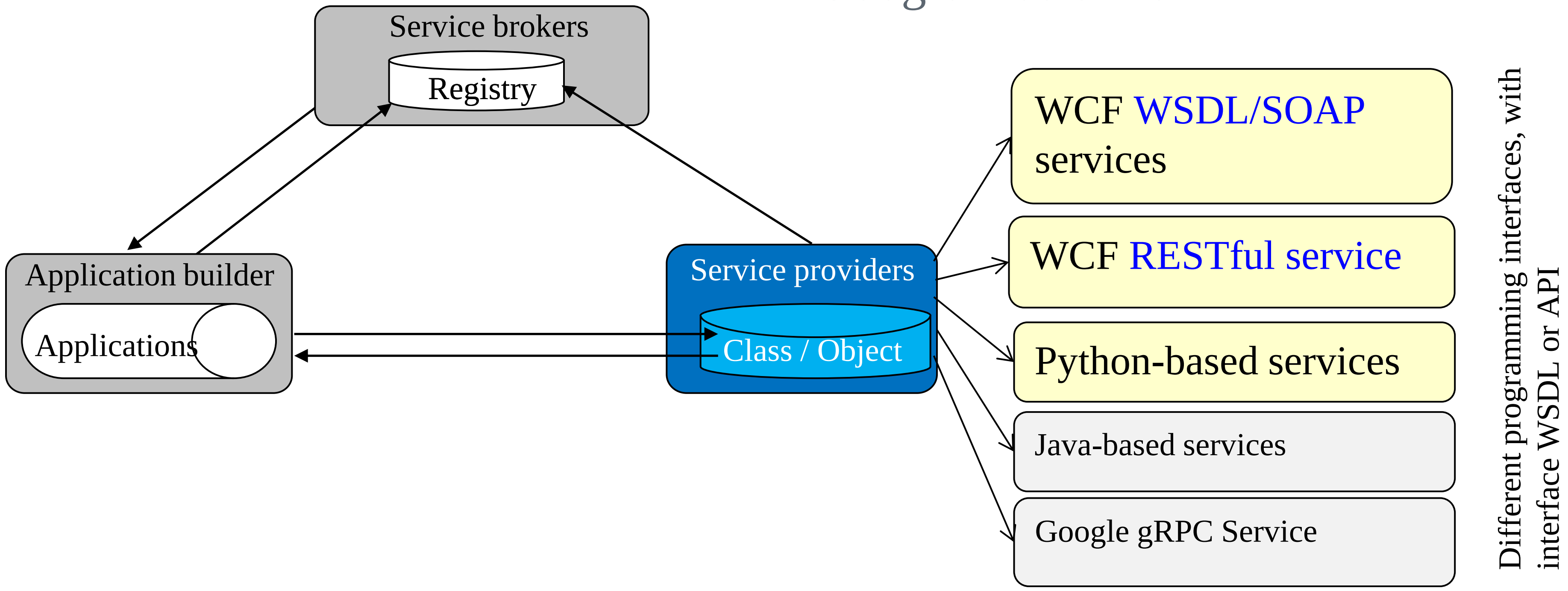
Service and Application Development Example

Ira A. Fulton Schools of Engineering
Arizona State University

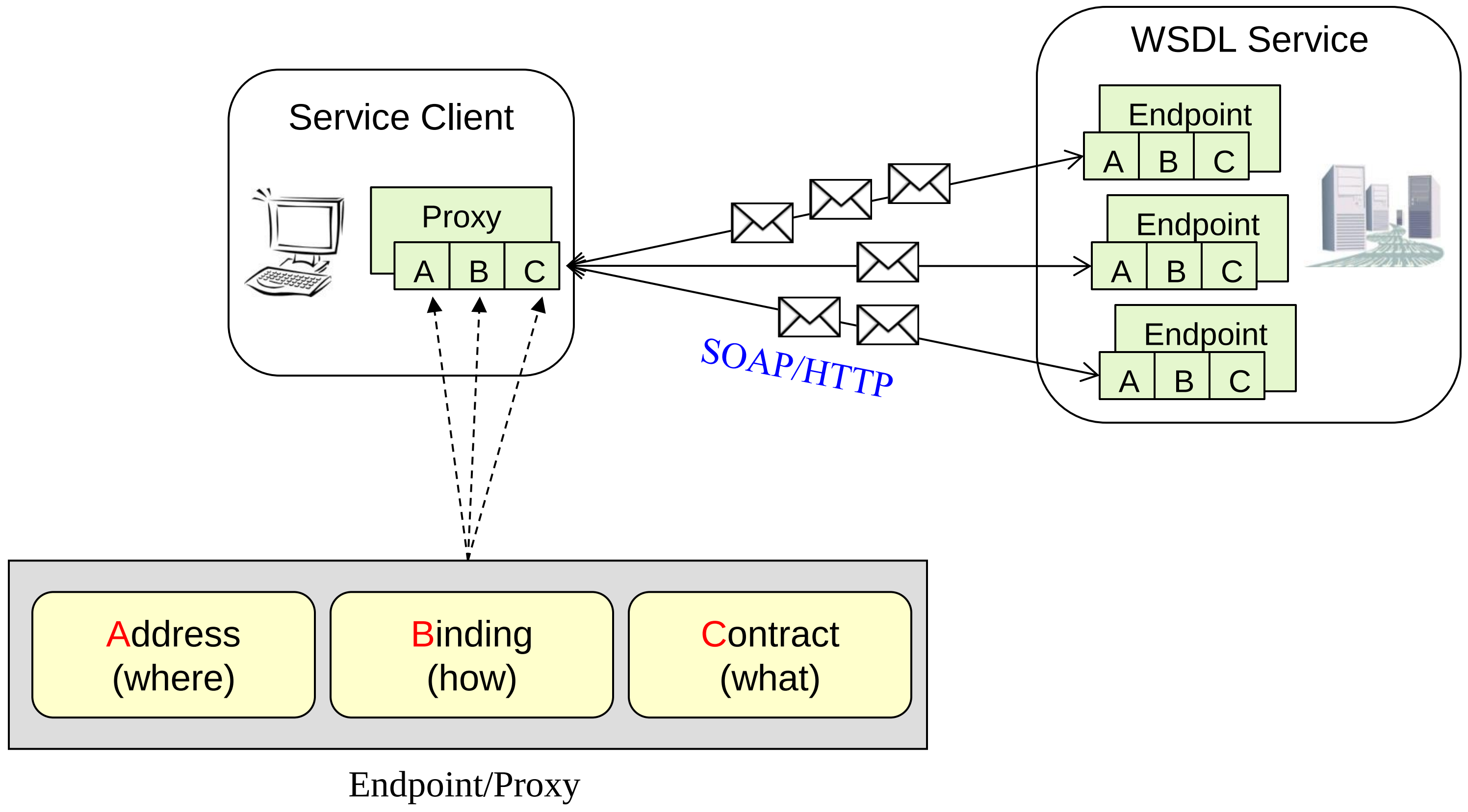
Service Development

Web Services can be developed using any major development environment:

- Visual Studio Platforms
- Python Platforms
- Java Platforms
- Google Platforms



Developing WSDL Web Services

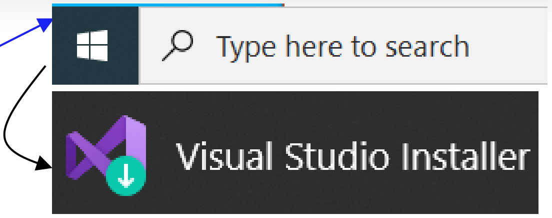


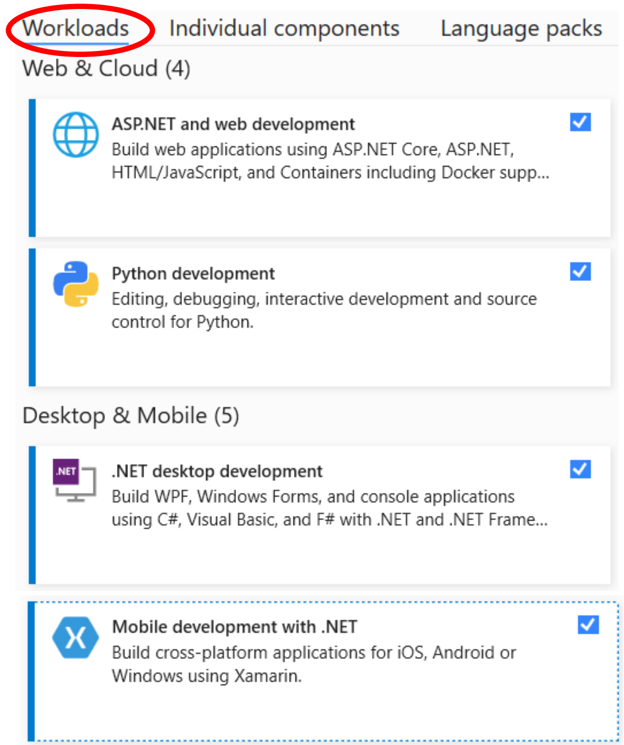
Install Visual Studio

1. **Download and Install Visual Studio Community version (free):**

   https://visualstudio.microsoft.com/downloads/

2. **Type Visual Studio Installer and make sure you install the following components:**



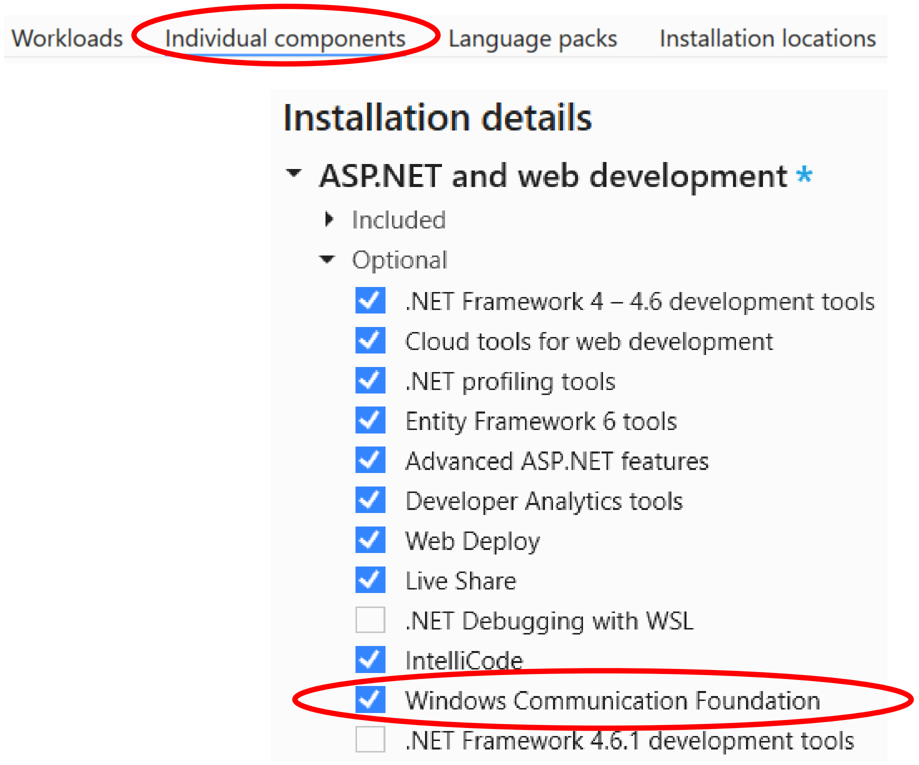

Workloads | Individual components | Language packs

Web & Cloud (4)

- ✓ **ASP.NET and web development** — Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
- ✓ **Python development** — Editing, debugging, interactive development and source control for Python.

Desktop & Mobile (5)

- ✓ **.NET desktop development** — Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...
- ✓ **Mobile development with .NET** — Build cross-platform applications for iOS, Android or Windows using Xamarin.

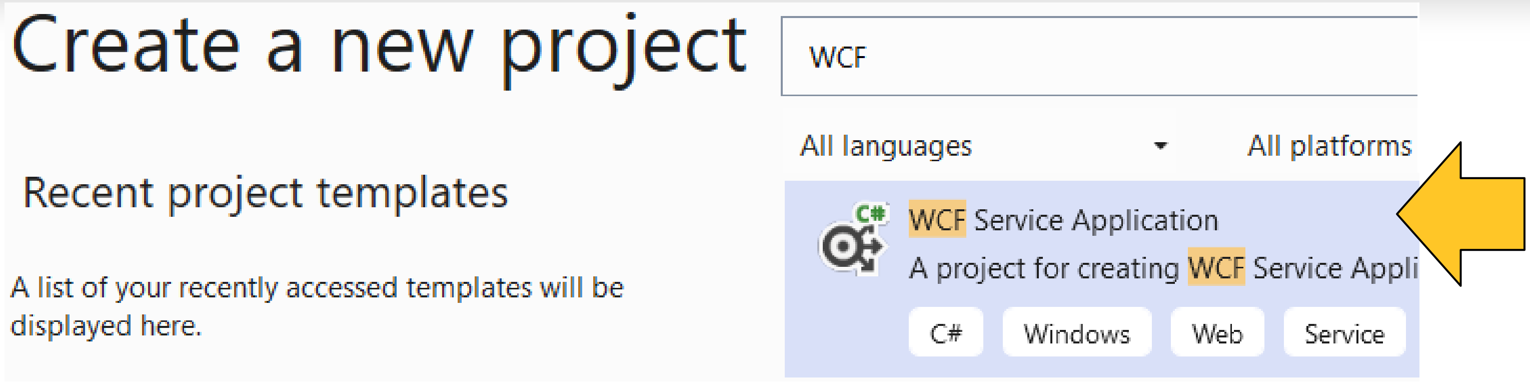

Workloads | Individual components | Language packs | Installation locations

Installation details

ASP.NET and web development *

- Included
- Optional
  - [✓] .NET Framework 4 – 4.6 development tools
  - [✓] Cloud tools for web development
  - [✓] .NET profiling tools
  - [✓] Entity Framework 6 tools
  - [✓] Advanced ASP.NET features
  - [✓] Developer Analytics tools
  - [✓] Web Deploy
  - [✓] Live Share
  - [ ] .NET Debugging with WSL
  - [✓] IntelliCode
  - [✓] Windows Communication Foundation
  - [ ] .NET Framework 4.6.1 development tools

Create a WCF Project



Create a new project

WCF

All languages ▾ All platforms

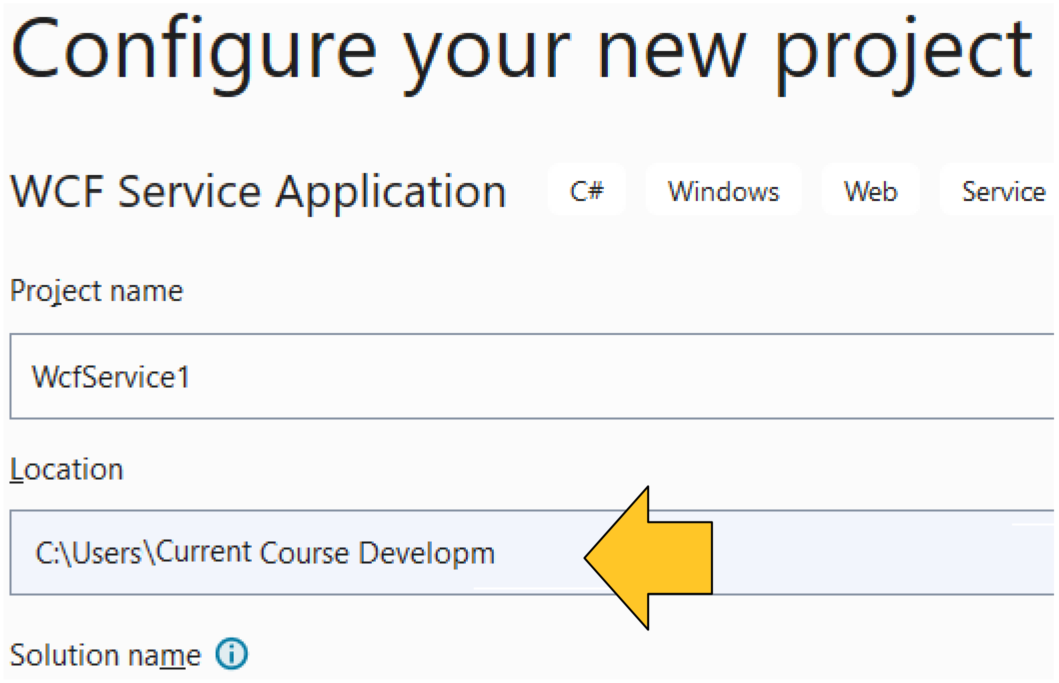
Recent project templates

A list of your recently accessed templates will be displayed here.

WCF Service Application

A project for creating WCF Service Appli

C# Windows Web Service



Configure your new project

WCF Service Application C# Windows Web Service

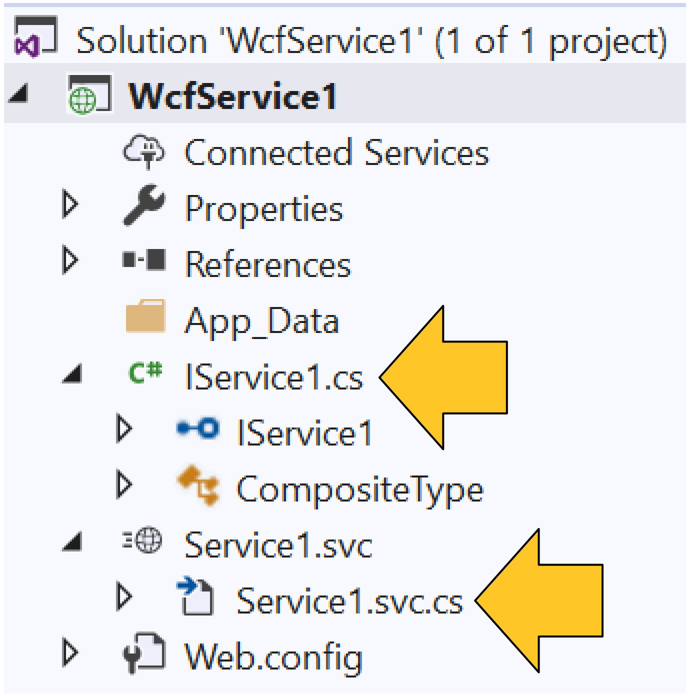
Project name

WcfService1

Location

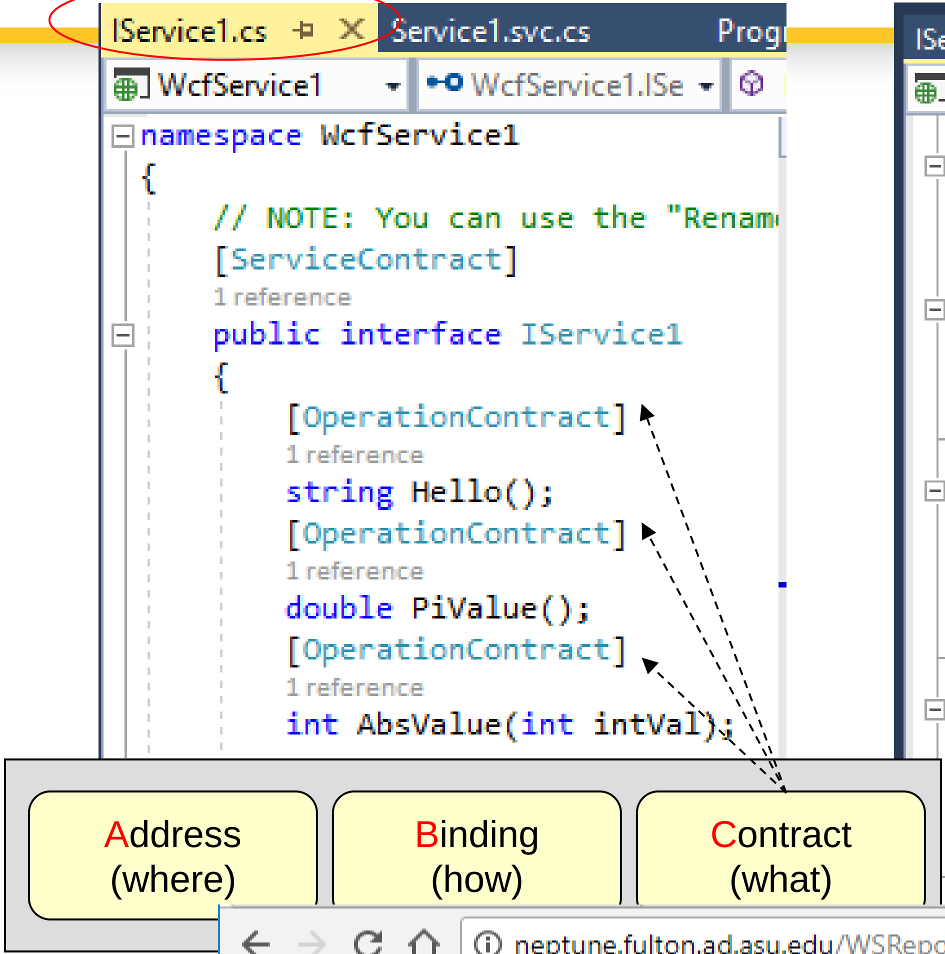
C:\Users\Current Course Developm

Solution name

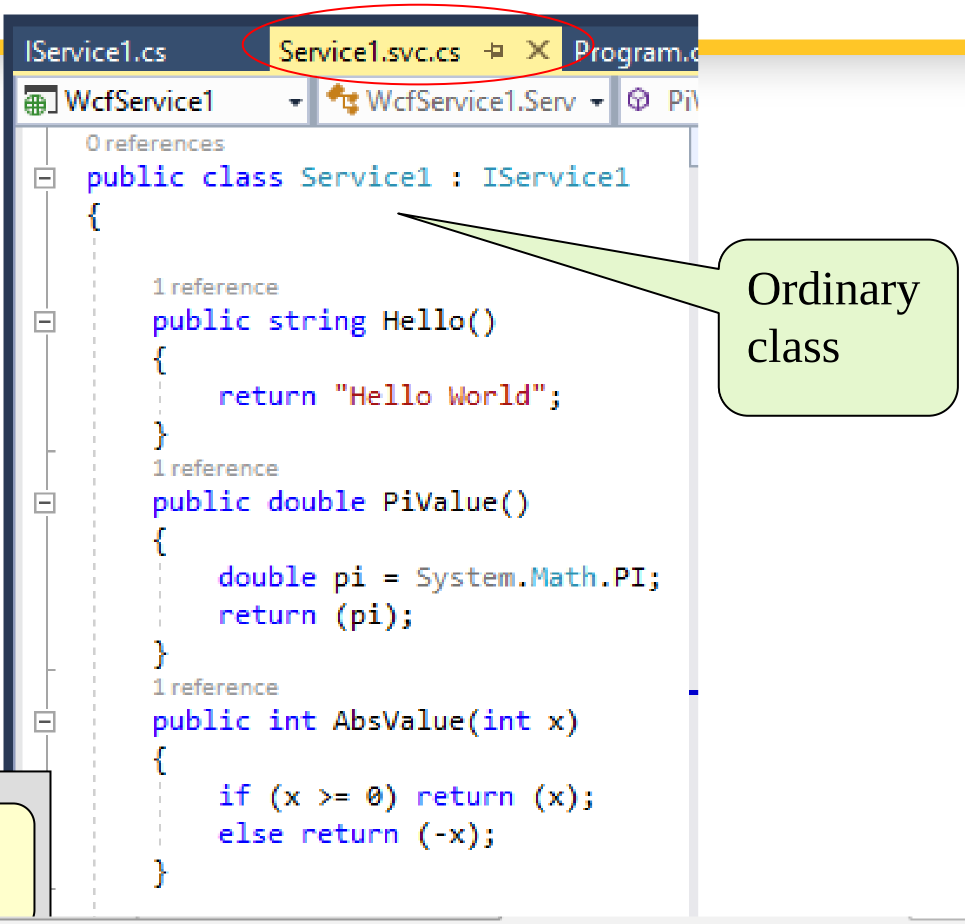


```
Solution 'WcfService1' (1 of 1 project)
◢ WcfService1
    Connected Services
  ▷ Properties
  ▷ References
    App_Data
  ◢ IService1.cs
    ▷ IService1
    ▷ CompositeType
  ◢ Service1.svc
    ▷ Service1.svc.cs
  ▷ Web.config
```

IService.cs and Service.cs Files

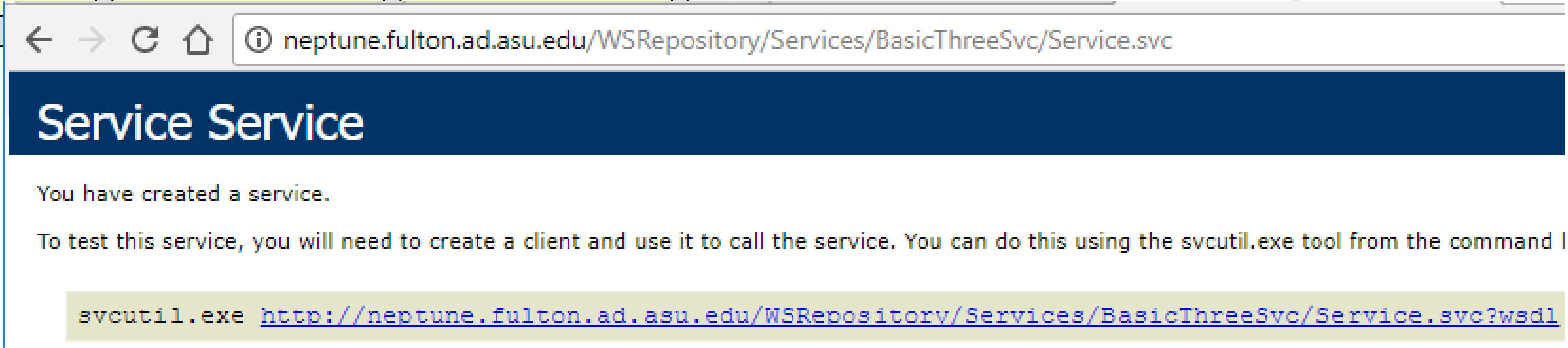


```
namespace WcfService1
{
    // NOTE: You can use the "Renam
    [ServiceContract]
    1 reference
    public interface IService1
    {
        [OperationContract]
        1 reference
        string Hello();
        [OperationContract]
        1 reference
        double PiValue();
        [OperationContract]
        1 reference
        int AbsValue(int intVal);
```



```
0 references
public class Service1 : IService1
{

    1 reference
    public string Hello()
    {
        return "Hello World";
    }
    1 reference
    public double PiValue()
    {
        double pi = System.Math.PI;
        return (pi);
    }
    1 reference
    public int AbsValue(int x)
    {
        if (x >= 0) return (x);
        else return (-x);
    }
```



neptune.fulton.ad.asu.edu/WSRepository/Services/BasicThreeSvc/Service.svc

Service Service

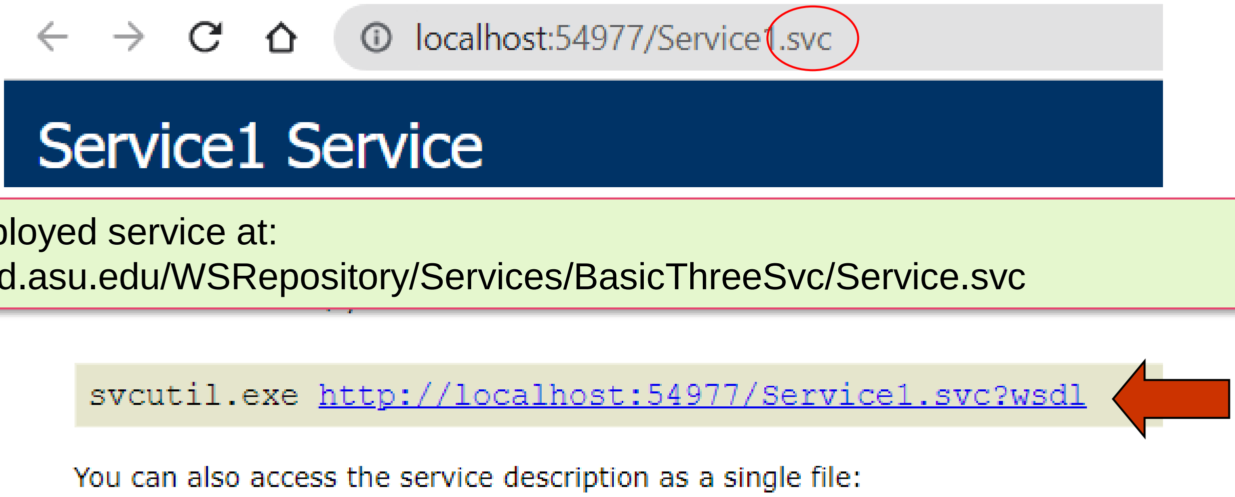
You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command l

```
svcutil.exe http://neptune.fulton.ad.asu.edu/WSRepository/Services/BasicThreeSvc/Service.svc?wsdl
```

Viewing and Testing a .svc service

(1) In Visual Studio, right click service file "Service1.svc" and choose: View in Browser



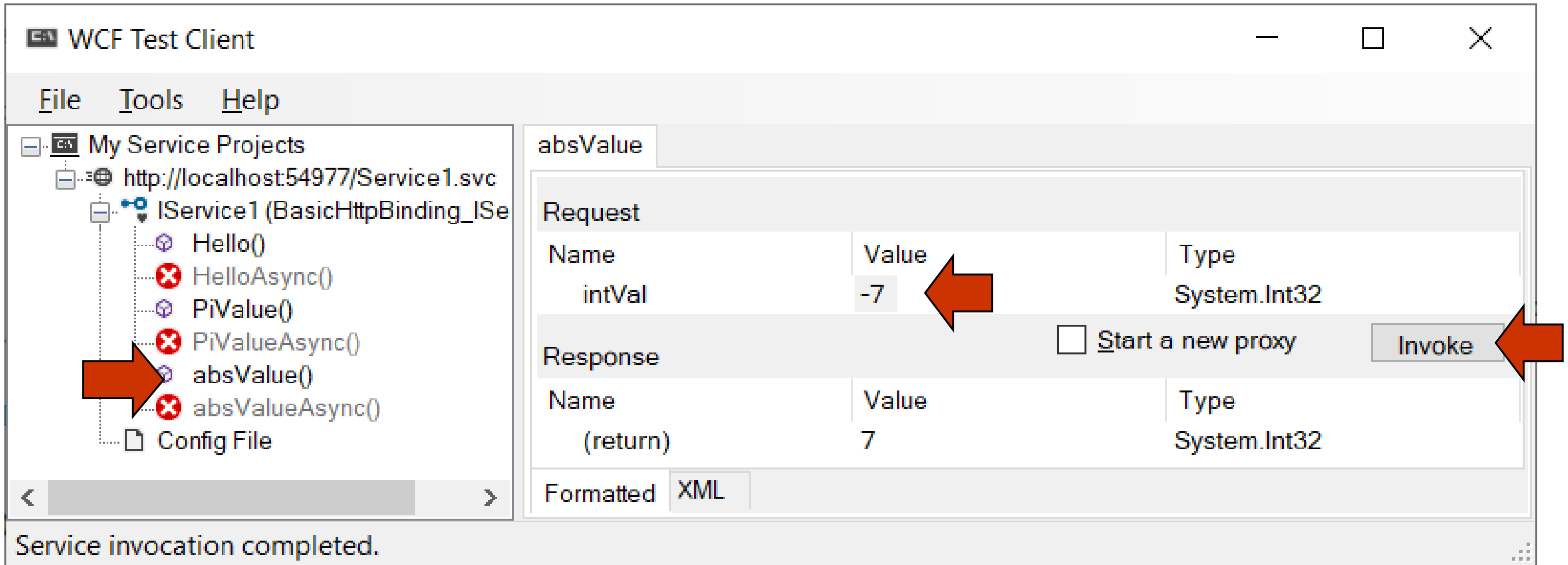
Service1 Service

Access the deployed service at:
http://venus.sod.asu.edu/WSRepository/Services/BasicThreeSvc/Service.svc

```
svcutil.exe http://localhost:54977/Service1.svc?wsdl
```

You can also access the service description as a single file:

(2) In Visual Studio, Menu: Debug → Start Without Debugging



Request

Name	Value	Type
intVal	-7	System.Int32

Response

Name	Value	Type
(return)	7	System.Int32

Service invocation completed.

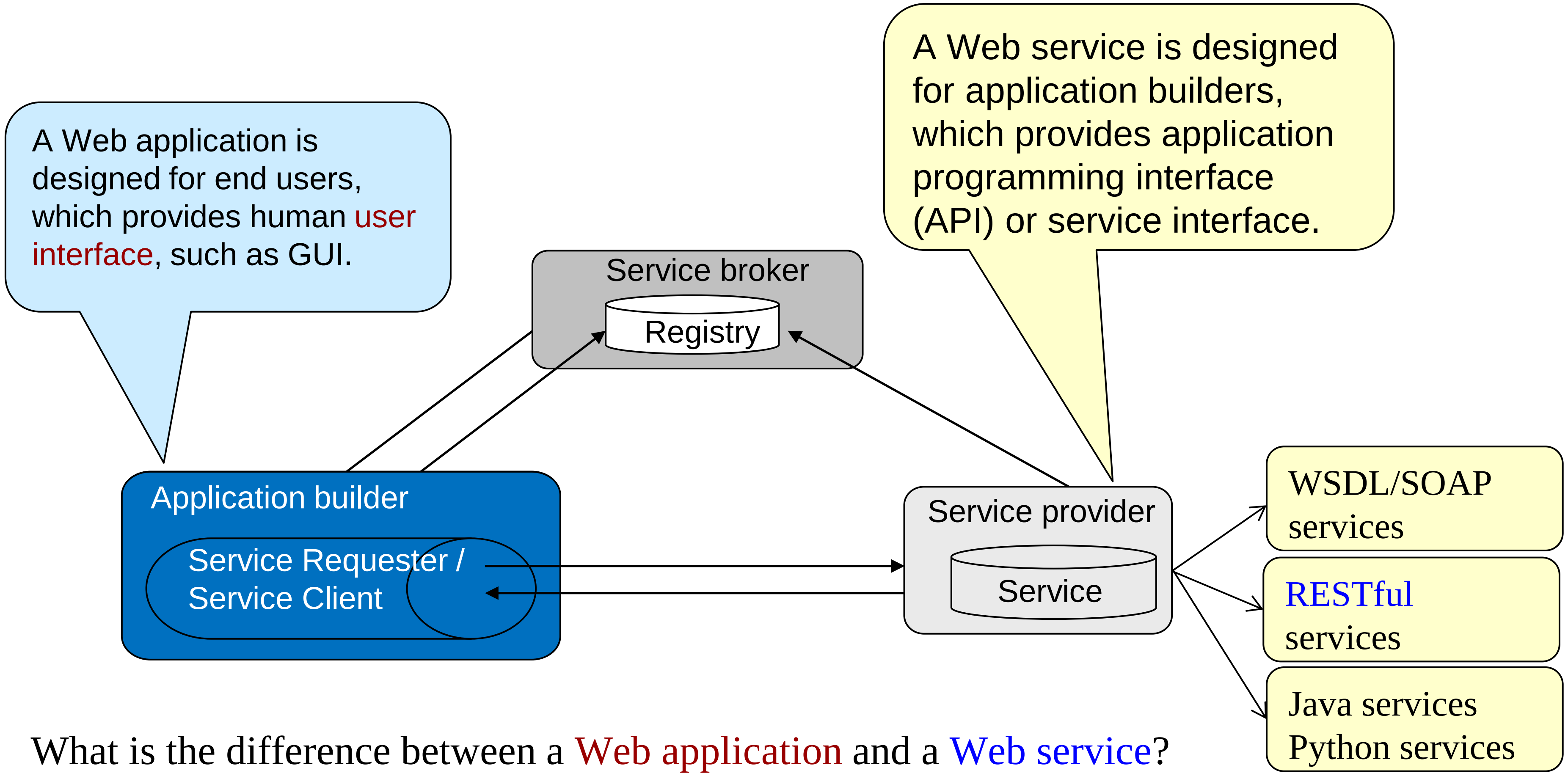
WSDL: Web Service Description Language

- WSDL is an industry-wide standard backed by MS, IBM, Oracle, AWS, …
- WSDL is used for describing Web services, including four critical aspects of Web services:
  1. Functionality description of services in standard taxonomy;
  2. Contract: service operation name, parameter, and return value;
  3. Binding information about the transport protocol to be used, usually, SOAP;
  4. Address information for locating the specified service.
- The last three aspects can be automatically generated.
- Web services described in WSDL can be searched, matched with the requirement.
- Web services described in WSDL provides the remote invocation detail.

As an Application Builder

Develop Windows (Desktop) Applications or

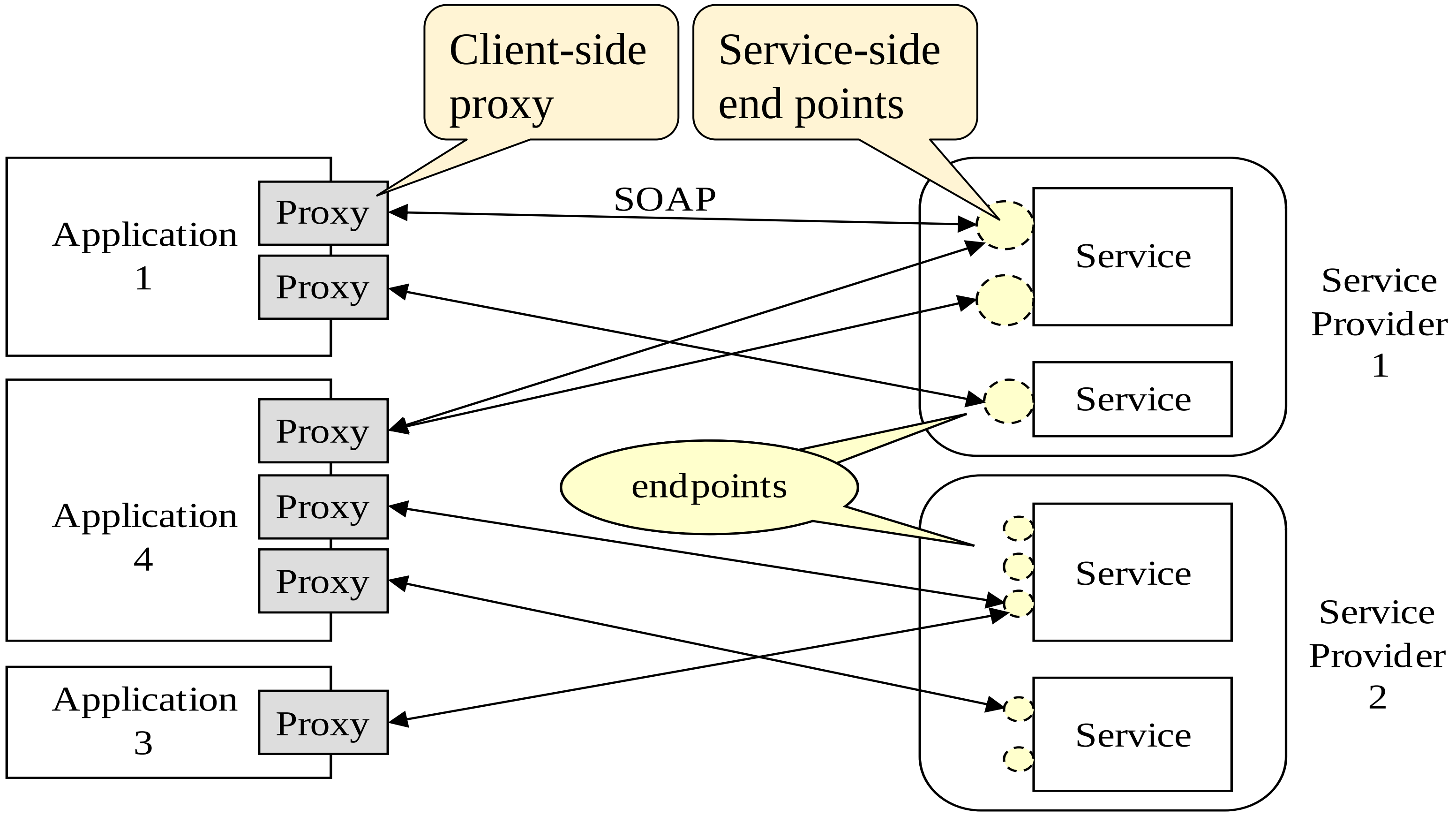
Web Applications Using ASP .Net



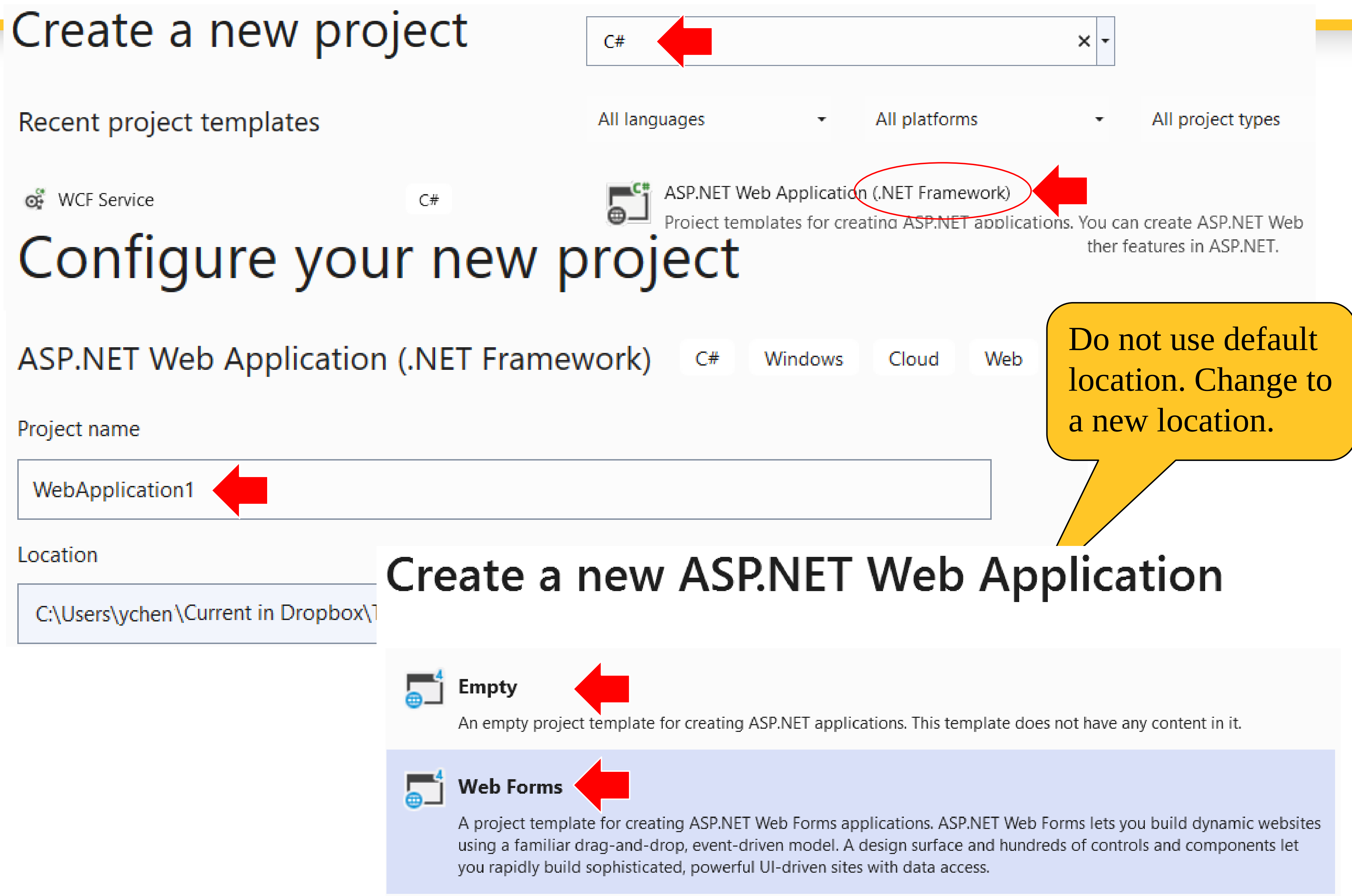
A Web application is designed for end users, which provides human user interface, such as GUI.
A Web service is designed for application builders, which provides application programming interface (API) or service interface.
Service broker
Registry
Application builder
Service Requester / Service Client
Service provider
Service
WSDL/SOAP services
RESTful services
Java services
Python services

What is the difference between a Web application and a Web service?

Applications Using WSDL Services Through Proxies



Create a New Web Application



Create a new project

C#

Recent project templates

All languages · All platforms · All project types

WCF Service — C#

ASP.NET Web Application (.NET Framework)

Project templates for creating ASP.NET applications. You can create ASP.NET Web ther features in ASP.NET.

Configure your new project

ASP.NET Web Application (.NET Framework) — C# · Windows · Cloud · Web

Project name

WebApplication1

Location

C:\Users\ychen\Current in Dropbox\

Do not use default location. Change to a new location.

Create a new ASP.NET Web Application

Empty

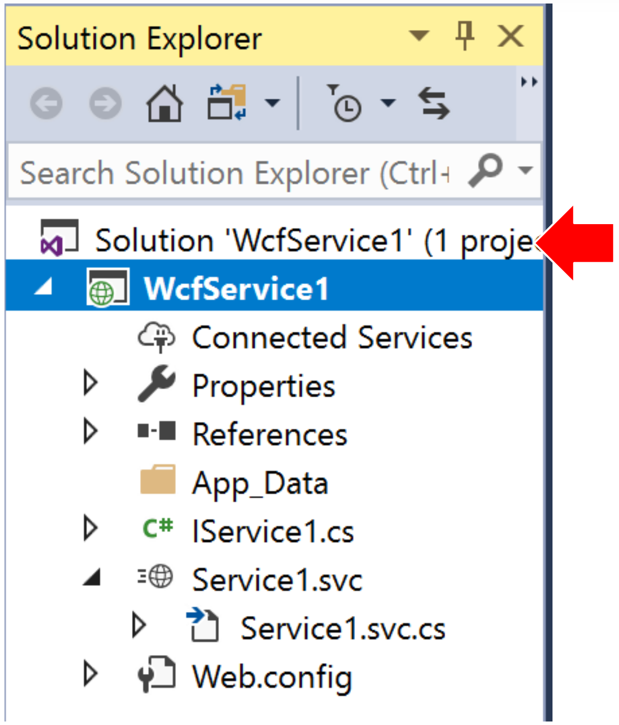
An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms

A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

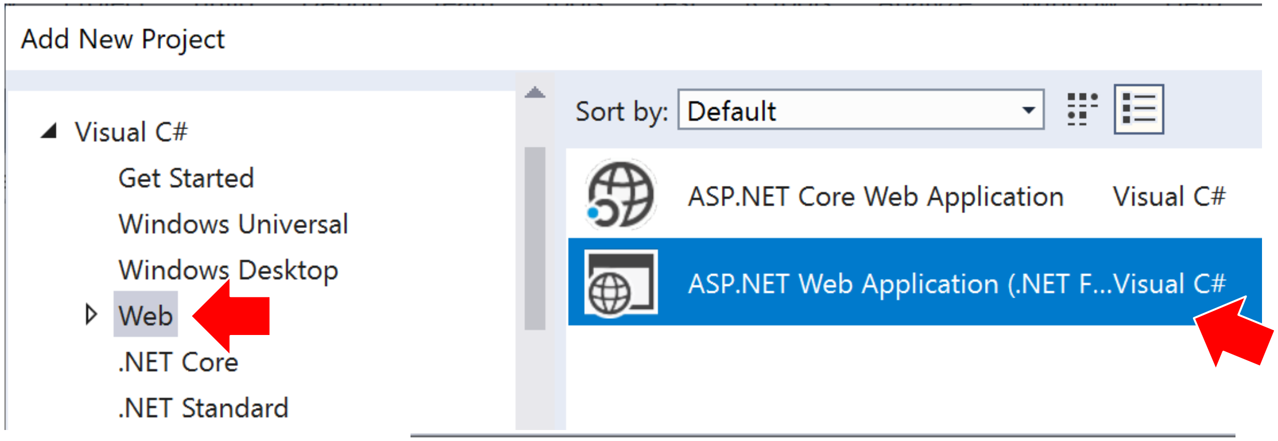
Or, Add into an Existing Solution

Right-click the Solution
Choose Add → New Project …


Solution Explorer

Search Solution Explorer (Ctrl+

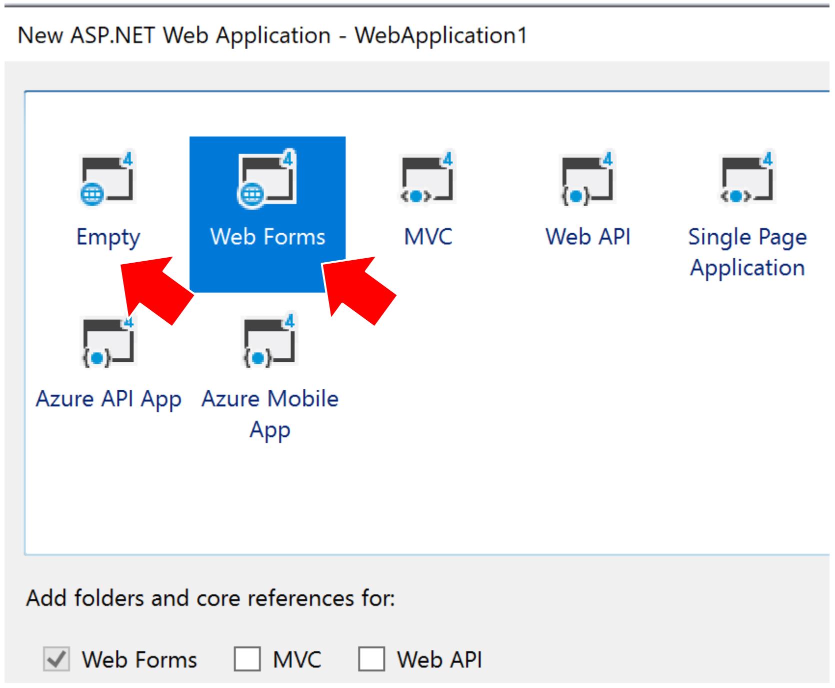
- Solution 'WcfService1' (1 proje
  - WcfService1
    - Connected Services
    - Properties
    - References
    - App_Data
    - IService1.cs
    - Service1.svc
      - Service1.svc.cs
    - Web.config


Add New Project

- Visual C#
  - Get Started
  - Windows Universal
  - Windows Desktop
  - Web
  - .NET Core
  - .NET Standard

Sort by: Default

ASP.NET Core Web Application	Visual C#
ASP.NET Web Application (.NET F...	Visual C#

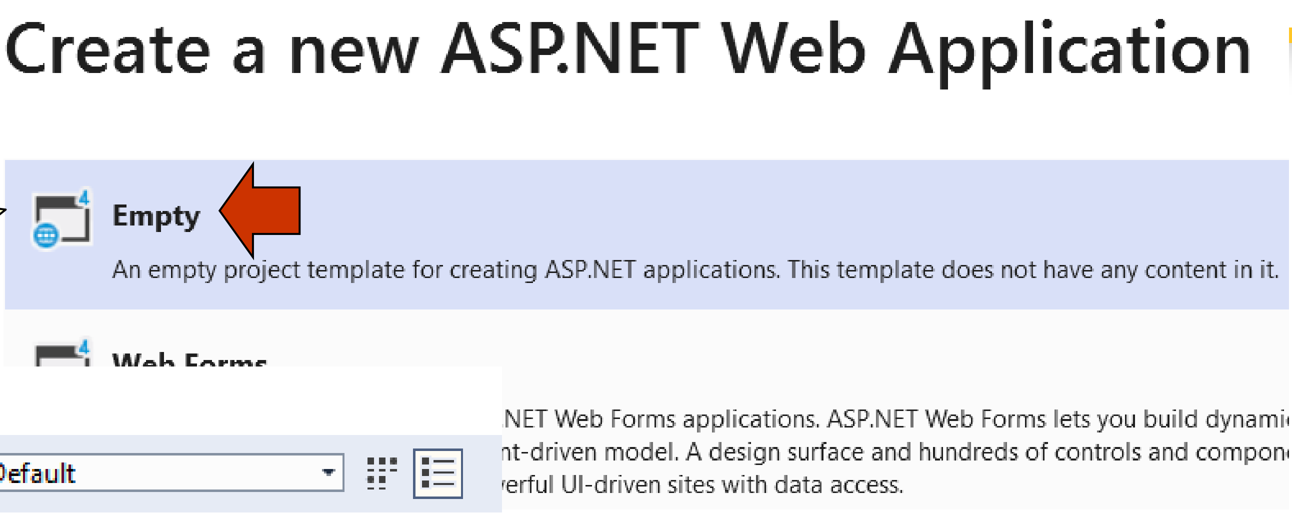

New ASP.NET Web Application - WebApplication1

Add folders and core references for:

[✓ Web Forms] MVC Web API

Start From An Empty Web Site

Create an Empty Web Site and then choose Add Web Form. You will not have the "Account" database created.



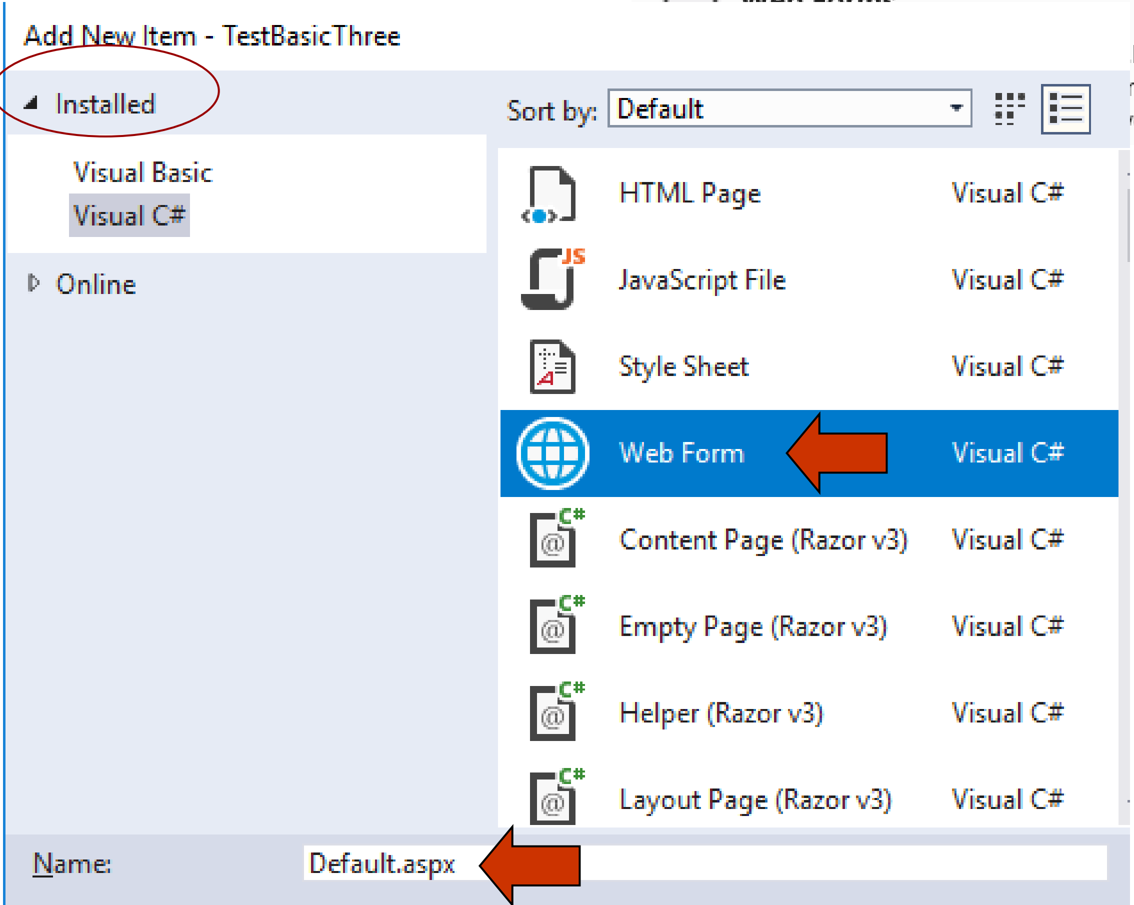
Create a new ASP.NET Web Application

Empty

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms

NET Web Forms applications. ASP.NET Web Forms lets you build dynami
nt-driven model. A design surface and hundreds of controls and compon
erful UI-driven sites with data access.



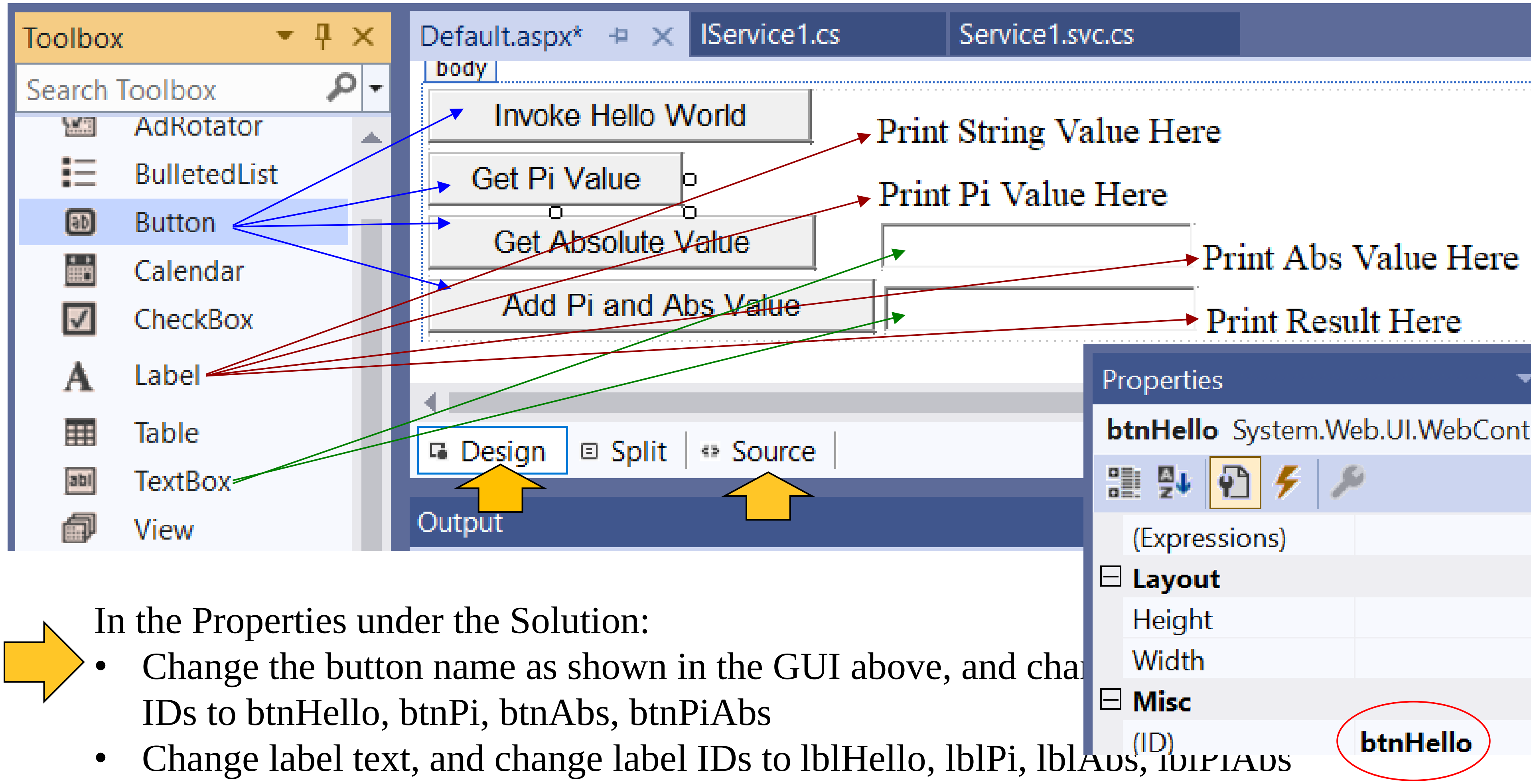
Add New Item - TestBasicThree

HTML Page	Visual C#
JavaScript File	Visual C#
Style Sheet	Visual C#
Web Form	Visual C#
Content Page (Razor v3)	Visual C#
Empty Page (Razor v3)	Visual C#
Helper (Razor v3)	Visual C#
Layout Page (Razor v3)	Visual C#

Name: Default.aspx

Right-click Project name
Choose Add → New Item …
Choose Web Form

GUI Design Using Web Form



In the Properties under the Solution:

- Change the button name as shown in the GUI above, and cha
  IDs to btnHello, btnPi, btnAbs, btnPiAbs
- Change label text, and change label IDs to lblHello, lblPi, lblAbs, lblPiAbs
- Change textbox IDs to txtAbs, txtPiAbs,
- Double click each button to create the code template behind each button.

The Code Templates Created. Do not Modify

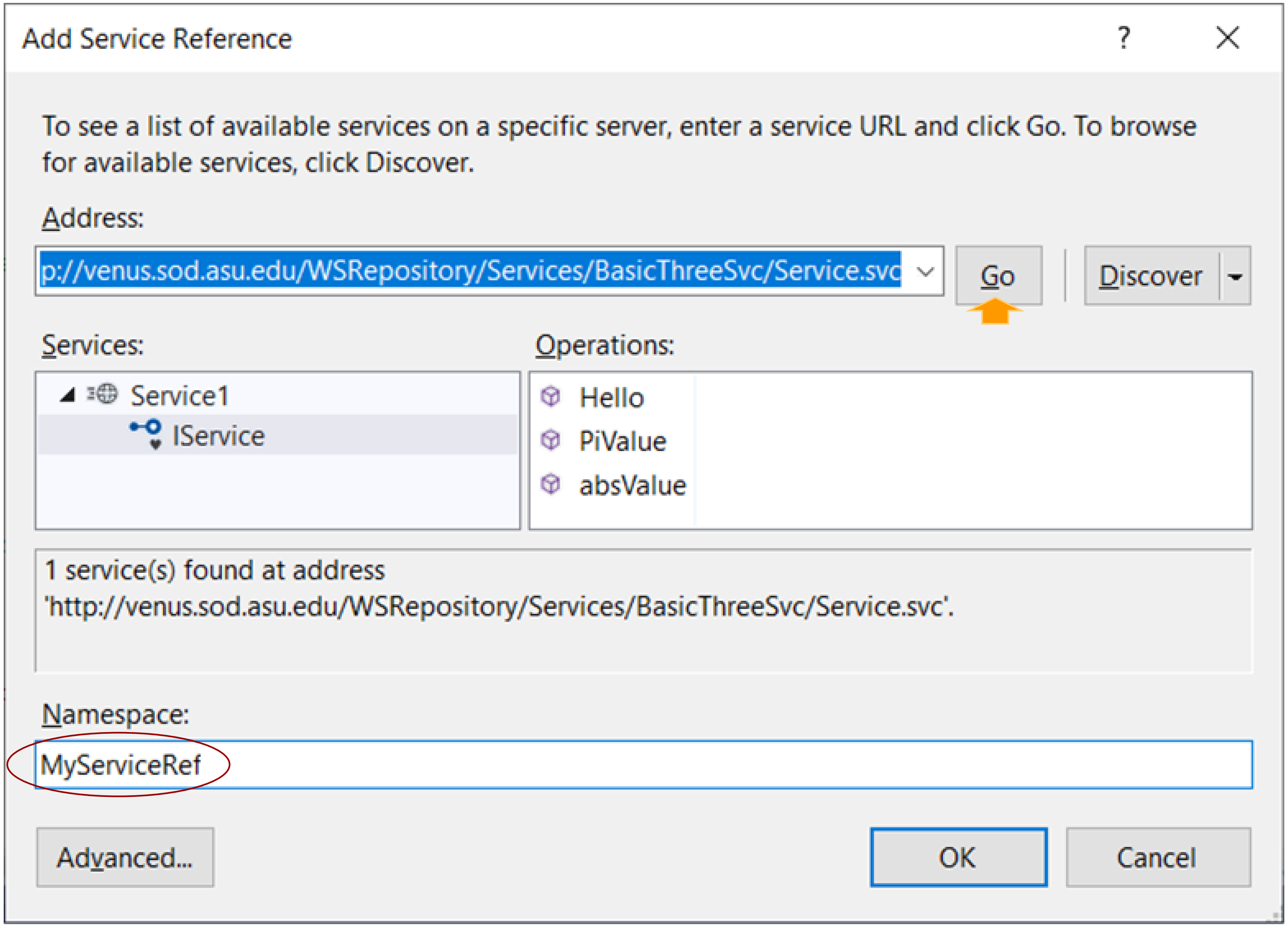
```
using System;
public partial class _Default : System.Web.UI.Page {
    protected void Page_Load(object sender, EventArgs e)
    {  // The code will be executed  every time when the page is loaded
    }
    protected void btnHello_Click(object sender, EventArgs e)
    {
    }
    protected void btnPi_Click(object sender, EventArgs e)
    {
    }
    protected void btnAbs_Click(object sender, EventArgs e)
    {
    }
    protected void btnPiAbs_Click(object sender, EventArgs e)
    {
    }
}
```

To be executed every time enter the page

Double click each button

Add Service Reference

http://venus.sod.asu.edu/WSRepository/Services/BasicThreeSvc/Service.svc


Add Service Reference

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: p://venus.sod.asu.edu/WSRepository/Services/BasicThreeSvc/Service.svc

Services:
- Service1
  - IService

Operations:
- Hello
- PiValue
- absValue

1 service(s) found at address 'http://venus.sod.asu.edu/WSRepository/Services/BasicThreeSvc/Service.svc'.

Namespace: MyServiceRef

Add Your Code into Templates

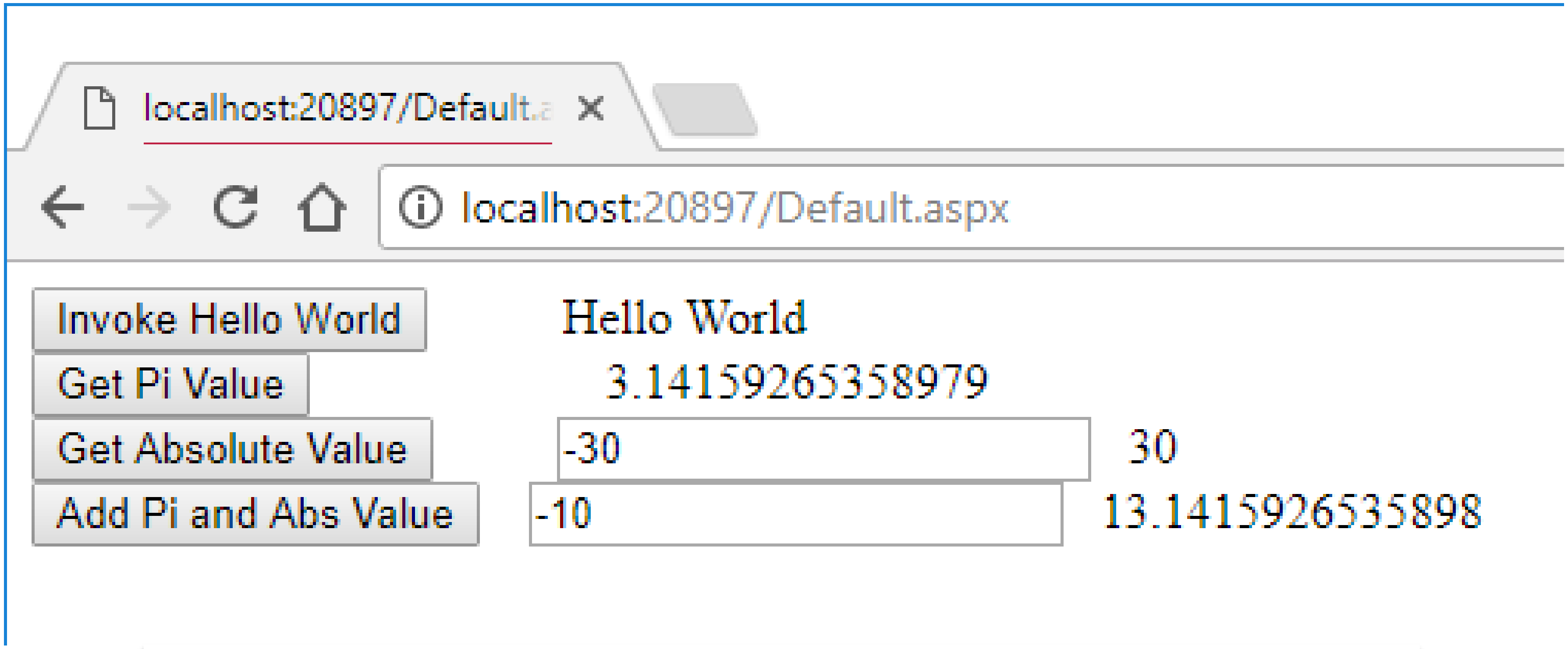
Template code must be generated by system: double click the button in the GUI Design page, to register the event

```
protected void btnHello_Click(object sender, EventArgs e) {
    MyServiceRef.ServiceClient prxy = new MyServiceRef.ServiceClient();
    lblHello.Text = prxy.Hello();
}
protected void btnPi_Click(object sender, EventArgs e)  {
    MyServiceRef.ServiceClient prxy = new MyServiceRef.ServiceClient();
    lblPi.Text = Convert.ToString(prxy.PiValue());
}
protected void btnAbs_Click(object sender, EventArgs e)  {
    MyServiceRef.ServiceClient prxy = new MyServiceRef.ServiceClient();
    String s = txtAbs.Text;
    Int32 x = prxy.absValue(Convert.ToInt32(s));
    lblAbs.Text = Convert.ToString(x);
}
protected void btnPiAbs_Click(object sender, EventArgs e) {
    MyServiceRef.ServiceClient prxy = new MyServiceRef.ServiceClient();
    String t = txtPiAbs.Text;
    Int32 y = prxy.absValue(Convert.ToInt32(t));
    Double result = y + prxy.PiValue();
    lblPiAbs.Text = Convert.ToString(result);
}
```

You enter the user code part only. If you enter the template code, the code is registered to the event of the button

Start Without Debugging

Right-click the “TestBasicThree” project and select “Set as Startup Project”
Menu: Debug → Start Without Debugging:

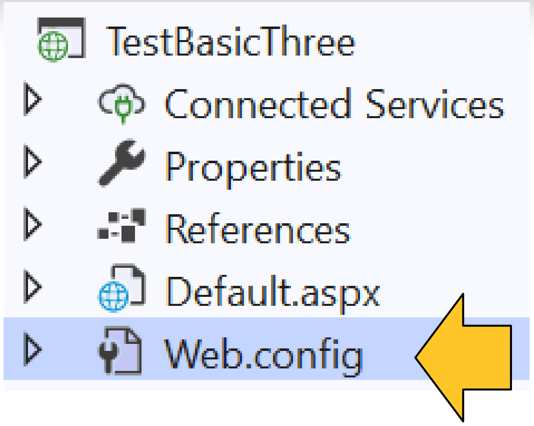


Hello World
3.14159265358979
-30 30
-10 13.1415926535898

You can try also the deployed application at:
http://venus.sod.asu.edu/WSRepository/Services/BasicThreeTryIt/

ASU WS: BasicHttpBinding → BasicHttpsBinding

- All ASU servers require https connection
- When creating an application to access ASU web service:
  - Open Web.config file and change BasicHttpBinding → BasicHttpsBinding

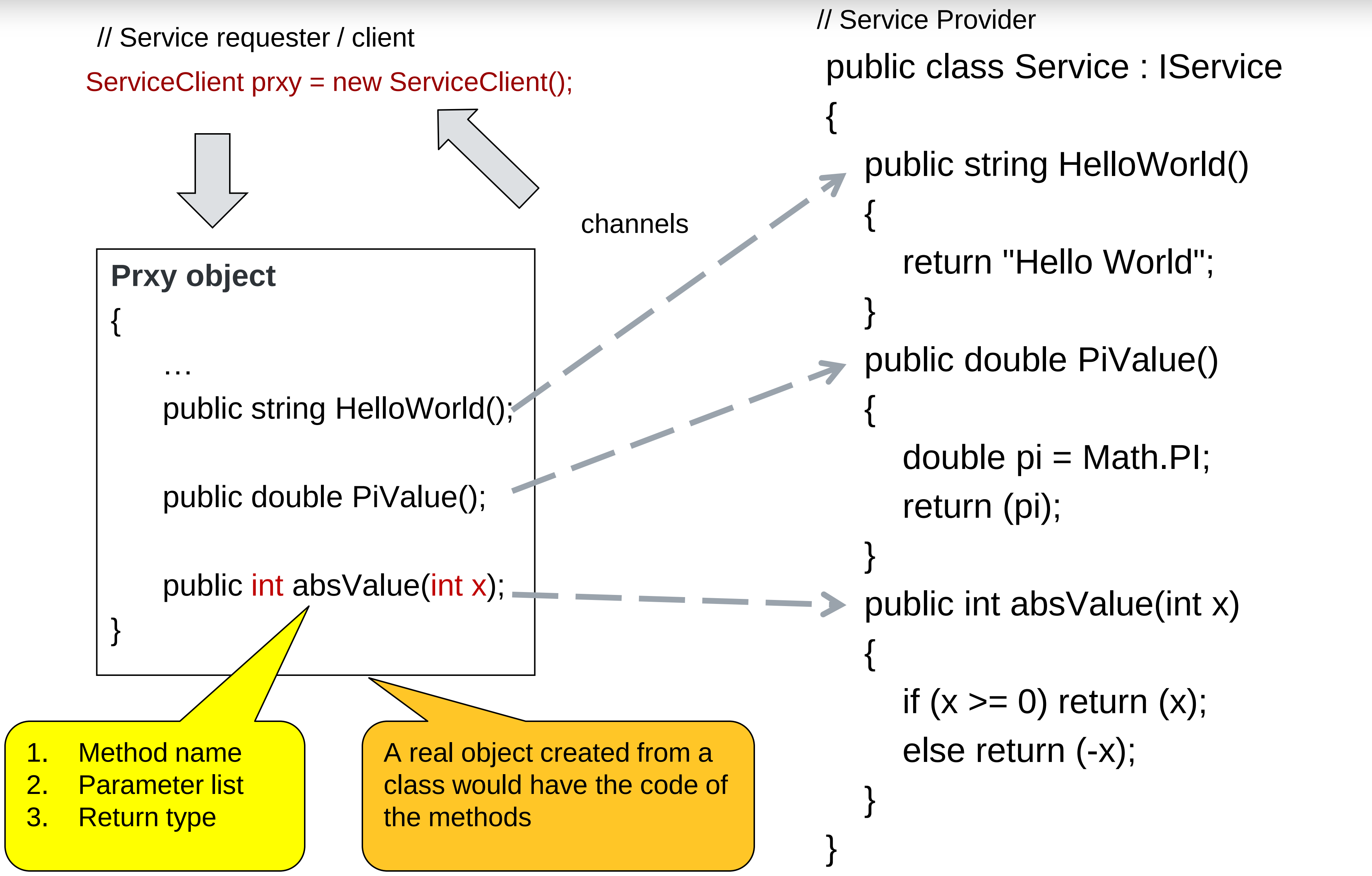


```
<configuration>
    <system.web>
        <compilation debug="true" targetFramework="4.0"/>
        <pages controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID"/>
        <httpRuntime targetFramework="4.7" />
    </system.web>

    <system.serviceModel>
        <bindings>
            <basicHttpsBinding>
                <binding name="BasicHttpsBinding_IService" />
            </basicHttpsBinding>
        </bindings>
        <client>
            <endpoint address="https://venus.sod.asu.edu/WSRepository/Services/BasicThreeSvc/Service.svc"
              binding="basicHttpsBinding" bindingConfiguration="BasicHttpsBinding_IService"

              contract="MyServiceRef.IService" name="BasicHttpsBinding_IService" />
        </client>
    </system.serviceModel>
</configuration>
```

Proxy Object: What does it represent?



// Service requester / client

```
ServiceClient prxy = new ServiceClient();
```

```
Prxy object
{
    …
    public string HelloWorld();

    public double PiValue();

    public int absValue(int x);
}
```

// Service Provider

```
public class Service : IService
{
    public string HelloWorld()
    {
        return "Hello World";
    }
    public double PiValue()
    {
        double pi = Math.PI;
        return (pi);
    }
    public int absValue(int x)
    {
        if (x >= 0) return (x);
        else return (-x);
    }
}
```

1. Method name
2. Parameter list
3. Return type

A real object created from a class would have the code of the methods

Static Binding vs. Dynamic Binding

- In object-oriented programming, dynamic binding allows a method call to be bound to the initial address of a method at run time, instead of at compilation time.
  - The method must be a virtual method, to allow the child class to redefine the method.
- In service-oriented programming, dynamic binding, also called dynamic proxy, allows a service to be bound to an application at run time, instead of at compilation time.
  - Do we need dynamic proxy?
  - Can we choose service and bind service at run time?

Summary of Module 2

- Service-Oriented Architecture and Concepts
  - Standards: XML, WSDL, SOAP
  - Service providers, Brokers, and Application builders
  - SOA Impact
- Put All Together Example
  - Create a Web service
  - Create a Web application using the Web services

Ira A. Fulton Schools of Engineering

Arizona State University