



---

## *SERVICIOS WEB CON LINKAR*

---

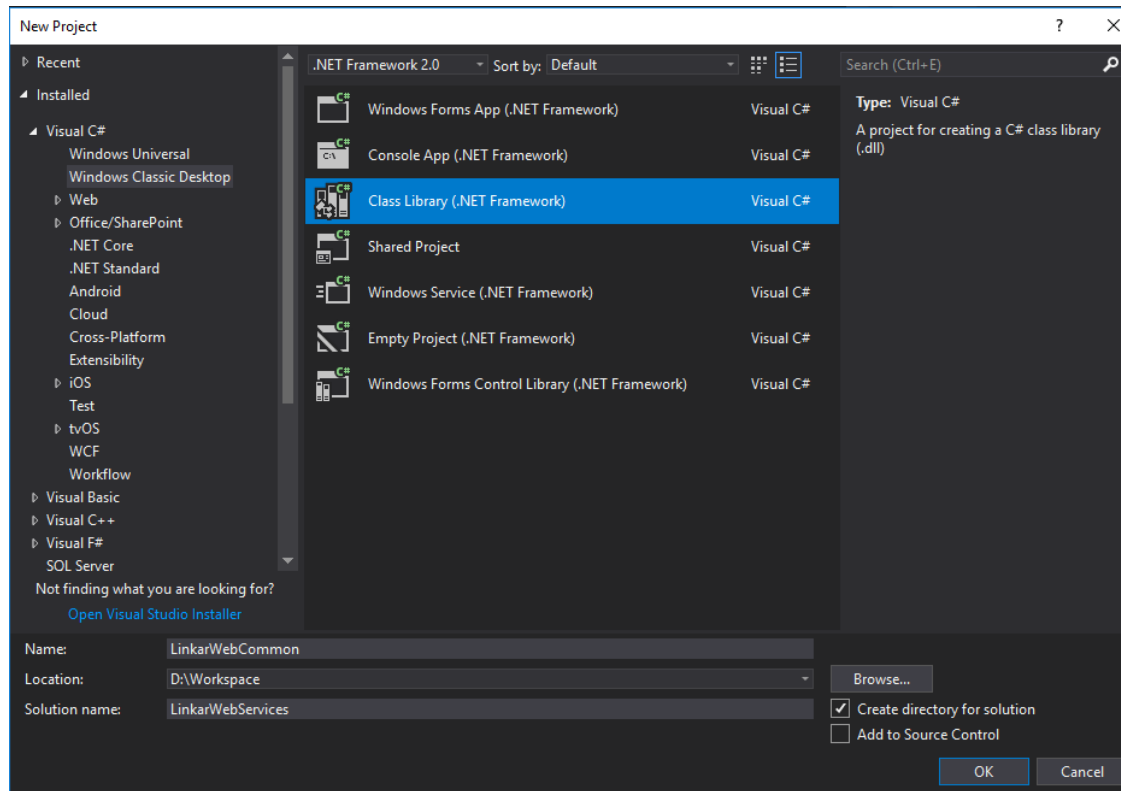
*Febrero 2019*

*by Kosday Solutions*



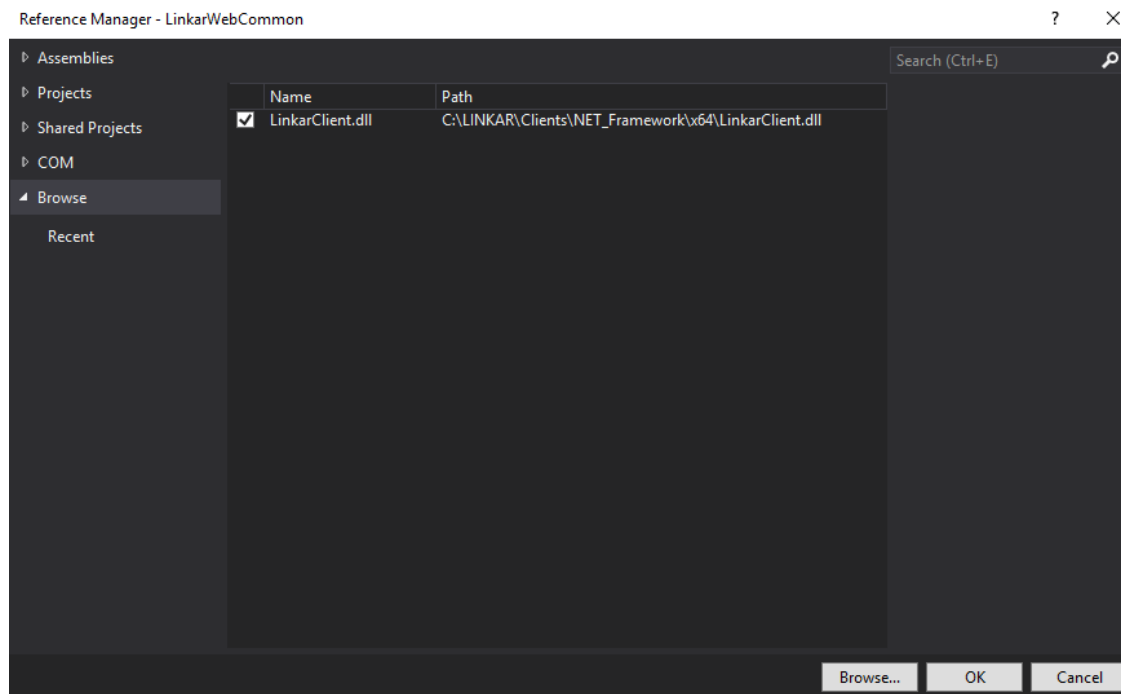
## SERVICIOS WEB CON LINKAR

En este artículo mostraremos como crear y publicar servicios web (REST API y WCF) que interactúen con Linkar Server en Visual Studio 2017. Para ello crearemos primero un nuevo proyecto común donde ubicar el código que será utilizado por ambos servicios. Creamos un proyecto de tipo Class Library (.NET Framework) en File -> New -> Project...



Abrimos el Solution Explorer y agregamos la referencia a Linkar Client en el apartado References de este nuevo proyecto.

## SERVICIOS WEB CON LINKAR



Cambiamos el nombre a la clase Class1.cs que nos crea de forma automática por Functions.cs y la alimentamos con el siguiente código:

```
using System;
using LinkarClient;
using LinkarCommon;

namespace LinkarWebCommon
{
    public class Functions
    {
        public static string SubDemoLinkar(string text, int seconds)
        {
            string error = "";
            string returnValue = "";
            try
            {
                //Remember to change your credentials to your own
                CredentialsOptions crd = new
                CredentialsOptions("192.168.1.1", 11300, "admin", "admin",
                "MYENTRYPOINT", "", "LINKAR WEB SERVICES");

                string args = seconds + ASCII_Chars.DC4_str + text +
                ASCII_Chars.DC4_str + "";
                LkData subResult = LinkarClnt.RunSubroutine(crd,
                "SUB.DEMOLINKAR", 3, args);
                if (subResult != null && !subResult.HasError() &&
                subResult.Arguments.Length == 3)
                    //Get Result Value
            }
        }
    }
}
```

## SERVICIOS WEB CON LINKAR

```
        returnValue = subResult.Arguments[2];
    else
    {
        //Manage Errors
        if (subResult.HasError())
            error = subResult.Error;
        else if (subResult.LkErrors != null &&
subResult.LkErrors.Length > 0)
            error = string.Join("\r\n", subResult.LkErrors);
        else if (subResult.Arguments.Length != 3)
            error = "Invalid Output number of Arguments";
        else
            error = "UNKNOW ERROR";
    }
}
catch (Exception ex)
{
    throw ex;
}
if (string.IsNullOrEmpty(error))
    return returnValue;
else
    throw new Exception(error);
}
}
```

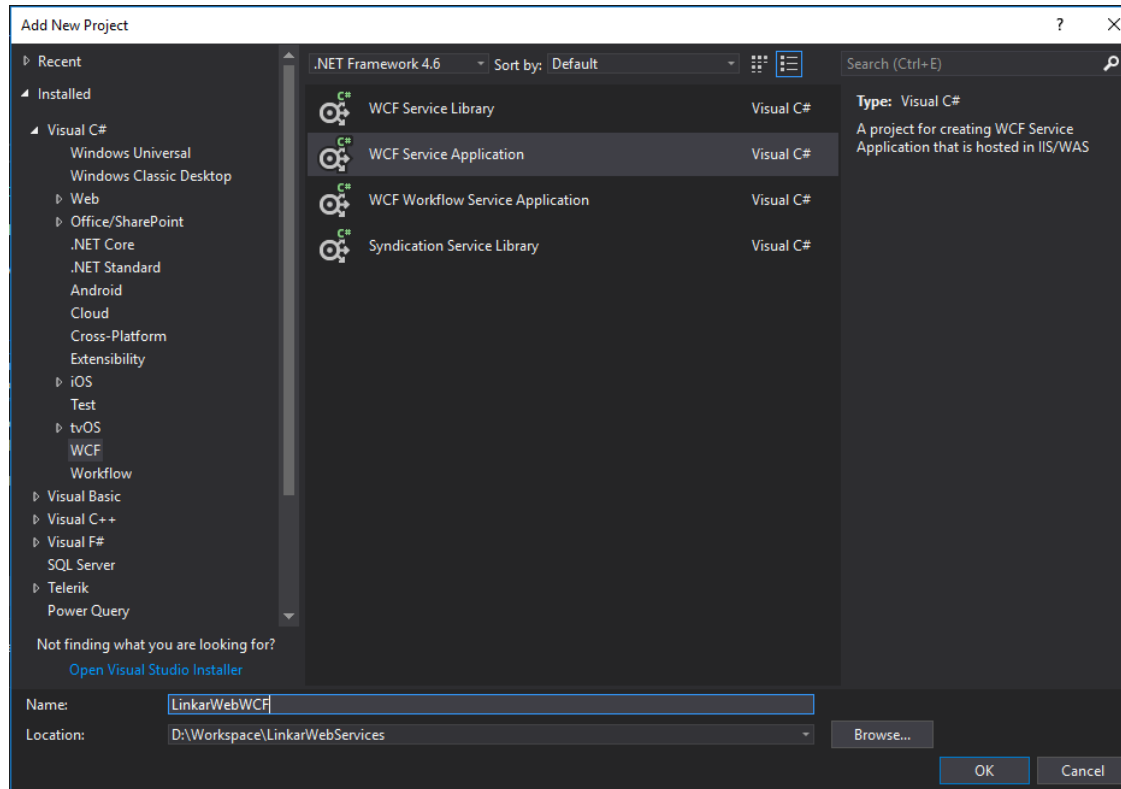
Con esta función SubDemoLinkar llamaremos a una subrutina de la cuenta demo a la que pasando un texto y un número de segundos nos devolverá el mismo texto en mayúsculas transcurridos los segundos indicados. Esto es solo un ejemplo, en casos reales tendremos tantas funciones como sean necesarias y con el uso que queramos (Select, Read, Update,Delete...). Recuerde cambiar las credenciales por las suyas propias.

Una vez completados los puntos anteriores pasaremos a crear los proyectos correspondientes a cada tipo de servicio.

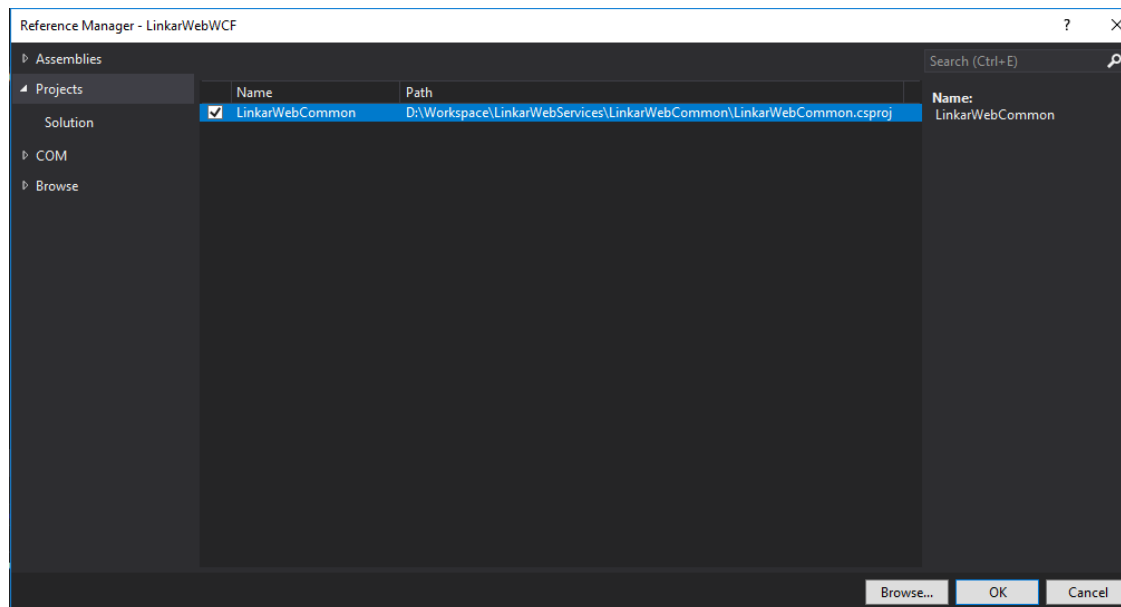
### CONSTRUIR SERVICIO WCF

En el Solution Explorer podremos crear un nuevo proyecto dentro de la misma solución haciendo click derecho sobre la solución -> Add -> New Project...

## SERVICIOS WEB CON LINKAR



Debemos añadir también una referencia a el proyecto LinkarWebCommon para poder usar la función definida allí.



## SERVICIOS WEB CON LINKAR

Este nuevo proyecto nos crea varios archivos por defecto, debemos fijarnos en IService1.cs y Service1.svc. Ambos traen un código de ejemplo que sustituiremos por los siguientes códigos:

IService1.cs

```
using System.ServiceModel;

namespace LinkarWebWCF
{
    [ServiceContract]
    public interface IService1
    {

        [OperationContract]
        string GetData(string text, int seconds);
    }
}
```

Service1.cs

```
using LinkarWebCommon;

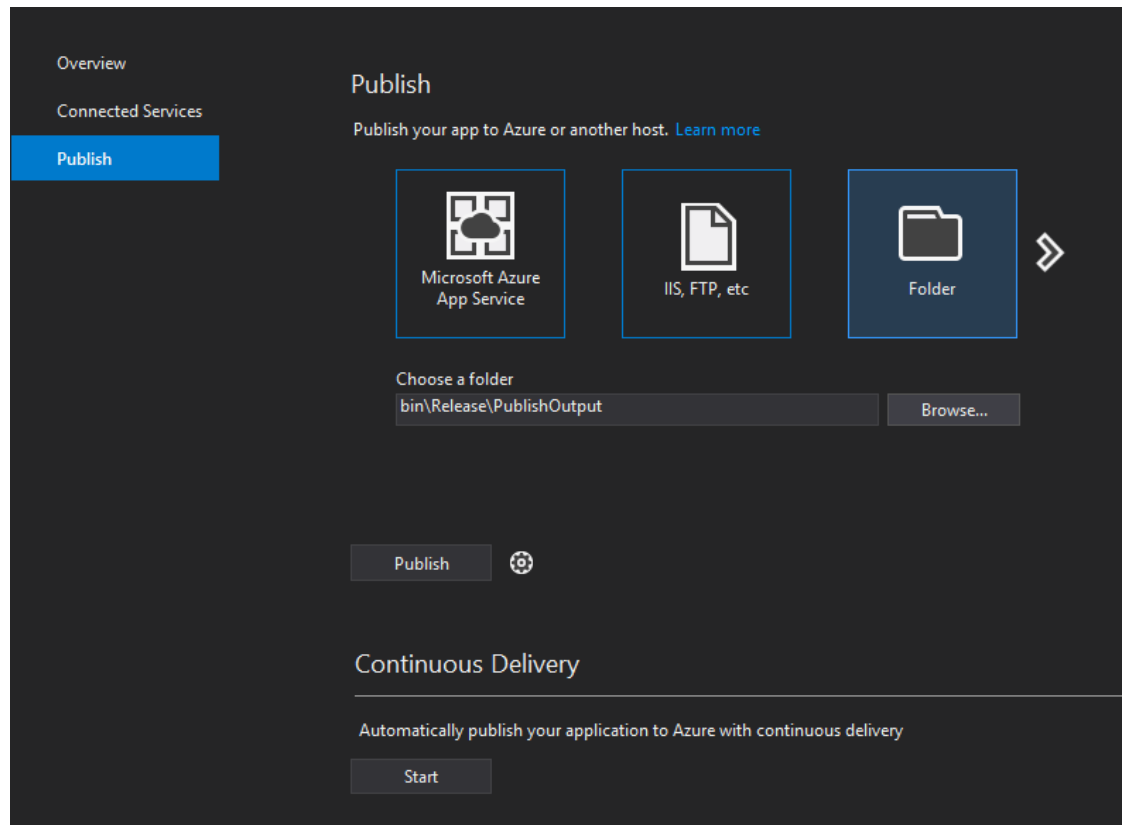
namespace LinkarWebWCF
{
    public class Service1 : IService1
    {
        public string GetData(string text, int seconds)
        {
            //Call common project
            return Functions.SubDemoLinkar(text, seconds);
        }
    }
}
```

Con este código exponemos en el servicio un único método GetData que llama a su vez a la función SubDemoLinkar que definimos anteriormente en LinkarWebCommon.

### PUBLICAR WCF

Para publicar el servicio WCF seguiremos el Asistente de Visual Studio. Abrimos el Solution Explorer, hacemos click derecho sobre el proyecto LinkarWebWCF y elegimos la opción Publish.

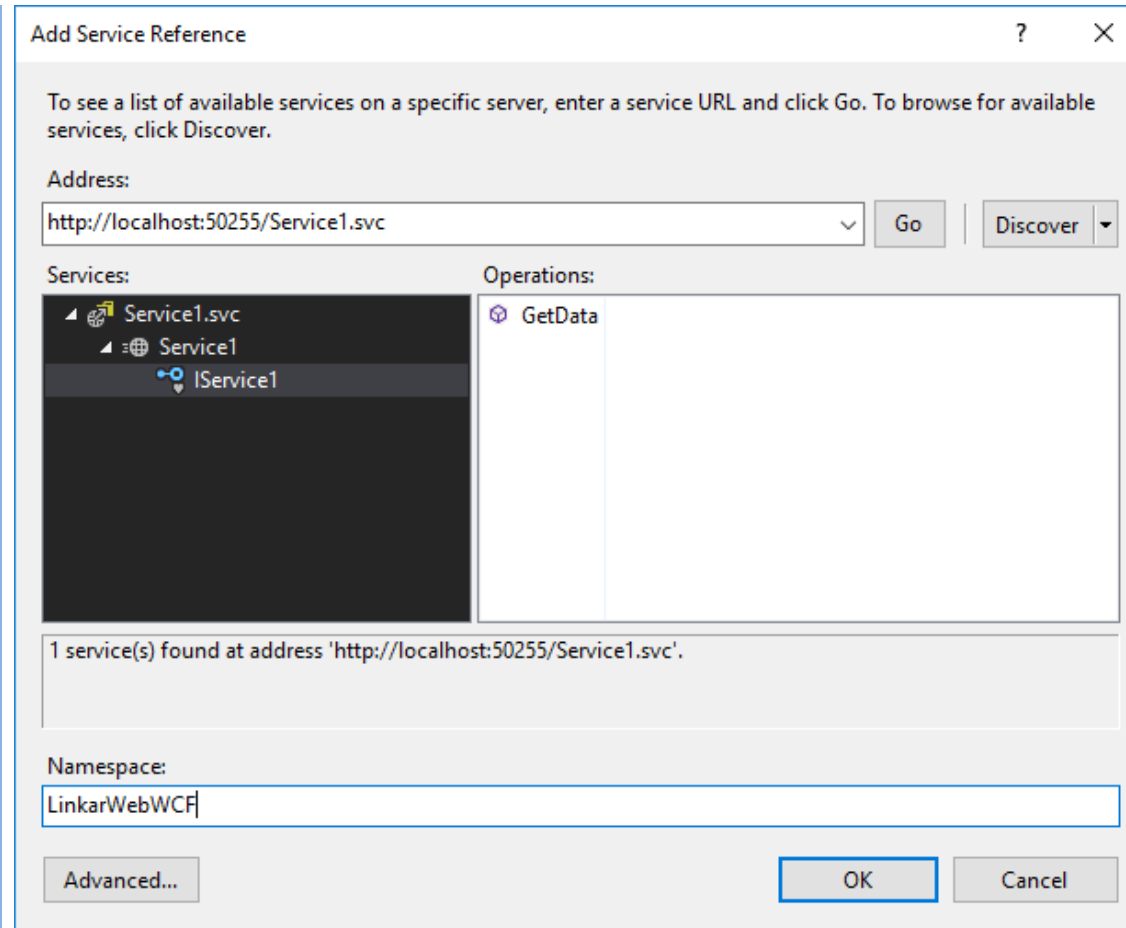
## SERVICIOS WEB CON LINKAR



### CONSUMIR SERVICIO WCF

La forma mas sencilla de consumir un servicio WCF desde Visual Studio es usar el asistente. Desde un proyecto pequeño como puede ser una aplicación de consola hacemos click derecho en su apartado References y seleccionamos Add Service Reference e indicamos la URL donde esta publicado el servicio.

## SERVICIOS WEB CON LINKAR



Hecho esto podremos hacer uso del mediante el siguiente código:

```
string text = "hello";
int seconds = 5;

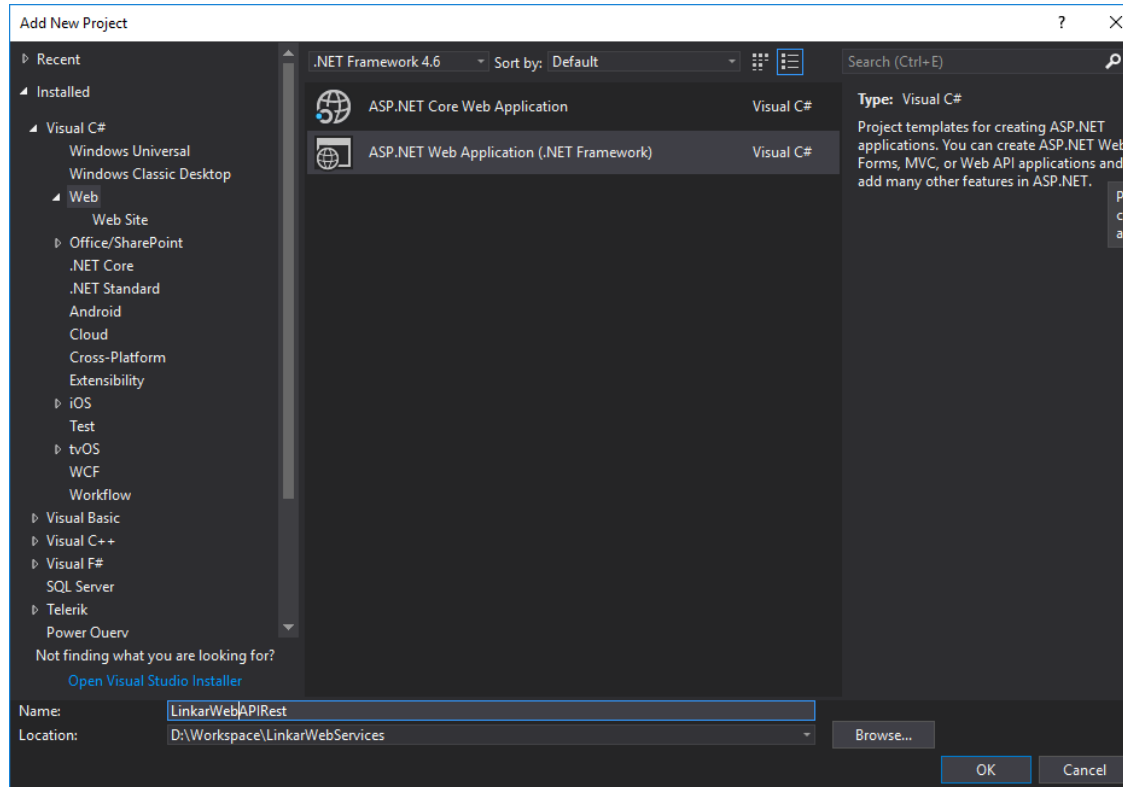
try
{
    LinkarWebWCF.Service1Client test = new
LinkarWebWCF.Service1Client();
    Console.WriteLine(test.GetData(text, seconds));
}
catch (Exception ex)
{
    Console.WriteLine("WCF ERROR: " + ex.Message);
}
```

CONSTRUIR API REST



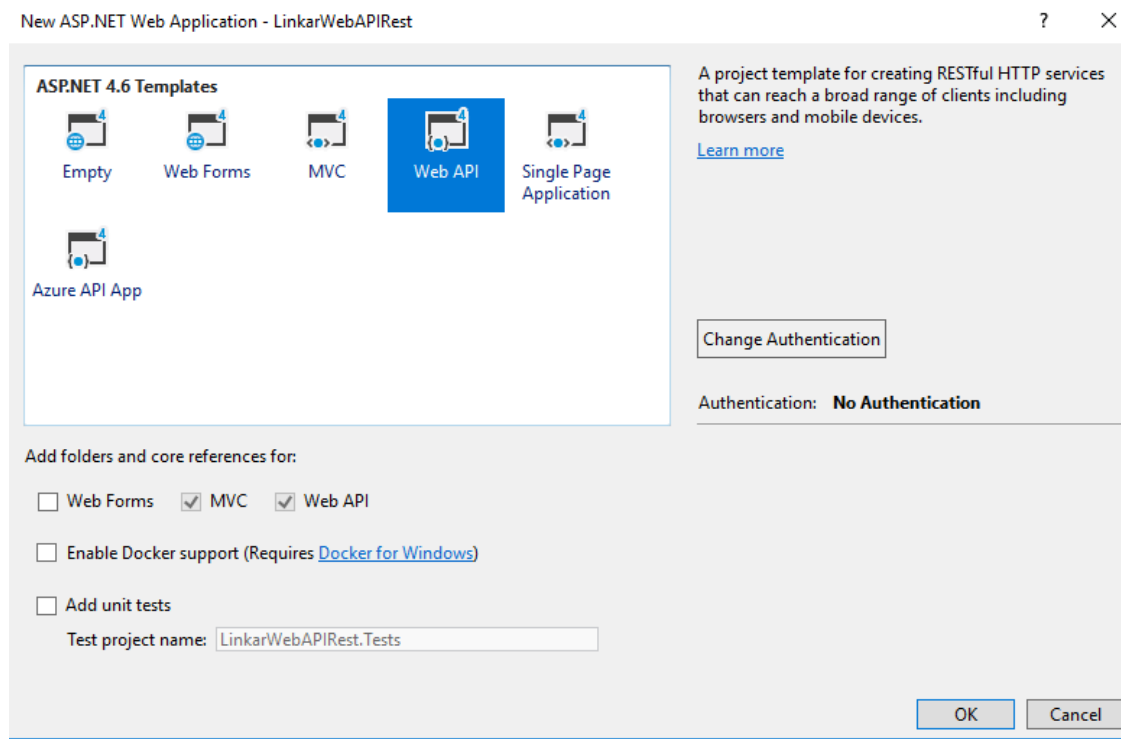
## SERVICIOS WEB CON LINKAR

En el Solution Explorer podremos crear un nuevo proyecto dentro de la misma solución haciendo click derecho sobre la solución -> Add -> New Project...

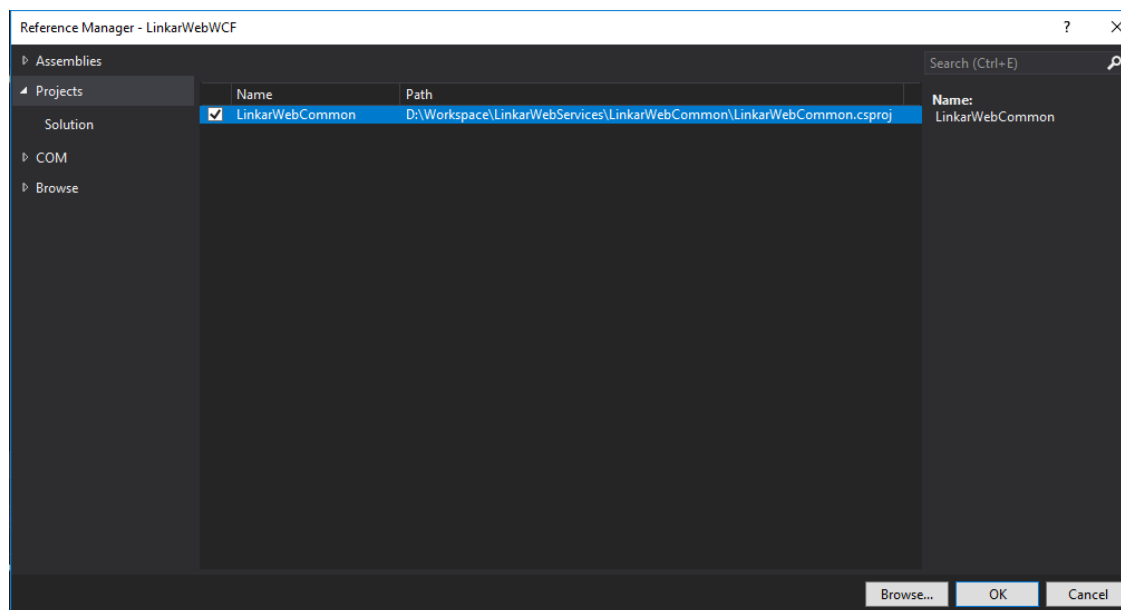


Seleccionamos la opción Web API.

## SERVICIOS WEB CON LINKAR



Debemos añadir también una referencia a el proyecto LinkarWebCommon para poder usar la función definida allí.



## SERVICIOS WEB CON LINKAR

Este nuevo proyecto nos crea muchos archivos por defecto, para una implementación básica nos fijaremos únicamente en el archivo ValuesController.cs ubicado en la carpeta Controllers. Dentro veremos un código autogenerated que cambiaremos por lo siguiente:

```
using LinkarWebCommon;
using System.Web.Http;

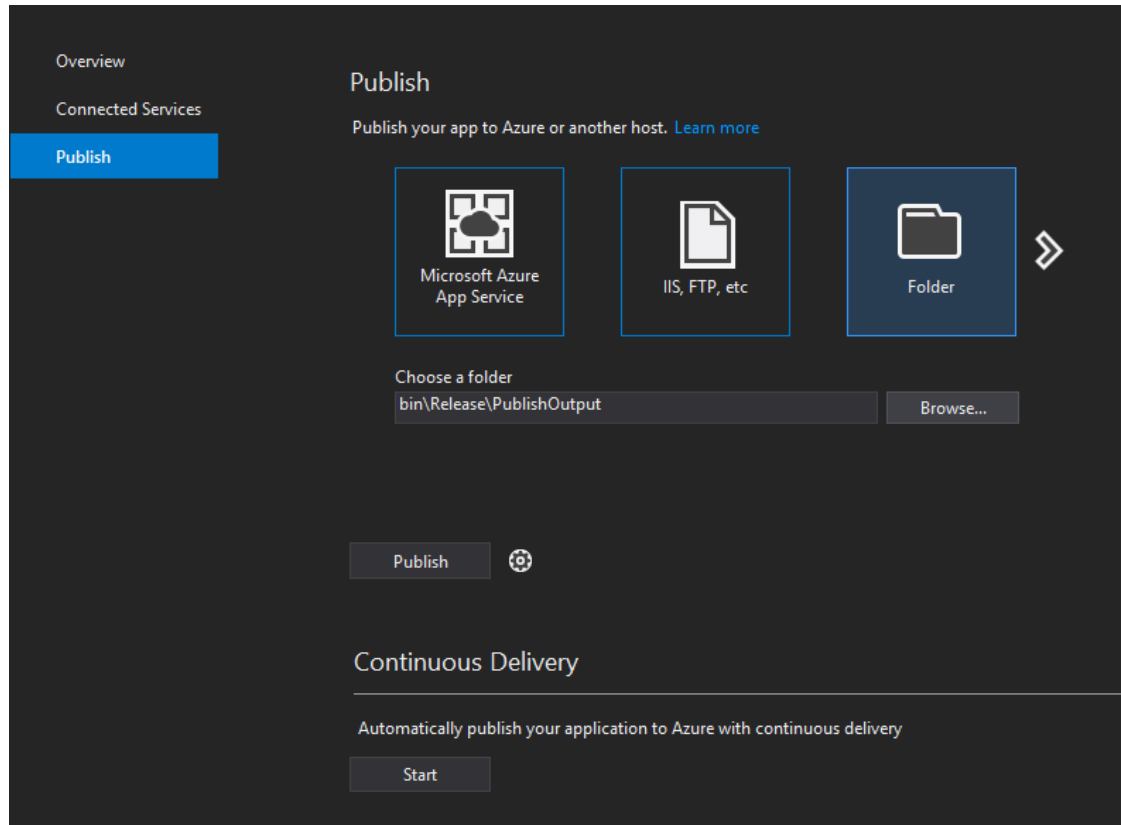
namespace LinkarWebAPIRest.Controllers
{
    public class ValuesController : ApiController
    {
        // GET api/values?text=hello&seconds=5
        public string Get(string text, int seconds)
        {
            //Call common project
            return Functions.SubDemoLinkar(text, seconds);
        }
    }
}
```

Con este código exponemos en la API un método Get dentro del controlador Values, que llama a su vez a la función SubDemoLinkar que definimos anteriormente en LinkarWebCommon.

### PUBLICAR API REST

Para publicar la API Rest seguiremos el Asistente de Visual Studio. Abrimos el Solution Explorer, hacemos click derecho sobre el proyecto LinkarWebAPIRest y elegimos la opción Publish.

## SERVICIOS WEB CON LINKAR



### CONSUMIR API REST

Hay múltiples formas de consumir un servicio de este tipo, por ejemplo, una aplicación de consola. Una vez sustituida la URL por la dirección donde este publicado el servicio este sería el código para hacer uso de él:

```
string text = "hello";
int seconds = 5;

try
{
    using (var client = new HttpClient())
    {
        client.BaseAddress = new Uri("http://localhost:50313/");
        HttpResponseMessage res = client.GetAsync("api/values?text="
+ text + "&seconds=" + seconds).Result;

        if (res.IsSuccessStatusCode)

Console.WriteLine(res.Content.ReadAsStringAsync().Result);
    }
}
else
```

## SERVICIOS WEB CON LINKAR

```
res.ReasonPhrase);
    Console.WriteLine("REST API ERROR: " +
    }
}
catch (Exception ex)
{
    Console.WriteLine("REST API ERROR: " + ex.Message);
}
```

## SERVICIOS WEB CON LINKAR

# GRACIAS

[www.linkarmv.com](http://www.linkarmv.com)

[sl@kosday.com](mailto:sl@kosday.com)