

Contents

CAPITOLUL 4	2
Servicii Web	2
1. Introducere	2
2. Crearea unei aplicații Web Service	3
Adaugarea unei descrieri pentru serviciul Web și pentru metodele acestuia.....	7
3. Crearea unui client de tip Console Application.....	8
Exerciții.....	11
Tema	12

CAPITOLUL 4

Servicii Web

1. Introducere

Un serviciu web este o colecție de protocoale și standarde utilizate pentru comunicarea între două dispozitive prin *Internet*. Acesta se bazează pe modelul client-server, astfel: serverul expune unele metode și clientul se conectează la server pentru a utiliza metodele. Ideea este de a crea componente, precum black box, care pot interacționa, fără a lua în considerare sistemul de operare sau limbajul de programare. Mai precis, serviciul web este o componentă sau un modul de cod executabil cu o interfață specială care face ca metodele să fie disponibile pentru alte programe utilizând o solicitare http. Cererea se face prin utilizarea http get, http post și simple Object Access Protocol (SOAP).

În figura de mai jos este prezentat modul de funcționarea a aplicațiilor client - server.

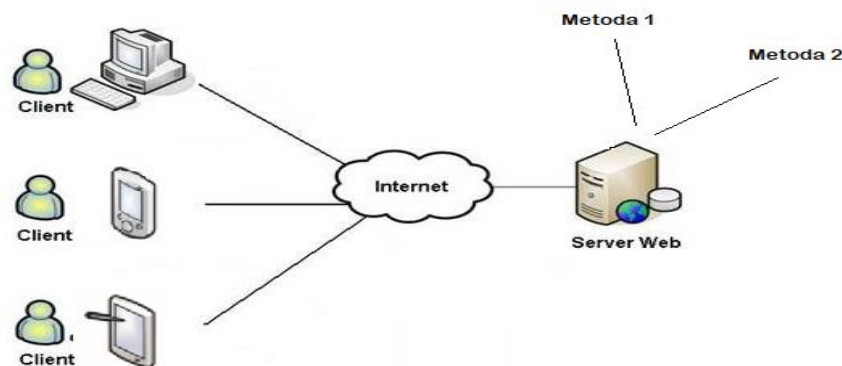


Fig. 1 Modul de funcționare al aplicațiilor client -server

Prin urmare, metodele sau mesajele pot fi publicate pentru a fi utilizate de pretutindeni. Pentru codarea datelor, serviciul web folosește Extensible Markup Language (XML) și pentru transportul acestuia, un protocol bazat pe XML numit SOAP. SOAP este folosit în aplicații care necesită schimb de informații prin HTTP. Un serviciu web poate fi localizat utilizând limbajul de definire a serviciului Web (WSDL). Arhitectura serviciilor web este prezentată în figura de mai jos.

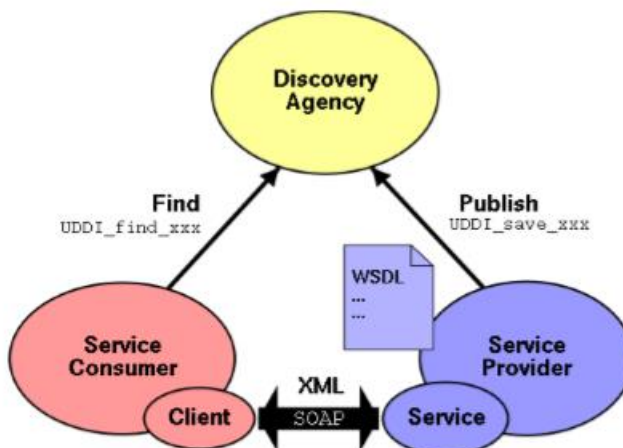


Fig. 2 Arhitectura serviciilor web

Serviciile Web aduc următoarele **avantaje** în cadrul unei aplicații distribuite:

- **Simplitate:** serviciile Web pot fi suportate de toate platformele întrucât folosesc protocoale cunoscute („open standards”) global;
- **Scalabilitate ridicată:** întrucât, un serviciu web poate răspunde la un număr mare clienți;
- Clientul nu este afectat dacă definițiile funcțiilor din interfața expusă de serviciul Web sunt modificate, totuși metodele trebuie să păstreze aceeași parametri și același tip returnat;
- **Interoperabilitatea:** un serviciu WEB poate fi folosit de către clienți implementați în limbaje de programare diferite.

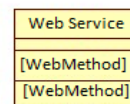
În continuare sunt prezentate câteva *exemple* de utilizare a serviciilor WEB.

- Majoritatea serviciilor Web reale sunt de tip SOAP, dar există și câteva servicii de **tip REST notabile (RESTful Web Services)** : Amazon S3 (Simple Storage Service) pentru memorarea și regăsirea de obiecte, rețeaua Twitter și alte site-uri de blog, în care se descarcă fișiere XML în format RSS sau Atom cu liste de legături către alte resurse.
- Ca exemple de **servicii SOAP (Simple Object Access Protocol)** larg utilizate sunt servicii pentru rate de schimb între diferite valute, servicii bancare pentru verificare și operare în conturi (ex: de către aplicații Web de comerț electronic), servicii de memorare și regăsire date, s.a.

2. Crearea unei aplicații Web Service

În primul rând, pentru a putea crea aplicații de tip Web, trebuie să avem adăugat pachetul de servicii necesare din Visual Studio Installer, și anume : ASP.NET and web development.

Construcția unui serviciu Web începe de la o clasă ce conține una sau mai multe metode web și cel puțin una dintre aceste metode va fi publicată și expusă prin interfața Web a serviciului.



Pentru a crea un serviciu Web, utilizând mediul de dezvoltare Visual Studio, putem opta între cele două posibilități: fie selectăm din Start Page **Create new project...**, fie din bara de meniu, după cum urmează: File → New → Project → Visual C# → Web → **ASP.NET Web Application**, precum în figura 3 și 4. După crearea proiectului, toate fișierele din cadrul proiectului pot fi găsite în Solution Explorer.

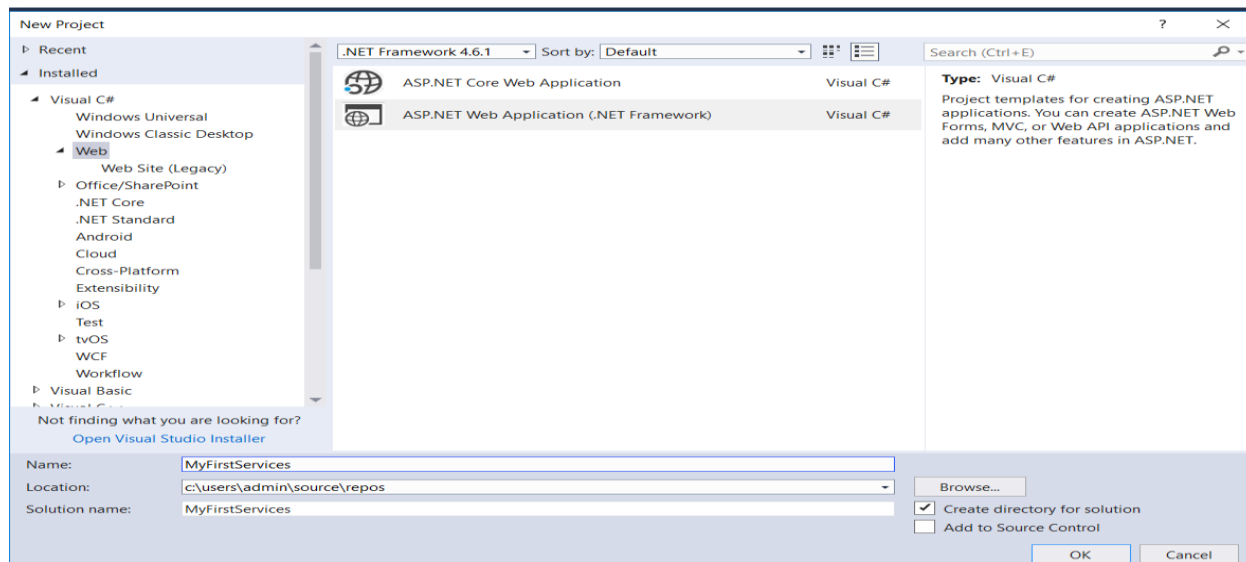


Fig.3 Crearea unui proiect de tip Web Service

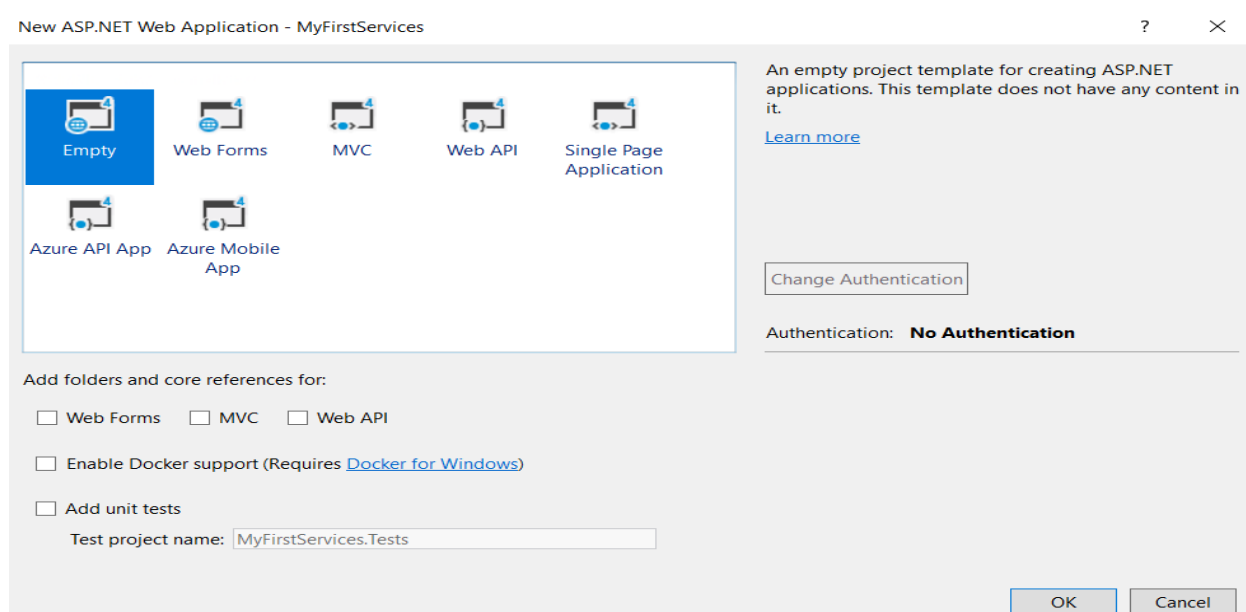


Fig.4 Crearea unui proiect de tip Web Service

După crearea aplicației de tip Web trebuie să ne adăugăm serviciul pentru a putea crea metode necesare. Acest lucru se realizează precum în figurile ce urmează (fig. 5, fig.6).

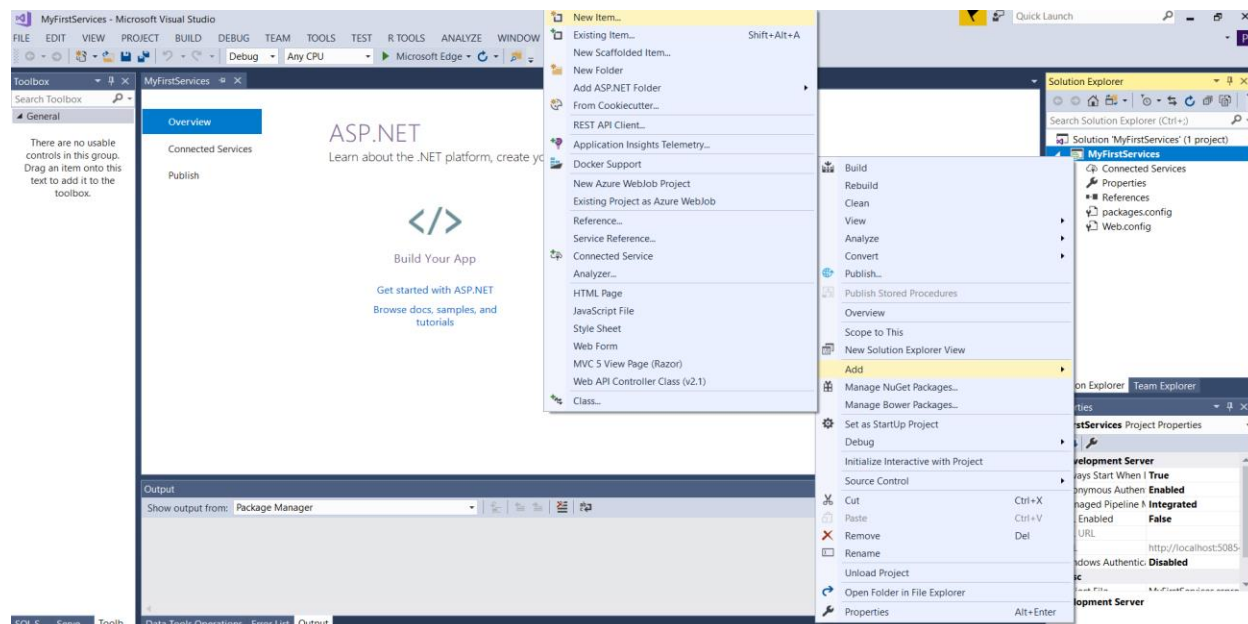


Fig.5 Adăugarea serviciului WEB

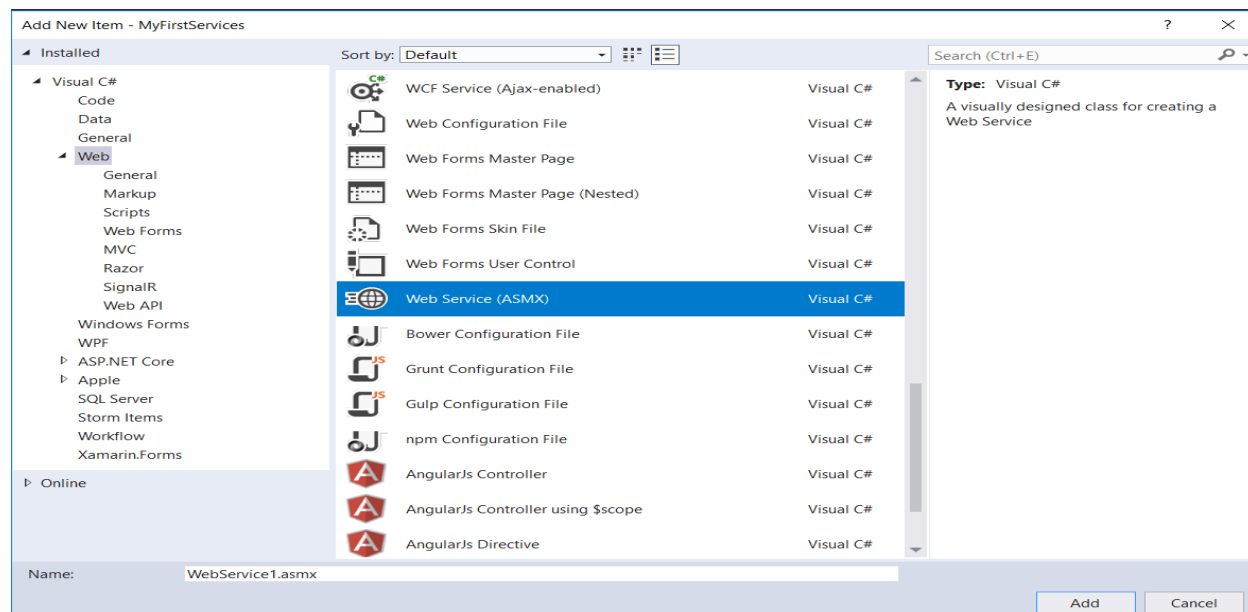



Fig.6 Adăugarea serviciului WEB

După adăugarea serviciului, în clasa `WebService1` este adăugată în mod automat prima metodă și anume, metoda ***HelloWorld()***, după cum putem observa din figura 7. Următorul pas îl constituie expunerea serviciului în interfața Web apăsând butonul Run și selectând browser-ul dorit, în cazul nostru acesta este *Microsoft Edge* ( **Microsoft Edge**).

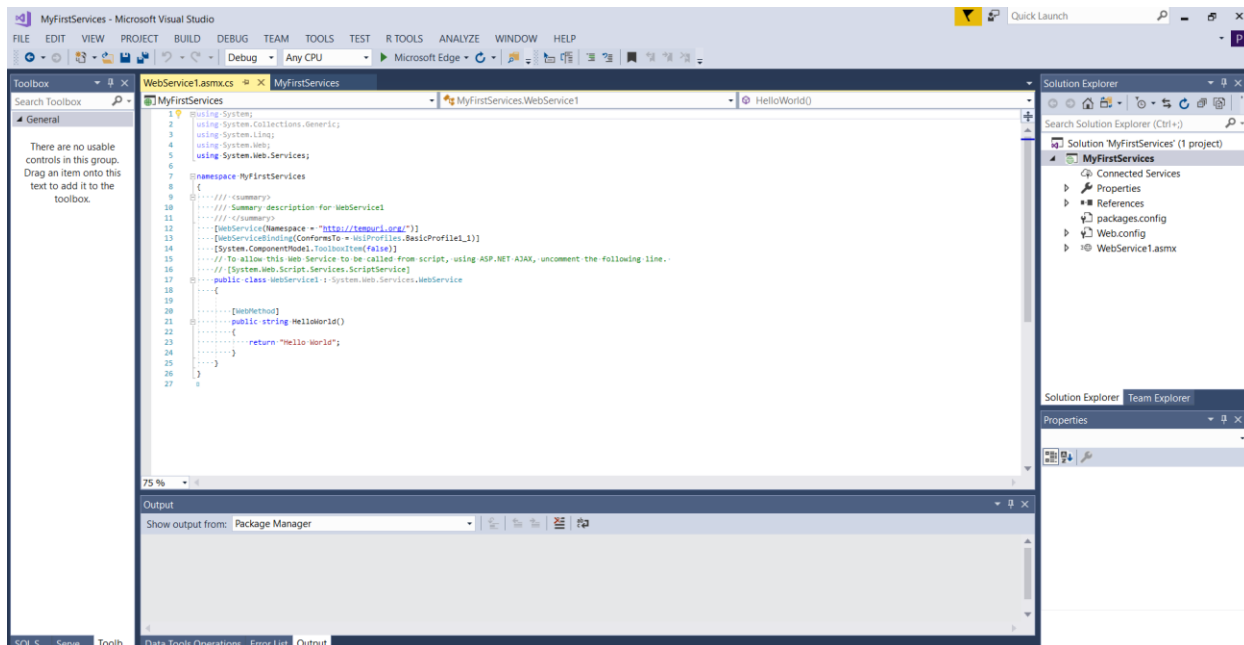


Fig.7 Codul sursă generat automat

Nota!: Testarea serviciului Web se face folosind un browser Web în care introducem adresa către fișierul .asmx. În browser v-a apărea o pagină de test generată, de runtime-ul ASP.NET. Înaintea testării aplicației în browser nu există nici o altă modalitate de a executa codul metodei serviciului Web.

În urma rulării serviciului web obținem:

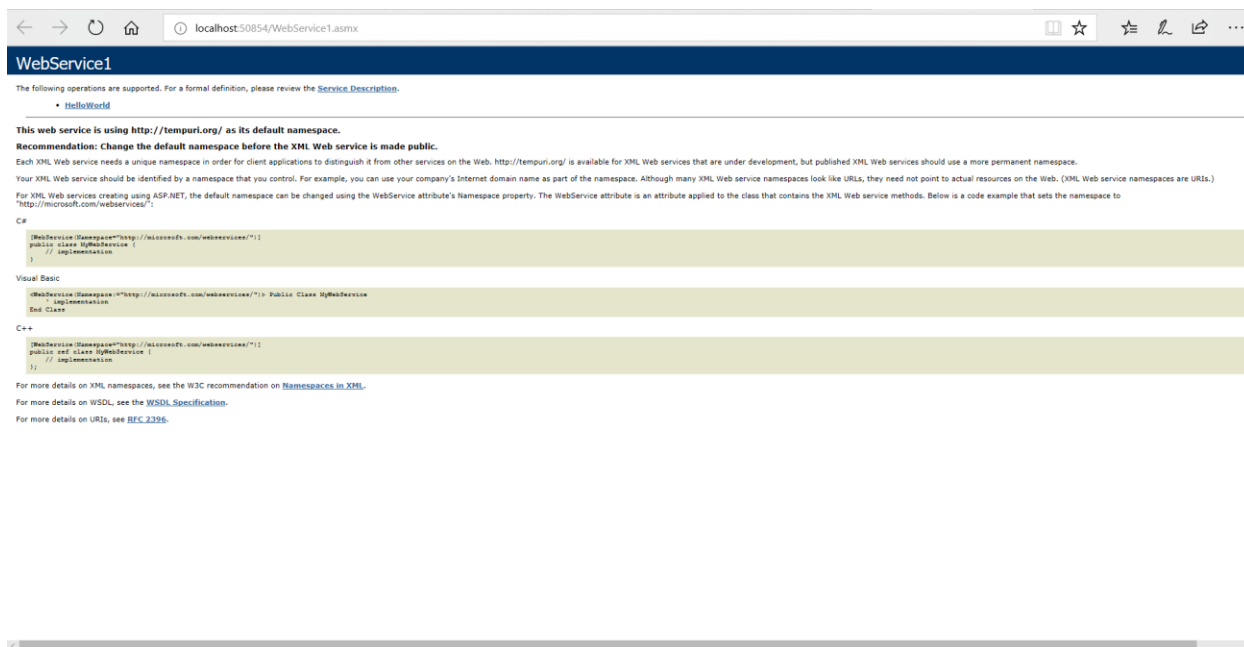


Fig. 8 Rularea serviciului Web

Observație!: Pentru ca o metodă să poată fi apelată (vizibilă) în aplicația client, aceasta trebuie declarată ca metodă web prin intermediul entității **[WebMethod]**. Pentru fiecare metoda în parte trebuie adăugată această directivă.

Adaugarea unei descrieri pentru serviciul Web și pentru metodele acestuia

ASP.NET oferă posibilitatea descrierii serviciului Web și a fiecărei metode în parte. Acest lucru se poate dovedi util pentru clienții noi ai unui serviciu Web. Documentarea se face prin decorarea cu atribute atât a clasei cât și a metodelor acesteia, după cum putem observa în cele ce urmează.

```
[WebService(Description="Serviciu Web pentru Cursul de C#",  
    Name="My First Web Service" , Namespace = "Web Services")]  
// [WebService(Namespace = "http://tempuri.org/")]
```

```
[WebMethod(Description = "This method displays a message")]  
public string HelloWorld()  
{  
    return "Hello World";  
}
```

Fig. 9 Adăugarea descrierii serviciului Web

Să adăugăm o altă metodă. De exemplu, vrem să calculăm suma a două numere. Pentru aceasta, implementăm o metodă după cum urmează:

```
[WebMethod]  
public int Add(int a, int b)  
{  
    return a + b;  
}
```

Prin construirea și derularea proiectului, vom avea primul nostru serviciu web, așa cum este prezentat în figura 10.

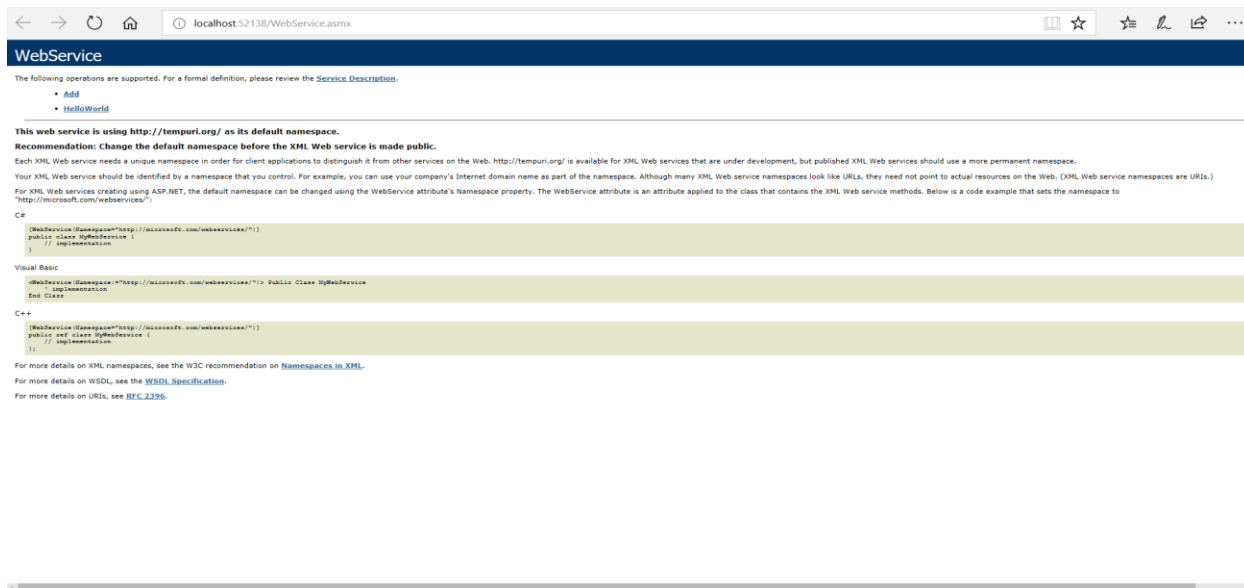


Fig. 10 Primul serviciu web

După dezvoltarea serviciului cu metodele necesare, pentru a le putea utiliza este nevoie să construim și aplicațiile client. În continuare este prezentat modul prin care ne putem dezvolta un client de tip Console Application în mediul de dezvoltare Visual Studio.

3. Crearea unui client de tip Console Application

Pentru a crea un client, deschidem o nouă fereastră Visual Studio și creăm un nou proiect de tip Console Application, precum în figura 11 și denumim proiectul myFirstClient.

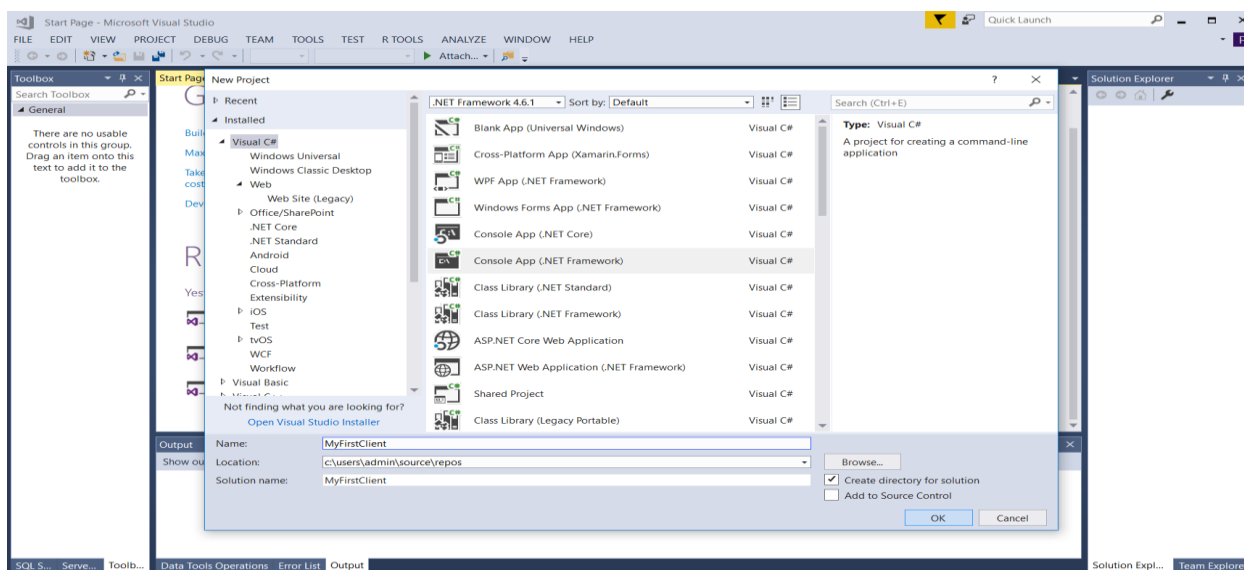


Fig. 11 Crearea clientului de tip Console Application

După crearea proiectului, așa cum spuneam, trebuie să **adăugăm referința serviciului**, mai exact să conectăm aplicația Web la client prin intermediul adresei URL a serviciului. Acest lucru îl putem observa în figura 12. După ce am obținut o referință la un serviciu Web al doilea pas, pentru un client este de a cunoaște modul de a interacționa cu serviciul adică: cunoașterea metodelor disponibile, a parametrilor și tipul acestora.

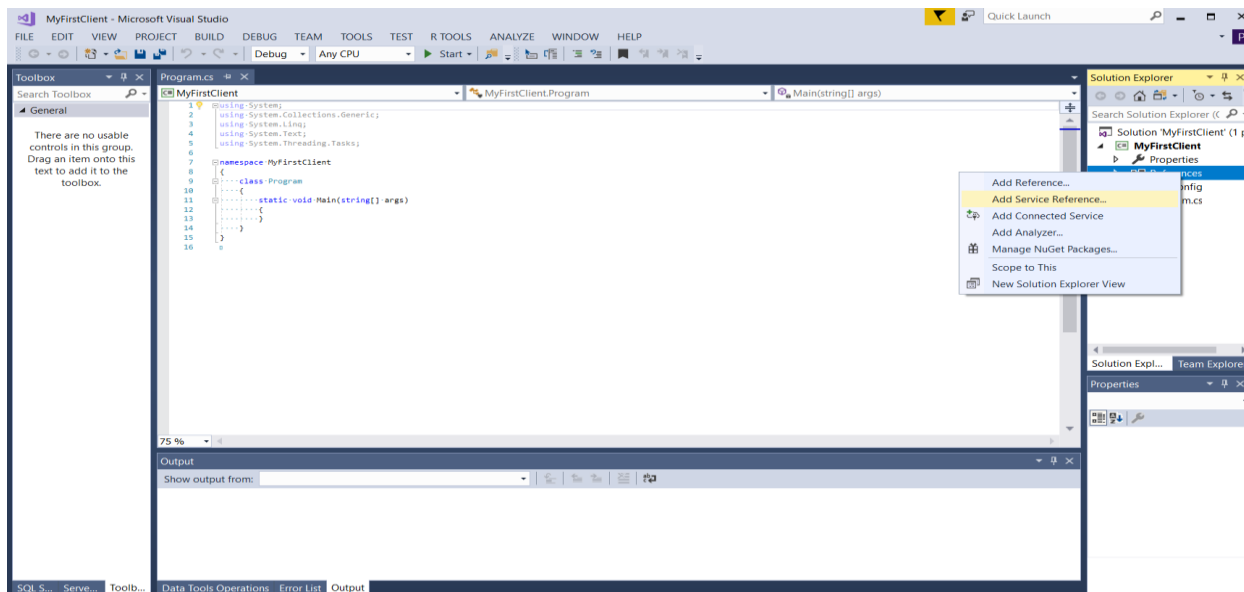


Fig. 12 Adăugarea unei referințe la serviciul WEB

Notă!: Pentru a putea adăuga serviciul în aplicația client, acesta trebuie să fie pornit (să ruleze) și mai trebuie să copiem adresa URL a acestuia din browser pentru a o introduce la pasul următor (figura 13), după care se apasă butonul Go. Dacă serviciul este găsit, acesta va apărea sub adresa URL, și vom putea vizualiza inclusiv metodele serviciului, apăsând dublu click pe numele serviciului.

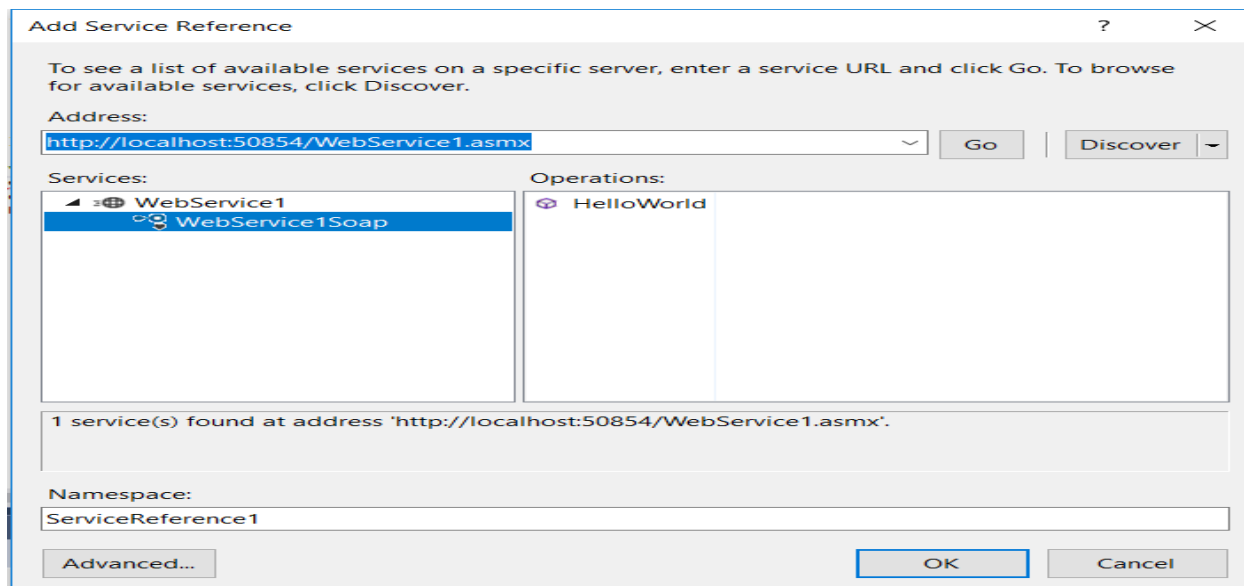


Fig. 13 Adăugarea unei referințe la serviciu WEB

Observație!: Atunci când adăugăm metode noi serviciului, pentru ca acestea să fie accesibile (vizibile) în aplicațiile client trebuie să facem update la referința serviciului. Acest lucru se poate vedea în figura ce urmează.

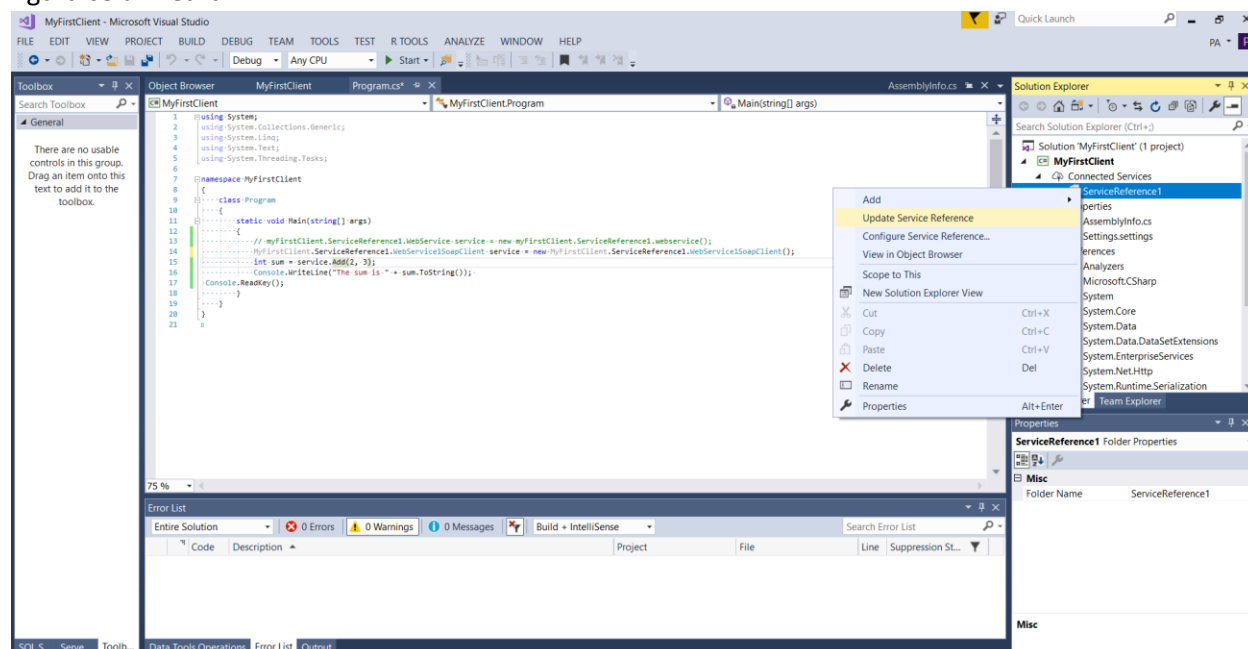


Fig. 14 Updatarea serviciului web în aplicația client

Acum, clientul este conectat la serviciu și poate utiliza metodele expuse de serviciu. În aplicația client trebuie să creem un obiect, cu ajutorul căruia apelăm metodele aplicației Web.

```
static void Main(string[] args)
{
    MyFirstClient.ServiceReference1.WebService1SoapClient service = new
    MyFirstClient.ServiceReference1.WebService1SoapClient();

    int sum = service.Add(2, 3);
    Console.WriteLine("The sum is " + sum.ToString());
    Console.ReadKey();
}
```

După implementarea codului, putem să testăm aplicația, apăsând butonul Run. Rezultatul rulării programului se poate observa în figura de mai jos.



Fig. 15 Rezultatul rulării aplicației client

Exerciții

1. Să se creeze un serviciu Web care să conțină următoarele metode :
 - Conversie F to C și invers;
 - Afișarea orei și datei exacte;
 - Afișarea unei liste cu 5 elemente;
 - Conversia între două valute (ex: lei - euro).
2. Să se creeze doi clienți unul de tip Console Application și unul de tip Windows Form Application care să utilizeze metodele serviciului.

OBS! Interfața grafică pentru aplicația de tip Win Form App ar putea să arate precum cea de mai jos:

The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar. The form contains the following elements:

- A list box labeled "listBox_Lista" on the left side.
- A button labeled "Add List" positioned below the list box.
- Three text input fields on the right side, labeled "Temp C", "Temp F", and "Rezultat" from top to bottom.
- Two buttons, "FtoC" and "CtoF", located below the "Rezultat" field.
- A label "Data" on the left side, below the "Add List" button.
- A label "Euro to Ron :" followed by two stacked text input fields.

Tema: Creați un serviciu Web care oferă acces la o bază de date care stochează informații. Apoi, creați un client de tip Windows Form App care utilizează serviciul. Serviciul va oferi metode de adăugarea și modificare a informațiilor. Tema aplicației este la alegere.