# Develop Guide

# Prepare

[**Nuget**](http://nuget.org/): The great Visual Studio extension. makes it easy to install and update open source libraries and tools.

[**WCF Web API**](http://wcf.codeplex.com/wikipage?title=WCF%20HTTP): We base on this to build REST WCF Service. Another link [here](http://blogs.msdn.com/b/endpoint/archive/2010/11/01/wcf-web-apis-http-your-way.aspx)

[**Entity Framework 4.2**](http://msdn.microsoft.com/en-us/library/gg696172%28v=VS.103%29.aspx): The Microsoft ORM product. We use [Code First](http://blogs.msdn.com/b/adonet/archive/2011/03/15/ef-4-1-code-first-walkthrough.aspx) methodology.

[**ServiceStack.Text**](http://www.servicestack.net/benchmarks/NorthwindDatabaseRowsSerialization.1000000-times.2010-02-06.html): Very fast json serializing and deserializing Api

[**NewtonSoft.Json**](http://james.newtonking.com/pages/json-net.aspx): This can worked in Silverlight for json serializing and deserializing. And this api can help us reduce the json serialize cotent size when use this api furture: NullValue Ignore and Default Value Ignore.

[**ProtoBuf.net**](http://code.google.com/p/protobuf-net/): Not used in current solution. But I believe it will help us improve the performance.

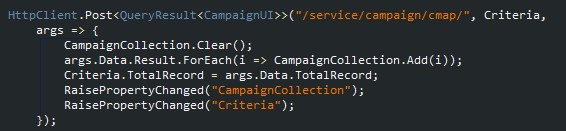
[**MVVM Light Toolkit**](http://www.galasoft.ch/mvvm/): Silverlight MVVM framework.

[**Code Stand**](http://www.idesign.net/idesign/download/IDesign%20CSharp%20Coding%20Standard.zip): IDesign C# coding standards is a pretty decent and compact (27 pages) Coding Standards Document. It covers a Naming conventions, Best practices and Framework specific guidelines.

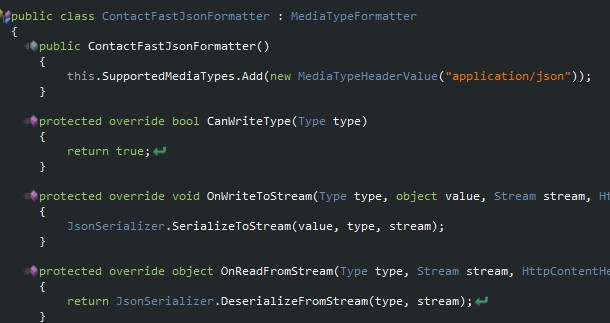
## Communications



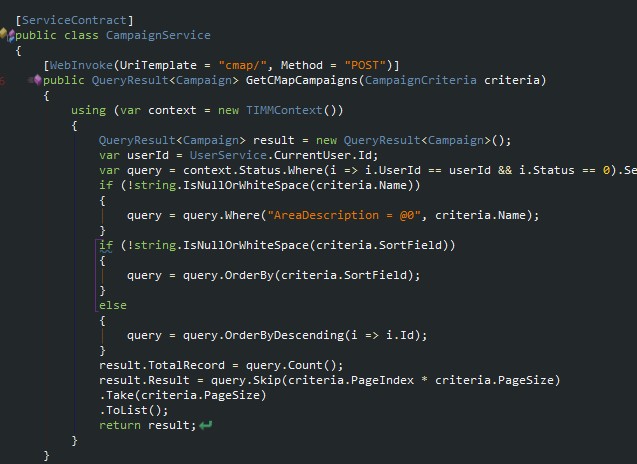
The base communication is Http. Silverlight call a helper method e.g.



The HttpClient serializer the parameter to json data and post use browser socket to the url (WCF REST Service). The silverlight can use browser or self socket for requests. I don’t know deep in these methods. But if you do not want manage cookie. Use browser socket is a better way.



The Web API will check the http header’s accept variables. When the service received the request. If the request is the supported media type (header accept value) is application/json. Web API will desializer from json to .net entity type and call the url service.

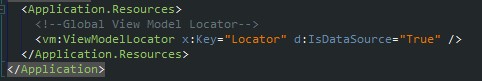


**Update: the ServiceStack.Text have a known issue: Can’t handle Entity have loop reference property. So the solution is to use NewtonSoft.Json.**

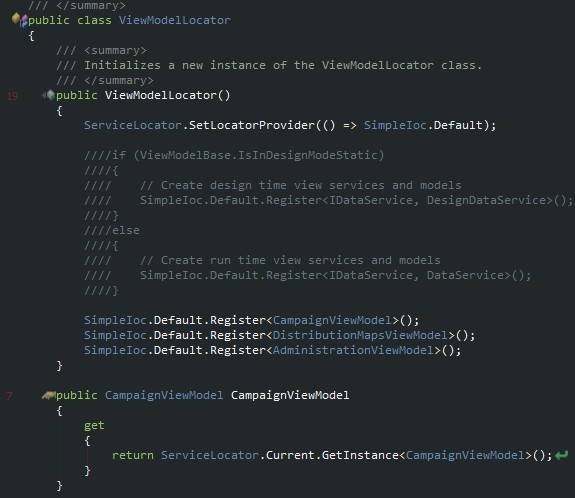
## Silvelright MVVM Pattern

Silverlight is not same as the winform. Even you can write code in winform way. In silverlight binding is everything. You can binding data event and so on. So use mvvm light toolkit is good choice to start the silverlight develop. The mvvm light toolkit is simple

You must define a application resource in app.xaml



And add the viewmodel in locator property



Then you can use all the viewmodel in any page

C:\Users\steve\AppData\Roaming\Tencent\Users\232256699\QQ\WinTemp\RichOle\K[Y5YKV))2L65VFQP8V)8_0.jpg

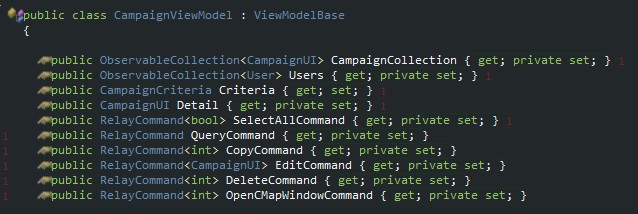
And bidning the data and method

C:\Users\steve\AppData\Roaming\Tencent\Users\232256699\QQ\WinTemp\RichOle\1UH(W9PB`HC16UW5J{RH8HP.jpg

C:\Users\steve\AppData\Roaming\Tencent\Users\232256699\QQ\WinTemp\RichOle\5$NU1M5I_T}B353(8)~6%QU.jpg

C:\Users\steve\AppData\Roaming\Tencent\Users\232256699\QQ\WinTemp\RichOle\))TQWYT_}Z_5$REEU7$WUN3.jpg

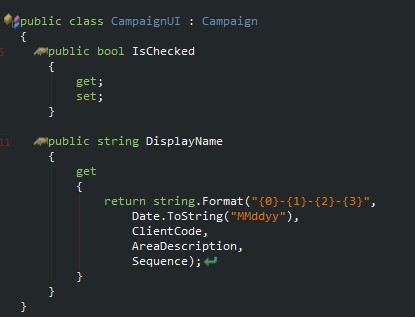
We define the data and method in viewmodel



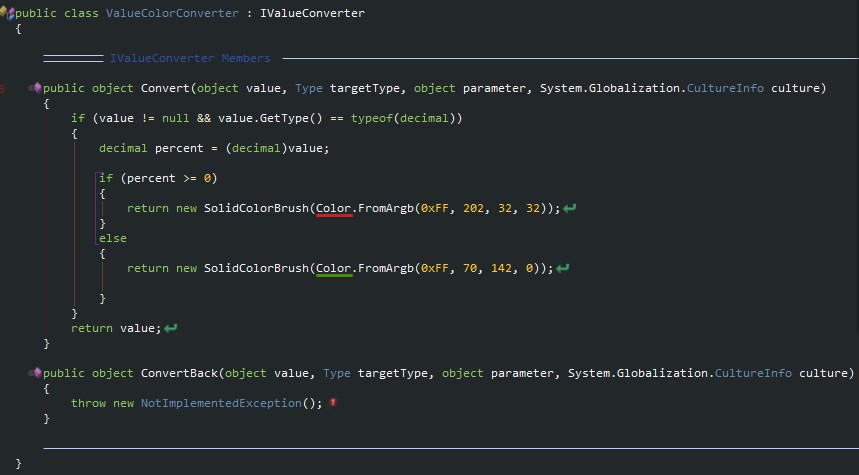
In some cause e.g. datagrid if we need add checkbox in first column



The DataGrid use a virtual row system to load thousand records. So you can’t operate row checked status by direct change the control in the DataGrid’s child controls. When you removed some rows. These rows will have wrong status. This is not a bug. It is only use the control in a wrong way. So we need add an IsSelected property to Campaign model. But this property is only need in silverlight. My solution is to inherit form Campaign.

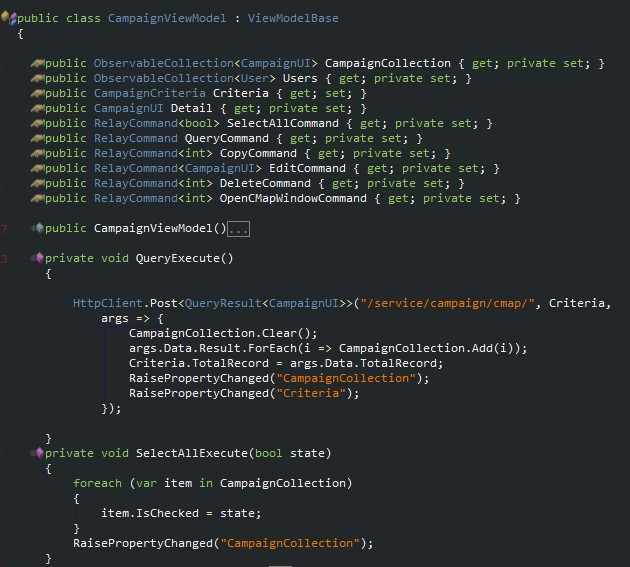


There is another problem in silverlight called multibinding. It is not support in silverlight. I use another property DisplayName for UI binding. This is an easy way to add more property for silverlight. And this is the reason why I don’t use ProtoBuf.net. But there are so many other solution to fix these problems. E.g. use convert is still a good way.



This is a color convert. When silverlight binding a decimal value to a brush. It can safe convert project completed percent to very different color. It is very helpful in some cause.

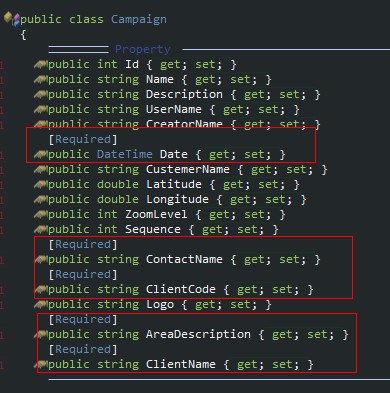
The ViewModel is where we put the logical code. It is like code behind. We write service call method. the ui logical



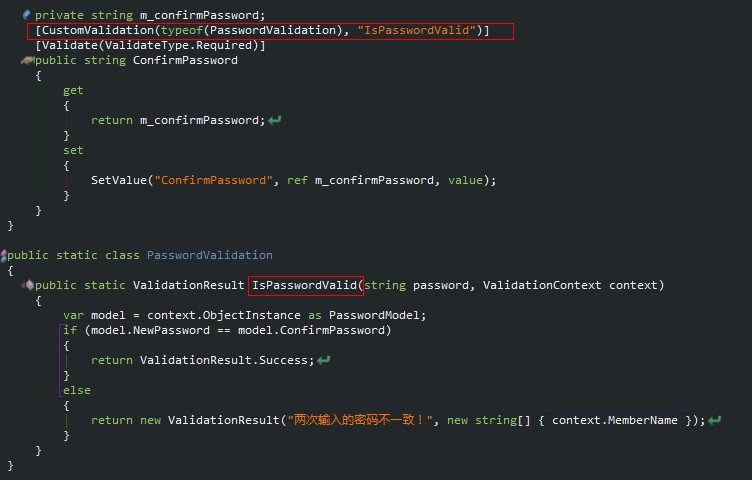
The CampaignViewModel define the campaign list view data. Create new campaing need sales and the SelectedAll command. So the View only focus on UI element. How the UI will be. The ViewModel only focus the logical. How I get the data. How I operate data.

Another very import things: validation!

This is very simple:



We only need add some attribute on the model property. If there is some more complex validation. So easy:



We define the custom validation and write the validation method. sorry is my another silverlight project. I will replace this section by our code later.

And remote validation is need on some kind situation. It is still easy. Some sample will add later.

## WCF REST Service

Use Web API is very helpful. We don’t need add endpoint. All these services must Initialize before use



The magic is WebActivator. We need add these code in asp.net global file without this api. Now everything is simple. But the WebActivator is only used in .net framework 4.0 and above.

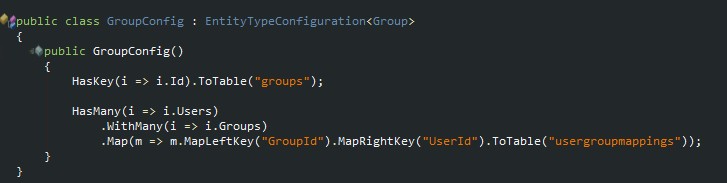


The REST Service need add [ServiceContract] attribute. And every method must define the http method. Get or Post. Consider the campaign service base address “service/campaign” defined in service initialize. So the end url is service/campaign/+UriTemplate.

## Entity Framework 4.2 Code First

Code first is a very great future released in Entity Framework 4.1. In old edmx file. There are many work need to do to implement some complex relationship. The fault is the edmx designer. This is not my think. It is from Microsoft. I must accept the code first is not easy like other Microsoft product.

Code first can use Annotations to define the mapping. But I want share these model in silverlight. So I only use the Fluent API.



These is a group mapping code.

HasKey means the group table have the primary key Id. ToTable(“groups”) means the Group entity is mapping to groups Table.

HasMany means Group Entity have a List<User> Users relationship

WithMany means the User Entity still have List<Group> Groups relationship

MapLeftKey("GroupId").MapRightKey("UserId").ToTable("usergroupmappings") means the mapping table is usergroupmappings GroupId is from left Group right User.

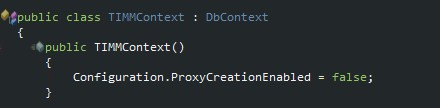
This is a many-to-many relationship.

Don’t worry about have there some different to the edmx. They are all base on the same entity framework core. And the edmx is a xml file. When you build the project. This xml file will generate a cs file. in this file there are entity class define and attribute. The Fluent API is these attribute underground function.

Entity Framework have the proxy function when you mark the property to virtual. This is means when EF load data will ignore these property. But the EF will replace the get method by another method. so why we must mark as virtual. When we call these property get method. there will raise a database connection to get data. But this function would not worked in serializer. Why?



I keep the most simple code in these query service. Using(var context = new TIMMContext()) This is very import. We must keep database connection close when finished query. But serializer is automatic processed after this code. The connection is closed. When serialser api try to access the campaign virtual property. The proxy will raise another database query. So it failed when connection is closed. And we don’t want every method will return all the relation data. If the UI only need campaign base property. Don’t return the SubMaps. We must keep the return values small enough. So we must turn off the proxy

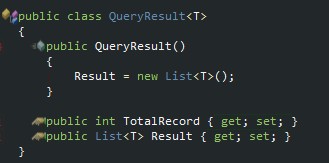


By the way the connection string is based on the Context class name

C:\Users\steve\AppData\Roaming\Tencent\Users\232256699\QQ\WinTemp\RichOle\UKE[@V{L`63L5__C`5}T{HQ.jpg

## Why not use WCF RIA Service

WCF RIA Service is still a WCF service. You can use the Fiddler to find out he svc endpoint. And if the silverlight marked use a WCF RIA Service. There will automatic generate entity class and request method to silverlight. The problem this generated entity. The entity can’t be Generics. It must use some ugly method to use another entity not defined in the Entity Framework model. E.g. I want use this class for wrap the return value



This class give us the pager info and list data in one response.

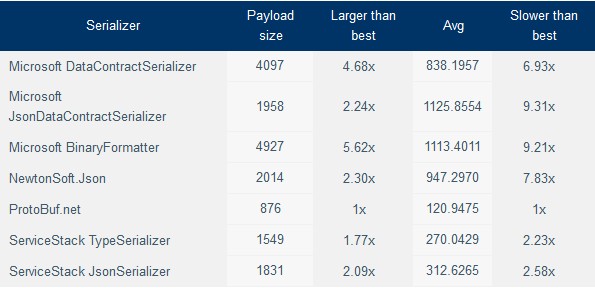
I am sure there will be some solution. You can safe skip this or use another way to our target. But the WCF RIA Service is out of control. I mean I can’t control it. I can’t change the serializer method to improve the performance. I can’t easy define any parameters in the service method. I can’t … there is so many limit.

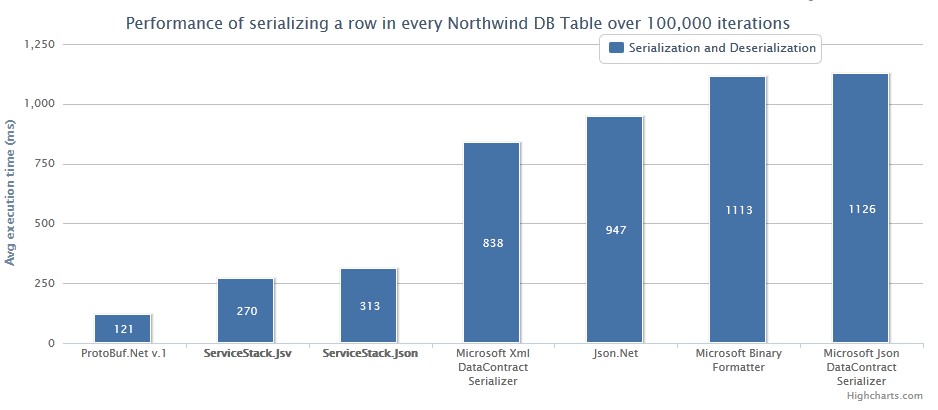
## Others

We can use any Http accept value and the Web API(WCF REST Service) can receive any type to deserializer to .net type. I believe use ProtoBuf.net for data serializer and deserializer will improve performance. And use some kind zip method will helpful. We must do some test to consider the serializer time + zip time + data transfer time ? serializer time + data transfer time.

The zip is a very heavy work for service. Especially in the many number of concurrent users. I design use another http accept like application/zipjson. When we know the request will return big result. E.g. request campaign with 300+ address. We can manual use another HttpClient method to force ask zip response. Or in the service do some check. And build the response header to mark the response data type. This is still in thinking.

Why use NewtonSoft.Json in Silvelright and use ServiceStack.Text in .net?





See more in here <http://www.servicestack.net/benchmarks/>

Use Web API default Microsoft JsonDataContractSerializer is bad idea. The best way is use ProtoBuf.net but I never used it in silverlight and .net. I believe we need more test before use it. Another thing is the WCF RIA Service use BinaryFormatter. The serializer size and performance are all bad. And I can’t replace the WCF RIA Service communicate method. The WCF RIA Service is design to simple use as I said. The WCF RIA is too simple to real world project.

The Http PUT and DELETE I have not added in this solution but I decide to add these method When I read this [article](http://www.artima.com/lejava/articles/why_put_and_delete.html). Next version is better choice.

Share Model in .net and silverlight. I think This is a right way to simple the project. We design the model in .net. use it in entity framework. Serializer to json and silverlight can easy deserializer to silverlight model. By the way the share model is not total fit silverlight. E.g. if we use a campaign list in silverlight

The secret is not the problem in this design. I checked the old version code. Use session to save use info in memory. Then check all asp.net requests. So we only host this silverlight in old solution page. Another question is use session to check user login state is not good way. This will cause more memory and bad performance. Use asp.net Membership is best practices. This is not the big problem. By the way save user password in clear word so bad design.

Entity framework is good ORM. Use 4.2 and code first is hard to say why. Another way is use stand edmx file and build POCO file. It is still worked. I don’t worry this. The real problem is how to define the entity framework model base current database. The NHibernate have some difference. I really want keep the Static modeling. This is the very import thing in my opinion. The old developer do a good job on the model.

The ORM is normal have three relationship: one-to-one one-to-many many-to-many. The first tow relationship is simple. Many-to-many relationship is more complexity. And inheritance have a lot of troublesome problems. The old developer is still not well fixed. I think these should be fixed in this version.