent être écrites à l'aide de toute technologie, sans restriction aucune : Flash, Java, Silverlight, Google Maps, iPhone, Android etc.

Il offre choix et ouverture. Notamment :

* Les applications clientes et les visualisations de tout type peuvent être écrites à l'aide de toute technologie ;
* Les données sont exposées dans une variété de formats ouverts : [AtomPub (*Atom Publishing Protocol*)](http://tools.ietf.org/html/rfc5023)[[16]](#footnote-16), [KML (*Keyhole Markup Language*)](http://www.opengeospatial.org/standards/kml/)[[17]](#footnote-17), [JSON (*JavaScript Object Notation*)](http://tools.ietf.org/html/rfc4627)[[18]](#footnote-18), [JSONP (*JSON with padding*)](http://www.json-p.org/)[[19]](#footnote-19) et [RDF (*Resource Description Framework*)](http://www.w3.org/RDF/)[[20]](#footnote-20).

Le site exemple à l’adresse <http://ogdifrance.cloudapp.net> est construit en utilisant les composantes du Kit de démarrage OGDI DataLab et fournit un accès interactif à certains jeux de données accessibles au public avec, à la clé, exemples de code et ressources pour écrire des applications en utilisant le modèle de programmation mis en œuvre par OGDI DataLab.

Le Kit de démarrage OGDI DataLab v6 correspond :

* D’une part, à l’adaptation en français accompagnée de certaines évolutions du kit de développement logiciel OGDI v2 disponible en téléchargement sous la même licence libre sur la [forge CodePlex](http://ogdi.codeplex.com/)[[21]](#footnote-21). Cette adaptation correspond au kit de démarrage OGDI v2a disponible sur le [Centre de téléchargement Microsoft France](http://www.microsoft.com/downloads/fr-fr/details.aspx?FamilyID=420837f8-25ec-47ba-b558-b6a7f64b3348)[[22]](#footnote-22).
* Et, d’autre part, à la fusion avec le kit de développement DataLab (OGDI v4) désormais disponible sur la [forge Github](https://github.com/openlab/datalab)[[23]](#footnote-23), qui constitue lui-même une évolution du kit de développement logiciel OGDI v2 disponible sur la forge CodePlex.



Le Kit de démarrage OGDI DataLab v6, fusion de ces deux branches de code, constitue le kit de développement DataLab v6 sur la forge Github ; ce dernier est ainsi l’unique référence communautaire.

# Présentation rapide du protocole OData

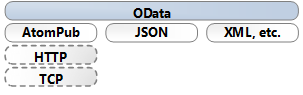


De vastes ensembles de données sont disponibles aujourd’hui et les données sont aujourd’hui collectées et conservées à un rythme encore jamais vu. Beaucoup, pour ne pas dire la plupart, de ces données sont verrouillées dans des applications et/ou des formats spécifiques les rendant de facto difficiles d'accès ou d'intégration en vue de nouvelles utilisations et innovations sociales et économiques.

Le [protocole ouvert de données OData (*Open Data Protocol*)](http://www.odata.org/developers/protocols)[[24]](#footnote-24) est un protocole Web pour requêter et effectuer des opérations sur des données. En permettant la création aisée de services orientés vers les données, il fournit un moyen de déverrouiller vos données et les exempte des silos qui existent aujourd'hui dans les applications.

OData s’applique les technologies Web telles que http, AtomPub, JSON, JSONP pour :

* Permettre aux sources de données, qu’il s’agisse de façon non exhaustive de bases de données relationnelles, de systèmes de fichiers, de systèmes de gestion de contenu, de sites Web traditionnels, de sites collaboratifs, etc., d'être exposées sur le Web de façon simple, sécurisée et interopérable ;
* Et fournir ainsi un accès à l’information depuis un large éventail d’applications, de services et de magasins/stockages de données.



Le protocole résulte et tire bénéfice des expériences de mise en œuvre de serveurs et de clients AtomPub dans une variété de produits au cours des dernières années.

Il exploite les identificateurs de ressource uniformes (*Uniform Resource Identifier* ou URI) pour l'identification des ressources (*Uniform Resource Locator* ou URL) et une interface uniforme http permettant d'interagir avec ces ressources, tout comme le ferait un utilisateur avec n'importe quelle application sur le Web.

Les (ensembles de) données et leur logique associée sont exposés comme un flux (*feed*) OData,

Ces derniers contiennent des données structurées (appelées entités) qui elles-mêmes contiennent des propriétés de types primitifs ou complexes. Par exemple, on peut définir un ensemble de données pour les points d'apport volontaire (PAV) des déchets ménagers comme celui mis à disposition par le [Service SIG Rennes Métropole](http://www.data.rennes-metropole.fr/les-donnees/catalogue/?tx_icsopendatastore_pi1%5buid%5d=47)[[25]](#footnote-25). Cet ensemble va contenir une liste de PAV, chacun ayant des propriétés telles que le code INSEE de la commune sur laquelle est situé le PAV, le type de PAV, le ou les flux de déchets pris en charge (verre, journaux-magazine, ordures ménagères ou multi-matériaux), si le PAV est implanté sur le domaine public ou pas, une longitude et une latitude en termes de coordonnées géo spatiales, etc.

Une fois les données exposées comme un flux OData via l’interface http, de simples requêtes http sur l’URL correspondante permettent de consommer les données. Il est bien entendu possible d’effectuer des opérations de filtrage, de tri ou d’agrégation en rajoutant certains paramètres à l’URL. Par exemple, la requête  http://ogdifrance.cloudapp.net:8080/v1/frOpenData/RennesPAV/?$filter=flux eq 'VE' pour reprendre l’illustration précédente permet de ne s’intéresser qu’aux PAV prenant en charge le verre (VE) comme flux de déchet.

Les données peuvent être consommées en format AtomPub, [JSON](http://json.org/)[[26]](#footnote-26), ou encore KML suivant le choix du consommateur : bibliothèques à destination des développeurs, produits dit de « *business intelligence* », composants de visualisation, etc.

Cet engagement et adhésion envers les principes fondamentaux du Web et ses standards permettent à OData d’offrir une intégration des données simplifiée et un nouveau seuil en termes d’interopérabilité au travers d’un large éventail de producteurs et de consommateurs.

Il existe ainsi aujourd’hui des bibliothèques en .NET, iPhone, Java, PHP, JavaScript, Ruby, etc. quel que soit le coté où l’on se place comme notamment décrit dans le billet [OData interoperability with .NET, Java, PHP, iPhone and more](http://blogs.msdn.com/b/interoperability/archive/2010/03/16/odata-interoperability-with-net-java-php-iphone-and-more.aspx)[[27]](#footnote-27). Une liste des ressources ainsi disponibles est proposée sur le [site OData.org](http://www.odata.org/developers/odata-sdk)[[28]](#footnote-28). Grâce à OData et son écosystème, les développeurs peuvent donc créer des applications et services innovants qui viennent enrichir l’écosystème applicatif des Administrations et Collectivités Territoriales choisissant de publier leurs données et ainsi fournir une valeur ajoutée certaine aux citoyens et usagers.

Les spécifications du protocole OData ainsi rapidement décrit sont disponibles sous [l’OSP (Open Specification Promise)](http://www.microsoft.com/openspecifications/en/us/programs/osp/default.aspx)[[29]](#footnote-29) qui permet à tout le monde, y compris les communautés et les projets open source, de créer des services de données pour tout environnement d’exécution ainsi que des clients pour consommer à leur tour ces mêmes services. Autrement dit, **ces spécifications** **sont ouvertes, publiques et libres d’implémentation**. De ce fait, au regard de l’article 4 de la loi n° 2004-575 du 21 juin 2004 pour la confiance dans l’économie numérique qui définit au niveau de l’article 4 un « standard ouvert » comme suit :

« On entend par standard ouvert tout protocole de communication, d’interconnexion ou d’échange et tout format de données interopérable et dont les spécifications techniques sont publiques et sans restriction d’accès ni de mise en œuvre. »

Le protocole OData est un standard ouvert et s’inscrit à ce titre pleinement dans la démarche Open Data. La démarche Open Data encourage en effet le recours à des « standards ouverts » chaque fois que cela s’avère possible de façon, d’un point de vue technique, à faciliter le partage, la réutilisation, la pérennité dans le temps, etc. des données ouvertes et applications ainsi mises à disposition.

**Ces spécifications font également l’objet d’une adoption croissante par l’industrie et d’un nombre d’implémentations en évolution constante tant en termes de producteurs que de clients consommateur.**

Remarque : Pour de plus amples informations, vous pouvez consulter le site dédié à <http://www.odata.org>.

Ainsi, face à une demande du marché de plus en plus importante quant à un accès facile aux données (ouvertes) à travers de multiples plateformes et appareils, une pression croissante pour créer un Web plus ouvert et programmable en ayant une approche commune afin d'exposer et des consommer des données, Citrix, IBM, Microsoft, Progress Software, SAP et WSO2 ont proposé la création d’un Comité technique (TC) OData au sein de l’OASIS (Organization for the Advancement of Structured Information Standards), un consortium international de standards ouverts : « To accomplish the goal of open data for the open Web, we have seen a push for support to enable access to and use of data across platforms, applications and devices,” déclare Jean Paoli, président de [Microsoft Open Technologies Inc](http://blogs.msdn.com/b/interoperability/archive/2012/04/12/announcing-one-more-way-microsoft-will-engage-with-the-open-source-and-standards-communities.aspx). “Taking steps to standardize OData through OASIS allows developers to act on the data in a more well-defined way. »

Le TC OData OASIS a donc pour mission de standardiser le protocole OData comme décrit dans la [proposition de charte pour le comité technique OData](http://lists.oasis-open.org/archives/oasis-charter-discuss/201205/msg00000.html). A la date de publication de ce document, le TC OData OASIS a approuvé la spécification DRAFT 01 de la version 4.0 d’OData et a lancé une [revue publique](https://lists.oasis-open.org/archives/odata/201305/msg00091.html)[[30]](#footnote-30) de cette dernière.

# Présentation rapide de Windows Azure



Windows Azure est une plateforme Cloud ouverte et flexible qui vous permet de générer, déployer et gérer rapidement des applications à travers un réseau global de centres de données gérés par Microsoft. Vous pouvez développer des applications à l'aide de n'importe quel langage, n'importe quel outil ou n'importe quelle infrastructure.

Ainsi, Windows Azure propose différents services et technologies pour l'intégration, le déploiement, la mise à l'échelle et la gestion d’applications ou de composantes de celles-ci dans le Cloud en s’inspirant du modèle qu’offre la plateforme Windows Server pour l’entreprise.

Cette plateforme comprend un socle d’exécution, Windows Azure, que l’on peut considérer comme l’équivalent d’un système d'exploitation dans le Cloud. De la même manière que l’on développe des applications pour Windows Server, on peut désormais développer des applications Cloud pour Windows Azure qui se charge de les exécuter.

Dans sa dimension Plateforme à la demande ou PaaS (Platform as-a-Service), Windows Azure est un système d'exploitation pour les services Cloud qui fait office d'environnement de développement, d'hébergement de services et de gestion de services pour la plateforme Windows Azure. Windows Azure vous fournit ainsi une plateforme Cloud flexible qui satisfera les besoins applicatifs. Cela vous permet d'héberger et de faire évoluer de façon fiable le code d'application dans les rôles de calcul. Vous pouvez stocker les données à l'aide de bases de données relationnelles SQL, de magasins de tables NoSQL et de magasins de blob non structurés, et éventuellement utiliser les services Hadoop et Business Intelligence (BI) pour les explorer. Vous pouvez tirer parti des fonctions de messagerie fiables de Windows Azure pour activer les applications distribuées évolutives, et fournir des solutions hybrides exécutées sur un Cloud et dans l'environnement d'entreprise local. Les services de mise en cache en mémoire distribuée et du réseau de distribution de contenu de Windows Azure vous permettent de réduire la latence et de garantir des performances d'application optimales n'importe où dans le monde.

Windows Azure vous permet d'utiliser n'importe quel langage, n'importe quelle infrastructure ou n'importe quel outil pour développer des applications. Les fonctionnalités et services sont exposés à l'aide de protocoles REST ouverts. Les bibliothèques clientes de Windows Azure sont disponibles pour plusieurs langages de programmation et sont publiées suivant les termes d'une licence source ouverte et hébergée sur GitHub.

Windows Azure fournit une infrastructure à la demande ou IaaS (Infrastructure as-a-Service) qui évolue et s'adapte aux nouveaux besoins d’une organisation. Windows Azure permet d'ajouter des réseaux virtuels et des ordinateurs virtuels Windows Server et Linux en quelques minutes seulement et d'adapter votre utilisation au fur et à mesure de l'évolution de vos besoins.

Pour de plus amples détails sur les différentes possibilités que vous offre Windows azure, vous pouvez consulter la vue d’ensemble proposée à l’adresse <http://www.windowsazure.com/fr-fr/solutions/>.

Windows Azure est actuellement commercialisé dans 41 pays. Vous pouvez accéder à Windows Azure avec un paiement à l’usage, sans engagement, ou avec des offres d’abonnements comme décrit à l’adresse <http://www.microsoft.com/france/windows-azure/Offres.aspx>, ou encore dans le cadre d’un accord d’entreprise Microsoft.

Au-delà de ces éléments, vous pouvez vous appuyer également sur les ressources mises à disposition par Microsoft Platform Ready (MPR) à l’adresse <http://microsoftplatformready.com/fr>. MPR a été conçu pour vous fournir les éléments dont vous avez besoin pour planifier, développer, tester et commercialiser vos solutions sur la plateforme Microsoft en entreprise et/ou dans le Cloud.

Les données ouvertes supposent une plateforme capable d’héberger un nombre croissant de données (et d’applications), totalement sécurisée, accessible à tous et tout type de plateforme ou environnement, et en mesure de supporter de forts pics de trafic.

Dans un tel contexte, l’utilisation de la plateforme Windows Azure pour ce kit de démarrage présente de multiples bénéfices :

* Comme évoqué ci-avant, la plateforme prend en charge une variété de normes et standards, de langages et de protocoles offrant ainsi une réelle interopérabilité. Jean Paoli en charge de la stratégie en matière d’interopérabilité chez Microsoft revient sur les éléments d’interopérabilité d’une plateforme Cloud dans le billet [Interoperability Elements of a Cloud Platform Outlined at OSCON](http://blogs.msdn.com/b/interoperability/archive/2010/07/22/interoperability-elements-of-a-cloud-platform-outlined-at-oscon.aspx)[[31]](#footnote-31) ;
* Comme fournisseur de services, Microsoft doit se conformer aux exigences réglementaires des entités gouvernementales pour les juridictions où la plateforme Windows Azure est opérée pour en assurer la conformité et gérer les risques en matière de sécurité. Windows Azure, qui fonctionne dans l'infrastructure de [Microsoft Global Foundation Services (GFS)](http://www.globalfoundationservices.com/)[[32]](#footnote-32) [certifiée ISO 27001](http://www.bsiamerica.com/en-us/Assessment-and-Certification-services/Management-systems/Certificate-and-Client-Directory-search/Search/Search-Results/?pg=1&licencenumber=IS+533913&searchkey=companyXeqXmicrosoft)[[33]](#footnote-33), est également [certifiée ISO 27001](http://www.bsiamerica.com/en-us/Assessment-and-Certification-services/Management-systems/Certificate-and-Client-Directory-search/Search/Search-Results/?pg=1&licencenumber=IS+577753&searchkey=licenceXeqX577753)[[34]](#footnote-34).

La norme ISO/IEC 27001 est reconnue dans le monde entier comme l'une des principales normes internationales en matière de gestion de la sécurité et des exigences associées. Dans le même temps, il est important de noter que Windows Azure va au-delà d’ISO 27001 avec l’application de [contrôles de sécurité additionnels](http://blogs.technet.com/b/gfs/archive/2009/06/16/response-to-question-about-sas-70-objectives.aspx)[[35]](#footnote-35).

Par ailleurs, Microsoft est signataire du [Safe Harbor](http://www.export.gov/safeharbor/eu/eg_main_018365.asp)[[36]](#footnote-36) et s'engage à remplir toutes ses obligations en vertu du Framework Safe Harbor.

D’autres certifications de l'industrie ont été obtenues ou sont en cours d’évaluation pour la plateforme. Pour de plus amples informations, nous vous invitons à consulter les ressources du [Centre de gestion de la confidentialité Windows Azure](http://www.windowsazure.com/fr-fr/support/trust-center/)[[37]](#footnote-37) et notamment le document [Réponse standard pour les demandes d'information, Sécurité et Respect de la vie privée](http://www.microsoft.com/en-us/download/details.aspx?id=26647)[[38]](#footnote-38) ;

* Une quantité illimitée d’informations publiques (données ouvertes) peut être hébergée dans le Cloud sans matériel serveur supplémentaire et gestion additionnelle. Les informations publiques deviennent encore plus précieuses avec une plus grande pertinence locale. Par exemple, les villes et les collectivités (région, conseil général, communauté urbaine, etc.) génèrent des données riches sur l’espace public (voirie, assainissement), le transport et ses perturbations, les espaces naturels (parcs, arbres), les projets de construction, etc. ;
* Le Cloud est hautement évolutif (*scalable*) et vous ne payez seulement que pour ce qui est utilisé. Ceci le rend idéal pour l'hébergement d’informations publiques qui peuvent susciter des volumes de transactions variés en fonction de l’intérêt porté. Windows Azure propose une approche économique des plus attractives pour démarrer rapidement et mettre à disposition des informations publiques ;
* [Windows Azure Marketplace](https://datamarket.azure.com/)[[39]](#footnote-39) permet aux entités publiques (gouvernement, ministères, collectivités locales et territoriales, mairies, services déconcentrés de l'Etat, etc.) de facturer les éditeurs d’applications pour la réutilisation et la consommation des informations publiques mises à disposition et aux éditeurs d’applications de facturer leurs utilisateurs pour les services à valeur ajoutée offerts au-dessus de ces données.

# Configuration requise pour le Kit de démarrage OGDI DataLab

## Configuration système

Le Kit de démarrage OGDI DataLab a été conçu pour la plateforme Windows Azure. Les prérequis exigés pour ce Kit de démarrage 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 3.0. 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.

[](http://www.microsoft.com/web/downloads/platform.aspx)

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 Vista SP2 (32 bits ou 64 bits) ou Windows Server 2008 SP2 (32 bits ou 64 bits) avec le [correctif logiciel KB971842 Correction des métadonnées WCF lors de l'hébergement derrière un programme d'équilibrage de charge](http://support.microsoft.com/kb/971842)[[40]](#footnote-40),
* 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)[[41]](#footnote-41).
* Microsoft Windows 8 (32 bits ou 64 bits) ou Windows Server 2012.

Remarque : Certaines fonctionnalités de Windows Server AppFabric, tels que le démarrage automatique, fonctionnent uniquement avec Windows 7 et Windows Server 2008 R2.

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 : Windows Azure requiert Microsoft .NET Framework 3.5 (ou ultérieur).

* Microsoft Visual Studio 2012 (ou Microsoft Visual Studio 2012 Express for Web) avec :
* [Correctif logiciel KB983301 Activation de Windows Azure IntelliTrace sur les systèmes d'exploitation 32 bits](http://archive.msdn.microsoft.com/KB983301)[[42]](#footnote-42) (non requis pour les systèmes x64) ;
* [Module de réécriture d'URL d'IIS 7.0](http://www.iis.net/download/URLRewrite)[[43]](#footnote-43) ;
* [Windows PowerShell 2.0](http://go.microsoft.com/fwlink/?LinkId=192992)[[44]](#footnote-44) (non requis pour Windows 7 et Windows Server 2008 R2).
* [Kit de développement logiciel Windows Azure 1.8 pour .NET (octobre 2012)](http://www.microsoft.com/windowsazure/sdk/).



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 sur le Microsoft .NET Framework, veuillez consulter le Centre de développement MSDN éponyme dédié à l’adresse <http://msdn.microsoft.com/fr-fr/netframework/aa663324.aspx>.

Pour des informations complémentaires sur la technologie WCF, veuillez consulter le Centre de développement MSDN 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.

Le Kit de démarrage repose sur la version 4.5 du Microsoft .NET Framework.

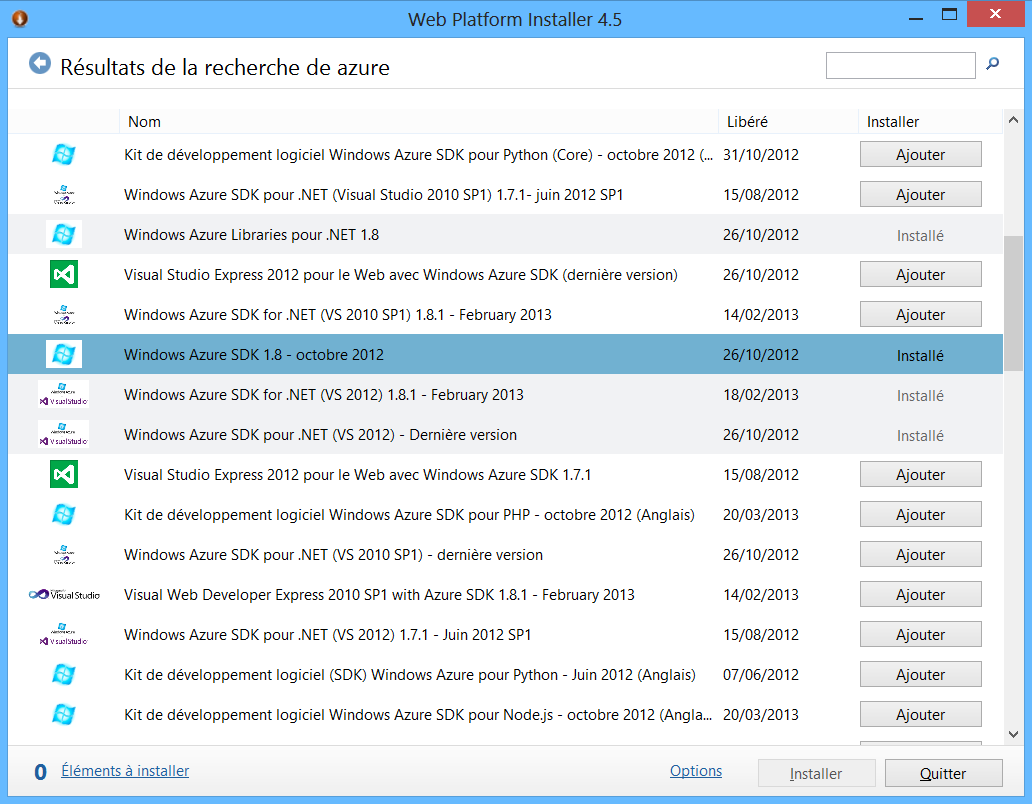
Le package redistribuable Microsoft .NET Framework 4.5 est disponible par téléchargement gratuit à l’adresse <http://www.microsoft.com/fr-fr/download/details.aspx?id=30653>.

Vous pouvez télécharger Microsoft Visual Studio Express 2012 for Web gratuitement sur <http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-for-web>.

Le package Windows Azure Tools pour Microsoft Visual Studio, qui comprend le kit de développement Windows Azure, étend Microsoft Visual Studio 2012 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 Windows Azure.

Le Kit de développement logiciel Windows Azure 1.8 pour .NET (octobre 2012) est disponible en téléchargement à l’adresse <http://www.windowsazure.com/en-us/downloads/?fb=en-us>. Sélectionner le SDK .NET puis cliquer sur le lien VS 2012.

Au-delà d’une installation manuelle (<http://www.microsoft.com/fr-fr/download/details.aspx?id=35448>) avec les prérequis et correctifs logiciels que cela suppose, est proposée une installation automatisée de l’ensemble via le programme d’installation de la plateforme Web Microsoft 4.5 depuis l’adresse <http://www.microsoft.com/web/gallery/install.aspx>. Le Kit de développement logiciel Windows Azure 1.8 pour .NET (octobre 2012) peut être ajouté dans la liste des produits et installés.



## 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 le Kit de démarrage sur une plateforme Windows 7 ou Windows Server 2008 R2 :

1. Installer Internet Information Services (IIS) 8.0,
2. Installer le Microsoft Framework .NET 4.5,
3. Installer Microsoft Visual Studio 2012,
4. Enfin, utiliser le programme d’installation de la plateforme Web Microsoft 4.5 pour installer le Kit de développement logiciel Windows Azure 1.8 pour .NET (octobre 2012), avec leurs prérequis et correctifs logiciels associés.

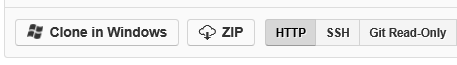
# Obtention des sources du Kit de démarrage OGDI DataLab

La solution communautaire OGDI DataLab est disponible sur la forge GitHub à l’adresse <https://github.com/openlab/DataLab>.

Les solutions communautaires évoluant avec l’investissement de la communauté, il est souhaitable de vos 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 le bouton 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)[[45]](#footnote-45) et [GitHub for Windows](http://blogs.msdn.com/b/ogdifrance/archive/2012/06/14/github-for-windows.aspx)[[46]](#footnote-46), tous deux présents sur le blog MSDN [Open Data France](http://aka.ms/OpenDataFrance)[[47]](#footnote-47).

Par ailleurs, L’environnement Visual Studio 2012 propose depuis l’[Update 2](http://www.microsoft.com/en-us/download/details.aspx?id=38188)[[48]](#footnote-48) avec les [outils Visual Studio pour Git](http://visualstudiogallery.msdn.microsoft.com/abafc7d6-dcaa-40f4-8a5e-d6724bdb980c)[[49]](#footnote-49) une intégration directe avec GitHub. L’article [Integrating and Using Github in Visual Studio 2012](http://www.codeproject.com/Articles/581907/IntegratingplusandplusUsingplusGithubplusinplusVis)[[50]](#footnote-50) vous propose un guide pas-à-pas dans ce contexte.

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

# Contenu du Kit de démarrage OGDI DataLab

Le Kit de démarrage OGDI DataLab se présente sous la forme d’une solution Visual Studio 2012 *ogdi.sln* à destination de la plateforme Windows Azure.

Cette solution s’articule autour de trois composantes de base :

1. Kit de développement logiciel interactif : DataBrowser.
2. Service de données.
3. Utilitaires de chargement de données.

Il convient également de noter enfin qu’un module Drupal est disponible pour intégrer le kit de démarrage OGDI DataLab avec Drupal, présentant les données ainsi mises à disposition sous forme à la fois de tableau complètement fonctionnel et de carte Bing. Ce module OGDI Field est téléchargeable directement depuis le [site Drupal](http://drupal.org/project/ogdi_field)[[51]](#footnote-51).

## Kit de développement logiciel interactif : DataBrowser

Le Kit de développement logiciel (*software development kit* ou SDK en abrégé) interactif est un site Web à destination des utilisateurs finaux et des développeurs. Il consomme les informations publiques mises à disposition par l’entrepôt de données en utilisant le service de données (Cf. ci-après).

Il s’agit d’une application Web [ASP.NET MVC 3.0](http://www.asp.net/mvc)[[52]](#footnote-52) conforme à l’architecture Modèle-Vue-Contrôleur (MVC). L’architecture MVC est 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.

[](http://jquery.com/)

L’application Web utilise [jQuery](http://jquery.com/)[[53]](#footnote-53) et une variété d'autres composants open source et permet aux utilisateurs de parcourir et d'interroger les données publiées. Les données peuvent être commodément visualisées dans des formats largement utilisés et reconnaissables tels que des tableaux, des cartes, des graphiques à barres ou des diagrammes circulaires. Ainsi, au lieu de télécharger un fichier et d’en étudier le contenu, les utilisateurs finaux peuvent interagir directement avec des outils visuels conviviaux qui présentent les données complexes d'une manière plus significative, intuitive et agréable. On peut ainsi citer le [complément PowerPivot à Excel 2010](http://msdn.microsoft.com/fr-fr/library/ee210644.aspx)[[54]](#footnote-54) qui permet de charger ces données ainsi exposées et les mettre à disposition d'objets de visualisation des données, comme des tableaux croisés dynamiques, dans une feuille de travail d’un classeur Excel.

Au-delà de la capacité de navigation dynamique dans les données, les développeurs peuvent rapidement apprendre comment utiliser les données publiées exposées par le socle OGDI DataLab depuis leurs propres applications à partir d’exemples de code prêts à l’exécution sur le site. Ces exemples sont disponibles dans une variété de langages et de bibliothèques/Frameworks largement utilisés sur le Web, incluant, entre autres, C#, Flex, JavaScript, PHP, Python, Silverlight.

## Service de données

Le kit de démarrage OGDI DataLab utilise et expose des services Web RESTful, en l’occurrence des services Web implémentés à l'aide du protocole http et des principes REST pour exposer des données pour la programmation.

Un outil Web de configuration initiale (*ConfigTool*) permet de définir au niveau d’un compte de stockage Windows Azure la configuration du Services de données. Celle-ci référence en particulier pour l’entrepôt de données du service de données les différents comptes de stockage Windows Azure utilisés pour le stockage des données OGDI DataLab.

Une fois l’entrepôt configuré, le kit retourne par défaut les données dans le format du protocole OData, une extension à AtomPub comme décrit à la section § Présentation rapide du protocole ODATA.

Le format de base des URLs pour invoquer un service du kit est la suivante : http://<*projet*>.cloudapp.net:8080/v1/<*conteneur*>/<*ensemble de données*>?<*requête*>

où:

* <*projet*> est le nom de projet de service pour la publication du service fondé sur le Kit de démarrage dans Windows Azure, par exemple *ogdifrance*,
* <*conteneur*> est le nom du conteneur, par exemple, *frOpenData* pour les ensembles de données mis à disposition par l’entrepôt de données,
* <*ensemble de données*> est le nom de l'ensemble de données, par exemple, *RennesPAV* vis-à-vis de l'ensemble des données localisant les points d'apport volontaire (PAV) des déchets ménagers mis à disposition par le [Service SIG Rennes Métropole](http://www.data.rennes-metropole.fr/les-donnees/catalogue/?tx_icsopendatastore_pi1%5buid%5d=47)[[55]](#footnote-55),
* <*requête*> est l’ensemble des paramètres de la requête, exprimé en utilisant un sous ensemble de la [syntaxe de requête de WCF Data Services](http://msdn.microsoft.com/fr-fr/library/dd673933.aspx)[[56]](#footnote-56).

Remarque : Le kit de démarrage OGDI DataLab supporte notamment les options de requête de WCF Data Services *$filter*, *$top, $skip, $orderby et $select*. Pour de plus amples informations, vous pouvez consulter le billet [OGDI DataLab évolue !](http://blogs.msdn.com/b/ogdifrance/archive/2013/04/22/ogdi-datalab-233-volue.aspx)[[57]](#footnote-57).

Remarque : Si une propriété d'un ensemble de données a une valeur nulle, elle sera inexistante dans le résultat retourné par le Service de données. Par exemple, si dans un ensemble de données vous avez la propriété ville et arrondissement et que votre requête ne retourne que des villes sans arrondissement, le champ arrondissement ne sera pas retourné. Votre application doit prendre en compte la possible absence d'un champ dans le résultat retourné.

Le Service de données peut aussi retourner les données au format JSON qui peut être plus facilement exploitables depuis JavaScript ou autre technologie. A ce propos, JSON est particulièrement bien adapté aux applications mobiles compte tenu de sa compacité vis-à-vis d’XML et permet donc d’économiser de la bande passante.

Pour retourner des données au format JSON, il suffit d'ajouter *format=json* à votre requête. Par exemple, pour retourner les points d'apport volontaire des déchets ménagers en verre pour la métropole de Rennes au format JSON :

http://ogdifrance.cloudapp.net:8080/v1/frOpenData/RennesPAV/?$filter=flux eq 'VE' &format=json

Pour diminuer la vulnérabilité en terme de sécurité liée aux attaques de type scriptage intersite (*Cross-site Scripting* ou XSS en abrégé), généralement les navigateurs empêchent le JavaScript d'une page Web venant d'un domaine X de faire des requêtes http vers un autre domaine Y. Cela empêche tout code JavaScript hébergé sur un domaine de faire des appels directs au Service de données, mais il existe plusieurs techniques qui peuvent être utilisées comme la [technique fondée sur des IFrames](http://softwareas.com/cross-domain-communication-with-iframes)[[58]](#footnote-58).

Le kit de démarrage OGDI DataLab fournit un support direct pour la technique JSONP. En utilisant cette technique, le Service de données appelle la fonction de rappel (*callback*) que vous aurez spécifiée, passant le résultat de votre requête au format JSON comme format d'entrée. Pour utiliser cette technique, il convient de construire votre requête avec les paramètres additionnels *format=json&callback=<fonction de rappel>* où <*fonction de rappel*> est le nom de votre fonction de rappel JavaScript définie dans la page émettant la requête.



Le Service de données prend également en charge le [format RDF](http://www.w3.org/TR/rdf-syntax-grammar/)[[59]](#footnote-59) (*Resource Description Framework*) du W3C et permet ainsi de retourner les données et leurs métadonnées associées de façon formelle (et validée par le [service de validation RDF](http://www.w3.org/RDF/Validator/)[[60]](#footnote-60) du W3C). RDF constitue le langage de base du [Web sémantique](http://www.w3.org/RDF/)[[61]](#footnote-61) tel que décrit par le W3C.

Pour retourner les données au format RDF, il suffit d’ajouter *format=rdf* à votre requête. Par exemple, pour retourner les points d’apport volontaire des déchets ménagers en verre pour la métropole de Rennes au format RDF :

http://ogdifrance.cloudapp.net:8080/v1/frOpenData/RennesPAV/?$filter=flux eq 'VE' &format=rdf

Enfin, la plupart des ensembles de données ainsi proposés est susceptible d’inclure également des données géo spatiales, qui peuvent être alors retournées dans le format KML, rendant le Kit de démarrage OGDI DataLab compatible avec des technologies de cartographie sur poste de travail ou sur le Web, comme Géoportail, Google Earth, Google Maps, Microsoft Bing Maps ou encore Yahoo! Maps.

Pour retourner des données géo spatiales au format KML, il suffit de rajouter *format=kml* à la requête. Par exemple, pour retourner les points d'apport volontaire des déchets ménagers en verre pour la métropole de Rennes au format KML :

http://ogdifrance.cloudapp.net:8080/v1/frOpenData/RennesPAV/?$filter=flux eq 'VE' &format=kml

Remarque : Si l'ensemble de données ne contient pas de données de géolocalisation, une requête de type KML au Service de données retournera un résultat vide.

Les développeurs utilisant Microsoft Visual Studio 2012 peuvent utiliser WCF Data Services pour accéder aux données ainsi exposées via des classes .NET. Avec Visual Studio, cela s’effectue simplement au travers de la fonctionnalité d’[ajout de référence à un service](http://msdn.microsoft.com/fr-fr/data/cc974504.aspx)[[62]](#footnote-62) (**Add Service Reference**).

Dans les faits, et d’une façon générale, les interfaces de programmation Web proposées par le Kit de démarrage OGDI DataLab sont accessibles à une grande variété de technologies côté client. Ainsi, par exemple, les développeurs Java peuvent utiliser la bibliothèque [Restlet Extension for OData](http://www.restlet.org/downloads/)[[63]](#footnote-63), une composante du projet Open Source Restlet mettant à disposition un Framework léger REST pour Java. Les développeurs PHP peuvent tirer parti du kit de développement logiciel [OData pour PHP](http://odataphp.codeplex.com/)[[64]](#footnote-64). Ces formats et technologies bien compris permettent aux développeurs de démarrer rapidement sur de nouvelles applications.

Le Service de données et le service sous-jacent de Tables Windows Azure supportent la pagination pour les résultats d'un volume important.

Remarque : Nous vous invitons à ce titre à consulter l’article MSDN [Query Timeout and Pagination](http://msdn.microsoft.com/en-us/library/dd135718.aspx)[[65]](#footnote-65) de la documentation Windows Azure.

Le Service de données et le service sous-jacent de Tables Windows Azure supportent la pagination pour les résultats d'un volume important.

## Utilitaires de chargement de données

Les utilitaires de chargement de données sont destinés à accompagner la mise en œuvre rapide du kit de démarrage pour en profiter pleinement. Un utilitaire de console prend des données au format CSV ou KML et les publie au niveau d’une mise en œuvre du service de données. Dans le processus de chargement des données, l'utilitaire peut créer un nouvel ensemble de données, ajouter des données ou mettre à jour les données dans un ensemble de données déjà publié. Étant donné que l'utilitaire est une application console qui est contrôlée par des paramètres en ligne de commande, le chargement de données peut être automatisé à l'aide de scripts et intégrer ainsi les processus ETL (*Extract, Transform, and Load*) existants. Une application graphique permet de réaliser les mêmes opérations au travers d’une interface claire et intuitive pour l’utilisateur.

# Organisation du code source du Kit de démarrage

L’intégralité du code source des sites, services et composants OGDI DataLab est située dans le sous-répertoire *Source* du répertoire d’extraction du Kit de démarrage OGDI DataLab.

Compte tenu de ce qui précède, ce répertoire contient 6 sous-dossiers principaux :

1. Le dossier *DataBrowser* correspondant au Kit de développement logiciel interactif. Il regroupe différents projets qui constituent, au final, l’application Web ASP.NET MVC3 destinée à l’environnement Windows Azure et permettant de visualiser les données publiées;
2. Le dossier *DataService* correspondant au Service de données. Il regroupe différents projets relatifs à la mise en œuvre de ces services Web RESTful ;
3. Le dossier *DataLoader* correspondant aux Utilitaires de chargement de données. Il regroupe différents projets relatifs ;
4. Le dossier *ConfigTool* correspondant à un simple outil de configuration pour le compte de stockage Windows Azure contenant la configuration du kit OGDI DataLab ;
5. Le dossier *DAISY* correspondant au plugin permettant de retourner des données stockées dans Windows Azure au format DAISY (*Digital Access Information SYstem*) ;
6. Le dossier *Shared* correspondant à un dossier partagé contenant 2 projets utilisés par le reste de la solution, à savoir *Ogdi.Azure* et *Ogdi.Config* qui servent essentiellement de point d’entrée au compte de stockage) ainsi que plusieurs bibliothèques utilisées par le reste de la solution.

La suite de cette section décrit les projets/éléments de configuration constituants de ces différents sous-dossiers de façon à mieux cerner la solution ainsi proposée par le kit de démarrage OGDI DataLab et ainsi vous l’accaparer 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 4.5 du Microsoft Framework .NET. Les fichiers de solution associés et présents dans ces dossiers sont destinés à l’environnement de développement Visual Studio 2012.

## Dossier DataBrowser (Kit de développement logiciel interactif)

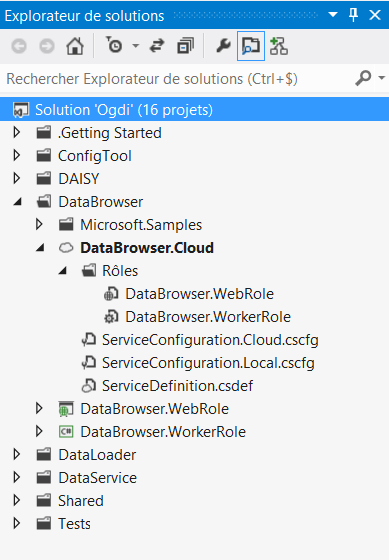
Cette première composante de la solution correspond au kit de développement logiciel interactif et comprend en particulier l’application Web ASP.NET MVC3 destinée à l’environnement Windows Azure.

Le document Déployer le Kit de démarrage OGDI DataLab dans Windows Azure précise les modalités de publication du kit de développement logiciel interactif dans Windows Azure.

Remarque importante : La publication requiert un compte et une souscription active avec Windows Azure. Vous pouvez accéder à Windows Azure avec un paiement à l’usage, sans engagement, ou bien des forfaits comme décrit à l’adresse <http://www.microsoft.com/france/windows-azure/Offres.aspx>.

Remarque : Il est possible de déployer le kit de développement logiciel interactif localement en entreprise moyennent une adaptation légère de la configuration de la solution.

### Organisation projet

* 1. Le kit de développement logiciel interactif est composé des 6 projets suivants :
  2. 

| **Projet** | **Description** |
| --- | --- |
| AspProviders | Ce projet contient les différents fournisseurs (*providers*) ASP.NET en matière de sécurité, à savoir fournisseurs d’authentification (*membership provider*), d’appartenances (*role provider*), de profil (*profile provider*) et d’état de session *(sessionstate provider*), adaptés à un contexte Windows Azure (solution architecturée avec les Tables de Windows Azure pour stocker les différentes données).  La technologie ASP.NET 2.0 et ultérieure propose, en effet, une architecture modulaire extensible en termes de mécanisme de sécurité notamment au travers de la définition de fournisseurs. A ce titre, nous vous conseillons également la lecture de l’article [ASP.NET 2.0 PROVIDER MODEL: INTRODUCTION TO THE PROVIDER MODEL](http://msdn.microsoft.com/en-us/library/aa479030.aspx)[[66]](#footnote-66) sur le modèle de fournisseurs. Le billet [ASP.NET 2.0 Membership, Roles, Forms Authentication, and Security Resources](http://weblogs.asp.net/scottgu/archive/2006/02/24/ASP.NET-2.0-Membership_2C00_-Roles_2C00_-Forms-Authentication_2C00_-and-Security-Resources-.aspx)[[67]](#footnote-67) regroupe également un ensemble de ressources qui vous permet de disposer rapidement d’une bonne vue d’ensemble des possibilités ainsi offertes. |
| DataBrowser.WebRole | Ce projet contient le site web frontal en ASP.NET MVC3 permettant de visualiser les données stockées dans les Tables Windows Azure sous forme de tableaux, cartes ou graphiques. |
| DataBrowser.Cloud | Ce projet standard Windows Azure est configuré pour supporter les deux rôles *DataBrowser.WebRole* et *DataBrowser.WorkerRole* (Cf. ci-dessus et ci-après). Ce projet contient la configuration de ces deux rôles qui composent l’application Cloud à déployer dans Windows Azure.  Il inclut le fichier de définition de service *ServiceDefinition.csdef* qui contient les métadonnées nécessaires à Windows Azure pour comprendre les exigences de l’application, tels que les deux rôles utilisés, leur niveau de confiance, les éventuels 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.  Les fichiers de configuration de service *ServiceConfiguration.Cloud.cscfg et ServiceConfiguration.Local.cscfg* spécifient le nombre d'instances à exécuter pour le ou les rôles configurés dans le projet (ici deux) 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 **Roles** 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 **Roles** mettra à jour les fichiers *ServiceDefinition.csdef,* ServiceConfiguration.Cloud.cscfg et ServiceConfiguration.Local*.cscfg* précédents.  Ce projet sert directement pour publier dans Windows Azure le paquet complet *DataBrowser*.  Dans un souci de réduction des coûts dans Azure ce projet exposera aussi l’application *DataService.WebRole* permettant de fournir les données (Cf. ci-après). Cette application sera exposée, par défaut, sur le port 8080 du même rôle que *DataBrowser.WebRole.*  L’ensemble de la configuration est définie dans le fichier ServiceDefinition.csdef notamment dans les parties <Sites> et <Endpoints>. |
| DataBrowser.WorkerRole | Ce projet de type Service est utilisé en interne dans Windows Azure pour convertir de façon asynchrone les données stockées dans les Tables Windows Azure en fichiers au format DAISY et CSV qui seront ensuite stockés dans un conteneur Blob (toujours dans Windows Azure). Ces fichiers pourront ensuite être téléchargés en toute transparence à travers une simple URL. |
| StorageInterface | Ce projet fournit la classe abstraite *IsdkStorageProviderInterface* qui définit des méthodes pour consommer en format XML, DAISY ou CSV les données ou métadonnées stockées dans les Tables Windows Azure. |
| WindowsAzureStorage | Ce projet fournit la classe *IsdkWindowsAzureStorageProvider* qui est une implémentation de la classe abstraite *IsdkStorageproviderInterface* du projet *StorageInterface* (Cf. ci-dessus). |

### Eléments de configuration

L’ensemble des paramètres de configuration du kit de développement logiciel interactif est regroupé au niveau :

* Des fichiers de configuration de service ServiceConfiguration.Cloud.cscfg et ServiceConfiguration.Local.cscfg du projet Windows Azure Databrowser.Cloud relatif aux rôles DataBrowser.WebRole et DataBrowser.WorkerRole ;
* Du fichier *WebConfig* du projet *DataBrowser.WebRole*. Ce fichier est situé à la racine du projet.

Remarque : Pour accéder à la valeur d’un paramètre depuis une vue utilisant le moteur de vues Razor pour ASP.NET MVC 3, il suffit d’utiliser par exemple la syntaxe suivante :

@Ogdi.Azure.Configuration.OgdiConfiguration.GetValue("*<parameter>*")

où <*parameter*> est le nom du paramètre considéré. La classe *Ogdi.Azure.Configuration.OgdiConfiguration* fait partie du projet *Ogdi.Azure* situé dans le dossier *Shared*.

Remarque : L’approche adoptée ici pour être étendue à d’autres paramètres qu’une personnalisation du kit de démarrage rendrait souhaitable.

Remarque importante : La plupart des paramètres des fichiers de configuration *ServiceConfiguration.Cloud.*cscfg et *ServiceConfiguration.Local.cscfg* du projet Windows Azure *DataBrowser.Cloud* sont dupliqués au niveau des fichiers *Web.Config* et *Web.Release.Config* du projet *DataBrowser.WebRole*, de façon à faciliter le cas échéant un fonctionnement du site Web frontal en ASP.NET MVC 3 indépendamment de l’environnement d’exécution de la plateforme Windows Azure.

La méthode statique *GetValue* de la classe *Ogdi.Azure.Configuration.OgdiConfiguration* permet de faire la distinction entre les situations d’exécution.

public sealed class OgdiConfiguration

{

/// <summary>

/// This method finds the value for the passed key from the AppSettings in XML or Windows Azure role environment.

/// </summary>

/// <param name="key">The value name.</param>

/// <returns>The value.</returns>

public static string GetValue(string key)

{

return

(RoleEnvironment.IsAvailable

? RoleEnvironment.GetConfigurationSettingValue(key)

: null)

?? System.Configuration.ConfigurationManager.AppSettings[key]

?? string.Empty;

}

}

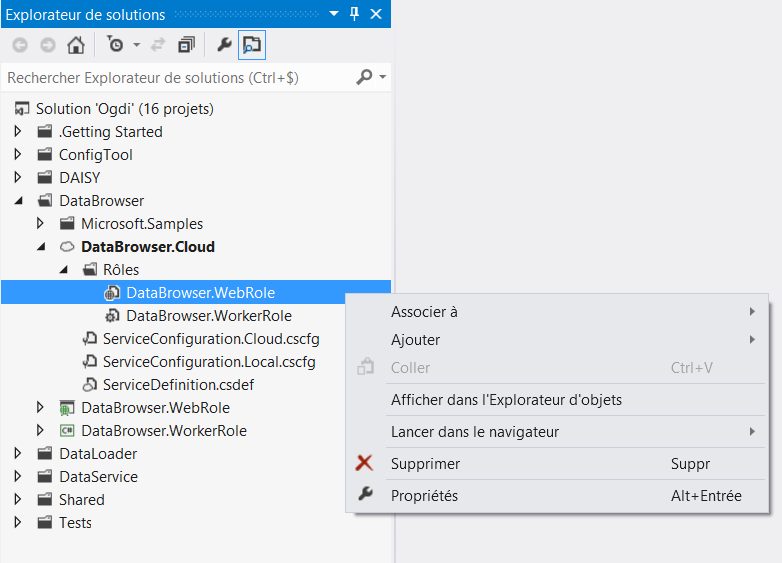
#### Paramètres des fichiers ServiceConfiguration.Cloud.cscfg et ServiceConfiguration.Local.cscfg

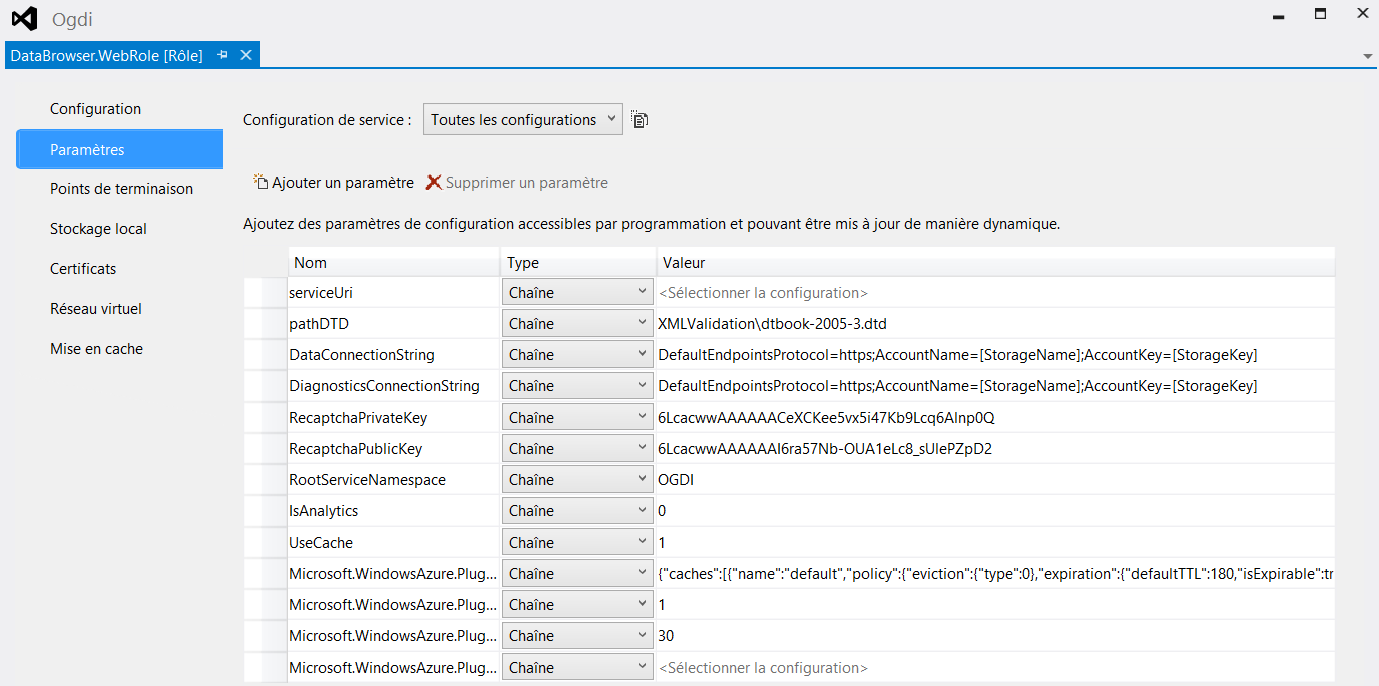
Les fichiers ServiceConfiguration.Cloud.cscfg et ServiceConfiguration.Cloud.cscfg contiennent les paramètres suivants :

| **Paramètre** | **Description** |
| --- | --- |
| DataConnectionString | Chaine de connexion pointant vers le compte de stockage Windows Azure gérant la configuration du kit OGDI DataLab.  Ce paramètre apparaît deux fois dans les fichiers *ServiceConfiguration.\*.cscfg*,une fois pour chacun des deux rôles *DataBrowser.WebRole* et *DataBrowser.WorkerRole* du projet Windows Azure.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**; AccountKey=**[StorageKey]** |
| DiagnosticsConnectionString | Chaine de connexion pointant vers le compte de stockage Windows Azure gérant la configuration du Kit OGDI DataLab.  Ce paramètre apparaît deux fois dans les fichiers *ServiceConfiguration.\*.cscfg*,une fois pour chacun des deux rôles *DataBrowser.WebRole* et *DataBrowser.WorkerRole* du projet Windows Azure.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**; AccountKey=**[StorageKey]** |
| serviceUri | URL du point de terminaison d’un service de données permettant de consommer les informations publiques.  Ce paramètre apparaît deux fois dans les fichiers *ServiceConfiguration.\*.cscfg*,une fois pour chacun des deux rôles *DataBrowser.WebRole* et *DataBrowser.WorkerRole* du projet Windows Azure.  La valeur par défaut est la suivante : [*http://data.govdata.eu:8080/v1/*](http://data.govdata.eu:8080/v1/) |
| pathDTD | 1. Chemin relatif vers le fichier *Digital Talking Book Document Type Definition*. Ce fichier permet de rendre compatible un texte avec le format DAISY. 2. Ce paramètre apparaît deux fois dans les fichiers *ServiceConfiguration.\*.cscfg*,une fois pour chacun des deux rôles *DataBrowser.WebRole* et *DataBrowser.WorkerRole* du projet Windows Azure. 3. La valeur par défaut est la suivante : *XMLValidation\dtbook-2005-3.dtd* 4. Ce paramètre ne doit pas être modifié. |
| IsAnalytics | Permet d’activer ou de désactiver la partie informations analytiques du site web. Il est à noter que l’activation de la partie informations analytiques peut engendrer des ralentissements lorsque le nombre de visites est important.  0 : Désactivé. Valeur par défaut   1. 1 : Activé |
| RecaptchaPrivateKey | 1. Clé privée pour le captcha. Le jeu de clés publique/privée doit être obtenu via le service reCaptcha à l’adresse <http://recaptcha.net/whyrecaptcha.html>. |
| RecaptchaPupblicKey | 1. Clé publique pour le captcha. Le jeu de clés publique/privée doit être obtenu via le service reCaptcha à l’adresse <http://recaptcha.net/whyrecaptcha.html>. |
| SmtpSettings | 1. Paramétrage SMTP (*Simple Mail Transfer Protocol*) pour l’envoi de message. Le paramétrage est structuré comme suit : 2. <hôte>,[<port>],[<ssl>],<de>,[<utilisateur>],[<mot de passe>] 3. où :  * <*hôte*> est le nom FQDN DNS du relai/serveur STMP, * <*port*> le port utilisé, * <*ssl*> est la chaîne de caractère « *ssl* » si la communication doit reposer sur un canal SSL/TLS chiffré et une chaîne de caractères vide dans le cas contraire, * <*de*> l’adresse de l’émetteur du mèl, * et <*utilisateur*>/<*mot de passe*> les crédentités pour l’authentification SMTP.  1. Les paramètres entre cochets ([]) sont optionnels ; les virgules (,) utilisées comme séparateur ne le sont pas. |
| UseCache | 1. Permet d’activer ou de désactiver la fonction de Caching sur Azure.   0 : Désactivé. Valeur par défaut   1. 1 : Activé |
| Microsoft.WindowsAzure.Plugins.Caching.CacheSizePercentage | 1. Taille du cache. C’est la quantité maximale de mémoire qui peut être utilisée par le cluster de cache sur les instances de l’ordinateur virtuel. 2. La valeur par défaut est 30. |
| Microsoft.WindowsAzure.Plugins.Caching.ConfigStoreConnectionString | Chaine de connexion pointant vers un compte de stockage Windows Azure. Généralement celui gérant la configuration du Kit OGDI DataLab.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**; AccountKey=**[StorageKey]**  En configuration Local, la valeur doit être :  UseDevelopmentStorage=true |

Les paramètres des rôles *Databrowser.WebRole* et *DataBrowser.WorkerRole* peuvent être modifiés directement depuis le projet *DataBrowser.Cloud*.

Pour ce faire, procédez comme suit :

1. Développez le dossier **Roles** du projet *DataBrowser.Cloud* dans l'Explorateur de solutions, faites un clic droit sur le projet *DataBrowser.WebRole* (ou le projet *DataBrowser.WorkerRole*) et puis sélectionnez **Propriétés**.
   * 1. 
2. Sélectionnez l’onglet **Paramètres.**



1. Modifiez le ou les paramètres souhaités.
2. Appuyez sur les touches CTRL+S pour sauvegarder la ou les modifications de la configuration.

#### Paramètres du fichier Web.Config du projet DataBrowser.WebRole

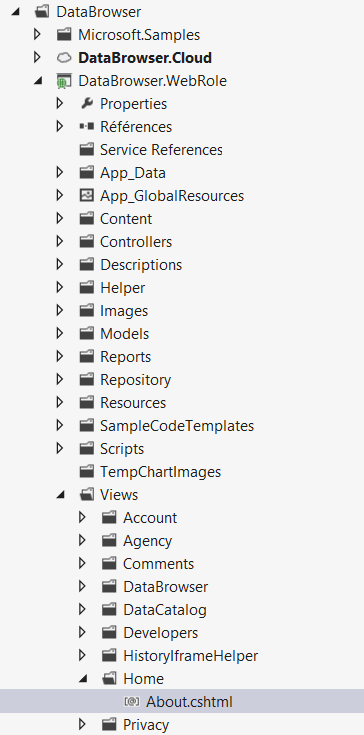
Les fichiers *Web.Config* et Web.Release.Config contiennent les paramètres suivants :

| **Paramètre** | **Description** |
| --- | --- |
| DataConnectionString | Chaine de connexion pointant vers le compte de stockage Windows Azure gérant la configuration du kit OGDI DataLab.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**; AccountKey=**[StorageKey]** |
| DiagnosticsConnectionString | Chaine de connexion pointant vers le compte de stockage Windows Azure gérant la configuration du Kit OGDI DataLab.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**; AccountKey=**[StorageKey]** |
| serviceUri | URL du point de terminaison d’un service de données permettant de consommer les informations publiques.  Ce paramètre apparaît deux fois dans les fichiers *ServiceConfiguration.\*.cscfg*,une fois pour chacun des deux rôles *DataBrowser.WebRole* et DataBrowser*.WorkerRole* du projet Windows Azure.  La valeur par défaut est la suivante : [*http://data.govdata.eu/v1/*](http://data.govdata.eu/v1/) |
| IsAnalytics | Permet d’activer ou de désactiver la partie informations analytiques du site web. Il est à noter que l’activation de la partie informations analytiques peut engendrer des ralentissements lorsque le nombre de visites est important.  0 : Désactivé. Valeur par défaut  1 : Activé |
| pathDTD | 1. Chemin relatif vers le fichier *Digital Talking Book Document Type Definition*. Ce fichier permet de rendre compatible un texte avec le format DAISY. 2. Ce paramètre apparaît deux fois dans les fichiers *ServiceConfiguration.\*.cscfg*,une fois pour chacun des deux rôles *DataBrowser.WebRole* et *DataBrowser.WorkerRole* du projet Windows Azure. 3. La valeur par défaut est la suivante : *XMLValidation\dtbook-2005-3.dtd* 4. Ce paramètre ne doit pas être modifié. |
| RecaptchaPrivateKey | 1. Clé privée pour le captcha. Le jeu de clés publique/privée doit être obtenu via le service reCaptcha à l’adresse <http://recaptcha.net/whyrecaptcha.html>. |
| RecaptchaPupblicKey | 1. Clé publique pour le captcha. Le jeu de clés publique/privée doit être obtenu via le service reCaptcha à l’adresse <http://recaptcha.net/whyrecaptcha.html>. |
| SmtpSettings | 1. Paramétrage SMTP (*Simple Mail Transfer Protocol*) pour l’envoi de message. Le paramétrage est structuré comme suit : 2. <hôte>,[<port>],[<ssl>],<de>,[<utilisateur>],[<mot de passe>] 3. où :  * <*hôte*> est le nom FQDN DNS du relai/serveur STMP, * <*port*> le port utilisé, * <*ssl*> est la chaîne de caractère « *ssl* » si la communication doit reposer sur un canal SSL/TLS chiffré et une chaîne de caractère vide dans le cas contraire, * <*de*> l’adresse de l’émetteur du mèl, * et <*utilisateur*>/<*mot de passe*> les crédentités pour l’authentification SMTP.  1. Les paramètres entre cochets ([]) sont optionnels ; les virgules (,) utilisées comme séparateur ne le sont pas. |

### A propos des vues proposées et de la localisation

Le site Web frontal du kit de développement logiciel interactif ASP.NET MVC 3 est entièrement localisé de façon à s’adapter dynamiquement à la culture de l’internaute.

Ceci s’applique à l’ensemble des vues proposées par le projet *DataBrowser.WebRole*. Prenons à titre d’illustration la page d’accueil dont la vue associée correspondant au fichier *About.cshtml* situé dans le sous dossier *Views\About*.



On peut voir au début du fichier le code suivant :

@Html.Raw(HomeResources.About.AboutPageHeading)

@Html.Raw(HomeResources.About.AboutPageContent)

Ce code tire parti de la localisation en ASP.NET qui est basée sur des fichiers de ressources .*resx*.

En observant l’arborescence du projet *DataBrowser.WebRole*, il convient de noter la présence d’un dossier *Resources*contenant une hiérarchie de sous-dossiers. Cette hiérarchie reprend la hiérarchie de premier niveau du projet. Ainsi, le sous-dossier *Resources\Views\About* vient en correspondance du dossier précédent *Views\About*.

Ce sous-dossier contient les fichiers de ressources relatives à la page/vue considérée.

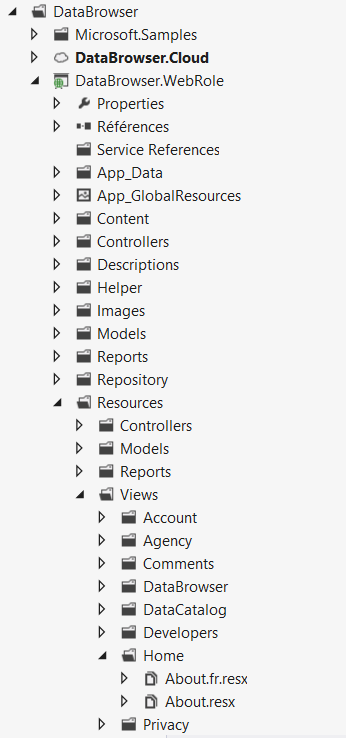
Ces fichiers de ressources respectent, pour pouvoir obtenir au final des contenus localisés, une convention de nommage, à savoir <*nom de fichier*>.<*culture*>.resx, où <*nom de fichier*> est le nom du fichier de la page/vue considérée et <*culture*> sert à indiquer la culture du fichier de ressource.

Il existe deux types de culture :

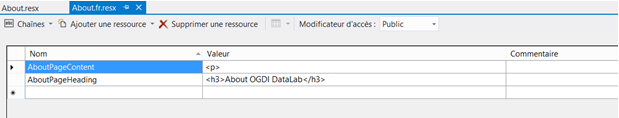
1. Les cultures dites neutres (aussi appelées invariantes). Elles contiennent seulement un code identifiant un langage, par exemple *fr*, *en*, etc.
2. Les cultures dites concrètes qui contient à la fois un code identifiant le langage et aussi un code identifiant la région, par exemple *fr-FR*, ou *fr-CA*, *en-UK* ou *en-US*.

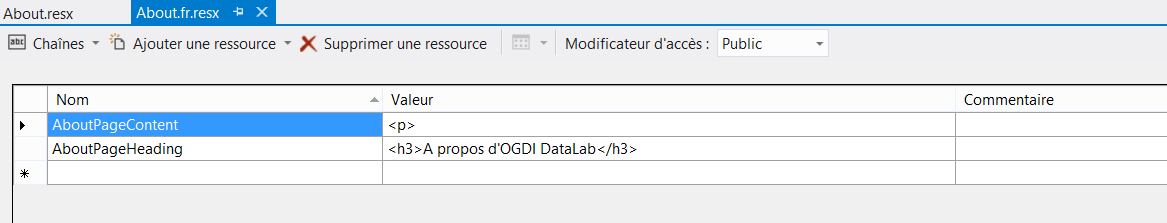
Dans le cas présent, le kit de démarrage propose systématiquement :

1. Un fichier de ressource par défaut, par exemple *About.resx* dans notre illustration,
2. Un fichier de ressource pour la langue française, par exemple *About.fr.resx* dans notre illustration.



Si l’on considère à présent le contenu de ces fichiers de ressources, on peut constater que ces derniers contiennent des paramètres sous forme clé/valeur. On retrouve les mêmes clés à l’identique dans les deux fichiers, seules les valeurs sont traduites.



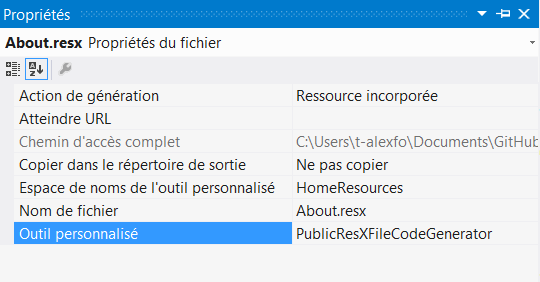


Remarque : Pour pouvoir accéder à ces fichiers de ressources, la valeur de la liste déroulante Modificateur d’accès doit être Public. Si la valeur est Internal, le fichier de ressource considéré sera alors invisible du reste du code. La raison en est que les fichiers de ressources sont compilés dans des assemblages satellites à l’assemblage principal qui contient le site Web frontal. Dès lors, un modificateur d’accès de type Internal pour les classes d’accès aux ressources générées automatiquement par Visual Studio n’est donc pas adapté.

Remarque : Il convient également de s’assurer, au niveau des propriétés du fichier de ressources que :

* L’attribut **Outil personnalisé** soit égal à *PublicResxFileCodeGenerator* ; ce qui permet de générer des classes de ressources fortement typées publiques en accord avec le modificateur d’accès précédent.
* L’attribut **Espace de noms de l’outil personnalisé** comporte un nom « *user-friendly* » afin de bénéficier d’un espace de noms facilement identifiable quand nous irons pointer vers les ressources dans notre code C#.

Ainsi, on bénéficie directement de la syntaxe *<Espace de noms de l’outil personnalisé>.<nom de fichier>.<clé>* pour aller chercher une ressource particulière, par exemple *HomeResources.About.AboutPageHeading* dans notre illustration.



Compte tenu de ces éléments, se pose maintenant la question « Comment le site Web va-t-il identifier la langue du client et donc se servir du fichier de ressource adéquat pour rendre le contenu approprié ? »

La réponse se trouve dans le fichier *global.asax* du site Web frontal. Ce fichier unique contient une classe héritant de la classe **HttpApplication** de l’espace nom **System.Web**. Cette classe permet de définir toutes les propriétés et les évènements qui sont communs à l’ensemble de l’application ASP.NET MVC 3.

Dans le cas présent, l’évènement qui nous intéresse est **AcquireRequestState**. Cet évènement se déclenche quand ASP.NET acquiert l’état courant associé à la requête courante, par exemple l’état de la session. Cet évènement s’avère donc particulièrement bien approprié pour aller chercher un contenu localisé.

Dans la classe héritant de **HttpApplication** dans le fichier *global.asax*, on peut donc définir une méthode ***Application\_AcquireRequestState*** qui va s’abonner à l’évènement **AcquireRequestState** et analyser la requête http du navigateur client afin d’en retirer, au niveau de ses en-têtes, la culture attendue en retour. Une fois la culture récupérée sous forme de chaine de caractères, il ne reste plus qu’à créer un objet de type **CultureInfo** et de faire pointer les propriétés **CurrentCulture** et **CurrentUICulture** du thread courant vers cet objet. Et le tour est joué !

protected void Application\_AcquireRequestState(object sender, EventArgs e)

{

if (HttpContext.Current.Request.UserLanguages != null)

{

var culture = HttpContext.Current.Request.UserLanguages[0];

if (!string.IsNullOrEmpty(culture))

{

CultureInfo ci = new CultureInfo(culture);

Thread.CurrentThread.CurrentCulture = ci;

Thread.CurrentThread.CurrentUICulture = ci;

}

}

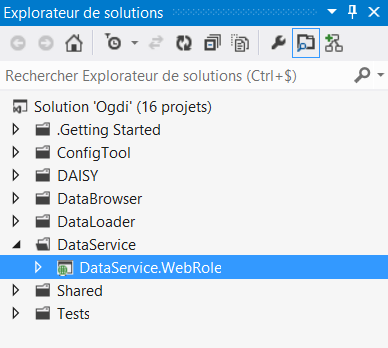
}

Nous vous invitons vivement à reprendre cette approche ainsi décrite pour toute personnalisation/extension du kit de démarrage OGDI DataLab.

## Dossier DataService (Service de données)

Cette seconde composante correspond au Service de données et aux différents services Web RESTful ainsi proposés par le socle. Le document Déployer le Kit de démarrage OGDI DataLab dans Windows Azure précise les modalités de publication du Service de données dans Windows Azure.

### Organisation projet

* 1. Le Service de données se compose du projet suivant :
  2. 

| **Projet** | **Description** |
| --- | --- |
| WebRole | Ce projet expose les données stockées dans les Tables Windows Azure. Les données sont exposées via le protocole OData. |

### Eléments de configuration

L’ensemble des éléments de configuration du Service de données est regroupé dans le projet *DataBrowser.Cloud* au niveau des fichiers de configuration de service *ServiceConfiguration.\*.cscfg* du projet Windows Azure *Services* comme expliqué précédemment.

Cependant il est possible de déployer séparément l’application DataService.WebRole en rajoutant le projet Windows Azure DataService.Cloud dans le dossier DataService et en supprimant la configuration de celui-ci dans le fichier *ServiceDefinition.csdef* du projet *DataBrowser.Cloud*.

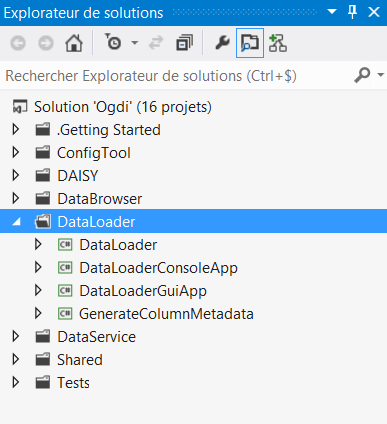
Le fichier *ServiceConfiguration.cscfg* du projet ajouté contient les paramètres suivants :

| **Paramètre** | **Description** |
| --- | --- |
| DataConnectionString | Chaine de connexion pointant vers le compte de stockage Windows Azure gérant la configuration du kit OGDI DataLab.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**; AccountKey=**[StorageKey]** |
| DiagnosticsConnectionString | Chaine de connexion pointant vers le compte de stockage Windows Azure gérant la configuration du kit OGDI DataLab.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**; AccountKey=**[StorageKey]** |

## Dossier DataLoader (Utilitaires de chargement de données)

Cette troisième composante correspond aux utilitaires de chargement d’informations publiques. Le document Mettre à disposition des informations publiques avec le Kit de démarrage OGDI DataLab précise les modalités d’utilisation de ces utilitaires.

### Organisation projet

* 1. Cette dernière composante est composée des 4 projets suivants :
  2. 

| **Projet** | **Description** |
| --- | --- |
| DataLoader | Ce projet de type bibliothèque contient toutes les classes permettant de télécharger un fichier de données texte au format CSV ou KML vers les Tables Windows Azure. |
| DataLoaderConsoleApp | Ce projet de type application en ligne de commande s’appuie sur la bibliothèque *DataLoader* pour télécharger un fichier de données texte au format CSV ou KML. Comme il s’agit d’une application en ligne de commande, il est possible de s’en servir pour créer un script automatisé. |
| DataLoaderGuiApp | Ce projet de type application graphique permet d’effectuer les mêmes opérations qu’avec l’application en ligne de commande précédente *DataLoaderConsoleApp* mais, cette fois-ci, avec des interactions avec l’utilisateur. |
| GenerateColumnMetadata | Ce projet de type application en ligne de commande peut être utilisé pour mettre à jour la génération RDF des ensembles de données précédemment stockés dans les tables de stockage. |

### Eléments de configuration

Les utilitaires de chargement de données qu’il s’agisse de la version graphique ou en ligne de commande reposent sur la configuration regroupée au niveau du fichier de configuration *App.config* des projets *DataLoaderGuiApp* (version graphique) et *DataLoaderConsoleApp* (version en ligne de commande).

Le fichier *App.config* contient les paramètres suivants :

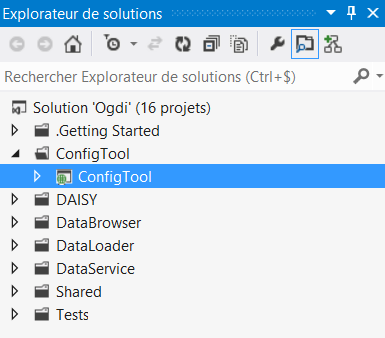
| **Paramètre** | **Description** |
| --- | --- |
| DataConnectionString | Chaine de connexion pointant vers le compte de stockage Windows Azure gérant les données du kit OGDI DataLab.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**;AccountKey=**[Storagekey]** |

En ce qui concerne l’utilitaire de mise à jour de la génération RDF (*GenerateColumnMetadata*), ce dernier dispose lui aussi d’un fichier *App.config* avec les paramètres suivants :

| **Paramètre** | **Description** |
| --- | --- |
| DataConnectionString | Chaine de connexion pointant vers le compte de stockage Windows Azure gérant la configuration du kit OGDI DataLab.  La valeur par défaut est la suivante :  DefaultEndpointsProtocol=https;AccountName=**[StorageName]**;AccountKey=**[StorageKey]** |
| DefaultDescription | Champ représentant la description par défaut qui sera associée avec les métadonnées des colonnes. Sa valeur peut être changée, mais les deux paramètres {0} et {1} sont obligatoires. {0} représente le nom de la colonne et {1} le nom de l’ensemble de données.  La valeur par défaut est la suivante :  It represents the {0} of a/an {1}. |

## Dossier ConfigTool

Ce dossier contient un sous-projet éponyme :



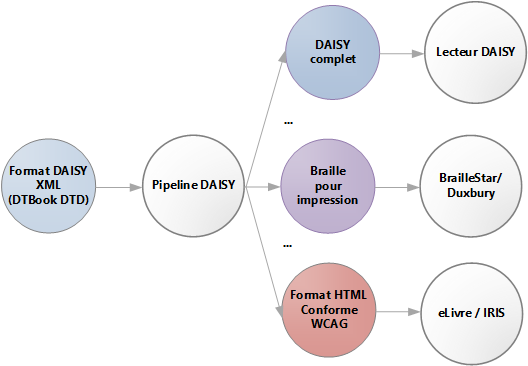
| **Projet** | **Description** |
| --- | --- |
| ConfigTool | Ce projet de type application Web met à disposition un simple outil de configuration pour le compte de stockage Windows Azure contenant la configuration du kit OGDI DataLab. |

## Dossier DAISY

Le format DAISY (*Digital Access Information SYstem*) XML est issu du [consortium DAISY](http://www.daisy.org)[[68]](#footnote-68) qui regroupe 70 associations (notamment l’Association BrailleNet en France) dont le but est, depuis 1996, de rendre tous les documents publiés accessibles aux personnes souffrant de déficiences visuelles, de dyslexie, ou encore de la maladie de Parkinson. Autant de handicaps qui rendent difficiles la lecture, la saisie sur un clavier et même la simple manipulation d'un livre.

Le format Daisy XML (DTBook DTD) est la base fondatrice du [standard DAISY 3/NISO (ANSI/NISO Z39.86 Specifications for the Digital Talking Book)](http://www.daizy.org/z3986)[[69]](#footnote-69), utilisée dans le monde entier pour lire et publier des contenus multimédias et constituer des livres numériques audio.

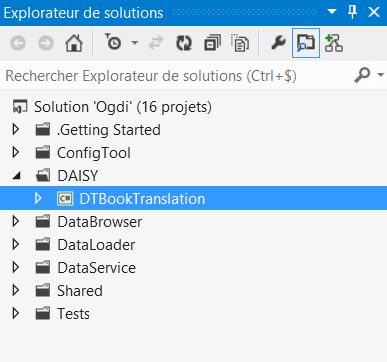
En effet, ce format pivot peut ensuite être converti dans de multiples formats au travers notamment du [Pipeline DAISY](http://www.daisy.org/projects/pipeline)[[70]](#footnote-70), une suite gratuite proposée par le consortium DAISY.



Les fichiers résultants sont ensuite non seulement lus par des synthétiseurs vocaux pour créer une forme de « braille électronique » mais, grâce au standard DAISY, les utilisateurs peuvent naviguer rapidement par en-tête ou numéro de page, utiliser des index et références, le tout avec des enregistrements audio et du texte classés et synchronisés.

Pour de plus amples informations, nous vous invitons à visionner/écouter le [film DAISY rend la lecture plus facile](http://www.microsoft.com/france/accessibilite/video/20080513-daisy-makes-reading-easier.asx)[[71]](#footnote-71).

Ce dossier contient un sous projet éponyme :

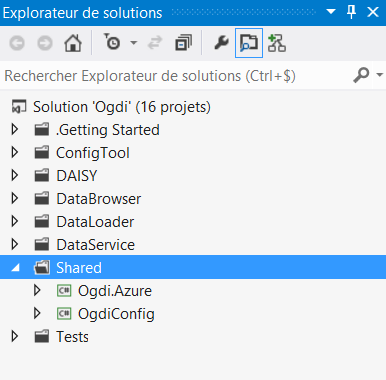


| **Projet** | **Description** |
| --- | --- |
| DAISY | Ce projet de type bibliothèque permet de retourner un ensemble de données stockées dans Windows Azure au format DAISY et ce par le biais de l’application d’une transformation XSL (*Extensible Stylesheet Language*).  La transformation repose sur les fichiers *ConvertToDTBook.xslt* et *Shematron.xsl* situés dans le sous-répertoire *Resources* du projet.  L’assemblage résultant est référencé dans le projet *DataBrowser.WebRole*  du kit de développement logiciel interactif. |

## Dossier Shared

### Organisation projet

Ce dossier contient deux sous-projets du Kit de démarrage OGDI DataLab, à savoir *Ogdi.Azure* et *Ogdi.Config* utilisés dans les solutions précédentes :



| **Projet** | **Description** |
| --- | --- |
| Ogdi.Azure | Ce projet de type bibliothèque sert de point d’entrée vers le compte de stockage Windows Azure contenant les données de configuration du kit OGDI DataLab. Il permet notamment d’accéder et d’interagir avec les tables intitulées *AnalyticInfo* et *ViewEntries* qui servent à collecter des statistiques sur les données exposées. |
| Ogdi.Config | Ce projet de type bibliothèque sert également de point d’entrée vers le compte de stockage Windows Azure contenant les données de configuration du kit OGDI DataLab mais permet d’accéder seulement à la table intitulée *AvailableEndpoints* ; celle-ci permet de lister les URL « racines » relatives aux différents ensembles de données définies dans le compte de stockage des données du kit OGDI DataLab. |

### Dépendances

Par ailleurs, ce dossier *Shared* est destiné à recevoir également les diverses dépendances du Kit de façon à pouvoir compiler la solution dans son ensemble.

A ce titre, le tableau ci-dessous précise l’ensemble des composants tiers utilisés dans la solution OGDI DataLab. Sauf indication contraire, ces composants doivent être copiés dans le dossier *Shared*. Copiez uniquement les fichiers spécifiés ci-dessous. Veuillez-vous assurer, par ailleurs, de revoir préalablement et d'accepter les termes de licence de chaque composant.

|  |  |  |
| --- | --- | --- |
| **Assembly/Composant** | **Emplacement de destination** | **Remarques** |
| [Newtonsoft.Json.dll](http://json.codeplex.com/Release/ProjectReleases.aspx?ReleaseId=25040) | %INSTALL\_ROOT%\Source\Shared\ |  |
| [CSharpFormat.dll](http://www.manoli.net/csharpformat/CSharpFormat.zip) | %INSTALL\_ROOT%\Source\Shared\ | Nécessite de reconstruire l’assemblage |
| [System.Threading.dll](http://go.microsoft.com/fwlink/?LinkId=182996) | %INSTALL\_ROOT%\Source\Shared\ | Après installation du paquet, l’assemblage à copier sous le dossier *Reactive Extensions\v1.0.2563.0\Net35folder* du répertoire racine d'installation par défaut. Ce dernier est *%PROGRAM FILES%\Microsoft Cloud Programmability*. |
| [System.Web.DataVisualization et System.Web.DataVisualizationDesign](http://www.microsoft.com/downloads/details.aspx?FamilyId=130F7986-BF49-4FE5-9CA8-910AE6EA442C&displaylang=en) | Référencé à partir de l'emplacement d'installation par défaut | Suppose d’installer le paquet .*msi* |
| [ASP.NET MVC 3.0](http://go.microsoft.com/fwlink/?LinkID=215693) | Référencé à partir de l'emplacement d'installation par défaut | Suppose d’installer le paquet .*msi* |
| [jquery.clipboard.min.js et jquery.clipboard.swf](http://bradleysepos.com/projects/jquery/clipboard/release/jquery.clipboard-2.0.1.zip) | %INSTALL\_ROOT%\Source\DataBrowser\  DataBrowser.WebRole\Scripts |  |
| [LumenWorks.Framework.IO.dll](http://www.codeproject.com/KB/database/CsvReader.aspx) | %INSTALL\_ROOT%\Source\Shared\ | Nécessite de suivre les instructions pour télécharger l’assemblage |

### Résolution des dépendances avec NuGet

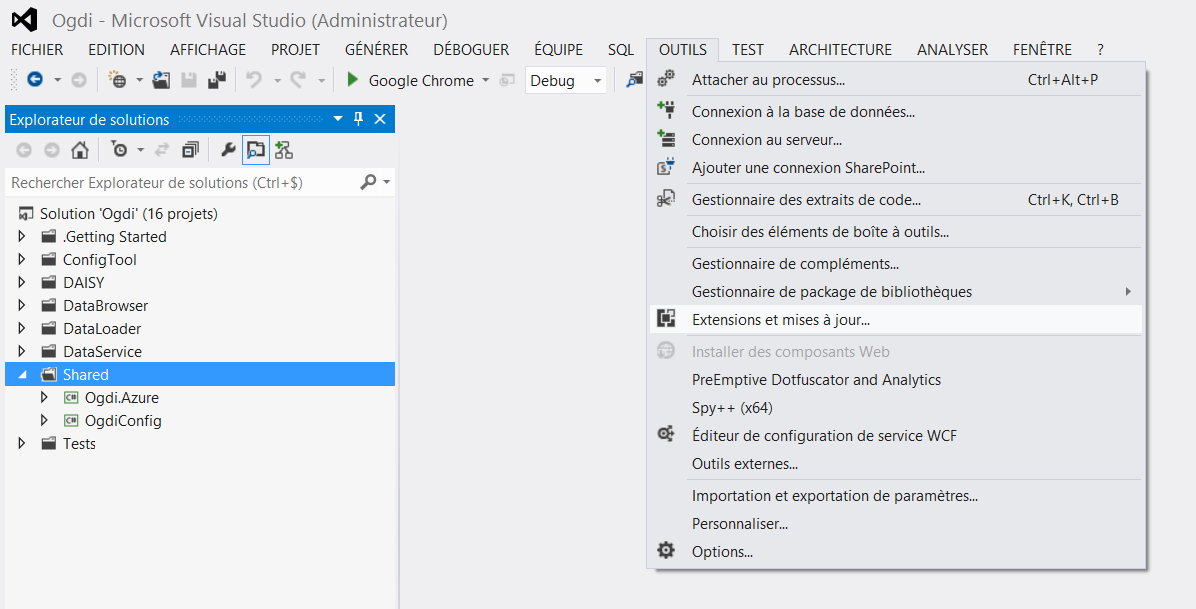


Les dépendances précédentes peuvent être résolues avec NuGet. NuGet est une extension qui vient s’ajouter à l’environnement Visual Studio 2012 et qui permet d’intégrer en quelques clics des bibliothèques et des outils sous licence libre à vos projets sous forme de paquets (package) installables.

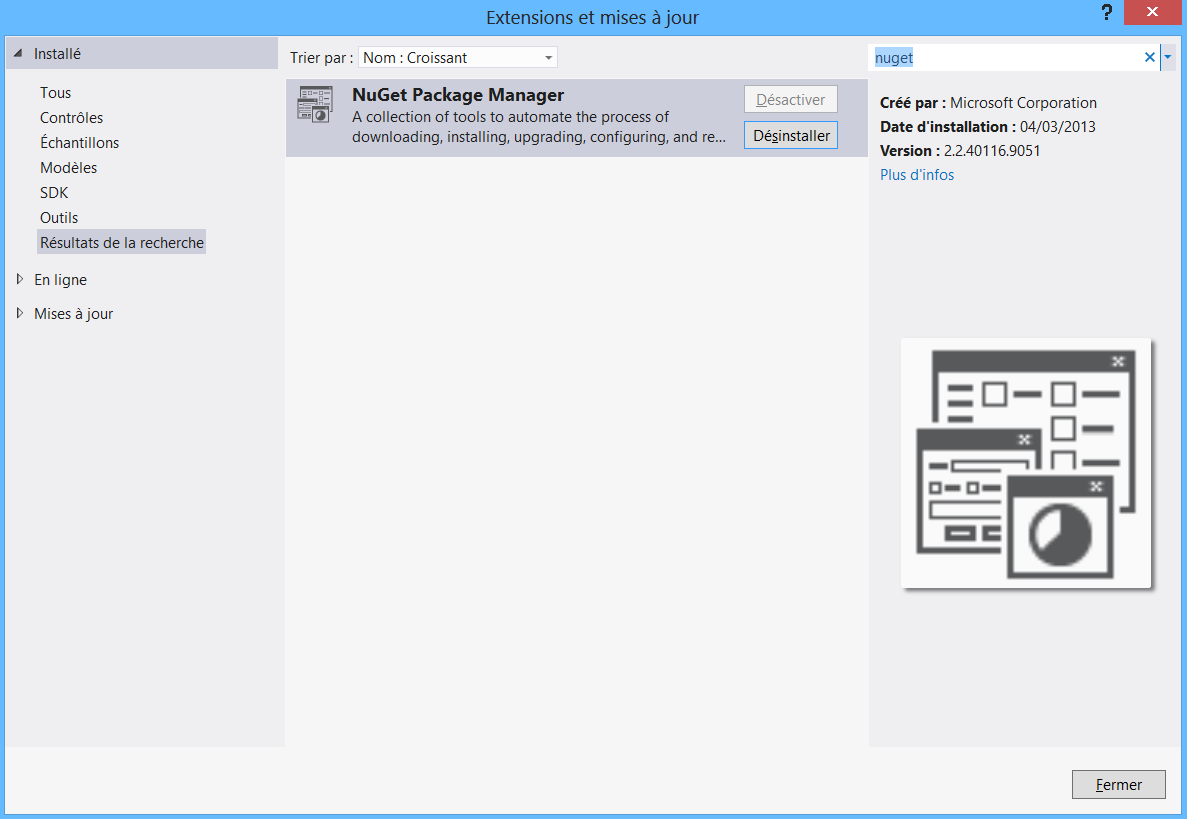
Les dépendances des projets de la solution sont disponibles sous forme de paquets depuis la [galerie en ligne](http://nuget.org/List/Packages)[[72]](#footnote-72) (online gallery) de NuGet.

Pour installer l’extension NuGet, procédez comme suit :

1. Lancez l’environnement de développement Visual Studio 2012.
2. Depuis la barre d’outils, cliquez sur **Outils**, puis **Extensions et mises à jour..**.



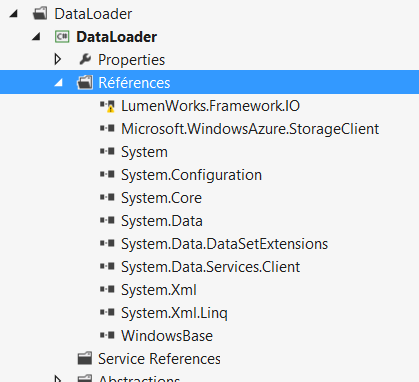
1. Une fois la fenêtre **Extensions et mises à jour**ouverte, sélectionnez l’onglet **En ligne**, puis, dans la barre de recherche, tapez simplement « *nuget* ».



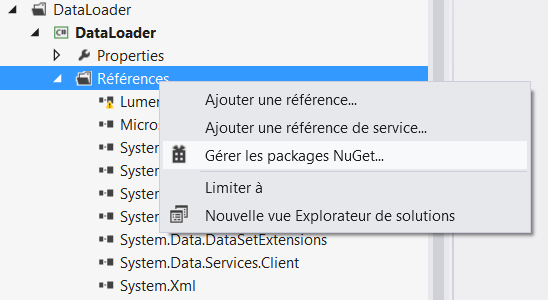
1. Cliquez ensuite sur **NuGet Package Manager** puis sur le bouton **Télécharger**. L’installation se lance alors. A l’issue de l’installation, un redémarrage de l’environnement est nécessaire.

Pour résoudre les dépendances précédentes des projets de la solution *Ogdi.sln*, procédez comme suit :

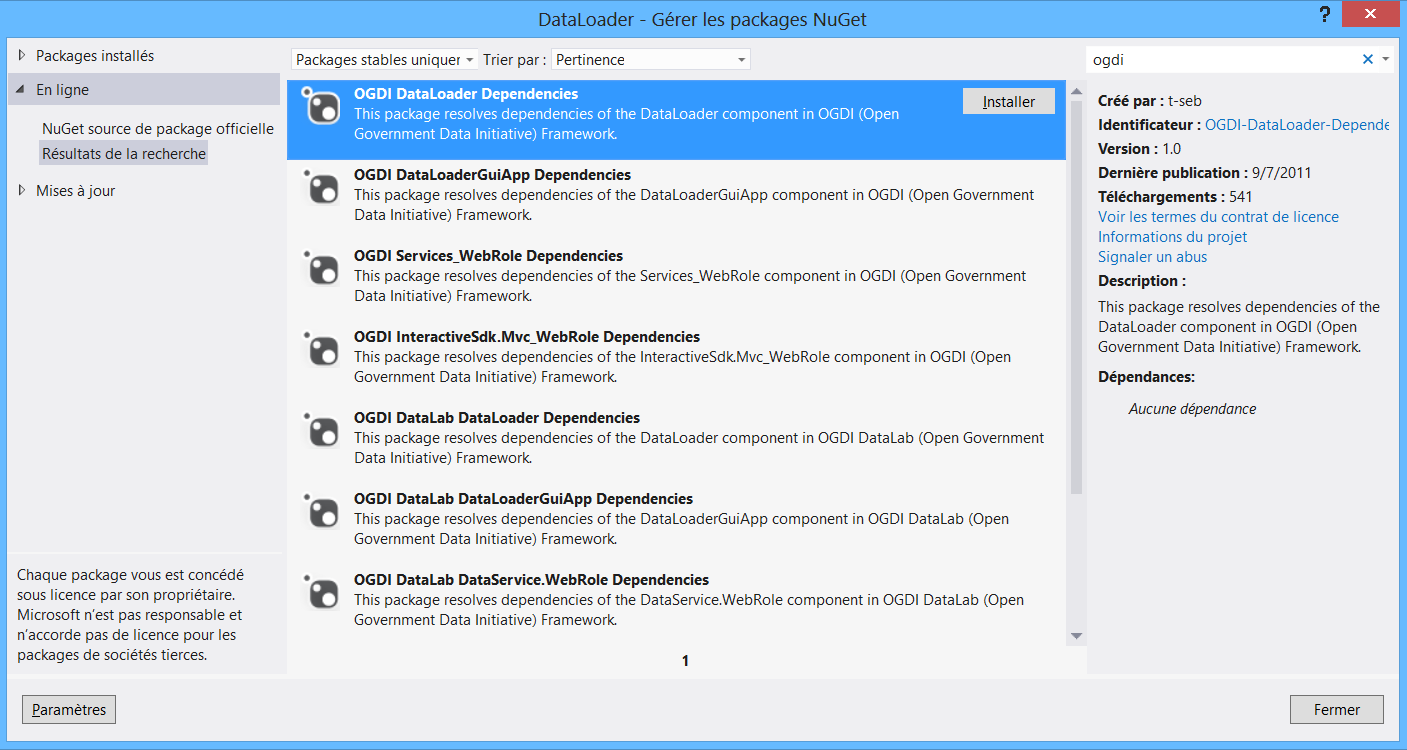
1. Ouvrez Windows Explorer avec des privilèges administratifs et placez-vous dans le sous-répertoire *Source* du répertoire d’installation du Kit de démarrage OGDI DataLab.
2. Double-cliquez sur l’icône du fichier *Ogdi.sln* situé à la racine du sous-répertoire *Source* pour ouvrir le fichier dans l’environnement Visual Studio 2012.
3. Depuis le volet **Explorateur de solutions**, sélectionnez par exemple le projet **DataLoader** et dépliez l’onglet **Références**. Vous pouvez constater qu’il y a une dépendance non-résolue vers la bibliothèque *LumenWorks.Framework.IO*.



1. Maintenant, faites un clic-droit sur **Références**, et cliquez sur **Gérer les packages NuGet…**



1. Dans la fenêtre **Gérer les packages NuGet** qui s’ouvre, sélectionnez l’onglet **En ligne** puis tapez ensuite « *ogdi* » dans la barre de recherche.

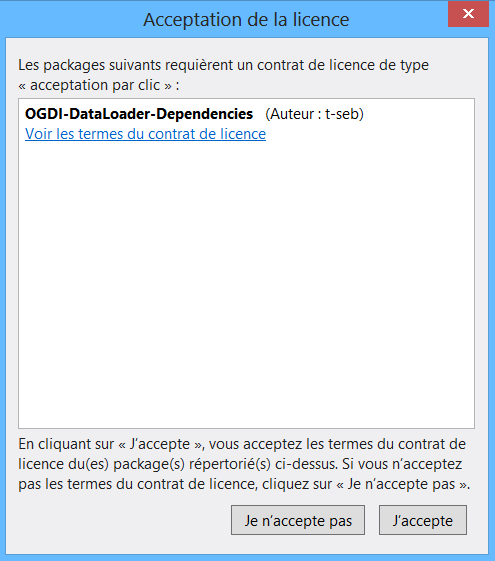


1. Comme vous pouvez le constater, 4 paquets sont disponibles et préfixés par OGDI DataLab, chacun installant les dépendances des projets de la solution OGDI DataLab dont ils portent le nom. Par exemple, dans le cas présent, nous cherchons à résoudre les dépendances du projet **DataLoader*.*** Sélectionnez donc le paquet **OGDI DataLab DataLoader Dependencies**, cliquez sur **Installer**.
2. Dans la boîte de dialogue Acceptation de la licence, cliquez sur le lien Voir les termes du contrat de licence.

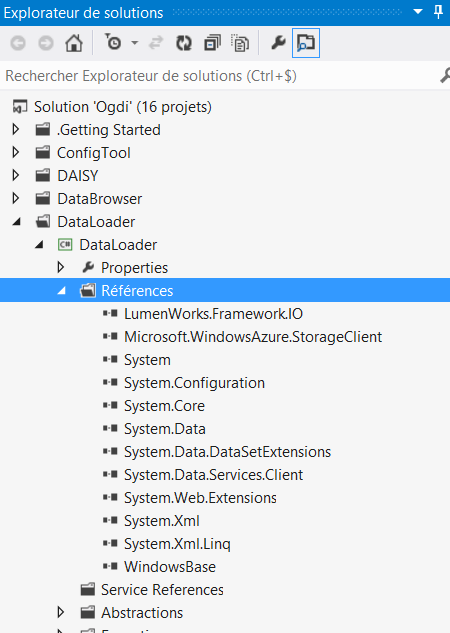
Remarque importante : Les composants logiciels contenus dans le paquet sont publiés sous leur différent contrat de licence respectif. Vous devez donc impérativement lire les termes de chaque contrat de licence qui s'applique et accepter les termes de chaque contrat avant d'accepter l'installation de ce paquet. Les liens afférents sont précisés pour faciliter cette opération.

Remarque importante : Il apparaît au total 8 paquets : 4 préfixés par *OGDI DataLab* et 4 autres seulement préfixés par *OGDI* ces derniers étant les paquets de l’ancienne version du kit de démarrage OGDI v2a. La version actuelle est le kit de démarrage OGDI DataLab v6. Il conviendra donc d’utiliser les paquets préfixés par OGDI DataLab dans votre solution.

1. Si vous acceptez les termes de chaque contrat qui s’applique, cliquez ensuite sur **J’accepte**afin matérialiser cette acceptation.



1. Cliquez maintenant sur **Fermer** dans la fenêtre **Gérer les packages NuGet**. Au niveau de la liste des références du projet, vous pouvez observer que la précédemment manquante bibliothèque *LumenWorks.Framework.IO*a été téléchargée et que la dépendance a donc été résolue.



La même opération peut être réalisée pour les projets *DataLoaderGuiApp*, *DataBrowser.WebRole* et Data*Service.WebRole* pour résoudre toutes les dépendances vers les bibliothèques sous licence libre utilisée dans le cadre du kit de démarrage OGDI DataLab.

# Compilation des éléments du Kit de démarrage OGDI DataLab

Les différentes composantes du Kit de démarrage OGDI DataLab peuvent être compilées en utilisant Visual Studio 2012, la commande *msbuild* en ligne de commande ou le script d’automatisation de compilation inclut. Les trois 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 du Kit de démarrage OGDI DataLab.
2. Saisissez *msbuild Ogdi.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 2012

Pour compiler une solution particulière via Visual Studio 2012, veuillez suivre ces étapes :

1. Ouvrez Windows Explorer avec des privilèges administratifs et placez-vous dans le sous-répertoire *Source* du répertoire d’installation du Kit de démarrage OGDI DataLab.
2. Double-cliquez sur l’icône du fichier *Ogdi.sln* situé à la racine du sous-répertoire *Source* pour ouvrir le fichier dans l’environnement Visual Studio 2012.
3. Dans le menu **Générer**, choisissez **Régénérer la 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.

## Compilation des éléments via le script d’automatisation

Pour compiler l’ensemble de la solution via le script, veuillez suivre ces étapes :

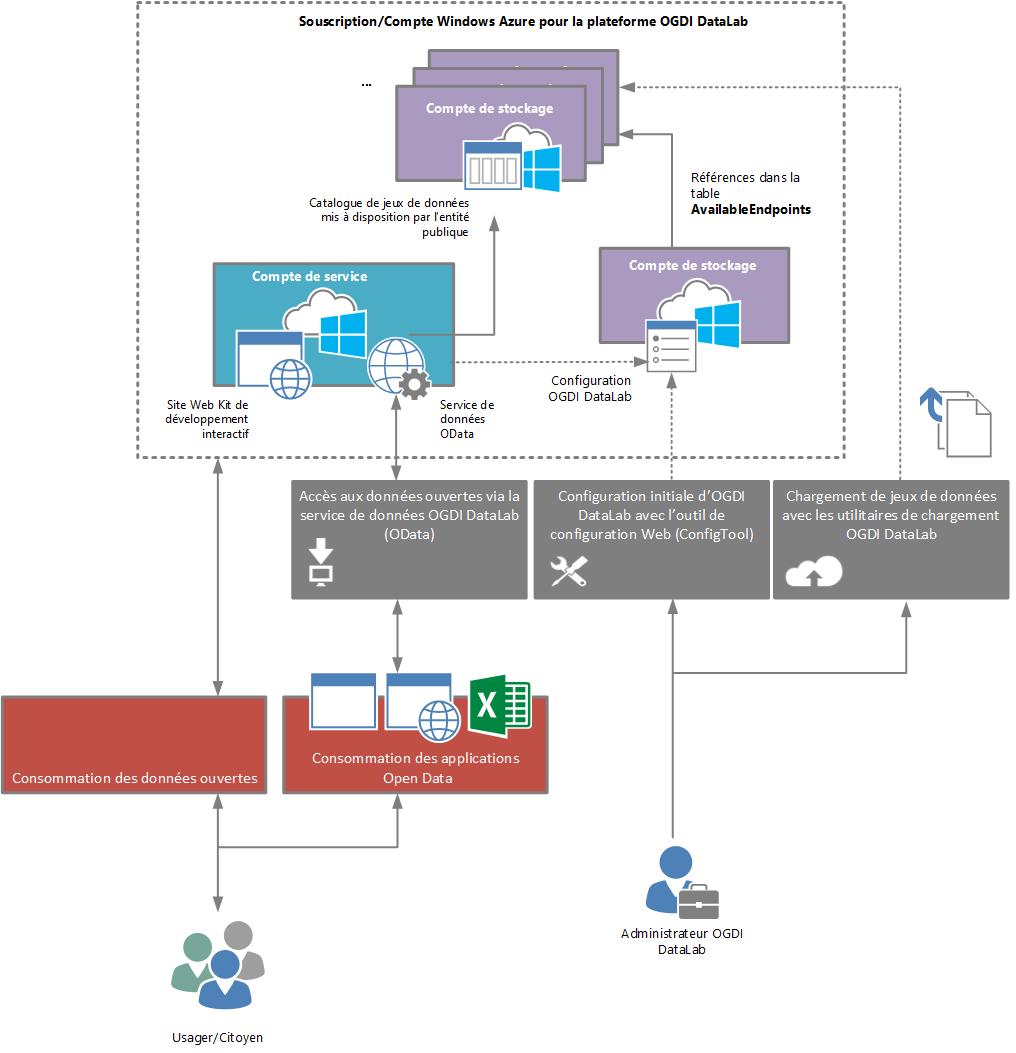
1. Ouvrez Windows Explorer et placez-vous dans le sous répertoire Source du répertoire d’installation du Kit de démarrage OGDI DataLab.
2. Double-cliquez sur le fichier *Build.cmd* situé à la racine du sous-répertoire *Source* pour lancer la compilation. Les éléments relatifs à la solution sont compilés dans le répertoire *Deployment* créé lors de la compilation à la racine du dossier *Source*.

# Mise en œuvre des éléments du Kit de démarrage OGDI DataLab dans Windows Azure

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

1. De récupérer, de s’approprier et de mettre en œuvre très rapidement et à moindre coût une solution ouverte de bout-en-bout de publication d’informations publiques via Windows Azure,
2. Et de publier et d’exposer sur ce socle les ensembles de données que l’entité souhaite mettre à disposition des usagers et citoyens.

La figure suivante illustre les éléments du kit de démarrage OGDI DataLab tels que décrits précédemment dans un contexte de mise en œuvre complète et opérationnelle dans Windows Azure :



Dans ce cadre, la mise en œuvre des éléments du kit de démarrage OGDI DataLab hors personnalisation particulière suppose les étapes suivantes :

1. Classification du ou des ensembles de données à rendre public en termes de type souhaité de réutilisation :

* Navigation  au sein des données brutes par un utilisateur final ;
* Analyse statistique sur des données brutes ;
* Type d'applications où les informations publiques pourraient être réutilisées.

1. Activation d’une souscription sur Windows azure. Vous pouvez accéder à Windows Azure avec un paiement à l’usage, sans engagement, ou bien des forfaits comme décrit à l’adresse <http://www.microsoft.com/france/windows-azure/Offres.aspx>.
2. Déploiement de la solution dans Windows Azure conformément au document Déployer le Kit de démarrage OGDI DataLab dans Windows Azure. Ce dernier propose une procédure guidée étape par étape pour la mise en œuvre opérationnelle du socle technologique proposé par le Kit de démarrage OGDI DataLab.
3. Publication du ou des ensembles de données en suivant les instructions du document Mettre à disposition des informations publiques avec le Kit de démarrage OGDI DataLab pour notamment la préparation des informations publiques de type texte dans un format prêt pour le téléchargement.

L’association d’informations en format binaire y est également abordée.

1. Prise de contact avec les créateurs d’applications potentiellement intéressés par la prise en compte des informations publiques ainsi disponibles dans leurs applications ou dans le développement de nouvelles applications qui utilisent ces informations publiques.

L’éventail des outils de développement disponibles pour l’accès aux informations publiques et à leur consommation facilite l’implication des acteurs quels qu’ils soient.

Cette étape peut, par ailleurs, conduire à la mise en place d’un concours d’idées avec « la vertu de l’exemple pour amorcer la pompe ». Le kit de démarrage peut accompagner une telle dynamique en offrant, par exemple, un socle d’accueil dédié aux applications pour Smartphone ainsi proposées dans ce contexte.

# Pour aller plus loin

Pour de plus amples information sur les solutions proposées par Microsoft à destination du Secteur Public, nous vous invitons à consulter le [site portail Web Microsoft Secteur Public](http://www.microsoft.com/france/secteur-public/default.aspx)[[73]](#footnote-73).



Pour de plus amples informations sur l'interopérabilité technique des produits et technologies Microsoft avec des logiciels et matériels d'autres fournisseurs, nous vous invitons à consulter le [site portail Web Microsoft Open](http://www.microsoft.com/france/openness/default.aspx)[[74]](#footnote-74).

Dans le contexte de la révolution numérique en marche, du foisonnement des technologies, des innovations permanentes, etc. l’interopérabilité, c’est-à-dire la capacité des produits, logiciels et matériels à communiquer entre eux, représente un enjeu fondamental aujourd’hui: à terme, tous les produits, logiciels, matériel doivent pouvoir communiquer entre eux.

Derrière la dimension technique de l’interopérabilité, le véritable enjeu est la liberté de chacun : celle de choisir les outils qui correspondent le mieux à ses besoins, celle de pouvoir accéder à ses données au fil du temps, indépendamment de la technologie utilisée.

Avec l’[annonce majeure](http://www.microsoft.com/en-us/news/press/2008/feb08/02-21ExpandInteroperabilityPR.aspx)[[75]](#footnote-75) faite au mois de février 2008, Microsoft s’est engagé, pour garantir la pérennité des données et assurer la conformité aux standards de l’industrie informatique, la prise en charge des principales normes internationales, dans une politique d’ouverture sans précédent dans l’histoire informatique, une démarche prospective qui prouve qu’innovation, propriété intellectuelle et interopérabilité ne s’opposent pas.

Cela se traduit notamment par la mise à la disposition de la communauté des développeurs, de plus de 50 000 pages d’informations techniques relatives aux protocoles de communication et aux interfaces de nos produits-phare, et par le développement des relations collaboratives avec les différents acteurs du secteur, y compris les communautés investies dans les logiciels Open source.

Pour de plus amples informations sur les standards d'interopérabilité et normes internationales que Microsoft soutient ou auxquels Microsoft apporte sa collaboration, nous vous invitons également à consulter le portail Microsoft Open.

Par ailleurs, afin de promouvoir l’industrie du logiciel sans favoriser un modèle de développement plutôt qu’un autre, nous contribuons depuis 2009 à l’évolution du noyau Linux sous licence GPL. Nous finançons également, et mettons gratuitement à disposition, en mode open source, des logiciels permettant d’échanger des documents issus de différentes suites bureautiques (Microsoft Office, Open Office, etc.). Nous investissons également dans le projet Apache Hadoop et contribuons à améliorer dans ce cadre les environnements de développement et d’exécution d’Hadoop sur Windows Azure et Windows Server.

Pour de plus amples informations sur les contributions, le financement et la mise à disposition sous licence libre de logiciels de Microsoft, nous vous invitons également à consulter le portail Microsoft Open.

1. Site portail Web Microsoft Secteur Public : http://www.microsoft.com/france/entreprises/secteur-public/solutions-pour-secteur-public.aspx [↑](#footnote-ref-1)
2. Solution Open Data clé en main: https://mspartner.microsoft.com/fr/fr/Pages/Solutions/Open-Data-Cle-en-main.aspx [↑](#footnote-ref-2)
3. Microsoft Windows Azure : http://www.microsoft.com/france/windows-azure/ [↑](#footnote-ref-3)
4. Protocole OData : http://www.odata.org [↑](#footnote-ref-4)
5. OGDI DataLab sur GitHub : https://github.com/openlab/DataLab [↑](#footnote-ref-5)
6. Solution Open Data clé en main: https://mspartner.microsoft.com/fr/fr/Pages/Solutions/Open-Data-Cle-en-main.aspx [↑](#footnote-ref-6)
7. Blog MSDN Open Data France : http://aka.ms/OpenDataFrance/ [↑](#footnote-ref-7)
8. Microsoft Windows Azure Training Kit : http://www.microsoft.com/downloads/en/details.aspx?FamilyID=413E88F8-5966-4A83-B309-53B7B77EDF78&displaylang=en [↑](#footnote-ref-8)
9. Microsoft Web Camps Training Kit : http://trainingkit.webcamps.ms/Default.htm [↑](#footnote-ref-9)
10. Windows Azure Team Blog : http://blogs.msdn.com/b/windowsazure/ [↑](#footnote-ref-10)
11. Edition RSLN du premier trimestre 2011 Open Data et nous, et nous et nous ? : http://www.rslnmag.fr/Content/Files/ff/UERGL1YyUkN6ZHdFaTZNSFJOQS5wZGY= [↑](#footnote-ref-11)
12. Dossier Quel sens donner au monde des données ouvertes ? : http://www.rslnmag.fr/blog/dossier/l-open-data\_et-nous\_et-nous\_et-nous\_/2011/4/19/quel-sens-donner-au-monde-des-donnees-ouvertes\_/ [↑](#footnote-ref-12)
13. Conférence [L’Open Data, et nous, et nous et nous ?](http://www.rslnmag.fr/blog/theme/les-rencontres-rsln/2011/2/10/l-open-data_et-nous_et-nous_et-nous_venez-imaginer-le-monde-de-l-abondance-des-donnees_/) : http://www.rslnmag.fr/blog/theme/les-rencontres-rsln/2011/2/10/l-open-data\_et-nous\_et-nous\_et-nous\_venez-imaginer-le-monde-de-l-abondance-des-donnees\_/ [↑](#footnote-ref-13)
14. World e.gov Forum : http://wegf.org/fr/ [↑](#footnote-ref-14)
15. Définition de REST sur Wikipedia : http://en.wikipedia.org/wiki/Representational\_State\_Transfer [↑](#footnote-ref-15)
16. RFC 5023 The Atom Publishing Protocol : http://tools.ietf.org/html/rfc5023 [↑](#footnote-ref-16)
17. OGC KML : http://www.opengeospatial.org/standards/kml/ [↑](#footnote-ref-17)
18. RFC 4627 The application/json Media Type for JavaScript Object Notation (JSON) : http://tools.ietf.org/html/rfc4627 [↑](#footnote-ref-18)
19. Site Defining Safer JSON-P : http://www.json-p.org/ [↑](#footnote-ref-19)
20. RDF : http://www.w3.org/RDF/ [↑](#footnote-ref-20)
21. OGDI : http://ogdi.codeplex.com [↑](#footnote-ref-21)
22. Kit de démarrage OGDI v2a: http://www.microsoft.com/downloads/fr-fr/details.aspx?FamilyID=420837f8-25ec-47ba-b558-b6a7f64b3348 [↑](#footnote-ref-22)
23. OGDI DataLab : https://github.com/openlab/DataLab [↑](#footnote-ref-23)
24. Documentation du protocole OData : http://www.odata.org/developers/protocols [↑](#footnote-ref-24)
25. Catalogue de données Rennes Métropole : http://www.data.rennes-metropole.fr/les-donnees/catalogue/?tx\_icsopendatastore\_pi1%5buid%5d=47 [↑](#footnote-ref-25)
26. Site Introducing JSON : http://json.org [↑](#footnote-ref-26)
27. Billet OData interoperability with .NET, Java, PHP, iPhone and more : http://blogs.msdn.com/b/interoperability/archive/2010/03/16/odata-interoperability-with-net-java-php-iphone-and-more.aspx [↑](#footnote-ref-27)
28. Kits de développement logiciel OData : http://www.odata.org/developers/odata-sdk [↑](#footnote-ref-28)
29. Open Specification Promise : http://www.microsoft.com/france/interop/approche/osp.aspx [↑](#footnote-ref-29)
30. 30-day Public Review for OData, OData Atom Format and OData JSON Format, V4.0 : https://lists.oasis-open.org/archives/odata/201305/msg00091.html [↑](#footnote-ref-30)
31. Billet Interoperability Elements of a Cloud Platform Outlined at OSCON : http://blogs.msdn.com/b/interoperability/archive/2010/07/22/interoperability-elements-of-a-cloud-platform-outlined-at-oscon.aspx [↑](#footnote-ref-31)
32. Microsoft Global Foundation Services (GFS) : http://www.globalfoundationservices.com/ [↑](#footnote-ref-32)
33. Certifications ISO 27001 Microsoft, Global Foundation Services Division : http://www.bsiamerica.com/en-us/Assessment-and-Certification-services/Management-systems/Certificate-and-Client-Directory-search/Search/Search-Results/?pg=1&licencenumber=IS+533913&searchkey=companyXeqXmicrosoft [↑](#footnote-ref-33)
34. Certification ISO 27001 Windows Azure : http://www.bsiamerica.com/en-us/Assessment-and-Certification-services/Management-systems/Certificate-and-Client-Directory-search/Search/Search-Results/?pg=1&licencenumber=IS+577753&searchkey=licenceXeqX577753 [↑](#footnote-ref-34)
35. Billet Response to Question about SAS 70 Objectives : http://blogs.technet.com/b/gfs/archive/2009/06/16/response-to-question-about-sas-70-objectives.aspx [↑](#footnote-ref-35)
36. U.S.-EU Safe Harbor : http://www.export.gov/safeharbor/eu/eg\_main\_018365.asp [↑](#footnote-ref-36)
37. Centre de gestion de la confidentialité Windows Azure : http://www.windowsazure.com/fr-fr/support/trust-center/ [↑](#footnote-ref-37)
38. Réponse standard pour les demandes d'information, Sécurité et Respect de la vie privée : http://www.microsoft.com/en-us/download/details.aspx?id=26647 [↑](#footnote-ref-38)
39. Windows Azure Marketplace : https://datamarket.azure.com/ [↑](#footnote-ref-39)
40. Correctif logiciel KB971842 Correction des métadonnées WCF lors de l'hébergement derrière un programme d'équilibrage de charge : http://support.microsoft.com/kb/971842 [↑](#footnote-ref-40)
41. Correctif logiciel cumulatif KB977420 pour Windows Communication Foundation : http://support.microsoft.com/kb/977420 [↑](#footnote-ref-41)
42. Correctif logiciel KB983301 Activation de Windows Azure IntelliTrace sur les systèmes d'exploitation 32 bits : http://archive.msdn.microsoft.com/KB983301 [↑](#footnote-ref-42)
43. Module de réécriture d'URL d'IIS 7.0 : http://www.iis.net/download/URLRewrite [↑](#footnote-ref-43)
44. Windows PowerShell 2.0 : http://go.microsoft.com/fwlink/?LinkId=192992 [↑](#footnote-ref-44)
45. 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-45)
46. GitHub for Windows: http://blogs.msdn.com/b/ogdifrance/archive/2012/06/14/github-for-windows.aspx [↑](#footnote-ref-46)
47. Blog MSDN Open Data France: http://aka.ms/OpenDataFrance/ [↑](#footnote-ref-47)
48. Visual Studio 2012 Update 2 : http://www.microsoft.com/en-us/download/details.aspx?id=38188 [↑](#footnote-ref-48)
49. Visual Studio Tools for Git : http://visualstudiogallery.msdn.microsoft.com/abafc7d6-dcaa-40f4-8a5e-d6724bdb980c [↑](#footnote-ref-49)
50. Integrating and Using Github in Visual Studio 2012 : http://www.codeproject.com/Articles/581907/IntegratingplusandplusUsingplusGithubplusinplusVis [↑](#footnote-ref-50)
51. Module OGDI Field : http://drupal.org/project/ogdi\_field [↑](#footnote-ref-51)
52. ASP.NET MVC 3.0 : http://www.asp.net/mvc [↑](#footnote-ref-52)
53. jQuery : http://jquery.com/ [↑](#footnote-ref-53)
54. Complément PowerPivot à Excel 2010 : http://msdn.microsoft.com/fr-fr/library/ee210644.aspx [↑](#footnote-ref-54)
55. Catalogue de données Rennes Métropole : http://www.data.rennes-metropole.fr/les-donnees/catalogue/?tx\_icsopendatastore\_pi1%5buid%5d=47 [↑](#footnote-ref-55)
56. Article MSDN Interrogation du service de données (WCF Data Services) : http://msdn.microsoft.com/fr-fr/library/dd673933.aspx [↑](#footnote-ref-56)
57. Billet OGDI DataLab évolue !: http://blogs.msdn.com/b/ogdifrance/archive/2013/04/22/ogdi-datalab-233-volue.aspx [↑](#footnote-ref-57)
58. Article Cross-Domain Communication with IFrames : http://softwareas.com/cross-domain-communication-with-iframes [↑](#footnote-ref-58)
59. W3C RDF/XML Syntax Specification : http://www.w3.org/TR/rdf-syntax-grammar/ [↑](#footnote-ref-59)
60. W3C RDF Validation Service : http://www.w3.org/RDF/Validator/ [↑](#footnote-ref-60)
61. W3C Resource Description Framework (RDF) : http://www.w3.org/RDF/ [↑](#footnote-ref-61)
62. Vidéo How Do I: Consume a WCF Data Service in a .NET Application : http://msdn.microsoft.com/fr-fr/data/cc974504.aspx [↑](#footnote-ref-62)
63. Bibliothèque Restlet extension for OData : http://www.restlet.org/downloads/ [↑](#footnote-ref-63)
64. Kit de développement logiciel OData pour PHP : http://odataphp.codeplex.com/ [↑](#footnote-ref-64)
65. Article MSDN Query Timeout and Pagination: http://msdn.microsoft.com/en-us/library/dd135718.aspx [↑](#footnote-ref-65)
66. Article ASP.NET 2.0 PROVIDER MODEL: INTRODUCTION TO THE PROVIDER MODEL : http://msdn.microsoft.com/en-us/library/aa479030.aspx [↑](#footnote-ref-66)
67. Billet ASP.NET 2.0 Membership, Roles, Forms Authentication, and Security Resources : http://weblogs.asp.net/scottgu/archive/2006/02/24/ASP.NET-2.0-Membership\_2C00\_-Roles\_2C00\_-Forms-Authentication\_2C00\_-and-Security-Resources-.aspx [↑](#footnote-ref-67)
68. Consortium DAISY : http://www.daisy.org [↑](#footnote-ref-68)
69. Standard DAISY/NISO : http://www.daizy.org/z3986 [↑](#footnote-ref-69)
70. DAISY Pipeline : http://www.daisy.org/projects/pipeline [↑](#footnote-ref-70)
71. Film DAISY rend la lecture plus facile : http://www.microsoft.com/france/accessibilite/video/20080513-daisy-makes-reading-easier.asx [↑](#footnote-ref-71)
72. Galerie en ligne NuGet : http://nuget.org/List/Packages [↑](#footnote-ref-72)
73. Site portail Web Microsoft Secteur Public : http://www.microsoft.com/france/secteur-public/default.aspx [↑](#footnote-ref-73)
74. Site portail Web Microsoft Interopérabilité : http://www.microsoft.com/france/openness/default.aspx [↑](#footnote-ref-74)
75. Communiqué de presse Microsoft Makes Strategic Changes in Technology and Business Practices to Expand Interoperability : http://www.microsoft.com/en-us/news/press/2008/feb08/02-21ExpandInteroperabilityPR.aspx [↑](#footnote-ref-75)