Linkar Suite


# *WEB SERVICES WITH LINKAR 2.0*

*March 2020*

*by Kosday Solutions*

kosday solutions

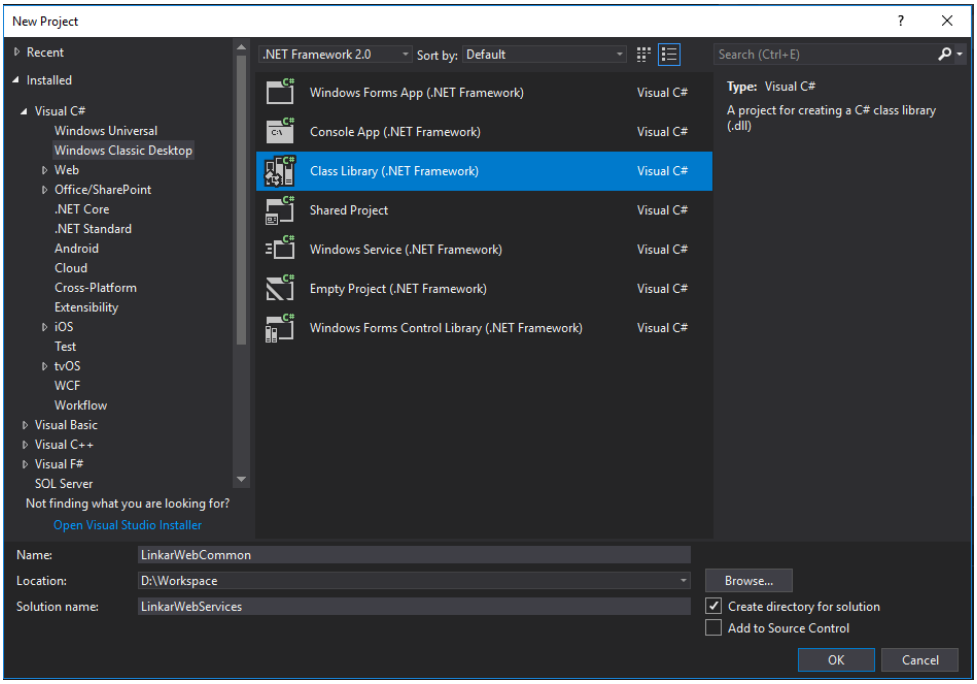
# WEB SERVICES WITH LINKAR 2.0

In this How To we will create and publish REST API and WCF web services working with Linkar in Visual Studio 2017.

We will start creating a new common project in order to store the code of both projects.

Create a Class Library (.Net framework) project.

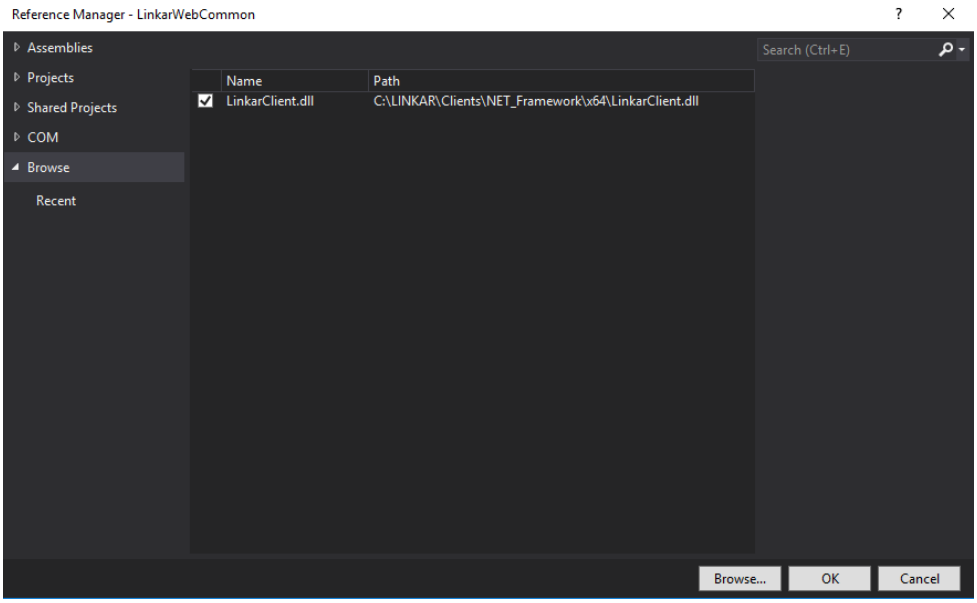
File -> New -> Project…




We need to add the Linkar Client Library (**LinkarClient.dll**. Open Solution Explorer and add the library to References for this project.

Save this project as **LinkarWebCommon**.

# WEB SERVICES WITH LINKAR 2.0




Change the class name *Class1.cs* that has been created by default to *Functions.cs* and add this code to it (change the connection credentials to your own):

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

namespace LinkarWebCommon
{
    public class Functions
    {
        public static string SubDemoLinkar(string text, int seconds)
        {
            string error = "";
            string returnValue = "";

            //Remember to change your credentials to your own
            CredentialsOptions crd = new CredentialsOptions("192.168.1.1",
"MYENTRYPOINT", 11300, "admin", "admin", "", "LINKAR DEMO SERVICES");

            string args = seconds + ASCII_Chars.DC4_str + text +
ASCII_Chars.DC4_str + "";
            LkData subResult = LinkarClt.RunSubroutine(crd, "SUB.DEMOLINKAR",
3, args);
            if (subResult != null && subResult.Arguments.Length == 3)
                //Get Result Value
                returnValue = subResult.Arguments[2];
            else
            {
                //Manage Errors
                if (subResult.Errors != null && subResult.Errors.Length > 0)
                    error = string.Join("\r\n", subResult.Errors);
```

# WEB SERVICES WITH LINKAR 2.0

```
                else if (subResult.Arguments.Length != 3)
                    error = "Invalid Output number of Arguments";
                else
                    error = "UNKNOWN ERROR";
            }
            if (string.IsNullOrEmpty(error))
                return returnValue;
            else
                throw new Exception(error);
        }
    }
    }
```

The above code will call a basic subroutine in your Linkar Demo MV account. The name of the subroutine is *SUB.DEMOLINKAR* and is located in LK.BP file:

```
SUBROUTINE SUB.DEMOLINKAR(SECONDS,DOWN,UPPER)
      *-----------------------------------------------------*
      * SECONDS=Seconds waiting for the action              *
      * DOWN=Any text in low case                           *
      * UPPER=Same text in upper case                       *
      *-----------------------------------------------------*
      SLEEP SECONDS
      UPPER=OCONV(DOWN,"MCU")
RETURN
END
```

With the *SubDemoLinkar* function, we call this subroutine by sending 3 arguments: *seconds* (number of seconds the subroutine will wait to perform the rest of the action), *text* (any text string in lower case), and an empty one. The function result is the same text in upper case.

This is only an example. You can perform any operation  in the subroutine.

You must remember to change your credentials in the code.

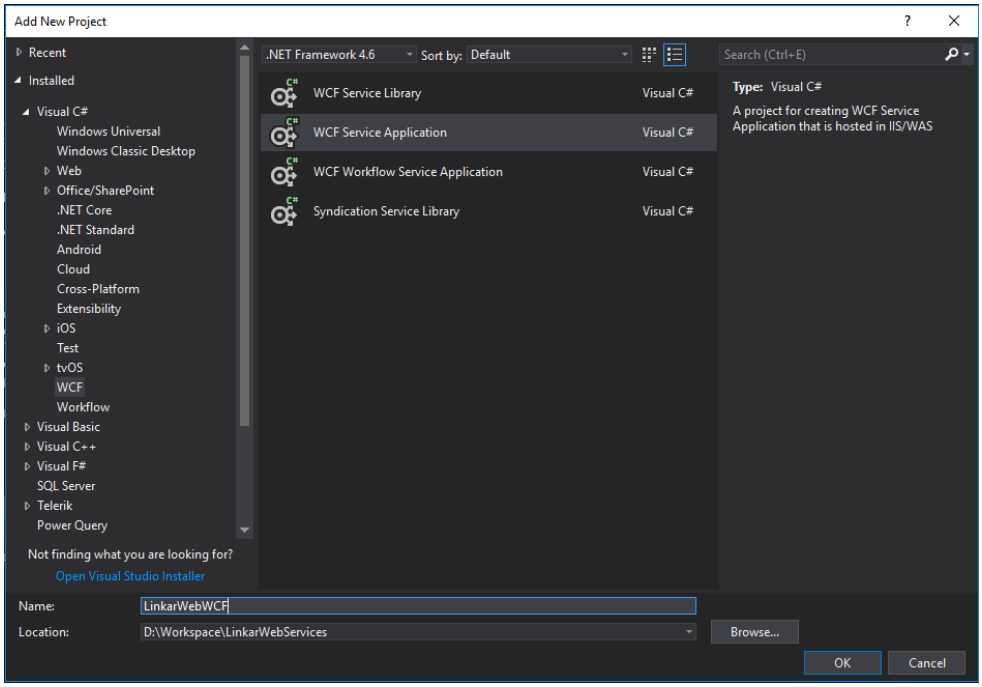
Now we are going to create each Project type.

### WCF PROJECT

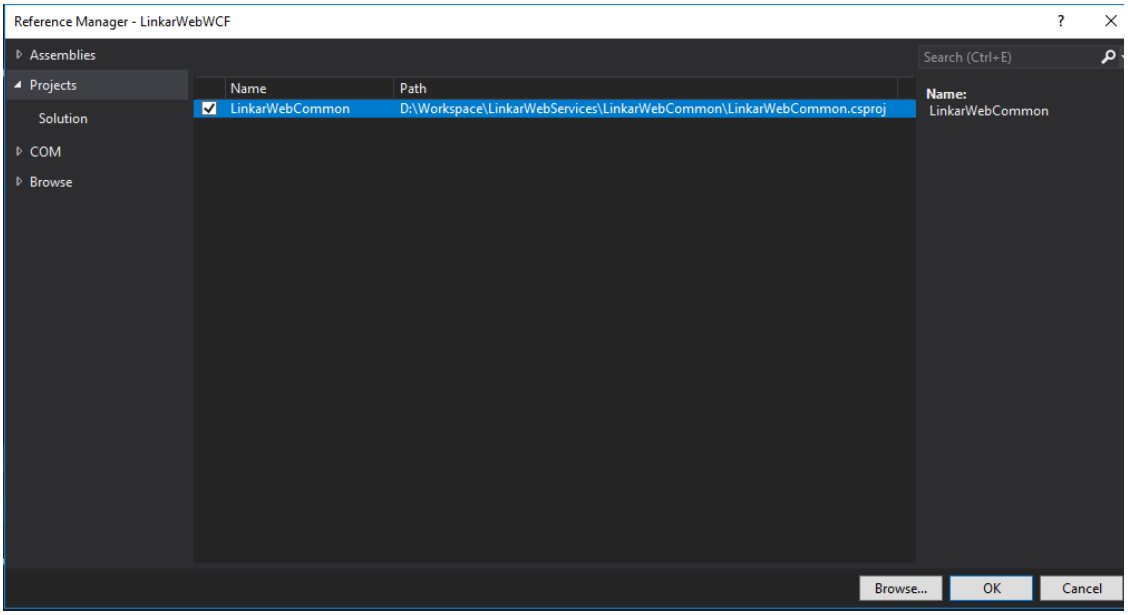
We will create a new Project in the same solution.

# WEB SERVICES WITH LINKAR 2.0

Right click on the Solution in Solution Explorer and choose Adda New Project …




We must add a Reference to the **LinkarWebCommon** project created before in order to use the *SubDemoLinkar* function defined there.




This new Project will create some files by default. We must pay attention to two of them, *IService1.cs* and *IService1.svc*.

We must replace their code with ours:

# WEB SERVICES WITH LINKAR 2.0

**IService1.cs**

```
using System.ServiceModel;

namespace LinkarWebWCF
{
    [ServiceContract]
    public interface IService1
    {

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

**Service1.cs**

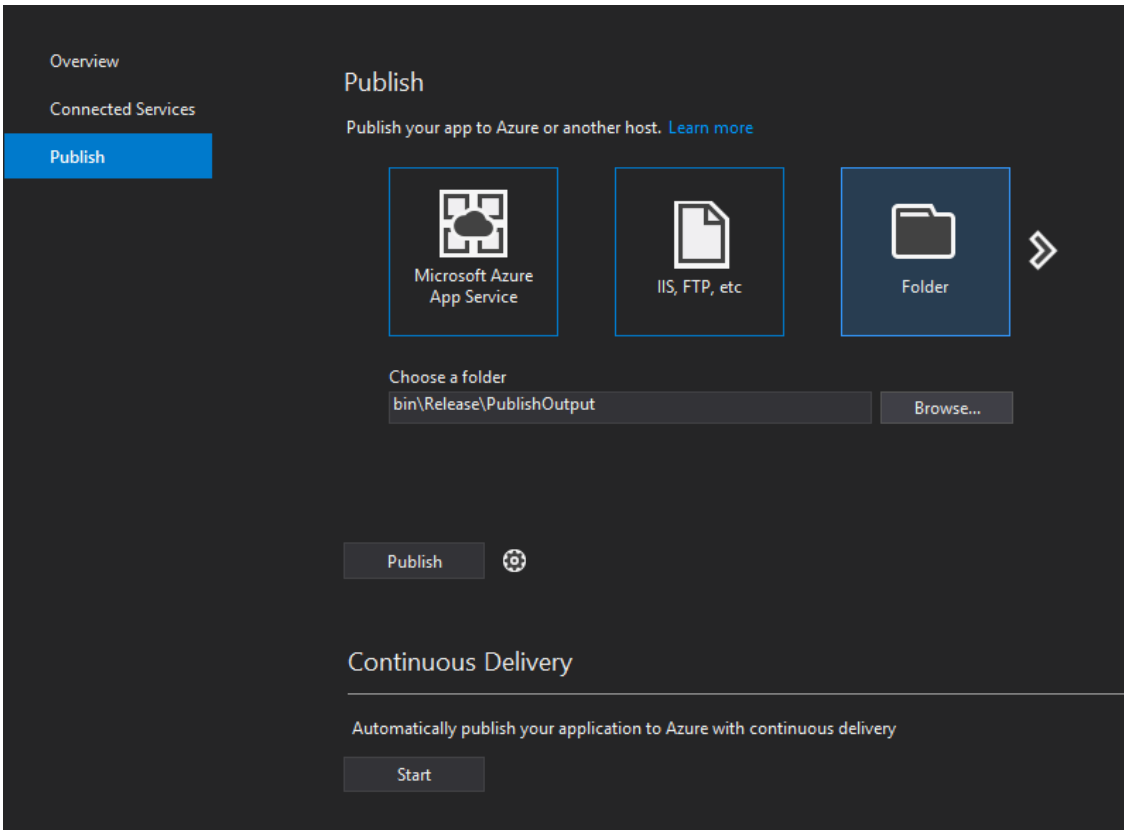
```
using LinkarWebCommon;

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

With this code we expose only one method in the service, which calls the *SubDemoLinkar* function defined previously in the *LinkaWebCommon* project, which then calls the SUB.DEMOLINKAR BASIC subroutine in your MV account.

### *PUBLISH THE WCF PROJECT*

We need to publish the WCF Project. The Visual Studio Assistant can be used. To open the Solution Explorer, right click on the *LinkarWebWCF* and select *Publish*.

# WEB SERVICES WITH LINKAR 2.0

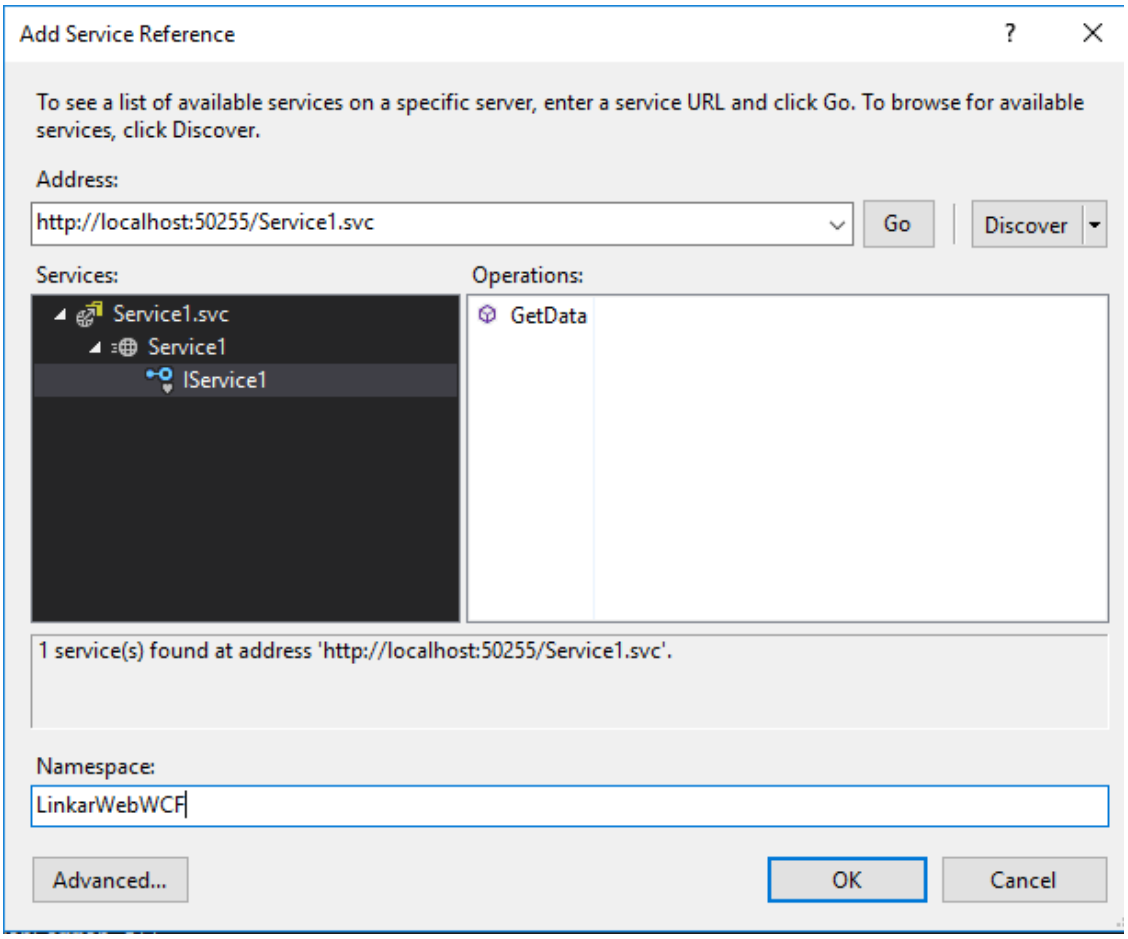



***USING THE WCF SERVICE***

The easiest way to use the WCF web service is to use the Visual Studio assistant.

Create a small project like a console app, right click on References, select Add Service and choose the Service URL you obtained in the previous step when you published the service.

# WEB SERVICES WITH LINKAR 2.0




Now, you can use it with this code:

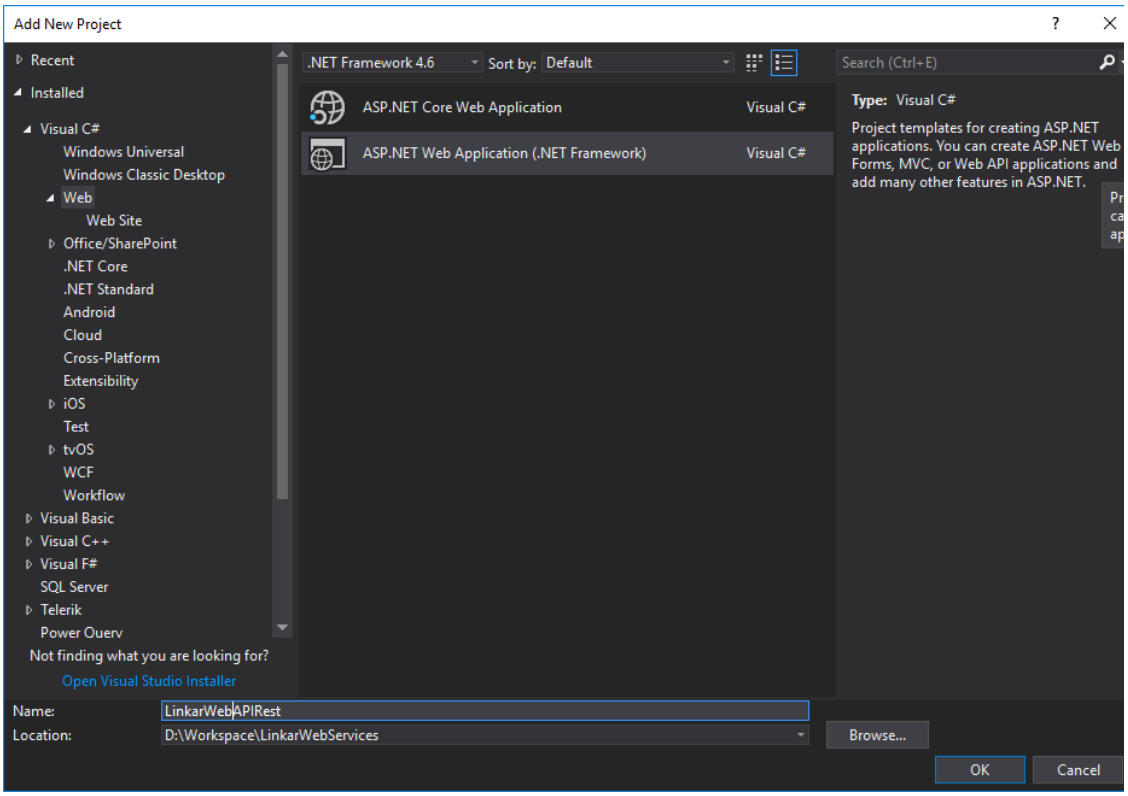
```
            string text = "hello";
            int seconds = 5;
            try
            {
                LinkarWebWCF.Service1Client test = new
LinkarWebWCF.Service1Client();
                Console.WriteLine(test.GetData(text, seconds));
            }
            catch (Exception ex)
            {
                Console.WriteLine("WCF ERROR: " + ex.Message);
            }
```

### API REST PROJECT

We will create a new Project in the same solution.

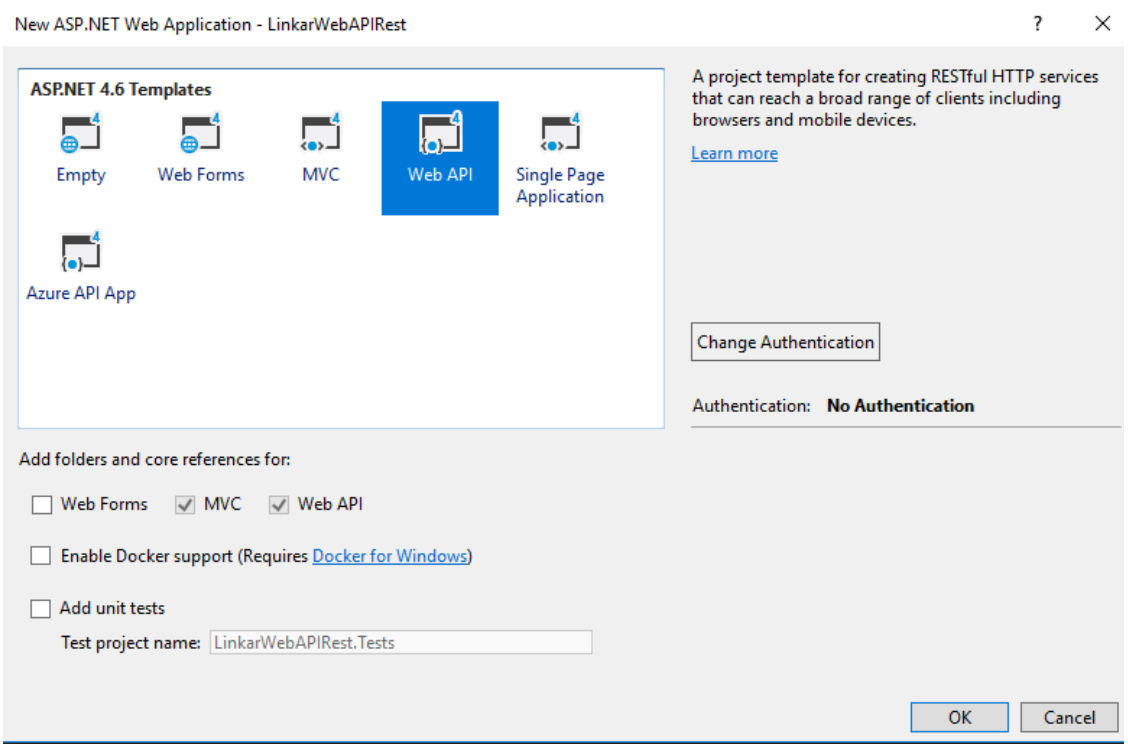
Right click on the Solution in Solution Explorer and choose Adda New Project …

# WEB SERVICES WITH LINKAR 2.0

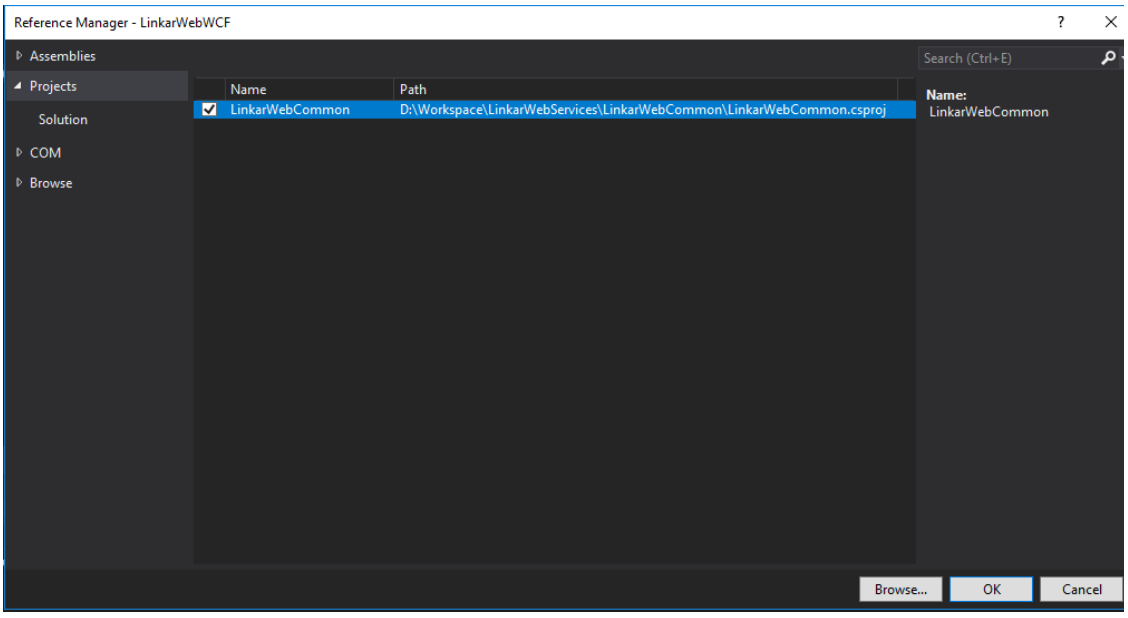



Now choose the Web API option.

# WEB SERVICES WITH LINKAR 2.0




We must add a Reference to the **LinkarWebCommon** project created earlier in order to use the *SubDemoLinkar* function defined there.




This new Project will create some files by default. We must pay attention to only one of them for a basic implementation: *ValuesController.cs* in the *Controllers* folder.

# WEB SERVICES WITH LINKAR 2.0

We must replace its code with ours:

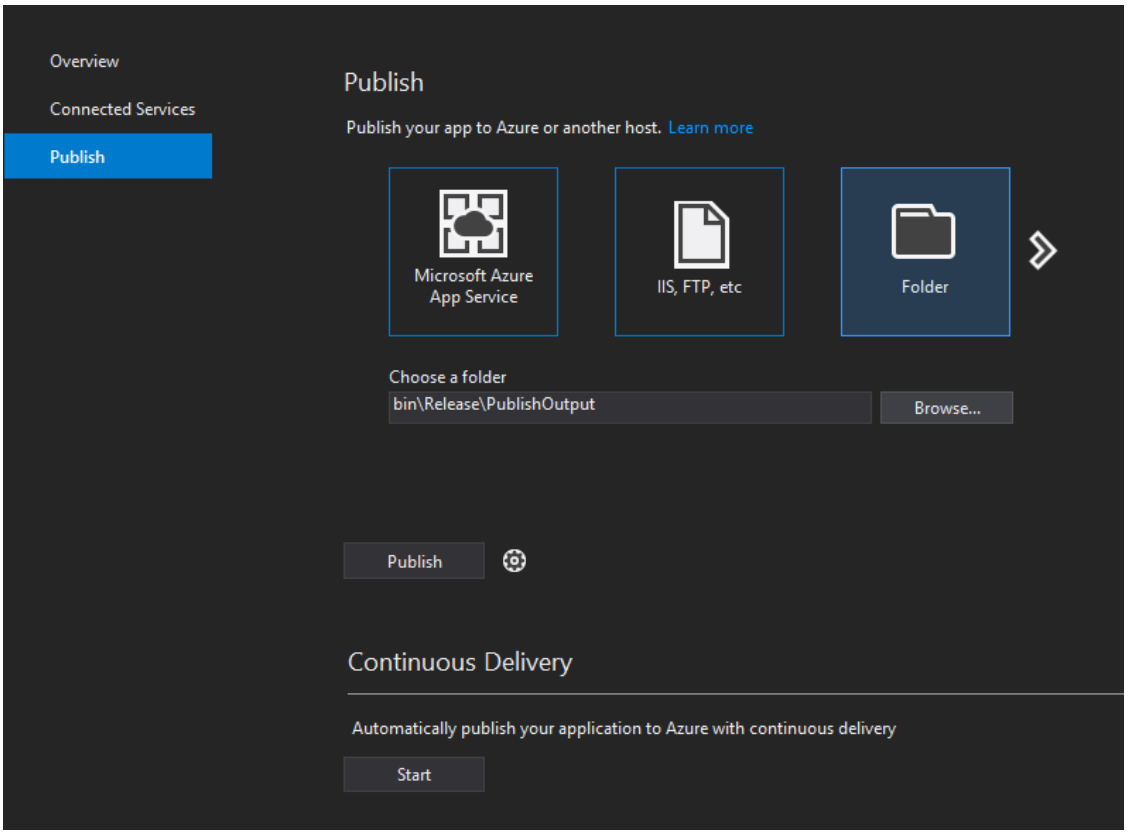
```
using LinkarWebCommon;
using System.Web.Http;

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);
        }
    }
}
```

With this code we expose one GET method in the API inside the *Values* controller, which calls the *SubDemoLinkar* function defined earlier in the *LinkaWebCommon* project, which also calls the SUB.DEMOLINKAR BASIC subroutine in your MV account.

***PUBLISH THE API REST SERVICE***

We need to publish the API REST Project. The Visual Studio Assistant can be used. Open Solution Explorer, right click on *LinkarWebAPIRest* and select *Publish*.

# WEB SERVICES WITH LINKAR 2.0




***USING THE API REST SERVICE***

There are many ways to call this webservice type, for instance in a Console app. You must change the URL with your own (obtained in the previous step)

```
            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
                        Console.WriteLine("REST      API      ERROR:      "      +
res.ReasonPhrase);
                }

            }
```

# WEB SERVICES WITH LINKAR 2.0

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

# WEB SERVICES WITH LINKAR 2.0

# THANK YOU

## www.linkarmv.com

## support@kosday.com