عنوان: **اتصال به سرویس WCF در NETCF 3.5**
نویسنده: مهدی پایروند
تاریخ: ۱۳۹۳/۱۱/۰۴ ۱۸:۱۰
آدرس: www.dotnettips.info
گروه‌ها: WCF, Windows Compact, .NET Compact Framework

با توجه به پیشرفتی که در حوزه اپلیکشن‌های وابسته به فریمورک دات نت بوجود آمده، ولی شاید حرکت عملی بزرگی از سمت تولیدکندگان در حوزه کامپکت صورت نگرفته و همچنان شاهد فرمانروایی سیستم عاملهایی چون Windows Compact 6.0 با استفاده از دات نت فریمورک‌هایی نهایت با نسخه 3.5 هستیم. البته میتوان ارزان‌تر بودن در خارج و مسئله تحریم در داخل را هم در نظر داشت و نمونه عینی این مورد را میتوان در دستگاههای وارد شده در حوزه Compact ، دید. البته شرکت‌های تولید کننده خارجی که عمدتا در کشورهای جنوب شرق و شرق آسیا هستند، جزو شرکت‌های مطرح در این زمینه هستند که بازارهای خوبی هم در کشورهای توسعه یافته‌ای چون آمریکا پیدا کرده‌اند.

در این بین برای عقب نماندن از تکنولوژی‌های جدید بوجود آمده در حوزه دات نت مانند WCF این مقاله کمکی هر چند کوچک برای استفاده از این قابلیت موثر در فریمورک کامپکت می‌تواند باشد.
پیشنیازهای لازم:
- Microsoft Visual Studio 2008 + Service Pack 1

- نصب Power Toys for .NET Compact Framework 3.5

پیاده سازی سرویس (بر روی سیستمی غیر از ویندوز کامپکت):

در ویژوال استودیو 2008 سرویس پک یک، پروژه ای از نوع class library را ایجاد کرده و سرویسی تستی را برای استفاده ایجاد میکنیم:

```
[ServiceContract(Namespace = "http://samples.wcf.cfnet.sample")]
    public interface ICalculator
    {
        [OperationContract]
        int Add(int a, int b);
    }
```

و پیاده سازی آن:

```
public class CalculatorService : ICalculator
    {
        public static int count;

        public int Add(int a, int b)
        {
            count++;
            Console.WriteLine(string.Format("{3}\tReceived 'Add({0}, {1})' returning {2}", a, b, a + b,
count));
            return a + b;
        }
```

سرور سرویس:

برای هاست این سرویس از یک برنامه‌ی کنسول که در سلوشن ایجاد میکنیم استفاده میکنیم. البته امکان‌های دیگر برای هاست سرویس در هر پروسس دات نتی را میتوان یاد آور شد. برای هاست کردن شروع یک سرویس WCF باید یک IP درون شبکه را که قابل دسترسی از سمت ویندوز کامپکت بوده و به سیستم انتساب داده شده، دریافت و استفاده کنیم:

```
var addressList = Dns.GetHostEntry(Dns.GetHostName());

string hostIP = addressList.AddressList.Single(x=>x.ToString().StartsWith("192.168.10.")).ToString();
```

```
Uri address = new Uri(string.Format("http://{0}:8000/Calculator", hostIP));
```

در قطعه بالا IP در رنج مناسب و قابل دسترسی انتخاب میشود چون ویندوز کامپکت (فارق از اینکه در شبیه ساز باشد یا واقعی) از طریق شبکه به سرور دسترسی پیدا میکند باید IP مناسب انتساب داده شده انتخاب شود.

```
ServiceHost serviceHost = new ServiceHost(typeof(CalculatorService),address);
serviceHost.AddServiceEndpoint(typeof(ICalculator), new BasicHttpBinding(), "Calculator");
```

در ادامه یک سرویس هاست را new کرده و سرویس و بایندینگ را به آن در سازنده پاس میدهیم.

```
var serviceMetadataBehavior =
new ServiceMetadataBehavior { HttpGetEnabled = true };
serviceHost.Description.Behaviors.Add(serviceMetadataBehavior);
```

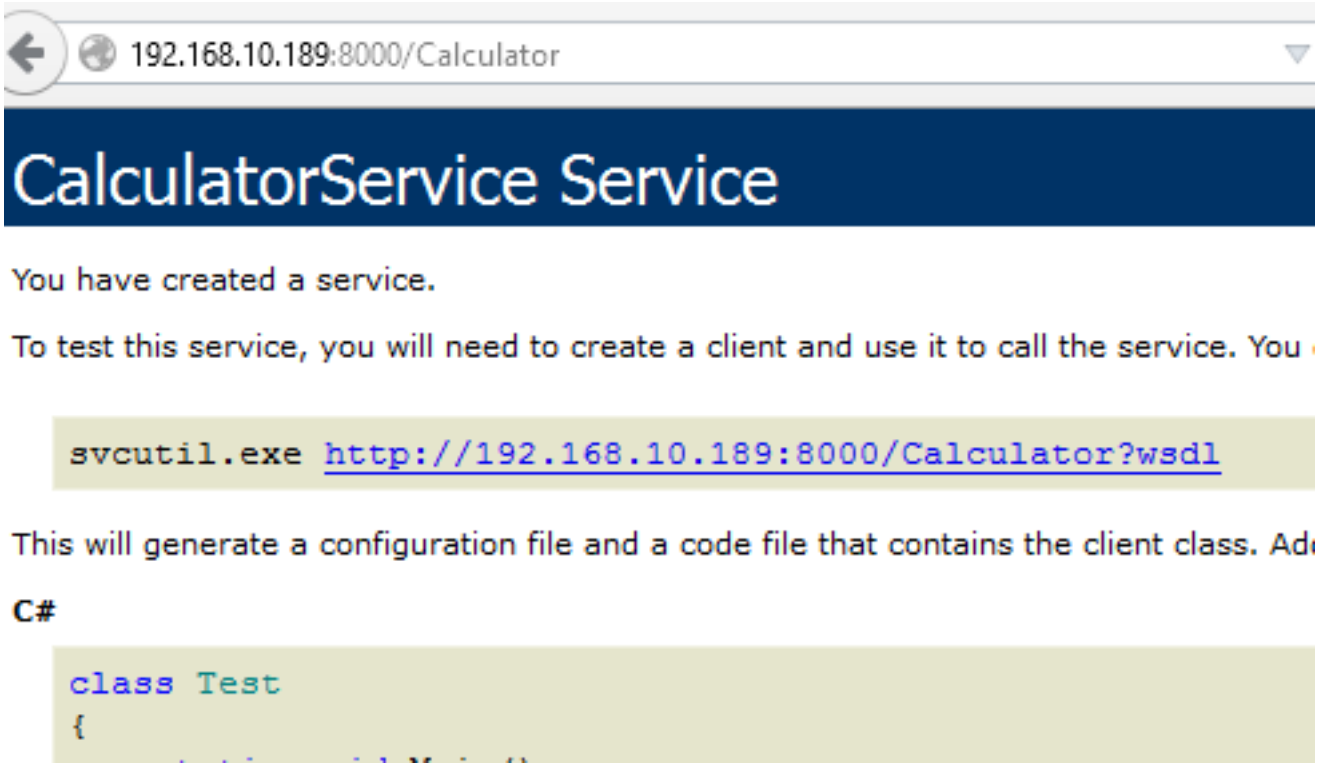
این قسمت برای ادامه کارکرد سرویس لازم نیست ولی در ادامه‌ی مقاله برای تولید کدهای سمت کلاینت باید این قابلیت فعال باشد و پس از آن دیگر احتیاجی نیست و میتوان این چند خط کد را کامنت کرد.

```
serviceHost.Open();
Console.WriteLine("CalculatorService is running at " + address.ToString());
Console.WriteLine("Press <ENTER> to terminate");
Console.ReadLine();
serviceHost.Close();
```

و در نهایت، شروع سرویس با فرمان Open و خاتمه آن با فرمان Close .

کلاینت سرویس (در داخل ویندوز کامپکت):

همراه با ارائه دات نت فریمورک 3.5 برای کار با سرویس WCF که از آن یک نسخه‌ی ارائه شده برای کامپکت نیز تهیه شده‌است، ابزاری مانند netcfSvcUtil.exe که در SDK نسخه‌ی کامپکت موجود است و کاربرد هندل کردن بعضی از موارد مانند تولید کد پروکسی‌های سمت کلاینت را دارد که در ادامه طرز استفاده از آن را بررسی خواهیم کرد. بعد از اجرای سرویس WCF با رفتار HttpGetEnabled = true برای بررسی سریع کارکرد صحیح سرویس، آدرس آن را در مرورگر میبینیم. تصویر زیر نتیجه‌ی آن در مرورگر است:



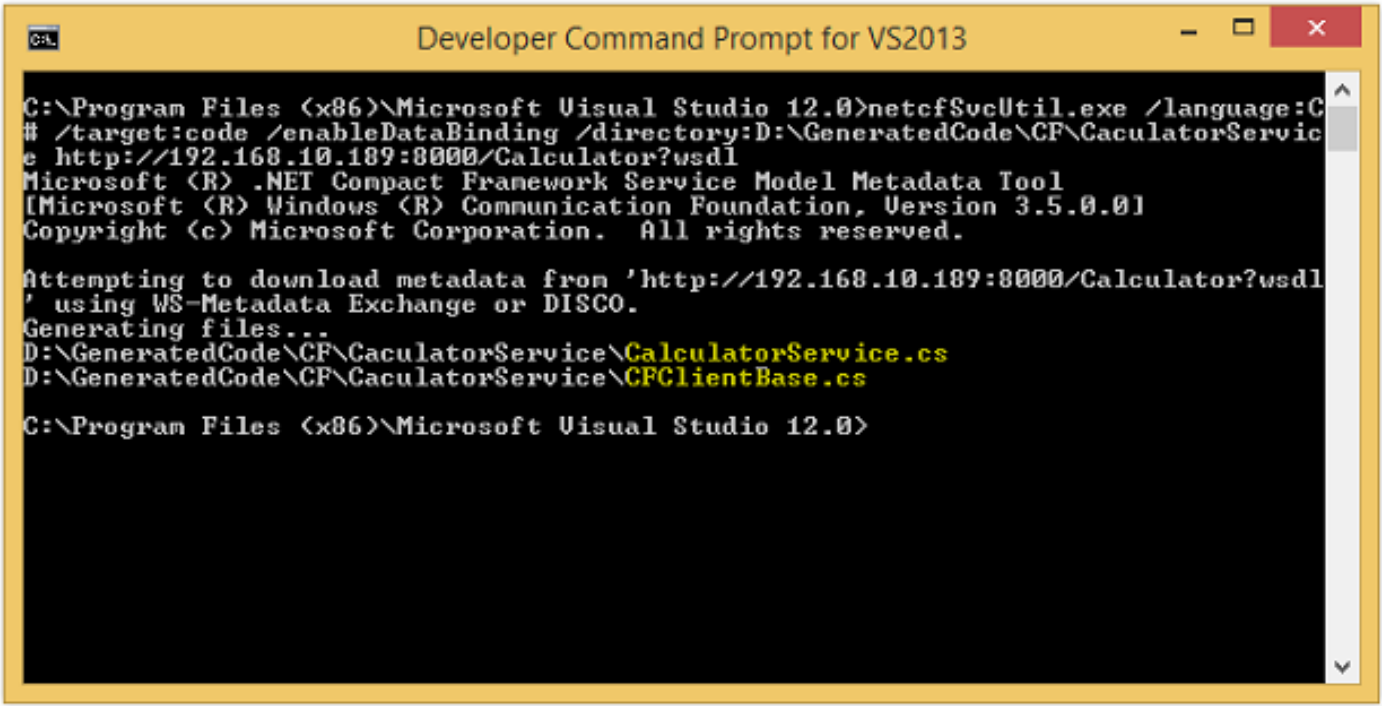
در خط فرمان به آدرس مربوط به این ابزار رفته (بسته به نسخه‌ی سیستم عامل ممکن است در پوشه‌های زیر یافت شود (:

```
(Windows Drive)\Program Files (x86)\Microsoft.NET\SDK\CompactFramework\v3.5\bin
(Windows Drive)\Program Files\Microsoft.NET\SDK\CompactFramework\v3.5\bin
```

و فرمان زیر را اجرا میکنیم:

```
netcfSvcUtil.exe /language:C# /target:code /directory:D:\GeneratedCode\CF\CaculatorService
http://192.168.10.189:8000/BooksService.svc?wsdl
```

البته ذکر IP شبکه در اینجا الزامی نیست؛ زیرا در صورت استفاده از آدرسهای داخلی سیستم، این فرمان به مشکلی بر نخواهد خورد. در این فرمان تولید کد با زبان c# و تولید کد که بصورت پیش فرض نیز وجود دارد و محل ذخیره سازی کدهای تولیدی را مشخص میکنیم و بعد از اجرای این فرمان، باید دو فایل در مسیر اشاره شده در فرمان تولید شود که اساس کار ما در سمت کلاینت خواهد بود:



کلاینت سرویس نیز با استفاده کدهای تولیدی بصورت زیر آماده سازی و اجرا میشود:

```
var addressList = Dns.GetHostEntry(Dns.GetHostName());
var localAddress = addressList.AddressList.Single(x =>
x.ToString().StartsWith("192.168.10.")).ToString();
```

دوباره IP مناسب در شبکه جاری استخراج میشود. بایندیگ مورد نیاز برای ارتباط با سرور ساخته میشود:

```
var binding = CalculatorClient.CreateDefaultBinding();
```

نکته‌ای که دراین قسمت باید مدنظر قرار گیرد این است که در زمان تولید کدها اگر از localhost یا 127.0.0.1 و یا آدرسهای دیگر انتساب داده شده به سرور استفاده کرده باشید در متد CreateDefaultBinding از همان آدرس استفاده میشود و برای اصلاح آن بصورت زیر عمل میکنیم:

```
string remoteAddress = CalculatorClient.EndpointAddress.Uri.ToString();
remoteAddress = remoteAddress.Replace("localhost", serviceAddress.Text);
```

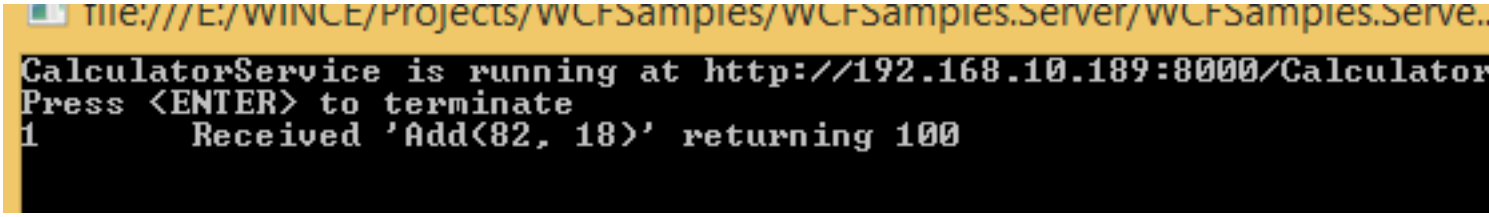
یک EndpointAddress با استفاده از این آدرس ساخته و به‌همراه بایندینگ، یک آبجکت از جنس CalculatorClient که در کدهای تولیدی داریم میسازیم:

```
CalculatorClient _client = new CalculatorClient(binding, endpoint);
```

برای تست نیز تنها متد این سرویس را با یک جفت عدد، صدا میزنیم:

```
var result = _client.Add(82, 18).ToString(CultureInfo.InvariantCulture);
```

به این ترتیب خروجی مورد نظر زیر را در کنسول سرویس مشاهده خواهیم کرد:


file:///E:/WINCE/Projects/WCFSamples/WCFSamples.Server/WCFSamples.Serve..
CalculatorService is running at http://192.168.10.189:8000/Calculator
Press <ENTER> to terminate
1 Received 'Add(82, 18)' returning 100