Telerik Academy

ASP.NET SignalR

ASP.NET MVC

Telerik Software Academy

http://academy.telerik.com

Telerik Academy

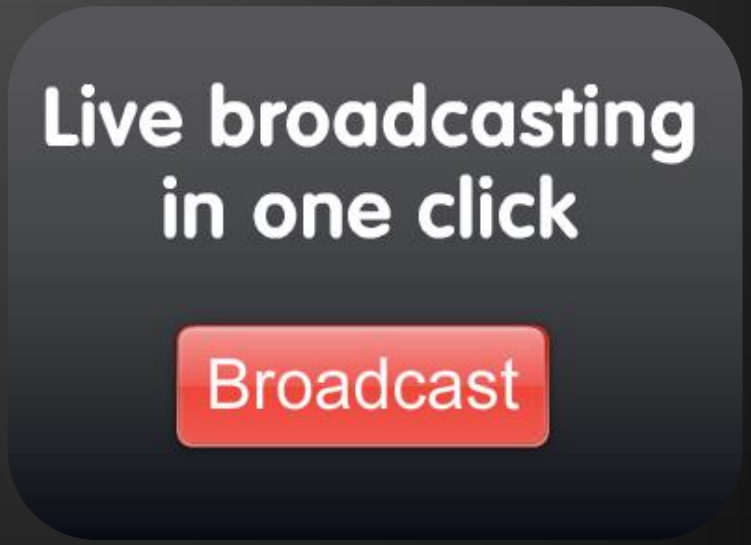
Table of Contents

- Web-based real-time communication
  - Problems and solutions
- ASP.NET SignalR - Overview & concepts
- Hubs
  - Server-side API
- Clients
  - Client-side API
- Chat Live Demo

Telerik Academy

Web based real-time communication

Now, now and now!



Problems & Solutions

- Users want data
  - Now & instant
  - Up-to-date
  - Delivered to any device, over any connection
- Examples
  - Live searches/updates
  - Stock streamers, auctions
  - Live scores, betting, interactive games
- Real-time feedback, real-time notifications

Telerik Academy

Problems & Solutions

- Developers need to provide real time data
  - But not only for web applications
  - What about mobile devices & apps?
  - What about traditional desktop applications?
  - What about server-to-server?
- Push communication beyond the web is a need
- Think, design & implement Push Services

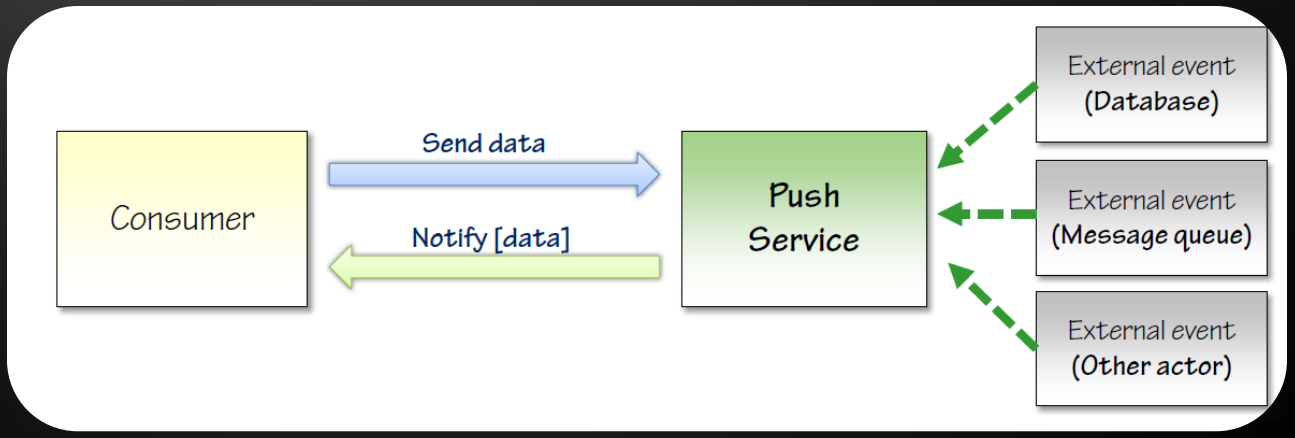
Telerik Academy

Push Services Pattern

Not official of course!

Push Services Pattern

- Push Services are not an official pattern
- Model a service that
  - accepts incoming connections from callers
  - is able to push data down to callers
  - abstracts from communication only details



HTTP Protocol

- Web communication – HTTP!
- Basically request-response
- Realize Push services with HTTP
  - Periodic Polling
  - Long Polling
  - Forever Frame
  - Server-Sent Events (SSE)
  - Web Sockets
- http://tinyurl.com/push-services

Web Sockets

- Web Sockets – latest piece of technology!
- Positives:
  - Easy
  - Constant connection
  - Send only small details
- Negatives
  - Only with Windows 8/Server 2012
  - Network considerations
  - Maybe some time, but not today

Telerik Academy

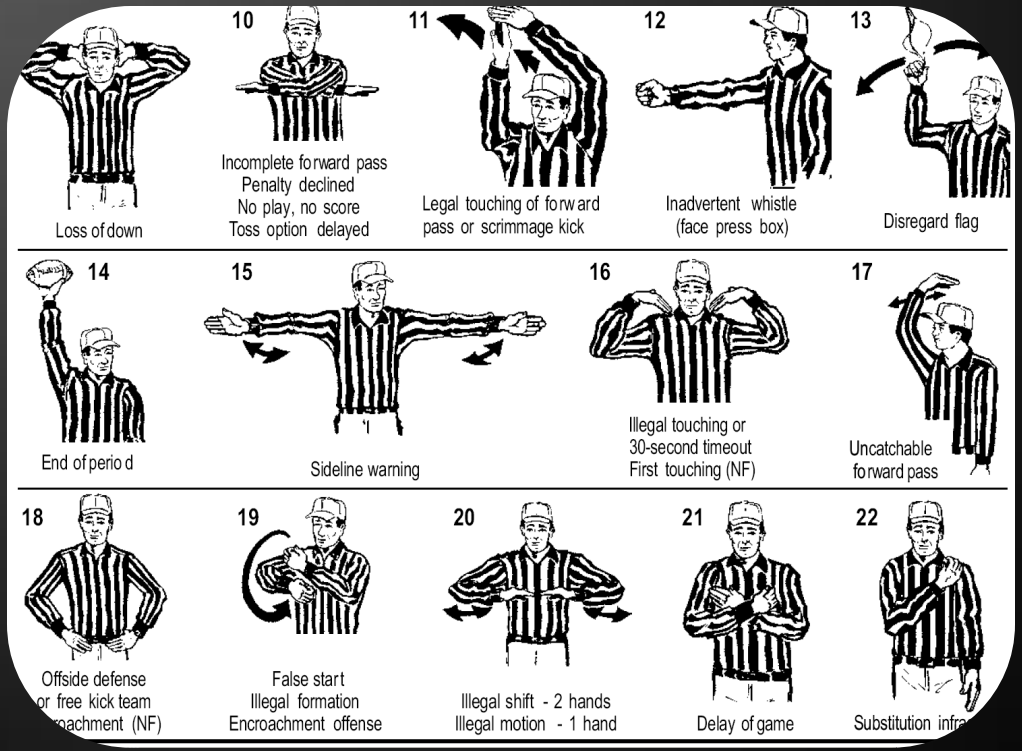
Web Sockets

Live Demo

Telerik Academy

SignalR Solution

Part of ASP.NET but not tied to it!



SignalR

- SignalR is:
  - A server-side framework to write push services
  - Clients for easy communication on any device
  - Optimized for asynchronous operations
- Operation:
  - Persistent connection
  - Data is sent as signal through connection ID
  - Volatile

SignalR

- Framework is based on Interfaces:
- Programming modes:
  - Persistent connection
  - Hubs
- Hubs offer predefined API for most scenarios
- SignalR example:
  - http://shootr.signalr.net/

SignalR

- Packages at NuGet:
  - Microsoft.AspNet.SignalR
  - Microsoft.AspNet.SignalR.Core
  - Microsoft.AspNet.SignalR.Owin
  - Microsoft.AspNet.SignalR.Js
  - Microsoft.AspNet.SignalR.Client
  - Microsoft.AspNet.SignalR.Utils

SignalR

- Installation
  - Download Microsoft.AspNet.SignalR
  - In Global.asax register the hubs

```
RouteTable.Routes.MapHubs();
```

  - Create directory Hubs
  - Add new SignalR classes to it
  - Done!
- To update from 1.x to 2.0

Telerik Academy

SignalR Hubs

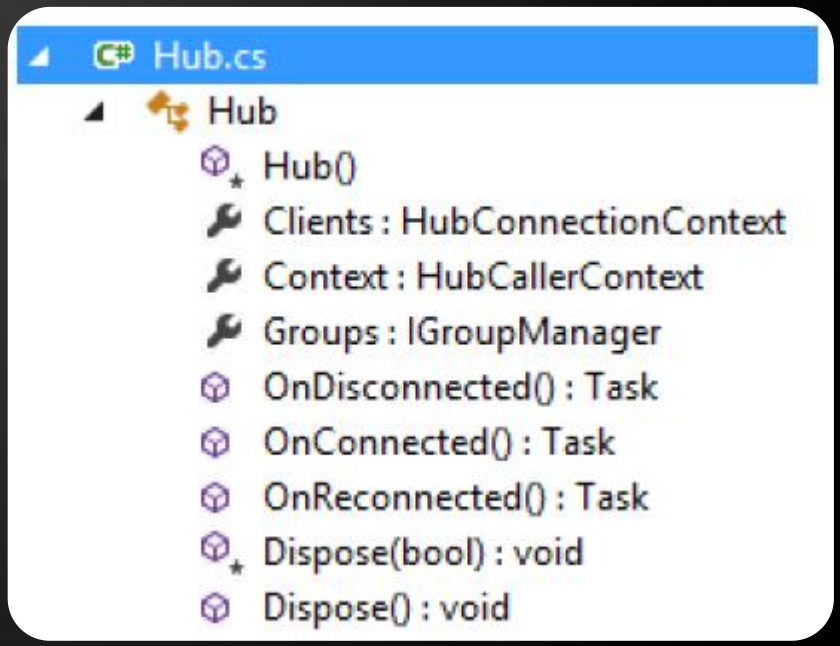
Server-side

Hubs

- Classes to implement push services
  - Abstraction on top of persistent connection
  - Convention over configuration
- Perfect for sending data from server to clients
- Conventions
  - Public methods are callable from clients
  - Send data by invoking client methods

Hub Programming

- Hub name reflected onto external API
- Return simple type, complex type or Task
- Objects and collection – automatically to JSON
- Context
  - ConnectionId
  - Request
  - Headers
  - RequestCookies
  - QueryString
  - User

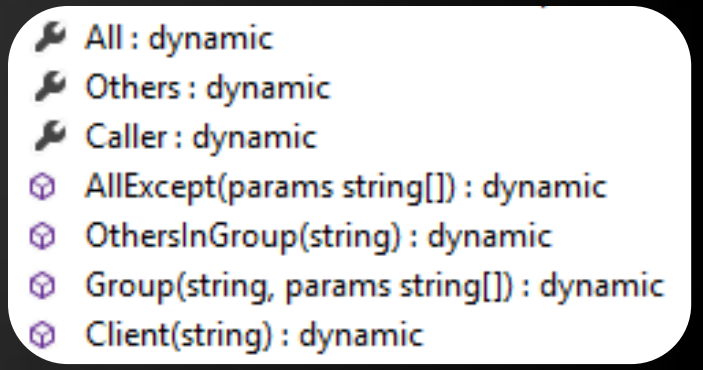


Hub Programming

- Base endpoint is /signalr
- JS metadata from /signalr/hubs
- Basically two protocol steps
  - negotiate: which transport do you support?
  - connect: OK, here is my persistent connection
- 'Best' transport negotiation
  - W. Sockets -> SSE -> Forever frame -> L. polling
- Any data is JSON encoded

Clients Property

- Clients property to send messages to clients
- Holds dynamic properties and methods
- Target method with parameters
  - Dynamically 'injected'
  - Serialized
  - Embedded into response




```
public void SendMessage(string message)
{
        var msg = string.Format("{0}: {1}",
                Context.ConnectionId, message);
        Clients.All.newMessage(msg);
}
```

Groups Property

- Groups - typical base pattern in push scenarios
- Add connections to groups
- And send messages to particular groups

```
public void JoinRoom(string room)
{
    Groups.Add(Context.ConnectionId, room);
}
public void SendToRoom(string room, string message)
{
    var msg = string.Format("{0}: {1}",
        Context.ConnectionId, message);
    Clients.Group(room).newMessage(msg);
}
```

- Groups are not persisted on server!

Telerik Academy

SignalR Clients

Client-side

Clients

- Consumers
  - Client applications
  - Other services/hubs
- SignalR provides a variety of client libraries
  - WinRT
  - Windows Phone 8
  - Silverlight 5
  - jQuery
  - C++

jQuery Client

- Automatic proxy code via /signalr/hubs
- Script generated based on hubs declaration
- Hubs become properties on $.connection
  - Example $.connection.chatHub
- Hub name camel cased
- $.connection.hub.start() – start connection

```
$.connection.hub.start({ transport: 'longPolling'});

var chat = $.connection.chat;
chat.server.joinRoom('private');
```

jQuery Client

- Define client side methods
  - They can be invoked by the hub

```
var chat = $.connection.chat;
chat.client.newMessage = onNewMessage;

function onNewMessage(message) {
        $('#messages').append(message);
    }
```

- Events for connection state handling
- Detect slow connections
- Cross-domain support

Telerik Academy

Chat

Live Demo

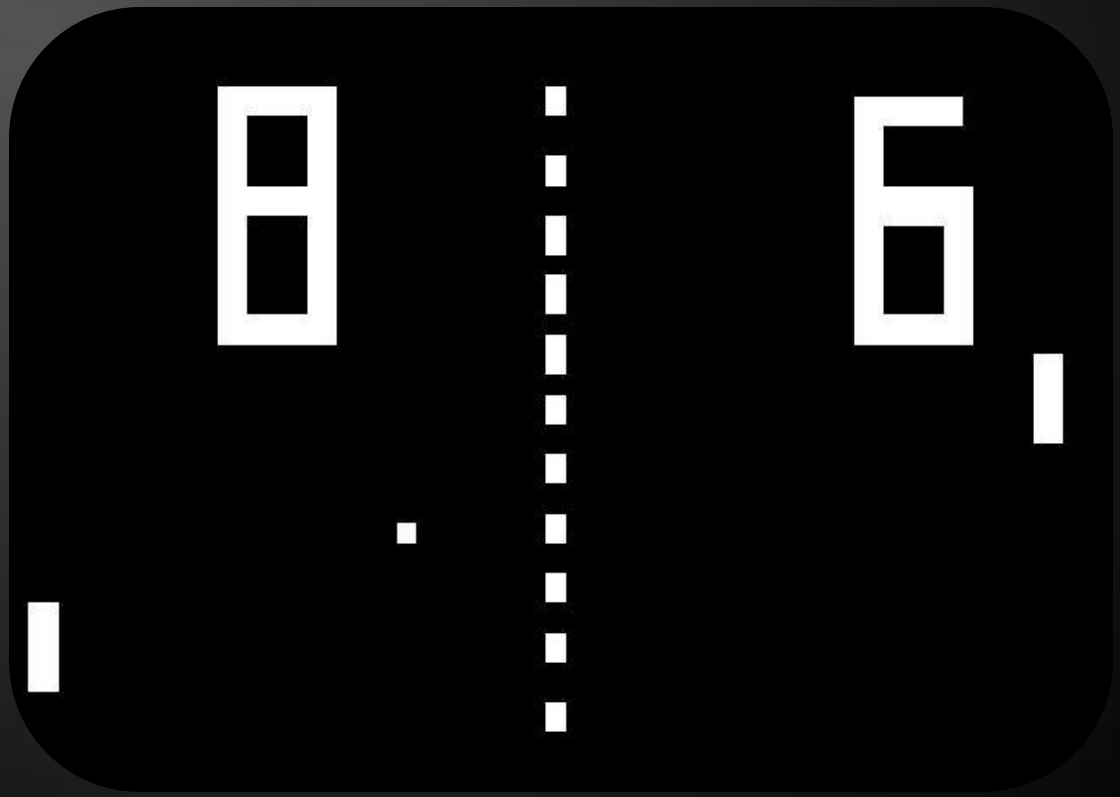
Telerik Academy

ASP.NET SignalR

http://academy.telerik.com

Homework

1. Create the Ping Pong game for two players with SignalR. You can use external libraries like CGWeb.



Free Trainings @ Telerik Academy

- "C# Programming @ Telerik Academy
  - csharpfundamentals.telerik.com

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com