



Febrero 2019

by Kosday Solutions





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.





Reference Manager - LinkarWe	ebCor	mmon		? ×	(
Assemblies				Search (Ctrl+E)	
▶ Projects		Name	Path		
Shared Projects	~	LinkarClient.dll	C:\LINKAR\Clients\NET_Framework\x64\LinkarClient.dll		
▶ COM					
▲ Browse					
Recent					
			Brows	se OK Cancel	1
					1
·					
Cambiamos el non	nbr	e a la clase Cla	iss1.cs que nos crea de forma automát	cica por Functions.cs	
y la alimentamos o	con	el siguiente co	ódigo:		
using Svs	ter	n;			
using Lir	ikai	rĆlient;			
using Lir	ıkaı	rCommon;			
		inkankleh Comm			
namespace	: L:	TIIKALMEDCOMMO	ווכ		
ι publi	.c (	class Functio	ons		
{	-				
F	ub:	lic static s	<pre>tring SubDemoLinkar(string text,</pre>	<pre>int seconds)</pre>	
(	•	string error	n = "";		
		string retu	rnValue = "";		
		try			
		( //Remember	to change your credentials to y		
		Credent:	ialsOptions crd	= new	
Credentia	ls(	Options("192	.168.1.1", 11300, "admi	n", "admin",	
"MYENTRYP	IIO	NT", "", "LII	NKAR WEB SERVICES");		
ACCTT Cha	nc	string	args = seconds + ASCII_Chars.D ".	0C4_str + text +	
ASCII_CHA	11.2	LkData	<pre>, subResult = linkarClt_R</pre>	unSubroutine(crd.	
"SUB.DEMC	LI	VKAR", 3, arg	gs);		
		if (su	bResult != null && !subResu	lt.HasError() &&	
subResult	: <b>.</b> Aı	rguments.Leng	gth == 3)		
		//G	et Result Value		





```
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;
            ł
            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...





Add New Project									? X
▶ Recent		.NET F	ramework 4.6 🔹 Sor	t by: Default	- #	E	Search (Ctrl+E)		م
<ul> <li>Installed</li> </ul>		œ	WCF Service Library		Visu	ual C#	Type: Visual C#		
▲ Visual C# Windows Unit Windows Class	versal sic Deskton	Ø.	WCF Service Applicatio	in	Visu	ual C#	A project for creati Application that is	ng WCF Sen hosted in IIS	vice S/WAS
<ul> <li>Veb</li> <li>♦ Office/ShareP</li> </ul>	oint	Q.	WCF Workflow Service	Application	Visu	ual C#			
.NET Core .NET Standard		Q	Syndication Service Lib	rary	Visu	ual C#			
Android Cloud Cross-Platfori	n								
Extensibility ▷ iOS									
Test ▷ tvOS WCF									
Workflow ▷ Visual Basic									
<ul> <li>Visual C++</li> <li>Visual F#</li> <li>SQL Server</li> </ul>									
▷ Telerik Power Query									
Not finding what y Open Visual S	ou are looking for? tudio Installer								
Name:	LinkarWebWCF								
Location:	D:\Workspace\Link	arWebSen	vices				Browse	or	Canad
								UK	Cancel

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

Reference Manager - LinkarW	/ebW0	CF										?	×
Assemblies													P
<ul> <li>Projects</li> </ul>		Name	Path							Name:			
Solution	~	LinkarWebCommon	D:\Workspac	ce\LinkarWebSe	ervices\Linkar	WebCommor	ı\LinkarWebCo	mmon.cs	proj	LinkarWe	ebCommo	n	
▶ COM													
Browse													
									Browse		OK	Can	cel





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.





Overview Connected Services Publish	Publish Publish your app to Azure or another host. Learn more	
	Microsoft Azure App Service IIS, FTP, etc	Folder
	Choose a folder hin\Release\PublishOutnut	Desures
	Publish 😨	
	Continuous Delivery	
	Automatically publish your application to Azure with continuous o	lelivery
	Start	

#### 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.





Add Service Ref	erence	?	$\times$
To see a list of services, click	available services on a specific server, enter a service URL and click Go. To browse Discover.	for availab	le
Address:			
http://localho	st:50255/Service1.svc 🗸 Go	Discover	• <b> </b> •
Services:	Operations:		
⊿ 🧬 Service ⊿ :⊕ Se	e1.svc Ø GetData rvice1		
-¢	IService1		
1 service(s) for	und at address 'http://localhost:50255/Service1.svc'.		
Namespace:			
LinkarWebWC	FI		
Advanced	ОК	Cancel	
Hecho esto po LinkarWebWCF	<pre>dremos hacer uso del mediante el siguiente código: string text = "hello"; int seconds = 5; try { LinkarWebWCF.Service1Client test = .Service1Client(); Console.WriteLine(test.GetData(text, seconds)); } catch (Exception ex) { Console.WriteLine("WCF ERROR: " + ex.Message); }</pre>		new
CONSTRUIR AI	PI REST		





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

Add New Project						? ×
▷ Recent	1	.NET Framework 4.6	<ul> <li>Sort by: Default</li> </ul>	- # E	Search (Ctrl+E)	م
<ul> <li>Installed</li> </ul>			e Web Application	Visual C#	Type: Visual C#	
<ul> <li>Visual C#</li> <li>Windows Univ</li> <li>Windows Class</li> <li>Web</li> </ul>	ersal sic Desktop	ASP.NET Web	b Application (.NET Framework)	Visual C#	Project templates for cr applications. You can c Forms, MVC, or Web A add many other feature	reating ASP.NET create ASP.NET Web Pl applications and es in ASP.NET. Pr
Veb Site     Office/SharePo     .NET Core     .NET Standard	pint					ар
Android Cloud Cross-Platforn Extensibility						
▷ iOS Test ▷ tvOS WCF						
Workflow ▷ Visual Basic ▷ Visual C++ ▷ Visual E#						
SQL Server Telerik Power Ouerv						
Not finding what yo	ou are looking for?					
Open Visual St	udio Installer					
Location:	D:\Workspace\Linka	arWebServices			Browse	
					ОК	Cancel

Seleccionamos la opción Web API.





New ASP.NET Web Application - LinkarWebAPIRest	? ×
ASP.NET 4.6 Templates Empty Web Forms MVC Web API Single Page Azure API App	A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices. Learn more Change Authentication Authentication: No Authentication
Add folders and core references for:	
Web Forms 🖉 MVC 👽 Web API	
Enable Docker support (Requires <u>Docker for Windows</u> )	
Add unit tests	
Test project name: LinkarWebAPIRest.Tests	
	OK Cancel

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

Reference Manager - LinkarV	VebWO	CF				?	×
Assemblies					rch (Ctrl+E)		م
Projects		Name	Path	Na	me:		
Solution	~	LinkarWebCommon	D:\Workspace\LinkarWebServices\LinkarWebCommon\LinkarWebCommon.csproj	Lin	nkarWebCommo	on	
▶ COM							
▶ Browse							
			Bro	wse	ОК	Can	cel





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 autogenerado 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.





Overview Connected Services Publish	Publish Publish your app to Azure or another host. Learn more	
	Microsoft Azure App Service IIS, FTP, etc	Folder
	Choose a folder bin\Release\PublishOutput Publish	Browse
	Continuous Delivery Automatically publish your application to Azure with continuous deli Start	very

#### **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
```











Linkar Suite sl@kosday.com www.linkarmv.com

**SERVICIOS WEB CON LINKAR** 

# GRACIAS

## www.linkarmv.com

## sl@kosday.com

