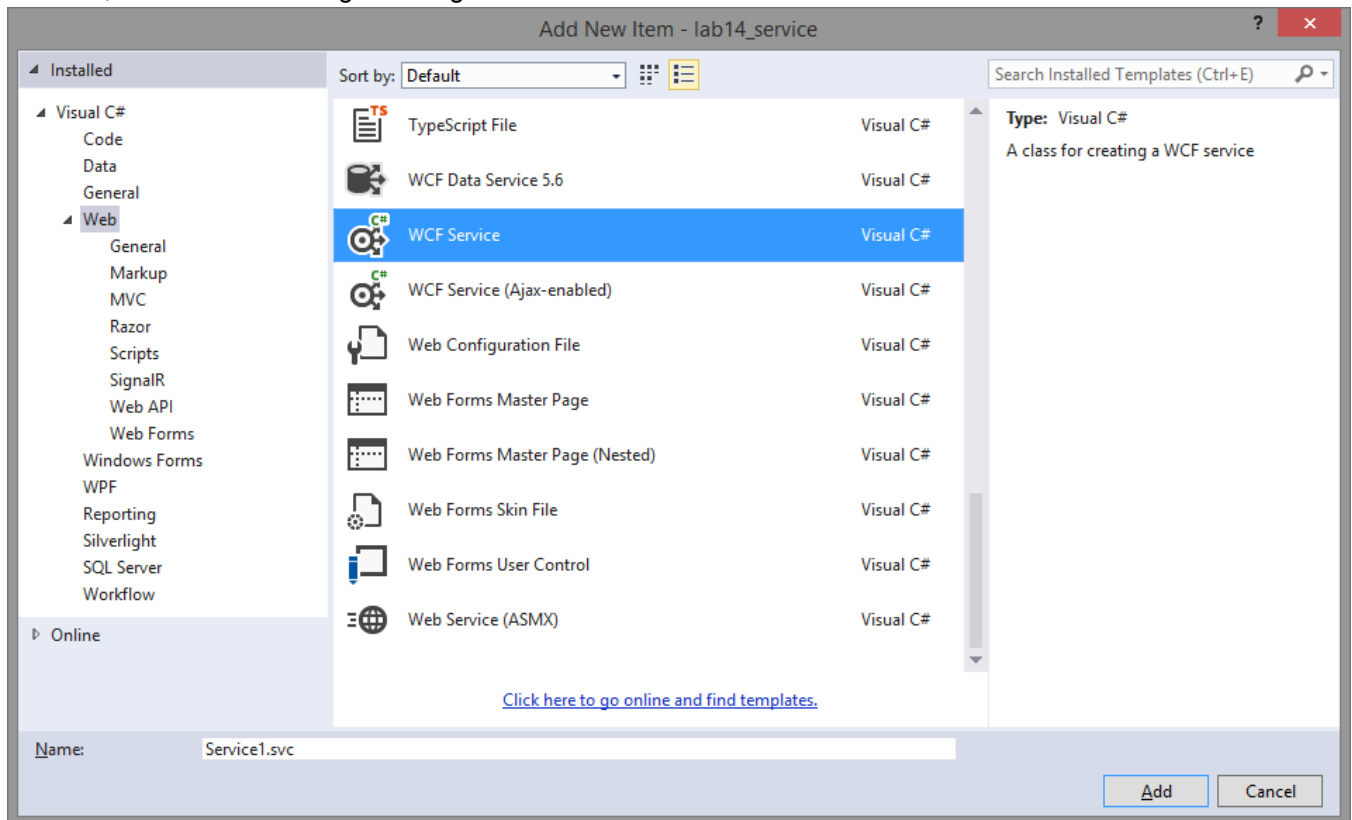


LAB 18: ASP.NET e Web Services

Este lab apresenta a criação e utilização de **Web Services** usando **C#** e aplicação em uma *página web*.

Exercício 1 – Criando um Web Service simples

1. Crie um novo projeto utilizando o template ASP.NET Empty Web Application com o nome “LAB14_Service”.
2. Clique com o botão direito sobre o projeto e adicione um novo item de acordo com o template “WCF Service”, de acordo com a figura a seguir.



3. Abra o arquivo “*IService1.cs*” via **Solution Explorer**. Altere o código da interface para se parecer com o código a seguir.

```
[ServiceContract]
public interface IService1
{
    [OperationContract]
    string SimpleMethod(string s);
}
```

4. Abra o arquivo “*Service1.svc*” via **Solution Explorer**. Altere o código da classe para se parecer com o código a seguir.

```
public class Service1 : IService1
{
    public string SimpleMethod(string s)
    {
        return "Hello " + s;
    }
}
```

5. Para verificar o web service, clique com o botão direito sobre o arquivo “Service1.svc” e selecione “View in Browser”. OVS abre uma página *Webcom* as informações sobre o serviço.

Você criou um serviço.

Para testar esse serviço, você precisará criar um cliente e usá-lo para chamar o serviço. Você pode fazer isso usando a ferramenta svcutil.exe na linha de comando com a seguinte sintaxe:

```
svcutil.exe http://localhost:11884/Service1.svc?wsdl
```

Você também pode acessar a descrição do serviço como um único arquivo:

```
http://localhost:11884/Service1.svc?singleWsdL
```

Isto gerará um arquivo de configuração e um arquivo de código que contém a classe cliente. Adicione os dois arquivos ao seu aplicativo cliente e use a classe cliente gerada para chamar o Serviço. Por exemplo:

C#

```
class Test
{
    static void Main()
    {
        ServiceClient client = new ServiceClient();

        // Use a variável 'client' para chamar as operações no serviço.

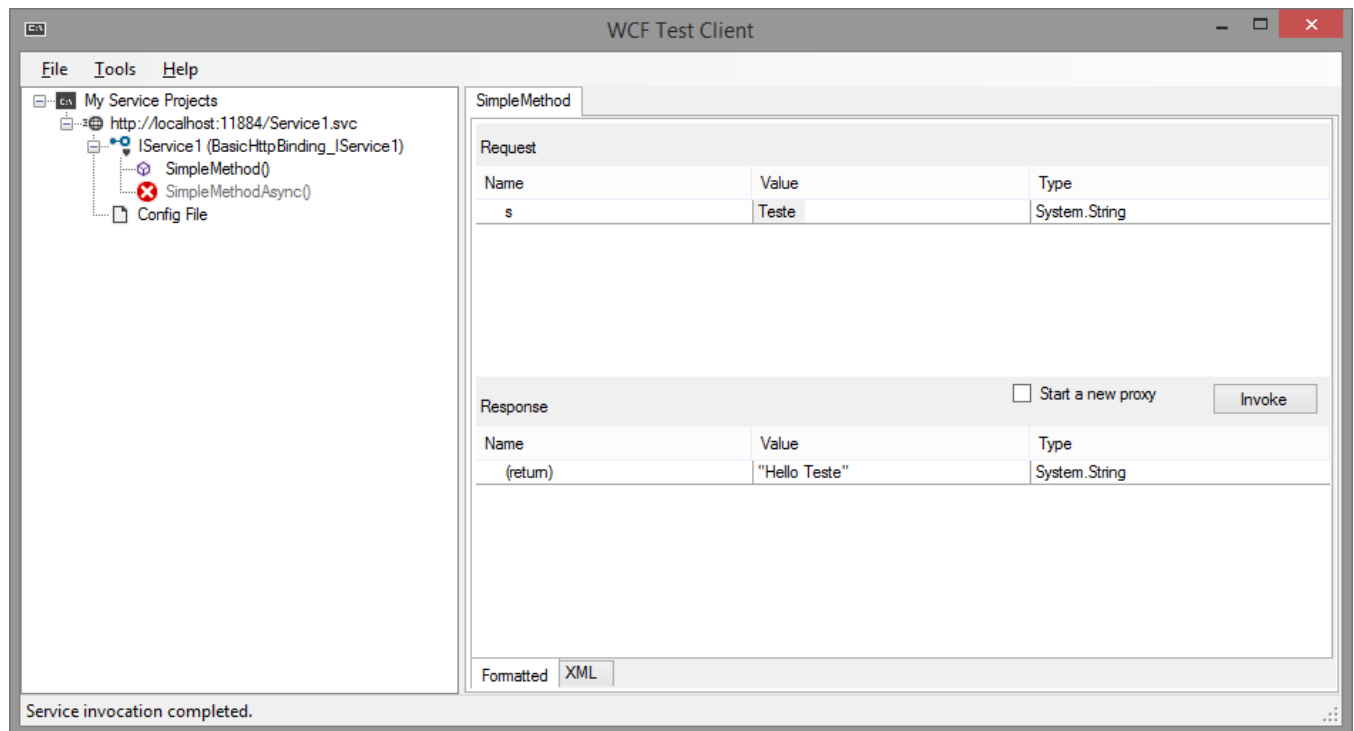
        // Sempre feche o cliente.
        client.Close();
    }
}
```

Visual Basic

```
Class Test
    Shared Sub Main()
        Dim client As ServiceClient = New ServiceClient()
        ' Use a variável 'client' para chamar as operações no serviço.

        ' Sempre feche o cliente.
        client.Close()
    End Sub
End Class
```

6. Para visualizar um aplicativo de teste, selecione o arquivo “Service1.svc” e clique em F5. Selecione o método que deseja testar, preencha os dados de envio e clique no botão “invoke”. Veja um exemplo na figura a seguir.



Exercício 2 – Acessando o Web Service

Este exercício ilustra a utilização (consumo) do **Web Service** anteriormente criado a partir de uma página *web*.

1. Crie um novo projeto do tipo **ASP.NET Empty Web Application** com o nome "LAB14_WebApp" na mesma solução onde foi criado o projeto anterior.
2. Para criar uma referência ao **Web Service**, selecione o comando de menu **Project | Add Service Reference**.
3. A janela de diálogo **Add Service Reference** aparecerá.

To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

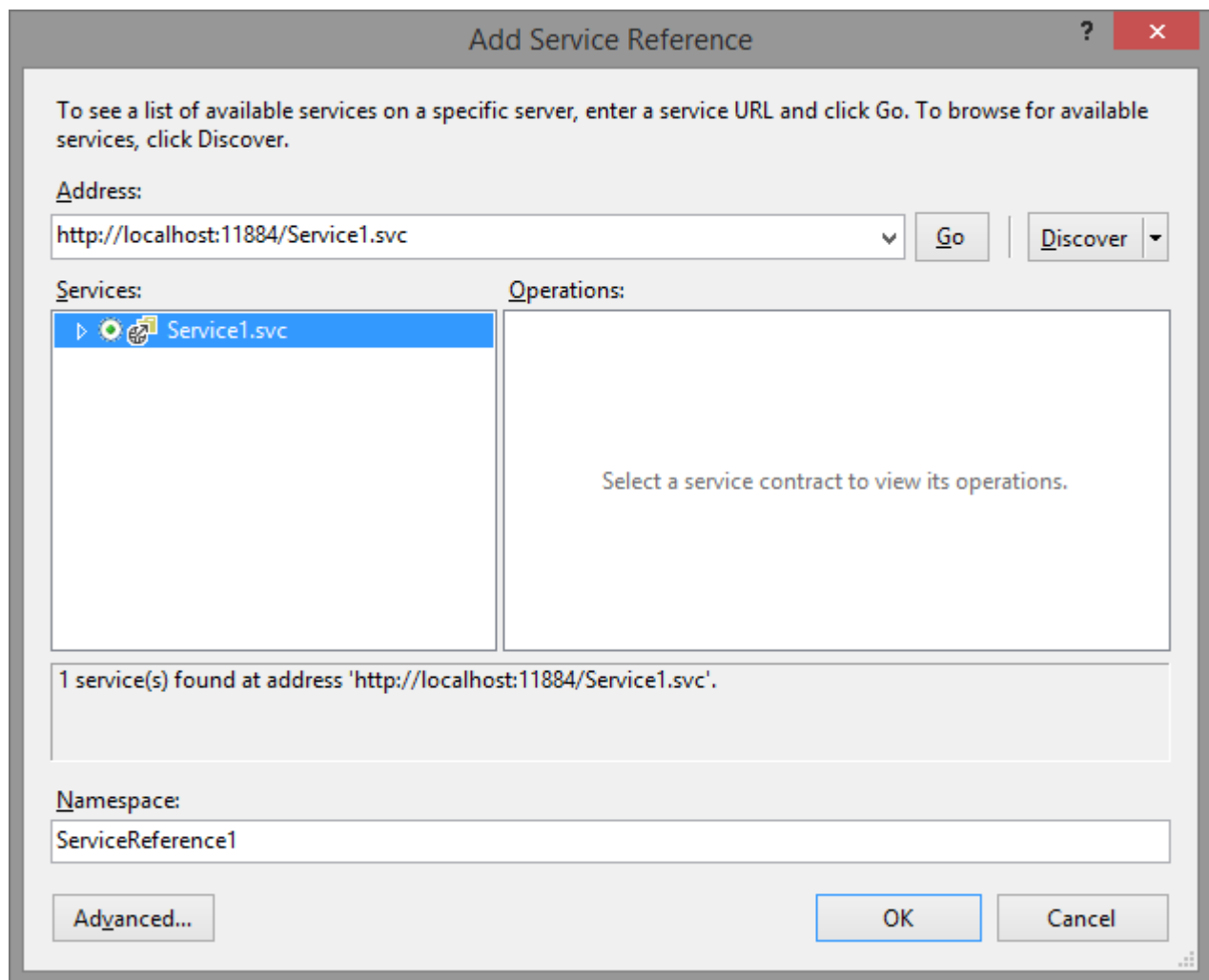
Address:

Services: **Operations:**

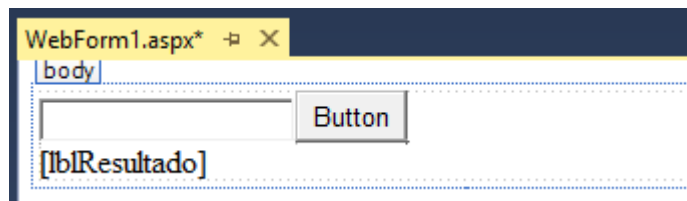
Namespace:

ServiceReference1

4. Clique **Discover** e selecione "Service1" OU digite a **URL do Web Service** que você criou. Clique em **Go**.



5. Para referenciar o **Web Service** clique no botão **OK**.
6. O projeto **WebApplication** terá uma novapasta chamada **Service References**. Estapasta conterá um nó **ServiceReference1** que é o nome do serviço referenciado.
7. Adicione uma nova página ao projeto.
8. Coloque um controle **Button** na página criada. Adicione também uma **Label** (ID: **lblResultado**) e uma **TextBox**.



9. Dê um duplo-clique no controle **Button** para acessar o método manipulador do evento **Click** deste botão.

10. Crie uma instância de sua classe de Proxy (“*ServiceReference1.Service1Client*”) e chame o método “*SimpleMethod()*”, passando como parâmetro uma string (a que for digitada no **textBox**). Mostre o retorno desta função na **label**. O método “*Button1_Click()*” deverá estar parecendo conforme segue:

```
protectedvoid Button1_Click(object sender, EventArgs e)
{
    ServiceReference1.IService1ws = newServiceReference1.Service1Client();
    lbResultado.Text = ws.SimpleMethod(TextBox1.Text);
}
```

11. Para compilar e rodar sua aplicação selecione o arquivo “WebForm1.aspx” e com o botão direito clique no menu “View in Browser”. Para testar o **Web Service**, preencha a **TextBox** clique no **Button**. O **Label** deve conter o texto “Hello “ seguido do texto digitado na **TextBox**.

Exercício 3 –Tipos Complexos de Dados

Este exercício mostra o retorno de um objeto complexo (array de objetos) a partir de um *web service*.

1. Clique com o botão direito sobre a solução e selecione **Add New Project** com o tipo **Class Library** e nome “*LAB14_ClassLibrary*”.
2. Adicione uma classe “*JogoFutebol*” ao projeto **ClassLibrary**.
3. Declare *quatro variáveis* para a classe como mostrado abaixo: duas do tipo **string** chamadas *casa* e *visitante* e duas do tipo **short** chamadas *golCasa* e *golVisitante*.

```
publicclassJogoFutebol
{
    publicshortgolCasa{get;set;}
    publicshortgolVisitante{get;set;}
    publicstring casa{get;set;}
    publicstringvisitante{get;set;}
}
```

4. Adicione uma referência ao projeto **ClassLibrary** no projeto “*LAB14_Service*”.
5. Abra o arquivo “*IService1.cs*”, adicione um novo método de acordo com o código a seguir:

```
[OperationContract]
JogoFutebol[] ComplexMethod();
```

6. Abra o arquivo “*Service1.svc*”, adicione um novo método de acordo com o código a seguir:

```
public JogoFutebol[] ComplexMethod()
{
    JogoFutebol[] resultadoJogo = new JogoFutebol[2];

    resultadoJogo[0] = new JogoFutebol();
    resultadoJogo[0].casa = "Barcelona";
    resultadoJogo[0].golCasa = 6;
    resultadoJogo[0].visitante = "Milan";
    resultadoJogo[0].golVisitante = 1;

    resultadoJogo[1] = new JogoFutebol();
    resultadoJogo[1].casa = "Barcelona";
    resultadoJogo[1].golCasa = 5;
    resultadoJogo[1].visitante = "Chelsea";
    resultadoJogo[1].golVisitante = 3;

    return resultadoJogo;
}
```

7. Teste o *Web Service* para observar como os objetos retornados do **Web Service** são serializados em **XML**.
8. Para compilar e testar o *Web Service*, clique com o botão direito do mouse sobre o projeto **WebService** e selecione **Set as StartupProject**. Pressione **CTRL+F5**.
9. Para testar o método clique no link "*ComplexMethod*" e então clique em **Invoke**. Examine como os objetos e o array são **serializados** em **XML**:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <ComplexMethodResponse xmlns="http://tempuri.org/">
      <ComplexMethodResult
        xmlns:a="http://schemas.datacontract.org/2004/07/LAB14_ClassLibrary"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:JogoFutebol>
          <a:casa>Barcelona</a:casa>
          <a:golCasa>6</a:golCasa>
          <a:golVisitante>1</a:golVisitante>
          <a:visitante>Milan</a:visitante>
        </a:JogoFutebol>
        <a:JogoFutebol>
          <a:casa>Barcelona</a:casa>
          <a:golCasa>5</a:golCasa>
          <a:golVisitante>3</a:golVisitante>
          <a:visitante>Chelsea</a:visitante>
        </a:JogoFutebol>
      </ComplexMethodResult>
    </ComplexMethodResponse>
  </s:Body>
</s:Envelope>

```

10. Localize a **ServiceReferenceServiceReference1** no projeto Web (**WebApplication**).
11. Dê um clique com o botão direito do mouse no nó **ServiceReference1** na pasta **Service References** e selecione **UpdateService Reference**. Isto irá atualizar a aplicação **WebApplication** de maneira a usar a nova funcionalidade de “**Service1**” e irá atualizar a classe de **Proxy** para incluir o novo método que você criou (“**ComplexMethod**”)
12. Para acessar a visualização de projeto do formulário dê um duplo-clique em “**Default.aspx**” na **Solution Explorer**.
13. Adicione a página uma **GridView** (**ID= “grvResultado”**) um segundo controle **button**
14. Dê um clique duplo sobre o botão que acabou de ser adicionado. Digite o seguinte código:

```

protectedvoid Button2_Click(object sender, EventArgs e)
{
    ServiceReference1.IService1ws = newServiceReference1.Service1Client();
    grvResultado.DataSource = ws.ComplexMethod();
    grvResultado.DataBind();
}

```

15. Abra a página “**WebForm1.aspx**” no navegador e teste a nova funcionalidade.