



گروه نرم افزاری Sweet

www.Sweetsoft.ir

ساخت وب سرویس های Soap و استفاده از آن در C#

نویسنده:

هادی امیرنهماوندی

(nahavandi.hadi@gmail.com)

مقدمه

یکی از روش های ارتباط بین نرم افزار ها استفاده از وب سرویس های سوپ می باشد. در این مقاله ساخت یک وب سرویس سوپ با استفاده از روش دابلوی.سی. اف در محیط دات نت ، نصب آن روی سرور های آی.آی.اس ویندوز و همچنین استفاده از آن به صورت همزمان و ناهمزمان در محیط دات نت با استفاده از یک پروژه نمونه توضیح داده می شود.

پروژه نمونه

در این مقاله ما سعی داریم یک وب سرویس برای یک شرکت ارایه دهنده بلیط قطارهای مسافربری بسازیم بتوان از طریق آن یک بلیط برای یک مسافر رزرو کرد ، و همچنین با وارد کردن شماره قطار ظرفیت آن قطار ، اطلاعات مسافری و تعداد مسافری آن قطار را دریافت کرد.

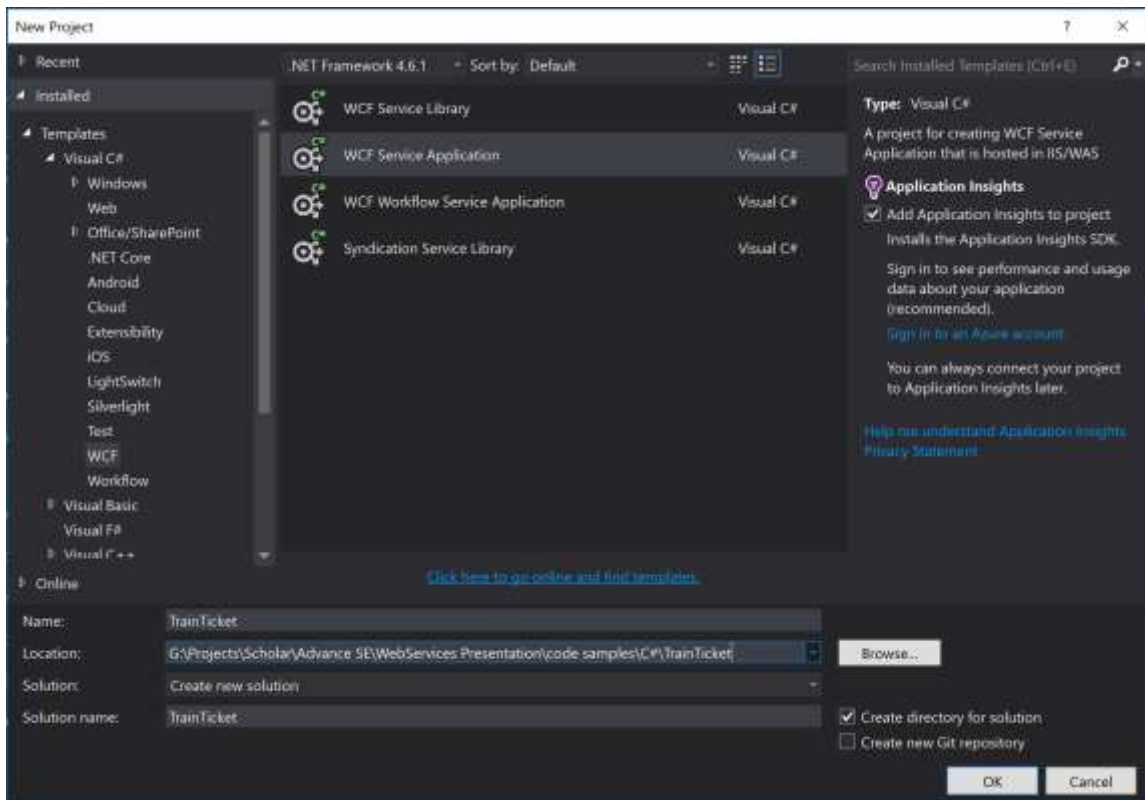
ساخت پروژه جدید برای وب سرویس در محیط ویژوال استودیو ۲۰۱۵

در محیط دات نت وب سرویس ها را می توان به صورت یک پروژه مستقل ایجاد نمود ، می توان آن را به صورت یک کتابخانه به سولوشن (راهکار) مورد نظر اضافه نمود و یا اینکه به طور مستقیم آن را به پروژه اضافه نمود. در این مقاله ما به ایجاد وب سرویس به صورت یک پروژه مستقل می پردازیم.

ایجاد پروژه وب سرویس WCF

برای ایجاد یک پروژه WCF Service Application می بایست قدم های زیر را انجام دهید:

- ۱- نرم افزار ویژوال استودیو ۲۰۱۵ را اجرا کنید.
- ۲- دکمه های Ctrl+Shift+N را فشار دهید تا صفحه ساخت پروژه جدید نمایش داده شود (شکل ۱).
- ۳- از سمت چپ از بخش سی شارپ روی گزینه WCF کلیک کنید.
- ۴- گزینه WCF Service Application را انتخاب کنید.
- ۵- در بخش Name نام پروژه مورد نظر خود را وارد کنید. (ما در اینجا نام آن را TrainTicket می گذاریم)



شکل ۱: ساخت یک پروژه نرم افزار سرویس دابلیو.سی.اف در ویژوال استودیو ۲۰۱۵

ساختار پروژه نرم افزار وب سرویس WCF

پس از تایید ، ویژوال استودیو یک پروژه جدید به همراه یک وب سرویس نمونه با نام Service1 را ایجاد خواهد کرد. هر وب سرویس در این روش دارای دو فایل می باشد: ۱- یک Interface با نام I[ServiceName] (منظور از ServiceName نام وب سرویس ما است) و یک فایل با نام [ServiceName].svc که خود شامل فایل [ServiceName].cs و فایل تنظیمات وب سرویس می باشد.

عنوان و نوع پارامترهای ورودی و خروجی هر وب سرویس و همچنین ساختار انواع پیچیده مورد استفاده در آن (complex types) می بایست در فایل interface آن تایین شود. البته انواع پیچیده را می توان در فایل های مجزا نیز معرفی نمود. انواع پیچیده و ساخت آن ها در بخش های بعدی این مقاله توضیح داده خواهد شد.

واضح است که برای اجرای وب سرویس می بایست یک کلاس interface آن را پیاده سازی (Implement) نماید. آن کلاس در فایل [ServiceName].cs قرار می گیرد.

ساختار فایل اینترفیس وب سرویس های WCF

در شکل ۲ اینترفیس وب سرویس نمونه ای که می خواهیم بنویسیم نمایش داده شده است، در ادامه به توضیح کد آن می پردازیم.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;

namespace TrainTicket
{
    [ServiceContract]
    interface ITrainTicket
    {
        [OperationContract]
        OperationResponse Reserve(int trainCode, Person p);

        [OperationContract]
        OperationResponse GetCapacity(int trainCode);

        [OperationContract]
        OperationResponse GetPassengers(int trainCode);

        [OperationContract]
        OperationResponse GetPassengerCount(int trainCode);
    }

    [DataContract]
    interface ITrainTicket
    {
        [DataMember]
        public class Person
        {
            private int id;
            private string name;
            private string family;

            [DataMember]
            public int ID
            {
                get { return id; }
                set { id = value; }
            }

            [DataMember]
            public string Name
            {
                get { return name; }
                set { name = value; }
            }

            [DataMember]
            public string Family
            {
                get { return family; }
                set { family = value; }
            }
        }
    }
}

```

شکل ۲: ساختار فایل Interface یک وب سرویس

اینترفیس وب سرویس از دو فضای نام زیر استفاده می کند:

۱- System.ServiceModel

۲- System.Runtime.Serialization

عبارت های معرفی عملیات وب سرویس در فضای نام System.ServiceModel و عبارت های معرفی داده های وب سرویس در فضای نام System.Runtime.Serialization تعریف شده اند.

برای مشخص کردن این که Interface نوشته شده interface یک وب سرویس می باشد می بایست قبل از تعریف آن عبارت [ServiceContract] را قرار داد.

در اینجا نام وب سرویس ما Service1 می باشد ، همانطور که توضیح داده شد نام اینترفیس آن باید IService1 باشد.
وب سرویس ما دارای چهار متد می باشد:

- ۱- متد Reserve که کد قطار و اطلاعات مسافر را دریافت نموده و نتیجه رزرو را به صورت یک بله یا خیر بر می گرداند.
- ۲- متد GetCapacity که کد قطار را دریافت نموده و ظرفیت قطار را بر می گرداند.
- ۳- متد GetPassengers که قطار را دریافت نموده و لیست اطلاعات مسافران را برمی گرداند.
- ۴- متد GetPassengerCount که قطار را دریافت نموده و تعداد مسافران موجود آن را بر می گرداند.

همانطور که در شکل ۲ مشاهده می کنید قبل از تعریف هر متدی که توسط وب سرویس ارائه می شود باید عبارت [OperationContract] را قرار دهید.

همانطور که مشاهده می کنید خروجی متد سوم از انواع داده اولیه نمی باشد و می بایست آن را تعریف کنیم. به دلیل اینکه این کلاس باید در کلاینت های استفاده کننده از این وب سرویس نیز ساخته شود می بایست تعریف آن را به همراه مشخصات عمومی آن با عبارت های مخصوصی مشخص نماییم. همانند کلاس های دیگر زبان سی شارپ می توانیم برای تعریف هر کلاس فایل مستقلی ایجاد کنیم و یا اینکه تعریف کلاس را در یک فایل موجود انجام دهیم. در اینجا تعریف کلاس را در همان فایل اینترفیس انجام می دهیم.

همانطور که در شکل ۲ مشاهده می کنید برای مشخص کردن کلاس مورد استفاده در وب سرویس می بایست قبل از تعریف آن عبارت [DataContract] را قرار دهید.

قبل از هر مشخصه عمومی کلاس نیز باید عبارت [DataMember] را قرار دهید. در مواردی مانند مثال بالا که مشخصات به صورت خصوصی (private) باشند می بایست عبارت یاد شده را قبل از متد های Accessor آن ها قرار دهید.

ساختار فایل پیاده سازی وب سرویس

پیاده سازی وب سرویس در فایلی که نام آن در قالب [ServiceName].cs می باشد و در پروژه نرم افزار وب سرویس می تواند با باز کردن فایل [ServiceName].svc انجام می شود. این کلاس یک کلاس معمولی سی شارپ می باشد و فایل اینترفیس وب سرویس را پیاده سازی می نماید. چون عملیات و داده های عمومی وب سرویس را در فایل اینترفیس مشخص نموده ایم ، دیگر نیازی به مشخص کردن متد ها با عبارت های مخصوص تعریف وب سرویس نمی باشد.

در شکل ۳ یک مثال برای پیاده سازی وب سرویس بلیط قطار ارائه شده است. در اینجا برای پیچیده تر نشدن کد در متد ها ذخیره و دریافت گزارش را از پایگاه داده داده انجام نمی دهیم.

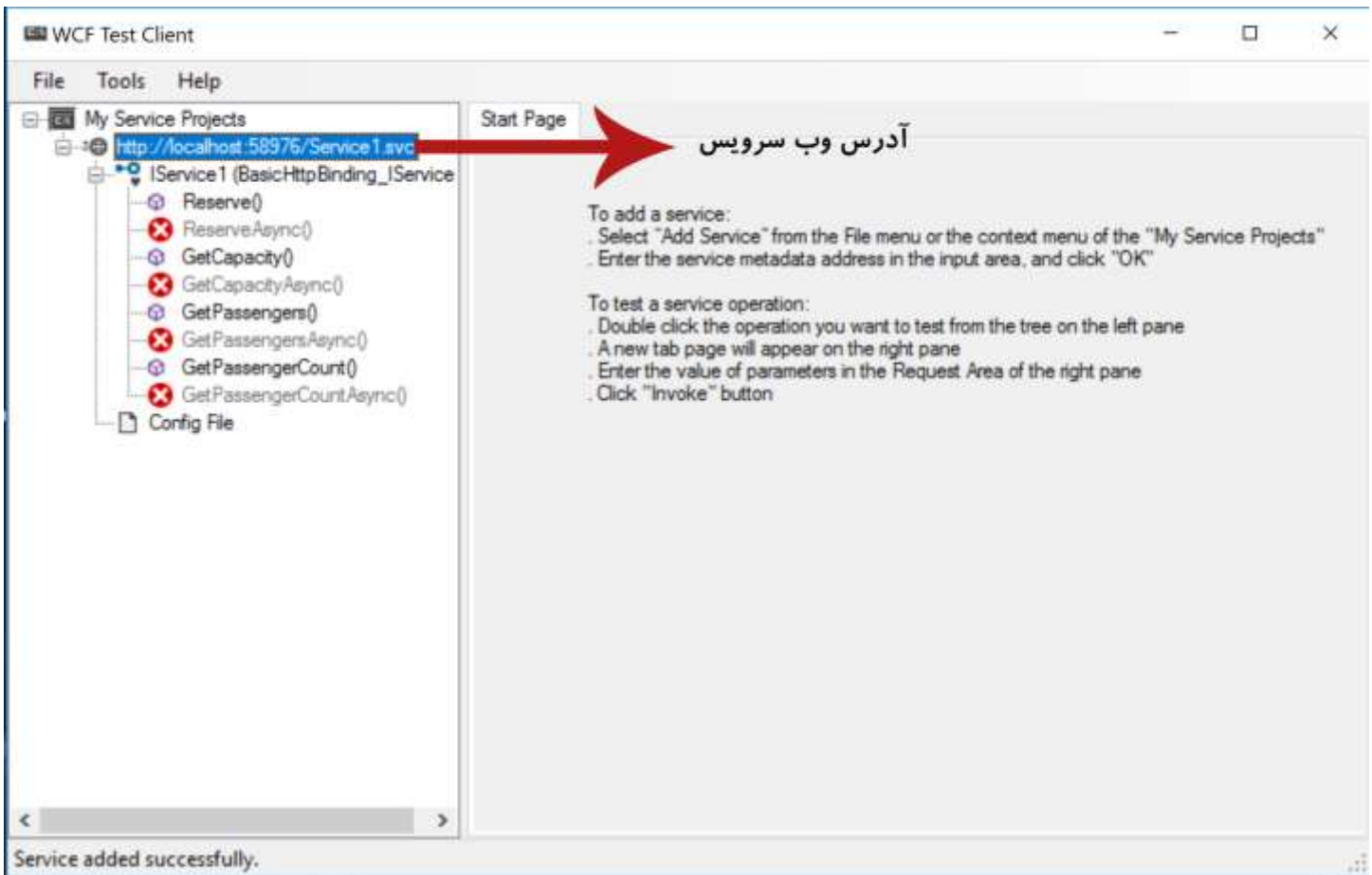
```
File Edit View Project Build Debug Team Tools Test Analyze Window Help Full Screen
ServiceFacade.cs ServiceFacade
TrainTicketServiceFacade
ReserveTicketFromCode_Person.pj

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Runtime.Serialization;
5 using System.ServiceModel;
6 using System.ServiceModel.Web;
7 using System.Text;
8
9 namespace TrainTicket
10 {
11     [WebService(Namespace = "http://tempuri.org/")]
12     [WebServiceBinding(ConformsTo = WSDL.Schema)]
13     public class Service1 : IService1
14     {
15         [OperationContract]
16         public int GetCapacity(int trainCode)
17         {
18             if (trainCode == 4)
19                 return 10;
20             else if (trainCode == 8)
21                 return 20;
22             else if (trainCode == 14)
23                 return 40;
24             else
25                 return 13;
26         }
27
28         [OperationContract]
29         public int GetPassengerCount(int trainCode)
30         {
31             if (trainCode == 4)
32                 return 7;
33             else if (trainCode == 8)
34                 return 9;
35             else if (trainCode == 14)
36                 return 30;
37             else
38                 return 4;
39         }
40
41         [OperationContract]
42         public List<Person> GetPassengers(int trainCode)
43         {
44             List<Person> pList = new List<Person>();
45             Person p1 = new Person();
46             p1.Name = "Hadi";
47             p1.Family = "Mehzad";
48             p1.ID = 1547;
49             pList.Add(p1);
50
51             Person p2 = new Person();
52             p2.Name = "Farhad";
53             p2.Family = "Mehzad";
54             p2.ID = 2224;
55             pList.Add(p2);
56             return pList;
57         }
58
59         [OperationContract]
60         public bool Reserve(int trainCode, Person p)
61         {
62             if (trainCode < 24)
63             {
64                 //Do Some Work in P
65                 return true;//Reserved
66             }
67             else
68                 return false;//Failed
69         }
70     }
71 }
```

شکل ۳: نمونه کد پیاده سازی وب سرویس

اجرای نرم افزار وب سرویس

برای آغاز به کار وب سرویس پروژه خود را اجرا نمایید. پس از اجرا مشابه شکل ۴ صفحه تست کردن وب سرویس نمایش داده می شود. می توانید آدرس وب سرویس خود را در این صفحه مشاهده نمایید. با وارد کردن آدرس وب سرویس در مرورگر اینترنت ، صفحه ای مانند شکل ۵ نمایش داده می شود.

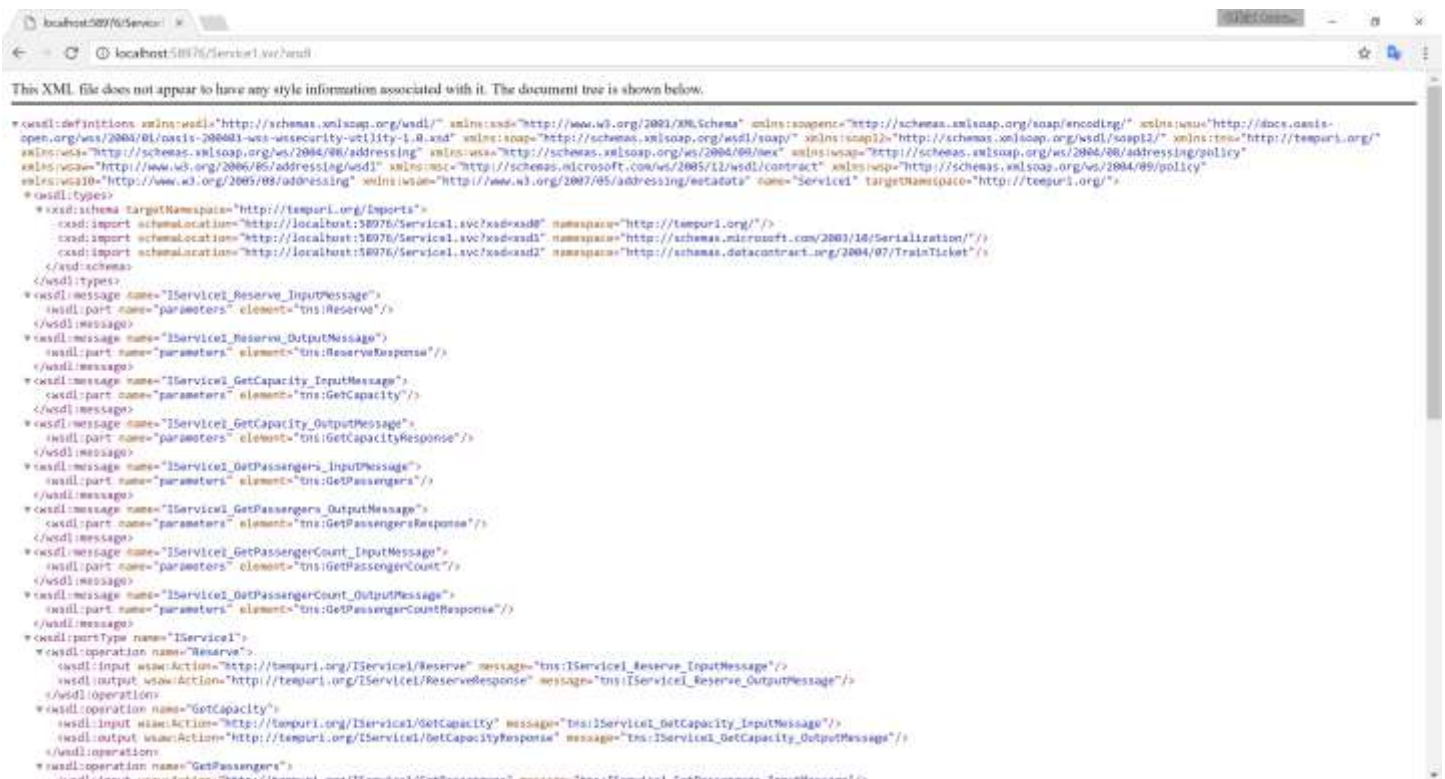


شکل ۴: کلاینت تست وب سرویس



شکل ۵: نمونه صفحه وب سرویس در مرورگر اینترنت

با اضافه کردن عبارت wsdل به لینک وب سرویس می توانید فایل wsdل وب سرویس را مشاهده نمایید که مانند شکل ۶ می باشد.



شکل ۶: نمونه فایل WSDL وب سرویس

استفاده از وب سرویس در سی شارپ

همانطور که اشاره شد وب سرویس های نوشته شده به روش سوپ فارغ از اینکه وب سرویس در چه محیط و یا زبان برنامه نویسی نوشته شده اند در زبان ها و محیط های برنامه نویسی مختلفی قابل دسترسی می باشند. در اینجا ما به نوشتن یک کلاینت برای وب سرویسی که برای بلیط قطار نوشتیم می پردازیم در محیط ویژوال استودیو ۲۰۱۵ و در زبان سی شارپ می پردازیم. نوشتن کد کلاینت در زبان های دیگر نیز به روش مشابه انجام می گیرد با این تفاوت که محیط ویژوال استودیو می تواند بیشتر کد مورد نیاز را تولید می کند و برنامه نویس فقط نیاز به استفاده از کد تولید شده دارد.

استفاده از وب سرویس به دو روش همزمان و غیر همزمان انجام می گیرد که در اینجا به هر دو روش به صورت مجزا می پردازیم.

استفاده از وب سرویس به صورت همزمان

در اینجا می خواهیم از وب سرویس نوشته شده برای بلیط قطار به صورت همزمان استفاده کنیم. همانطور که می دانید در این روش نرم افزار پس از ارسال درخواست به وب سرویس می بایست برای ادامه کار منتظر دریافت پاسخ از وب سرویس بماند و ارتباط بین وب سرویس و کلاینت تا زمان دریافت پاسخ قطع نمی شود.

همانطور که می دانید می توانیم از وب سرویس در انواع نرم افزار ها استفاده کنیم. به طور مثال ما در اینجا یک پروژه ویندوز فرم می سازیم و می خواهیم از وب سرویس در آن استفاده نماییم.

برای استفاده از وب سرویس از مسیر زیر وارد صفحه اضافه کردن ارجاع وب سرویس می شویم.

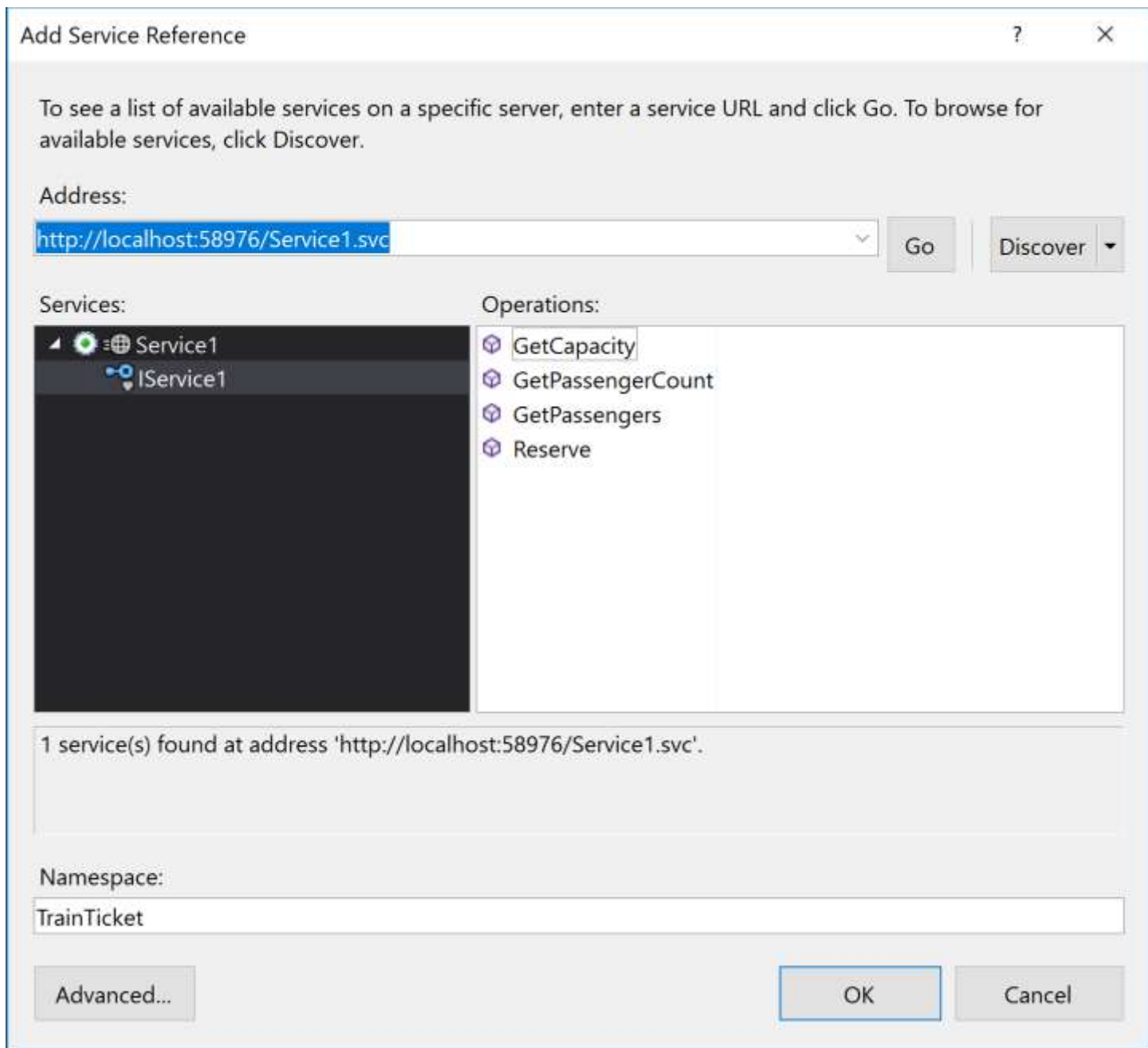
Project>Add Service Reference

در شکل 7 این صفحه نمایش داده شده است.

در این صفحه در بخش آدرس ، آدرس وب سرویس مورد نظر را وارد نمایید و با کلیک روی دکمه Go و ویژوال استودیو تلاش می کند که متدهای وب سرویس را تشخیص دهد. این کار را باید زمانی انجام دهید که وب سرویسی که آدرس آن را وارد می کنید در حال اجرا باشد.

حال در بخش Namespace نام فضای نامی که می خواهید در نرم افزارتان از طریق آن به وب سرویس دسترسی پیدا کنید را وارد کنید با کلیک روی دکمه OK ، ویژوال استودیو کد مورد نیاز برای دسترسی به وب سرویس وارد شده را تولید خواهد کرد. کدهای تولید شده را می توانید در بخش Service References پروژه تان مشاهده نمایید.

حال برای دسترسی به وب سرویس می بایست یک شی پراکسی از وب سرویس بسازید. پراکسی در سی شارپ به صورت یک کلاس با نام Client[ServiceName].[ServiceReferenceNamespace] ساخته می شود. متدهای موجود در پراکسی مشابه متدهای وب سرویس می باشند با این تفاوت که ویژوال استودیو در پیاده سازی متدهای پراکسی به جای کد آن متد کد ارسال و دریافت اطلاعات به متد هم نام از آن وب سرویس را قرار می دهد(شکل ۸) . به طور مثال برای نوشتن کد دریافت تعداد مسافران قطار شماره ۱۴ می توانیم کدی مانند شکل ۹ را بنویسیم.



شکل ۷: صفحه افزودن ارجاع به وب سرویس

```

public partial class Service1Client : System.ServiceModel.ClientBase<TrainTicketSyncClient.TrainTicket.IService1>, TrainTicketSyncClient.TrainTicket.IService1 {
    1 reference
    public Service1Client() {
    }

    0 references
    public Service1Client(string endpointConfigurationName) :
        base(endpointConfigurationName) {
    }

    0 references
    public Service1Client(string endpointConfigurationName, string remoteAddress) :
        base(endpointConfigurationName, remoteAddress) {
    }

    0 references
    public Service1Client(string endpointConfigurationName, System.ServiceModel.EndpointAddress remoteAddress) :
        base(endpointConfigurationName, remoteAddress) {
    }

    0 references
    public Service1Client(System.ServiceModel.Channels.Binding binding, System.ServiceModel.EndpointAddress remoteAddress) :
        base(binding, remoteAddress) {
    }

    2 references
    public bool Reserve(int trainCode, TrainTicketSyncClient.TrainTicket.Person p) {
        return base.Channel.Reserve(trainCode, p);
    }

    2 references
    public System.Threading.Tasks.Task<bool> ReserveAsync(int trainCode, TrainTicketSyncClient.TrainTicket.Person p) {
        return base.Channel.ReserveAsync(trainCode, p);
    }

    2 references
    public int GetCapacity(int trainCode) {
        return base.Channel.GetCapacity(trainCode);
    }

    2 references
    public System.Threading.Tasks.Task<int> GetCapacityAsync(int trainCode) {
        return base.Channel.GetCapacityAsync(trainCode);
    }

    2 references
    public TrainTicketSyncClient.TrainTicket.Person[] GetPassengers(int trainCode) {
        return base.Channel.GetPassengers(trainCode);
    }

    2 references
    public System.Threading.Tasks.Task<TrainTicketSyncClient.TrainTicket.Person[]> GetPassengersAsync(int trainCode) {
        return base.Channel.GetPassengersAsync(trainCode);
    }

    2 references
    public int GetPassengerCount(int trainCode) {
        return base.Channel.GetPassengerCount(trainCode);
    }

    2 references
    public System.Threading.Tasks.Task<int> GetPassengerCountAsync(int trainCode) {
        return base.Channel.GetPassengerCountAsync(trainCode);
    }
}

```

شکل ۸: نمونه کد پراکسی ساخته شده در کلاینت وب سرویس

```

1 reference
private void button1_Click(object sender, EventArgs e)
{
    TrainTicket.Service1Client Proxy = new TrainTicket.Service1Client();
    MessageBox.Show(Proxy.GetCapacity(14).ToString());
}

```

شکل ۹: نمونه کد دریافت ظرفیت قطار شماره ۱۴

استفاده از وب سرویس به صورت غیر همزمان

همانطور که اشاره شد اتصال به وب سرویس به صورت همزمان دارای معایب زیادی می باشد و در بسیاری از موارد باعث فشار به سرور می شود همچنین در بسیاری از فرایندها ممکن است دریافت پاسخ از وب سرویس مدت زیادی طول بکشد و به همین دلیل برنامه می بایست زمان زیاد منتظر بماند. به همین دلیل در این به استفاده غیر همزمان از وب سرویس می پردازیم.

در روش غیر همزمان نرم افزار در زمان فراخوانی پراکسی یک متد به صورت ورودی به آن می دهد که پس از دریافت پاسخ از وب سرویس آن متد اجرا خواهد شد. در این روش نرم افزار می تواند بدون منتظر ماندن به فعالیت خود را ادامه می دهد

برای استفاده از وب سرویس به صورت غیر همزمان صفحه افزودن ارجاع وب سرویس را باز کنید (شکل ۷) حال پس از وارد کردن آدرس، بررسی و وارد کردن فضای نام ارجاع روی دکمه Advanced کلیک کنید و در صفحه باز شده (شکل ۱۰) گزینه Generate Asynchronous Operations را انتخاب نمایید. پس از تایید ویزوال استودیو کدهای لازم برای ارتباط با وب سرویس به صورت غیر همزمان را تولید خواهد کرد.

در شکل ۱۱ نمونه کد دریافت ظرفیت قطار شماره ۱۴ ارایه شده است. در ویزوال استودیو برای هر متد وب سرویس یک شنونده یا EventHandler با نام [OperatinName]Completed ساخته می شود و باید یک شنونده به آن اضافه کنیم. همانطور که می دانید در سی شارپ در زمان ساخت EventHandler می بایست یک متد استاتیک با دو ورودی از نوع object و EventArgs به عنوان ورودی به آن بدهیم. در اینجا ورودی متد را از نوع [OperatinName]CompletedEventArgs قرار می دهیم که خود از EventHandler ارث برده است.







همانطور که مشاهده می کنید با استفاده از مشخصه Result این پارامتر ورودی متد می توانیم به خروجی وب سرویس دسترسی پیدا کنیم. واضح است که مشخصه Result کلاس [OperatinName]CompletedEventArgs هم نوع خروجی متد وب سرویس مورد نظر است که در این مثال از نوع عدد است.

Client

Access level for generated classes: Allow generation of asynchronous operations Generate task-based operations Generate asynchronous operations

Data Type

 Always generate message contractsCollection type: Dictionary collection type: Reuse types in referenced assemblies Reuse types in all referenced assemblies Reuse types in specified referenced assemblies:

-  Microsoft.CSharp
-  mscorlib
-  System
-  System.Core
-  System.Data
-  System.Data.DataSetExtensions
-  System Deployment

Compatibility

Add a Web Reference instead of a Service Reference. This will generate code based on .NET Framework 2.0 Web Services technology.

شکل ۱۰: صفحه تنظیمات پیشرفته افزودن ارجاع به وب سرویس

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Threading;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace TrainTicketSyncClient
12 {
13     [Serializable]
14     public partial class Form1 : Form
15     {
16         [Serializable]
17         public Form1()
18         {
19             InitializeComponent();
20         }
21
22         [Serializable]
23         private void button1_Click(object sender, EventArgs e)
24         {
25             TrainTicket.ServiceClient proxy = new TrainTicket.ServiceClient();
26             proxy.GetCapacityCompleted += new EventHandler<TrainTicket.GetCapacityCompletedEventArgs>(saveResult);
27             proxy.GetCapacityAsync(14);
28         }
29
30         [Serializable]
31         static void saveResult(object sender, TrainTicket.GetCapacityCompletedEventArgs e)
32         {
33             System.IO.File.WriteAllText("r\\capacity.txt", e.Result.ToString());
34         }
35     }
36 }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
```

شکل ۱۱: نمونه کد ارتباط غیرهمزمان با وب سرویس