



Dotnet France
Technologies Sharepoint, SQL Server & .NET

Association Dotnet France

Présentation d'ADO .Net Data Services

Version 1.0



James RAVAILLE

<http://blogs.dotnet-france.com/jamesr>

Jean-Christophe VASSELON

Sommaire

1.	Introduction.....	3
1.1	Avant ADO .NET Data Services	3
1.2	Avec ADO .NET Data Services	3
1.2.1	Infrastructure d'accès aux données	4
2.	Les technologies utilisées par ADO .NET Data Services	5
2.1	Le langage JSON.....	5
2.2	Le langage Atom	6
2.3	L'architecture REST.....	6

1. Introduction

1.1 Avant ADO .NET Data Services

Depuis la création du Framework .Net, des efforts ont été déployés pour faciliter les interactions entre les applications et les bases de données. La première version du Framework .NET, proposait :

- L'utilisation de services Web.
- Le remoting.

Depuis la version 3.0 du Framework .NET (sorti fin 2006), Microsoft propose Windows Communication Foundation (WCF), qui apporta de nouveaux composants, permettant de faciliter la création d'applications basées sur des architectures distribuées. Si vous souhaitez en savoir plus sur WCF, vous trouverez des cours à cette adresse : http://www.dotnet-france.com/www/Cours/Cours.aspx?id_cours=11.

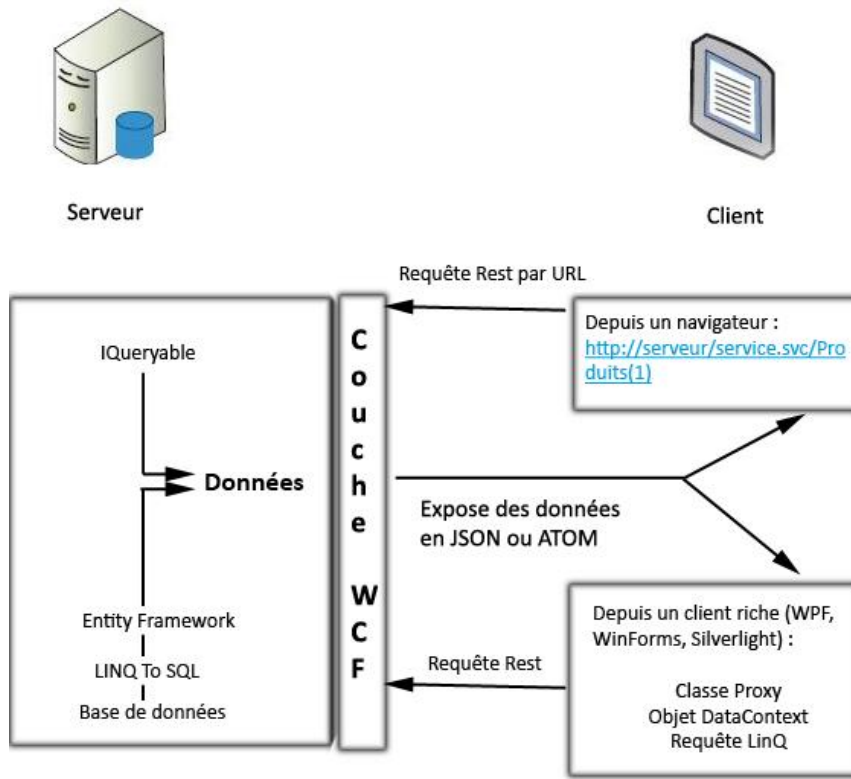
En 2007, Microsoft propose le projet « Astoria », connu aujourd'hui sous le nom d'ADO .NET Data Services.

1.2 Avec ADO .NET Data Services

ADO .NET Data Services est un ensemble de composants du Framework .NET, permettant donc la création de services génériques, interopérables et utilisables dans de diverses applications (Silverlight, Winforms, ASP .NET, ...).

ADO .NET DS repose sur l'utilisation combinée de WCF, LINQ To SQL, Entity Framework et ADO.Net ainsi que l'architecture ReST. Cela permet de requêter une base de données, de gérer les données en mode CRUD (Create Read Update et Delete), en envoyant des requêtes HTTP à destination de services WCF déployés sur serveur IIS.

1.2.1 Infrastructure d'accès aux données



Nous pouvons observer que la couche WCF, propose des services, exposant et permettant de gérer les données d'une base de données. Ces données sont donc accessibles :

- Directement depuis un navigateur.
- Depuis une application .NET, il suffira de créer une référence de service, générant une classe Proxy, se chargeant d'exposer toutes les méthodes que l'on pourra utiliser sur nos données au travers du service distant.

2. Les technologies utilisées par ADO .NET Data Services

ADO .NET Data Services utilise deux langages, pour transmettre les données contenues dans une source de données, à une application cliente :

- Le langage JSON (**J**ava**S**cript **O**bject **N**otation).
- Le langage ATOM (**A**tom Publishing Protocol).

ADO .NET Services est basée sur l'architecture REST (**R**epresentational **S**tate **T**ransfer).

2.1 Le langage JSON

Le langage JSON (JavaScript Object Notation) permet de créer un flux de données. Il constitue une alternative au langage XML, car il est plus concis et nativement interprété par le JavaScript. Il est principalement utilisé lors du développement d'applications Web avec Ajax.

Il est basé sur deux « structures générales » :

- Les collections de paires nom/valeur.
- Les listes ordonnées de valeurs.

On retrouve alors dans un flux JSON :

- Des objets {nom1:"valeur1", nom2:"valeur2", ...}
- Des tableaux : [valeur1, valeur2, valeur3 ...]
- Des valeurs.

Voici un exemple de flux de données JSON :

```
// Flux de données JSON
{ "d" : [
  {
    "__metadata" : {
      "uri" : http://localhost:49923/Silverlightoria/Astoria.svc
      "type" : "TestAstoriaModel.Produit"
    },
    "IDProduit": "1",
    "Marque": "Adadas",
    "Produit": "Chaussures courses",
    "Taille": "42",
    "Stock": "4",
    "Prix": "130",
  }
] }
```

Pour en savoir plus sur le langage JSON, nous vous invitons à consulter le site web suivant : <http://www.json.org>.

2.2 Le langage Atom

Le langage ATOM (**Atom** Publishing Protocol) est un langage basé sur le langage XML. De ce fait, il respecte ses spécifications. Le groupe de travail IETF (Internet Engineering Task Force) a défini les spécifications de ce langage.

Voici un exemple de flux de données ATOM :

```
<!--xml -->
< ?xml version="1.0" encoding="utf-8" ?>
<feed xml:base="http://localhost:49923/Silverlightoria/Astoria.svc/"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns="http://www.w3.org/2005/Atom">
<content type="application/xml">
<m:properties>
<d:IDProduit m:type="Edm.Int32">1</d:IDProduit>
<d>Type xml:space="preserve">Adadas</d:Marque>
<d:Marque xml:space="preserve">Chaussures courses</d>Type>
<d:Taille xml:space="preserve">42</d:Taille>
<d:Prix m:type="Edm.Int32">4</d:Stock>
<d:Stock m:type="Edm.Int32">130</d:Prix>
</m:properties>
</content>
```

Vous retrouverez les spécifications de ce langage à l'URL suivante : <http://www.ietf.org/rfc/rfc5023.txt>

2.3 L'architecture REST

L'architecture REST (**R**epresentational **S**tate **T**ransfer) est une architecture orientée ressources. Autrement dit, il faut voir chaque donnée dans notre base comme une ressource à laquelle on peut accéder au travers d'un service. Une base de données est considérée comme un dépôt de ressources.

Le principal objectif d'ADO.NET Data Services est de créer une infrastructure basée sur REST (**R**epresentational **S**tate **T**ransfer) pour permettre à des applications clientes :

- D'obtenir des données, pour les afficher, les traiter...
- D'utiliser des services centrés sur les données.

Pour cela, ADO .NET Data Services utilise des commandes HTTP liées au protocole REST :

- **POST** : permet d'ajouter, de mettre à jour ou de supprimer une ressource du dépôt (Create, Update, Delete).
- **GET** : permet de lire une ressource (Read).
- **PUT** : permet de créer ou de mettre à jour une ressource (Create, Update).
- **DELETE** : permet de supprimer une ressource (Delete).

Voici un schéma présentant cette infrastructure :

