

Silverlight 4 + RIA Services - Prêt pour les affaires

Exposer des services WCF (SOAP/WSDL)

par Brad Abrams ([Blog](#)) Deepin Prayag (Traduction) ([Home](#))

Date de publication : 14/02/2012

Dernière mise à jour : 20/02/2012

Cet article fait partie d'une série de traductions d'articles de Brad Abrams sur le développement d'applications métier avec Silverlight 4 et .NET RIA Services.

Cette série se concentre uniquement sur la base des applications métier : l'interrogation, la mise à jour, la validation et la sécurisation de vos données métier importantes.

Elle sera également utilisée pour mettre à jour certains billets de la **série Silverlight 3**.

Traduction.....	3
Prérequis.....	3
Exposer des services WCF (SOAP/WSDL).....	3
Conclusion.....	6
Liens.....	6
Remerciements.....	6

Traduction

Cet article est la traduction la plus fidèle possible de l'article original de **Brad Abrams, Silverlight 4 + RIA Services - Ready for Business: Exposing WCF (SOAP/WSDL) Services**

Prérequis

La procédure pas à pas requiert :

- **Visual Studio 2010** (ou la **version express gratuite**)
- **Silverlight 4 Tools** (inclut **RIA Services**)

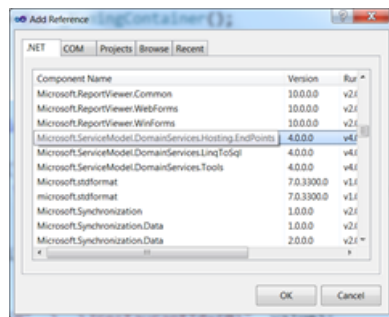
Vous pouvez **télécharger l'application complète**.

J'ai implémenté cela avec Silverlight 4 RC, mais je m'attends à ce que cela fonctionne avec Silverlight 4 RTM.

Exposer des services WCF (SOAP/WSDL)

Poursuivant **notre série**, je voulais aborder la façon dont un RIA Services peut être exposé comme un service SOAP/WSDL. Cela est très utile si vous voulez permettre qu'exactement la même logique métier / logique d'accès aux données soit disponible aux clients autres que Silverlight. Par exemple pour une application WinForms ou WPF ou même une application console. SOAP est un modèle particulièrement adapté pour l'interopérabilité avec le monde Java/JEE également.

Tout d'abord, vous devez ajouter une référence à l'assembly Microsoft.ServiceModel.DomainServices.Hosting.EndPoints à partir du RIA Services Toolkit.



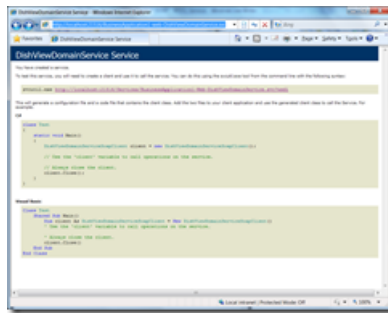
Ensuite vous devez modifier la section « endpoints » de la configuration domainServices dans le fichier web.config. Ci-dessous je vous montre les « endpoints » SOAP et OData activés.

```
<system.serviceModel>
  <domainServices>
    <endpoints>
      <add name="OData"
        type="System.ServiceModel.DomainServices.Hosting.ODataEndpointFactory, System.ServiceModel.DomainSer
      <add name="Soap"
        type="Microsoft.ServiceModel.DomainServices.Hosting.SoapXmlEndpointFactory, Microsoft.ServiceModel.D
    </endpoints>
  </domainServices>
```

Maintenant, tout ce qui vous reste à faire est d'aller vers cette URL dans votre navigateur :

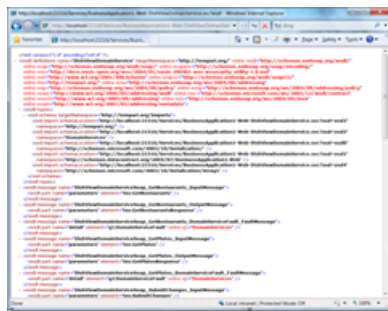
<http://localhost:21516/BusinessApplication1-web-DishViewDomainService.svc>

Notez que le motif est [namespace]-[typename].svc

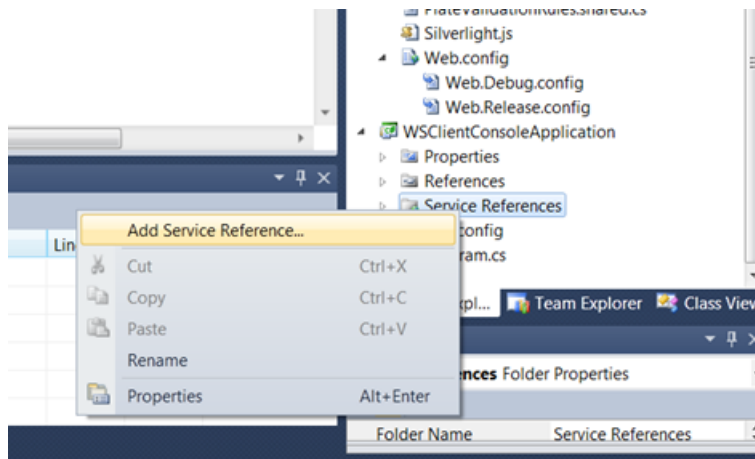


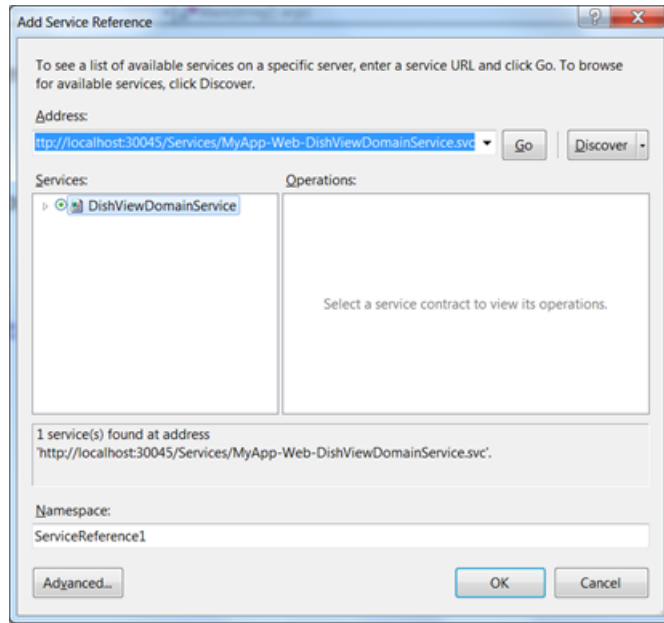
Et nous obtenons la très familière page de débogage du service WCF.

Et si vous voulez y aller en profondeur, voici le WSDL complet décrivant le service.



Vous avez fini sur le serveur, maintenant nous allons voir comment est-ce que vous consommez cela sur le client. Par souci de simplicité, j'ai créé une nouvelle application console. Faites un clic droit sur le projet et sélectionnez *Ajouter une référence de service* (en anglais, « *Add Service Reference* »).



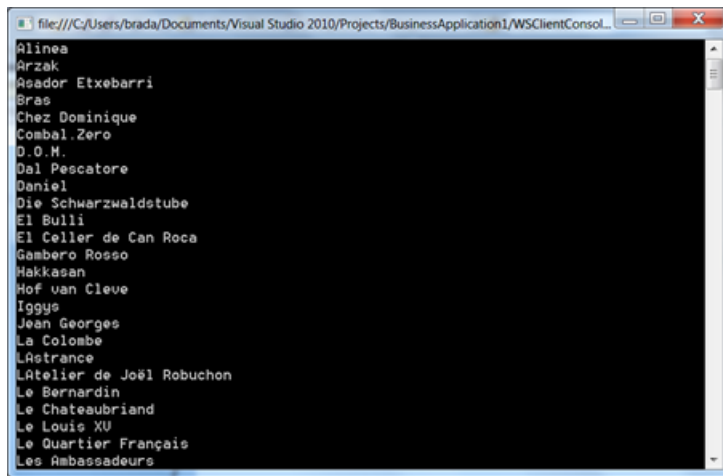


```

1: var context = new DishViewDomainServiceSoapClient();
2: var results = context.GetRestaurants();
3: foreach (var r in results.RootResults)
4: {
5:     Console.WriteLine(r.Name);
6: }
7:

```

Le code pour la requête est très simple. La ligne 1 crée le proxy client, la ligne 2 fait un appel synchrone au serveur. Ensuite nous devons tout simplement boucler à travers les résultats.



Mettre à jour les données est un peu plus intéressant. J'ai besoin de créer un ChangeSet que je remplis avec les mises à jour. Dans ce cas je mets à jour une seule valeur. Je dois aussi renvoyer la valeur d'origine vers le serveur.

```

1: var changeSet = new ChangeSetEntry();
2: var org = new Restaurant();
3: org.ID = entity.ID;
4: org.Name = entity.Name;
5: entity.Name = "Updated:" + entity.Name;
6: changeSet.Entity = entity;
7: changeSet.OriginalEntity = org;
8: changeSet.Operation = DomainOperation.Update;
9: var updateResults = context.SubmitChanges(new ChangeSetEntry[] { changeSet });
10:

```

Et vous pouvez voir ce qui résulte du fait d'exécuter cela plusieurs fois :



Pour plus d'informations, voir le billet de Deepesh Mohnani intitulé [Configuring your DomainService for a Windows Phone 7 application](#).

Pour la V1.0, ce billet décrit l'histoire de RIA Services pour les applications WPF. Même si elle fonctionne, ce n'est clairement pas la solution complète et de bout en bout qu'offre RIA Services pour Silverlight. Par exemple, vous voudrez peut-être l'expérience complète : le DataContext, entités avec validation, requêtes LINQ, le suivi des modifications etc. Ce niveau de support pour WPF est sur la feuille de route mais ne sera pas de la V1. Mais si vous aimez l'approche RIA Services, vous pourriez envisager [DevForce d'IdeaBlade](#) ou [CSLA.NET](#) qui fonctionnent à la fois pour WPF et Silverlight. Les gens sympa d'IdeaBlade m'ont offert 10 licences gratuites à distribuer. Si vous êtes intéressés, s'il vous plait faites le moi savoir.

Conclusion

Ceci conclut la neuvième partie de cette série. Dans la prochaine partie nous verrons comment exposer des services JSON.

Liens

- [Visual Studio 2010](#)
- [Visual Studio 2010 Express](#)
- [Silverlight 4 Tools](#)
- [RIA Services](#)
- [Télécharger l'application complète](#)

Remerciements

Je tiens ici à remercier [Brad Abrams](#) pour son aimable autorisation de traduire l'article.
Je remercie [tomlev](#) pour sa relecture technique et ses propositions.
Je remercie également [xyz](#) pour sa relecture orthographique et ses propositions.