

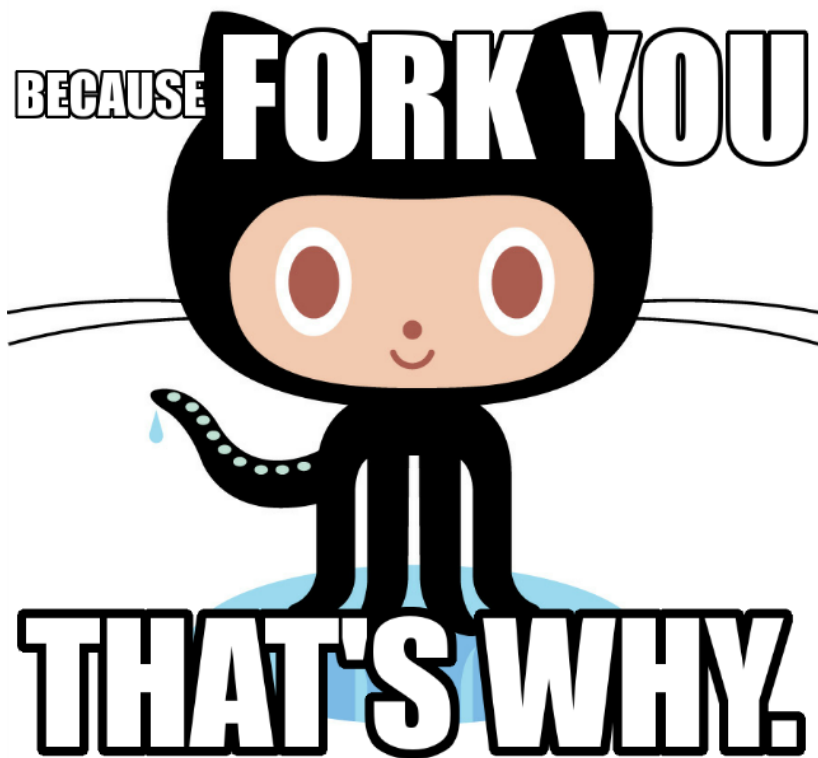
# SignalR: Add real-time to your applications

*by Eugene Zharkov*



**HOT**CODE

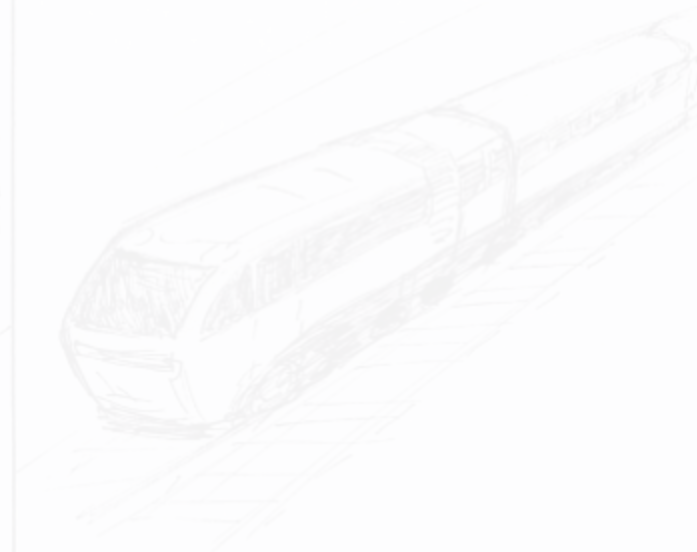
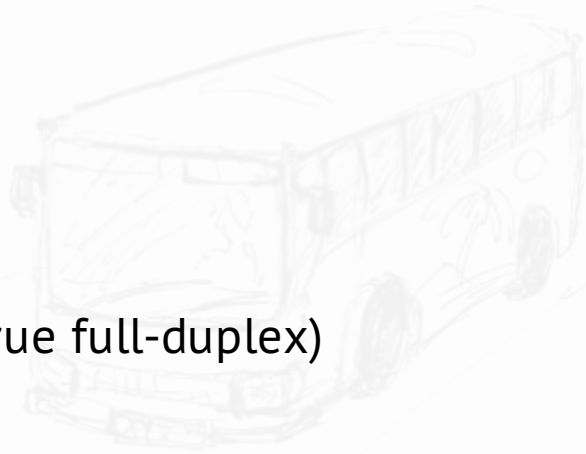
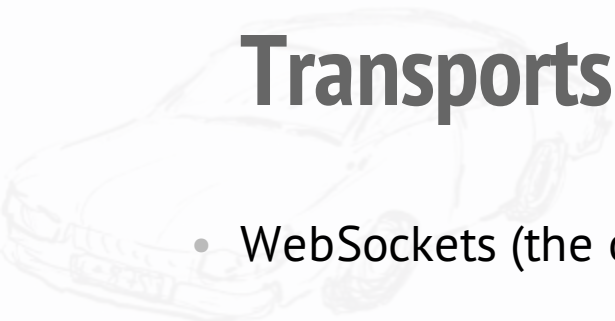
# Why to use SignalR?



<https://github.com/SignalR/SignalR>

# Transports

- WebSockets (the only true full-duplex)
- Server Sent Events
- Forever Frame
- Long polling



# WebSockets

- html5 feature
- works on top of TCP protocol
- full-duplex
- low latency, low overhead
- you need >IE10 and IIS8
- not all browsers / servers/ proxies support it

# Server Sent Events

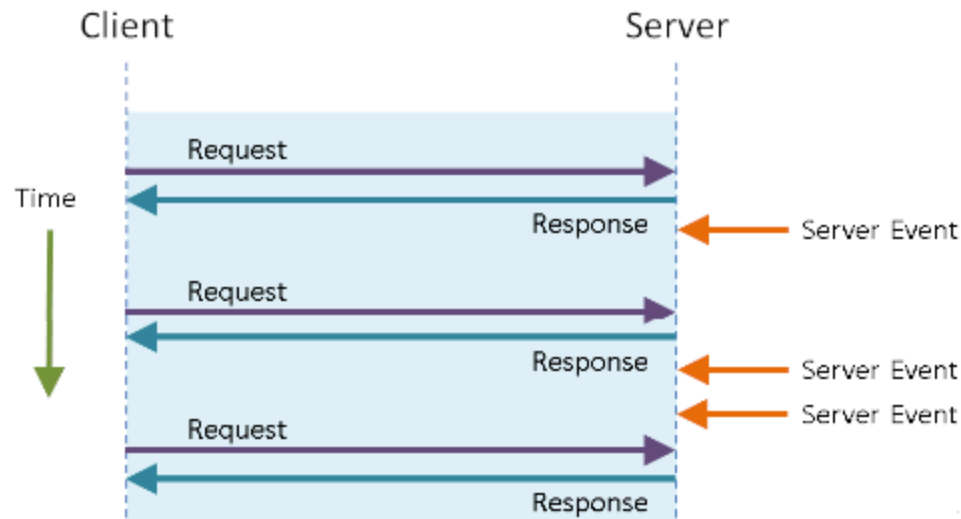
- requires a single connection between client-server
- uses JS API EventSource through which client can request a particular URL to receive data stream
- no need to reconnect
- works in server-2-client direction only

# Forever Frames

- only for IE
- hidden iframes are in the load loop with chunks of data

# Long Polling

- high overhead on requests/response (headers etc.)
- medium latency



# Transports

**Web Browser Transport Requirements**

Transport	Internet Explorer	Chrome (Windows or iOS)	Firefox	Safari (OSX or iOS)	Android
WebSockets	10+	current - 1	current - 1	current - 1	N/A
Server-Sent Events	N/A	current - 1	current - 1	current - 1	N/A
ForeverFrame	8+	N/A	N/A	N/A	4.1
Long Polling	8+	current - 1	current - 1	current - 1	4.1



# Clients

- iOS ([SignalR-ObjC](#))
- Android ([SignalA](#))
- Windows RT / Phone
- [jQuery](#), [Backbone](#)
- Mono
- QT ([SignalR-QT](#))

# Requirements

- Windows Server 2008 R2/2012, Windows 7/8
- IIS 8. IIS 7 and 7.5. Support for [extensionless URLs](#) is required.
- $\geq$  .NET 3.5, WinRT, Silverlight

**Does NOT support IE 6/7**



# Persistent Connection

- the raw connection
- IHTTP handler (OnConnection, OnDisconnection, OnReconnection)
- custom logic must be implemented

# Hub

- high-level API on top of Persistent Connection
- route automatically mapped
- clients can be divided by groups

# Add to project

The screenshot displays the NuGet Package Manager interface. On the left, a sidebar shows the navigation menu with 'Online' selected, and 'Search Results' highlighted. The main area shows search results for 'SignalR'. At the top, there are filters for 'Stable Only' and 'Sort by: Relevance'. The search results list several packages, with 'Microsoft ASP.NET SignalR JavaScript Client' at the top, featuring an 'Install' button. Below it are 'Microsoft ASP.NET SignalR Core Components', 'Microsoft ASP.NET SignalR', 'Microsoft ASP.NET SignalR OWIN', and 'Microsoft ASP.NET SignalR System.Web Components'. On the right, a detailed view for the 'Microsoft ASP.NET SignalR JavaScript Client' is shown, including its creator (Microsoft), ID, version (1.1.1), last published date (5/20/2013), download count (74897), and a description: 'JavaScript client for ASP.NET SignalR.' It also lists tags like 'Microsoft AspNet SignalR' and 'AspNetSignalR JS', and a dependency on 'jQuery (≥ 1.6.4)'.

Installed packages

Stable Only Sort by: Relevance

SignalR

Online

- All
- Local Test
- NuGet official package source
- Search Results

Updates

Microsoft ASP.NET SignalR JavaScript Client  
JavaScript client for ASP.NET SignalR. Install

Microsoft ASP.NET SignalR Core Components  
Core server components for ASP.NET SignalR.

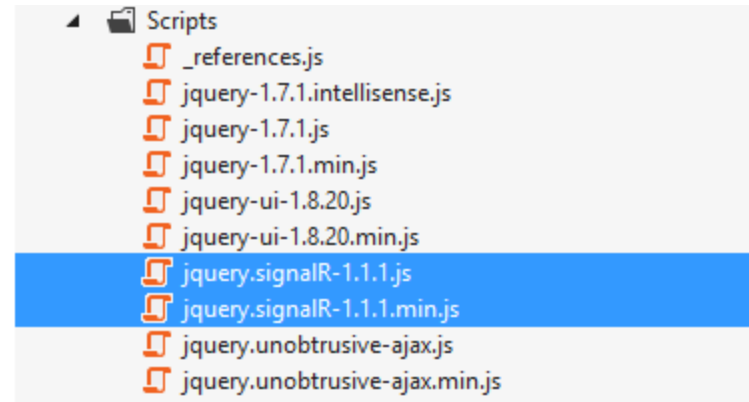
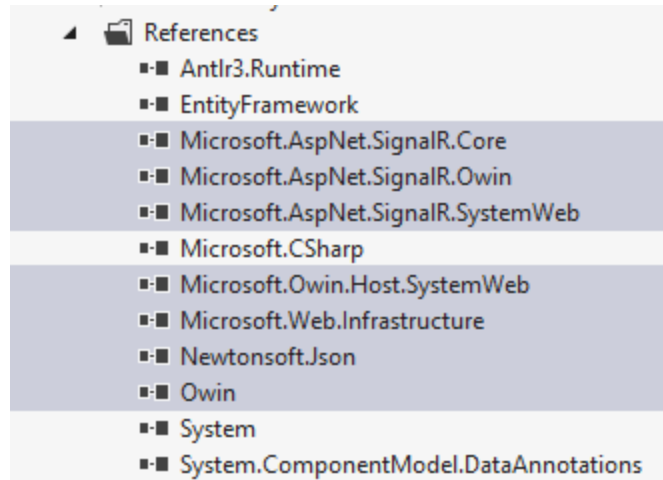
Microsoft ASP.NET SignalR  
Incredibly simple real-time web for .NET.

Microsoft ASP.NET SignalR OWIN  
OWIN componenets for ASP.NET SignalR.

Microsoft ASP.NET SignalR System.Web Components

Created by: Microsoft  
Id: Microsoft.AspNet.SignalR.JS  
Version: 1.1.1  
Last Published: 5/20/2013  
Downloads: 74897  
[View License Terms](#)  
[Project Information](#)  
[Report Abuse](#)  
Description:  
JavaScript client for ASP.NET SignalR.  
Tags: Microsoft AspNet SignalR  
AspNetSignalR JS  
Dependencies:  
jQuery (≥ 1.6.4)  
Each item above may have sub-

# New References and Scripts



## Hub registration : C#

```
01. public class MvcApplication : System.Web.HttpApplication
02. {
03.     protected void Application_Start()
04.     {
05.         RouteTable.Routes.MapHubs();
06.         AreaRegistration.RegisterAllAreas();
07.         WebApiConfig.Register(GlobalConfiguration.Configuration);
08.         FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
```



## Hub object | C#

```
01. public class Documents : Hub
02. {
03.     public void Send(string documentName)
04.     {
05.         Clients.All.addMessage(documentName);
06.     }
07. }
```

# Subscribe to Hub messages | JS

```
01. var documentHub = $.connection.documents;  
02. documentHub.client.addMessage = function (documentName) {  
03.     $("#documents").append("<li>" + documentName + "</li>");  
04. };
```

# Hub object | JS

```
01. $.connection.hub.logging = true;
02. $.connection.hub.start().done(function () {
03.     $("#addDocument").click(function () {
04.         documentHub.server.send($("#document").val());
05.     });
06. });
```

## Working with Hub | WinRT

```
01. var connection = new HubConnection("http://localhost:59576")
02. var hub = connection.CreateHubProxy("documents");
03. hub.On("addMessage", message =>
04.     Dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>
05.     {
06.         DocumentsCounter.Text = (++_newDocuments).ToString()
07.     }));
```

## Working with Hub | Android (SignalA)

```
01. HubConnection con = new HubConnection("http://localhost:59570")
    this, new LongPollingTransport())
02. IHubProxy hub = con.CreateHubProxy("documents");
03. hub.On("addMessage", new HubOnDataCallback() {
04.     @Override
05.     public void OnReceived(.. args) {
06.         { //... });
07. con.Start();
```

# That's all folks

- @2j2e
- eu.zharkov@gmail.com
- <http://bit.ly/signalr0>