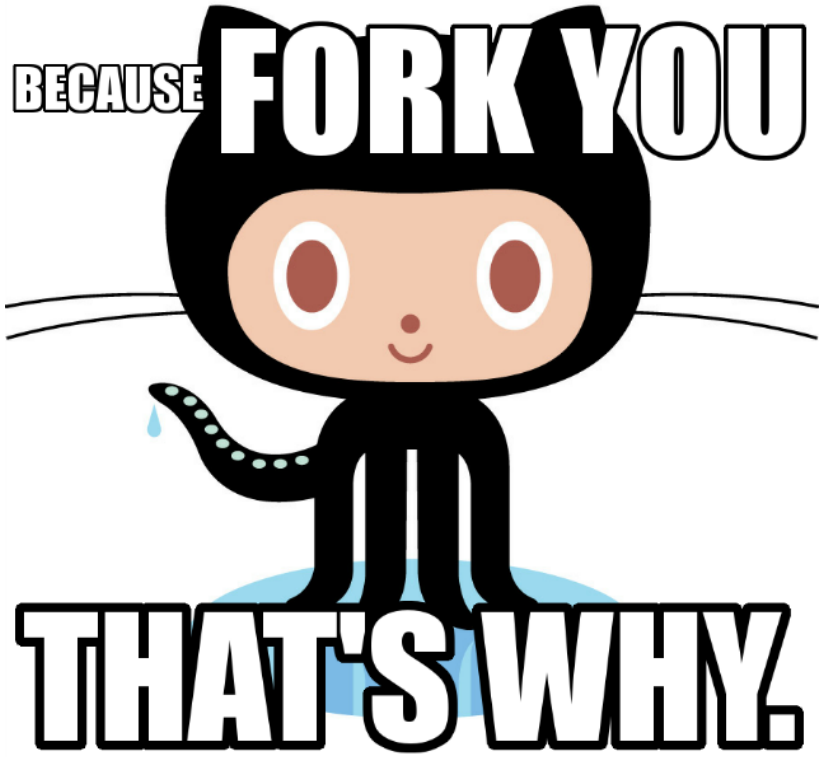
SignalR: Add real-time to your applications

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

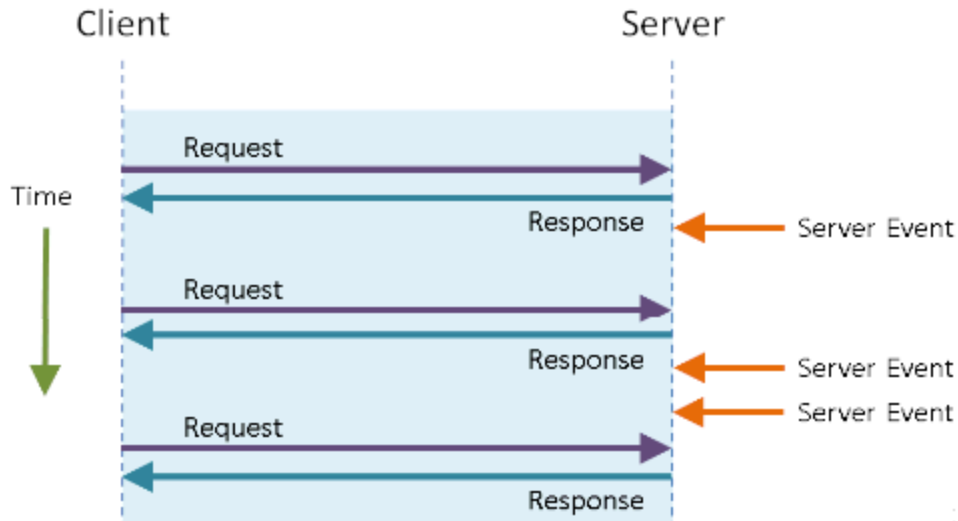
- requires a single connection between client-server
- uses JS API EventSource through which client can request a particular URL to receive data stream
- no need to reconnect
- works in server-2-client direction only

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.
- >= .NET 3.5, WinRT, Silverlight

Does NOT support IE 6/7

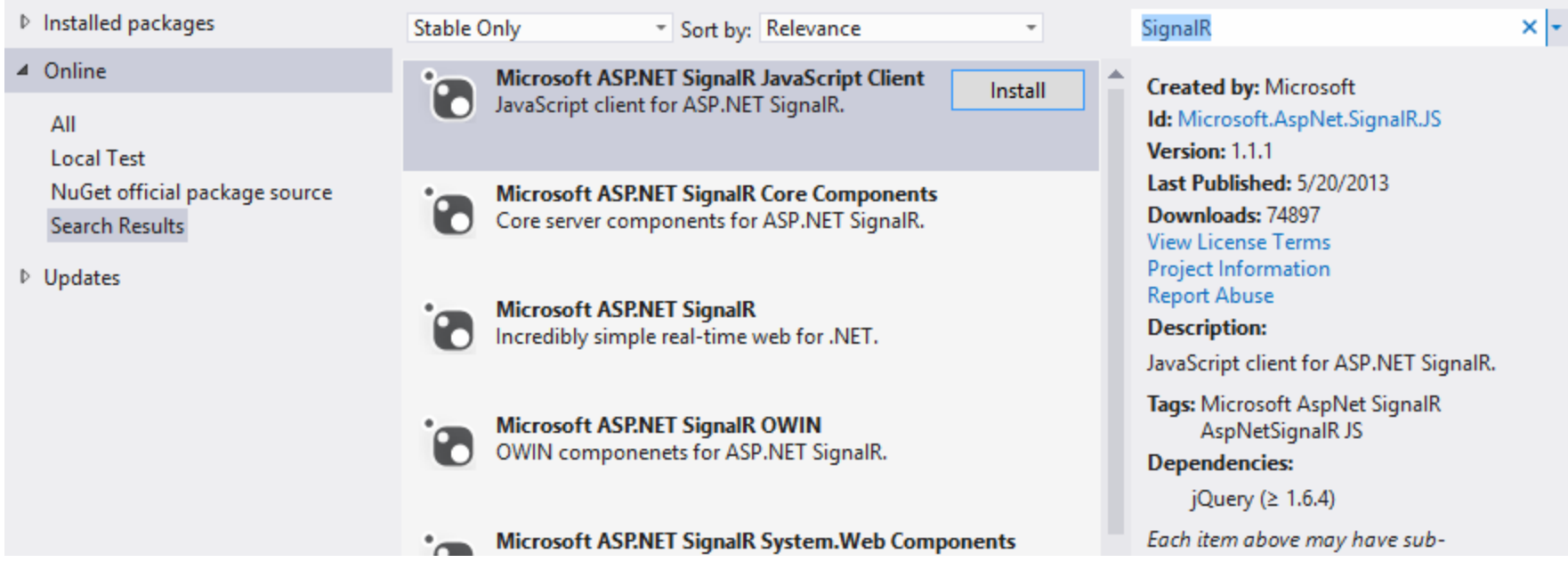
Persistent Connection

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

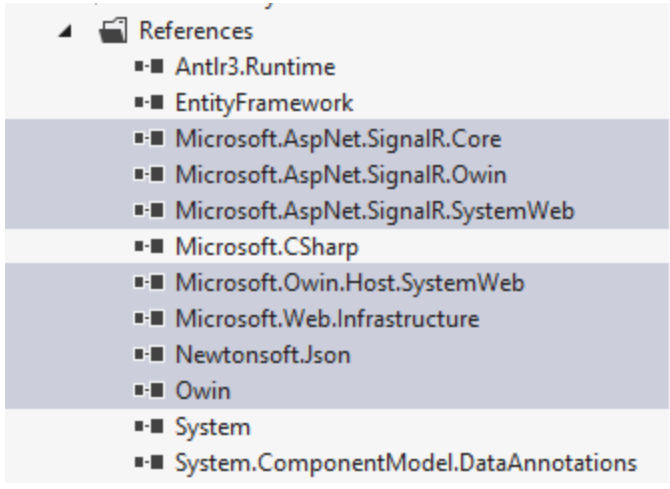
Hub

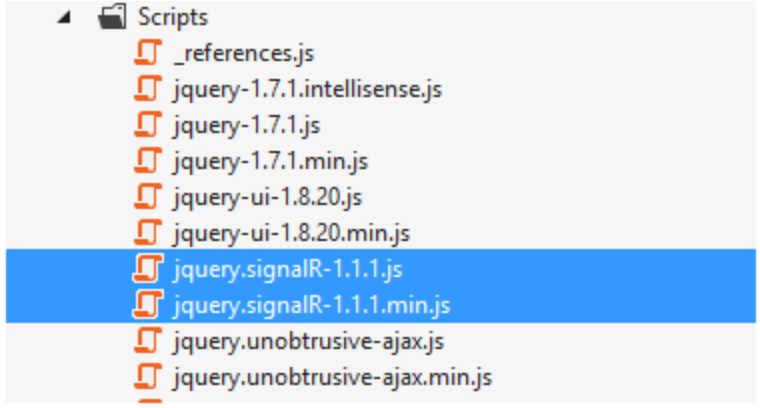
- high-level API on top of Persistent Connection
- route automatically mapped
- clients can be divided by groups

Add to project



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.Configurat
08.         FilterConfig.RegisterGlobalFilters(GlobalFilters.Fi
```

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:5957
                    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