



# SignalR

Ricardo Vasquez Sierra

[www.ricardovasquez.postach.io](http://www.ricardovasquez.postach.io)



**Microsoft**  
**CERTIFIED**  
Professional

# Agenda

- ❖ SignalR
- ❖ Configurando SignalR
- ❖ Hubs
- ❖ Conexiones
- ❖ Implementación



# ¿Que es SignalR?

- ❖ Biblioteca de ASP.Net para crear aplicaciones en tiempo real.
- ❖ Abstracción sobre HTTP de los transportes requeridos para realizar apps en tiempo real.
- ❖ Librerías para servidor y cliente.

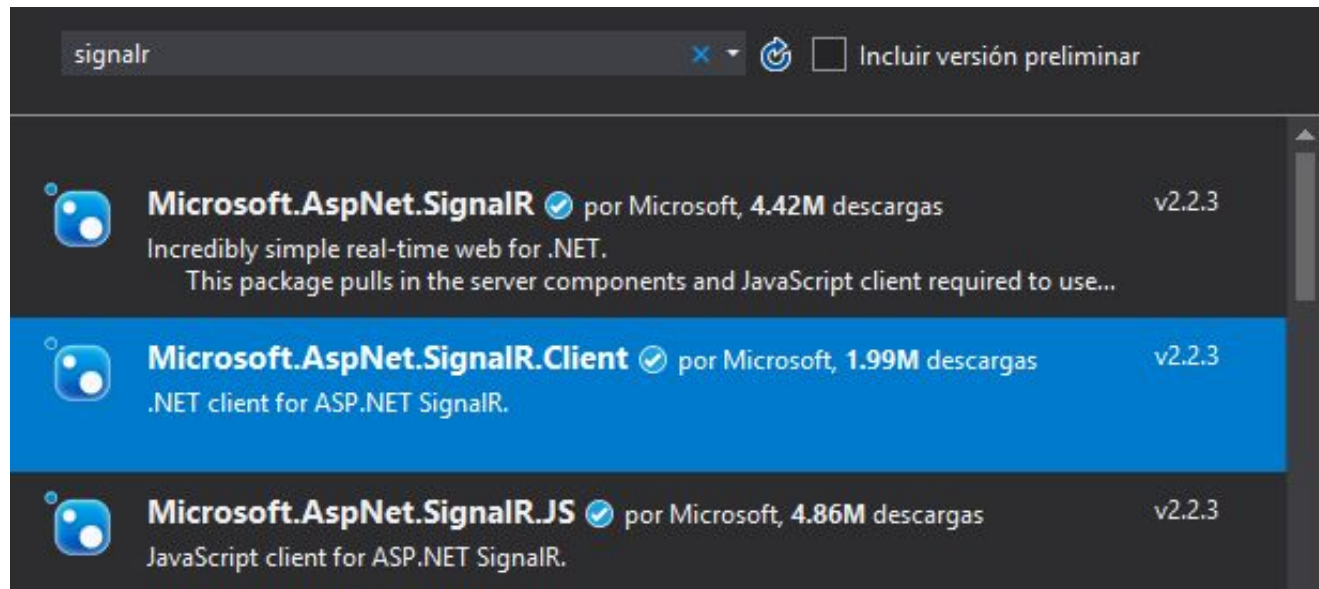


# ¿Cuándo Usar SignalR?

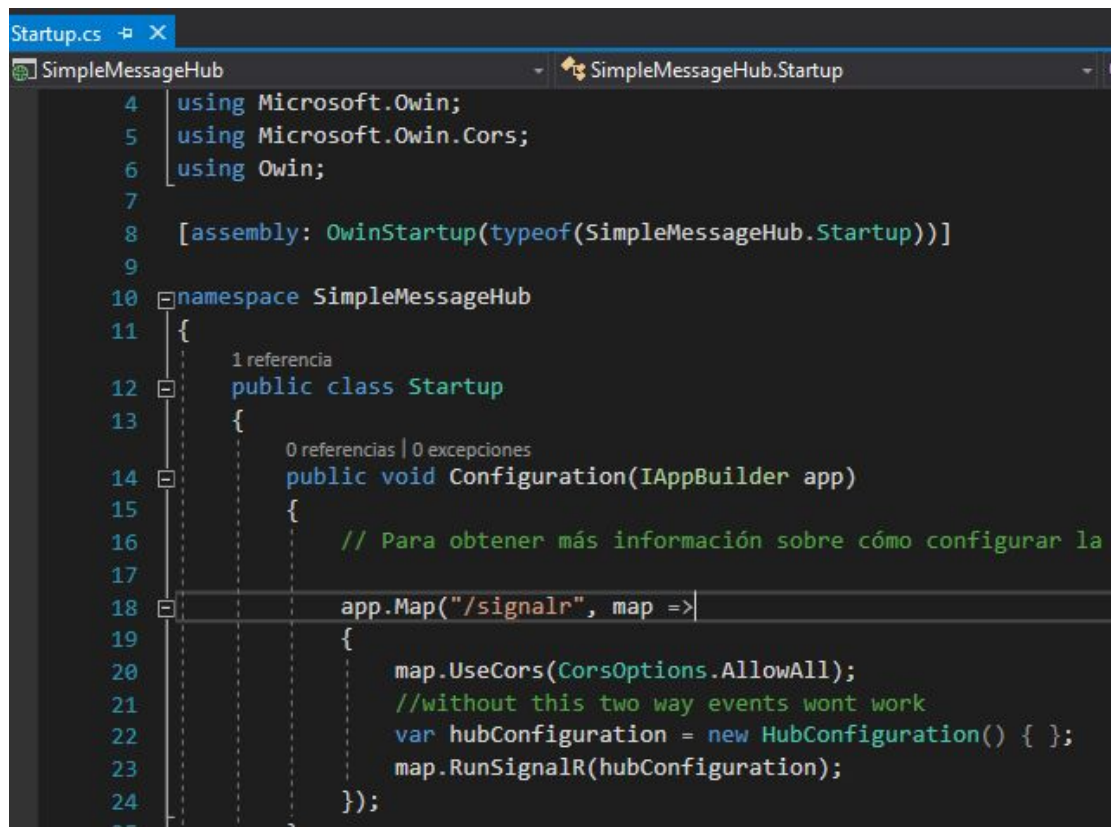
- ❖ Juegos
  - ❖ Aplicaciones sociales
  - ❖ Colaboracion de negocios
  - ❖ Mensajeria
  - ❖ Formas en tiempo real
- 
- ❖ **Todo lo que necesite data en tiempo real**

# Configurando SignalR

- ❖ <http://www.asp.net/signalr>
- ❖ NuGet
  - OWIN
  - Scripts
- ❖ Github



# Inicio de SignalR

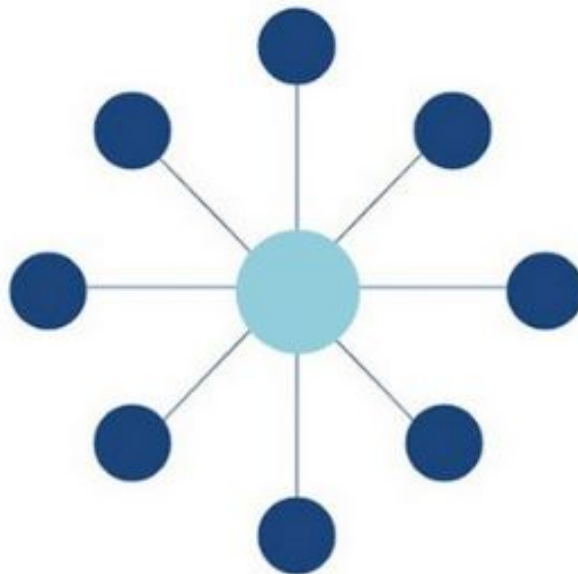


```
Startup.cs
SimpleMessageHub
SimpleMessageHub.Startup

4 using Microsoft.Owin;
5 using Microsoft.Owin.Cors;
6 using Owin;
7
8 [assembly: OwinStartup(typeof(SimpleMessageHub.Startup))]
9
10 namespace SimpleMessageHub
11 {
12     1 referencia
13     public class Startup
14     {
15         0 referencias | 0 excepciones
16         public void Configuration(IAppBuilder app)
17         {
18             // Para obtener más información sobre cómo configurar la
19             app.Map("/signalr", map =>
20             {
21                 map.UseCors(CorsOptions.AllowAll);
22                 //without this two way events wont work
23                 var hubConfiguration = new HubConfiguration() { };
24                 map.RunSignalR(hubConfiguration);
25             });
26         }
27     }
28 }
```

# Hubs de SignalR

- ❖ Microsoft.AspNet.SignalR.Hub
- ❖ Librería de Servidor
- ❖ Permite conectividad bidireccional



# Eventos de Hub

0 referencias | 0 excepciones

```
public void MetodoEjemplo()  
{  
    Clients.Caller.MetodoEjemploEjecutado("parametro1","parametro2","parametro3");  
}
```



# Transportes de Hubs (TCP's)

## ❖ Transportes

- **WebSockets** es el único transporte que establece una conexión bidireccional entre cliente y servidor (Santogrial).
- **Eventos de Servidor (SSE)** es el transporte en el que el explorador recibe actualizaciones a través de una conexión HTTP (Estandarizado en HTML5 con llamadas XHR).
- **Forever Frame (IE)** crea una conexión permanente mientras que el mismo no sea cerrado, la cual es usada por el servidor para enviar actualizaciones continuas y el cliente utiliza llamadas AJAX tradicionales para comunicarse con el servidor.
- **AJAX Long Polling** establece una conexión AJAX en la que se están enviando datos de manera continua desde el servidor hacia el cliente para mantener la conexión activa y es común en navegadores antiguos. Es el último transporte viable en caso de no haber otro.

## ❖ SignalR automáticamente utiliza el primero que encuentre soportado

# Demostración

Configurando Hub SignalR y Conectando Clientes  
Xamarin a Hub



# Implementando SignalR en Servidor (IIS)

- ❖ Instalar SDK de Azure
- ❖ Habilitar WebSockets
- ❖ Habilitar .Net v4.5



# Gracias! :)

Referencias:

MSDN  
WintellectNOW



**Microsoft**  
**CERTIFIED**  
Professional