

زمانیکه صحبت از برنامه‌های بلادرنگ می‌شود با کاربرانی سر و کار داریم که نیاز دارند تا اطلاعات مورد نیاز خود را همواره و بلافاصله در آخرین وضعیت به روز آن مشاهده کنند. در این بین، کلاینت می‌خواهد یک برنامه وب باشد یا سیلورلایت و یا یک برنامه نوشته شده با WPF. حتی برنامه‌های موبایل را نیز باید به این لیست اضافه کرد. در اینجا کلمه بلادرنگ به معنای ارسال اطلاعات از طرف سرور به کلاینت‌ها با فاصله زمانی بسیار کوتاهی از به روز رسانی اطلاعات صورت گرفته در سمت سرور است.

نمونه‌ای از این برنامه‌ها شامل موارد ذیل هستند:

- اطلاع رسانی همزمان به گروهی از کاربران
- جستجوهای زنده و به روز رسانی‌هایی از این دست
- نمایش بلادرنگ قیمت‌ها و وضعیت تجاری محصولات و سهام‌ها
- بازی‌های تعاملی
- برنامه‌های گروهی و تعاملی (مانند برنامه‌های Chat)
- برنامه‌های شبکه‌های اجتماعی (برای مثال پیام جدیدی دارید؛ شخص خاصی آنلاین یا آفلاین شد و امثال آن)

بنابراین به صورت خلاصه قصد داریم به ارائه بازخوردها و اطلاع رسانی‌های بلادرنگ یا نسبتاً سریع و به روز از سمت سرور به کلاینت‌ها برسیم.

برای مثال یک دیتاگرید را در نظر بگیرید. دو کاربر در شبکه صفحه یکسانی را گشوده‌اند و یکی از آن‌ها مشغول به ویرایش و یا حذف اطلاعات است. در ارتباطات بلادرنگ کاربر یا کاربران دیگر نیز باید (یا بهتر است) در زمانیکه گرید یکسانی را گشوده‌اند، بلافاصله آخرین تغییرات را ملاحظه کنند. یا حتی حالتی را در نظر بگیرید که شخصی SQL Server management studio را گشوده و در آنجا مشغول به تغییر اطلاعات گردیده است. در این حالت نیز بهتر است آخرین تغییرات بلافاصله به اطلاع کاربران رسانده شوند.

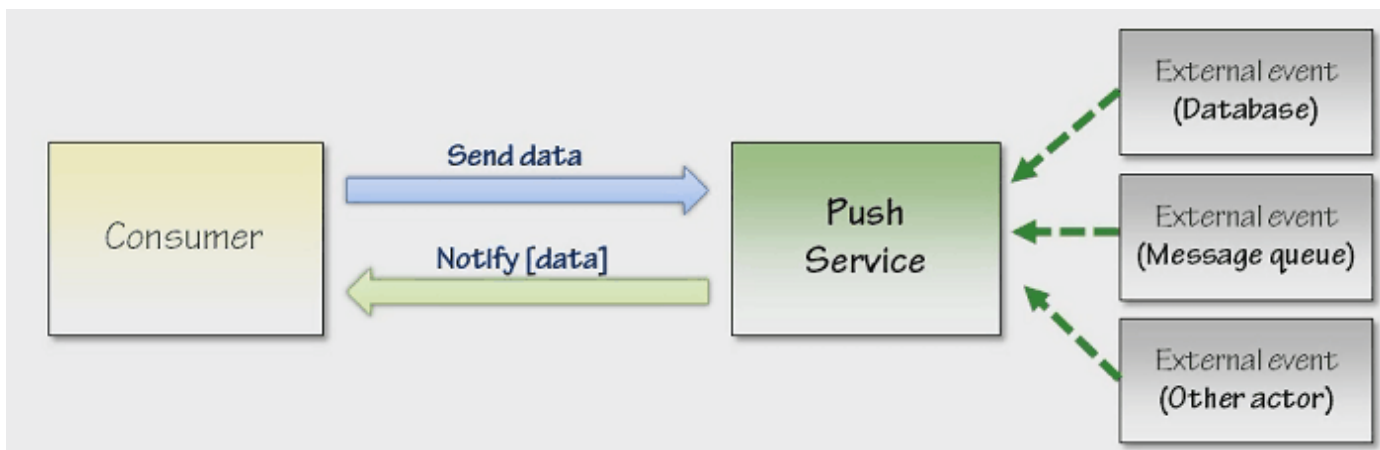
معرفی الگوی Push service

البته باید دقت داشت که الگوی push service یک الگوی رسمی ذکر شده در گروه‌های مرسوم الگوهای طراحی نیست، اما مفهوم آن سرویسی است که چندین کار ذیل را انجام می‌دهد:

الف) پذیرش اتصالات از چندین مصرف کننده. مصرف کننده‌ها در اینجا الزاماً محدود به کلاینت‌های وب یا دسکتاپ نیستند؛ می‌توانند حتی یک سرور یا سرویس دیگر نیز باشند.

ب) قادر است اطلاعات را به مصرف کننده‌های خود ارسال کند. این سرویس می‌تواند یک برنامه ASP.NET باشد یا حتی یک سرویس متداول ویندوز.

ج) در اینجا چندین منبع خارجی مانند یک بانک اطلاعاتی یا تغییرات رخ داده توسط یک سخت افزار که می‌تواند سبب بروز رخدادهایی در push service گردند نیز می‌تواند وجود داشته باشند. هر زمان که تغییری در این منابع خارجی رخ دهد، مایل هستیم تا مصرف کننده‌ها را مطلع سازیم.



پروتکل HTTP و ارتباطات بلادرنگ

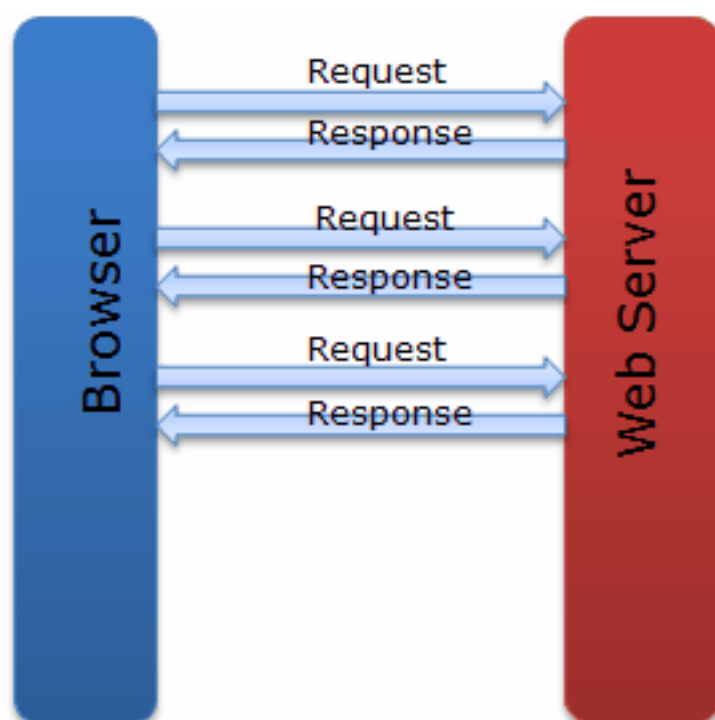
پروتکلی که در ارتباطات بلادرنگ مبتنی بر SignalR مورد استفاده قرار می‌گیرد، HTTP است و از قابلیت‌های Request و Response آن در اینجا بیشترین بهره برده می‌شود. پیاده سازی Push عموماً بر مبنای یکی از روش‌های متداول زیر است:

Periodic polling (1)

به این معنا که مثلاً هر 10 ثانیه یکبار، کاری را انجام بده؛ مانند ارسال متناوب: آیا تغییری رخ داده؟ آیا تغییری رخ داده؟ و به همین ترتیب. این روش اصلاً بهینه نبوده و منابع زیادی را خصوصاً در سمت سرور مصرف خواهد کرد. برای مثال:

```

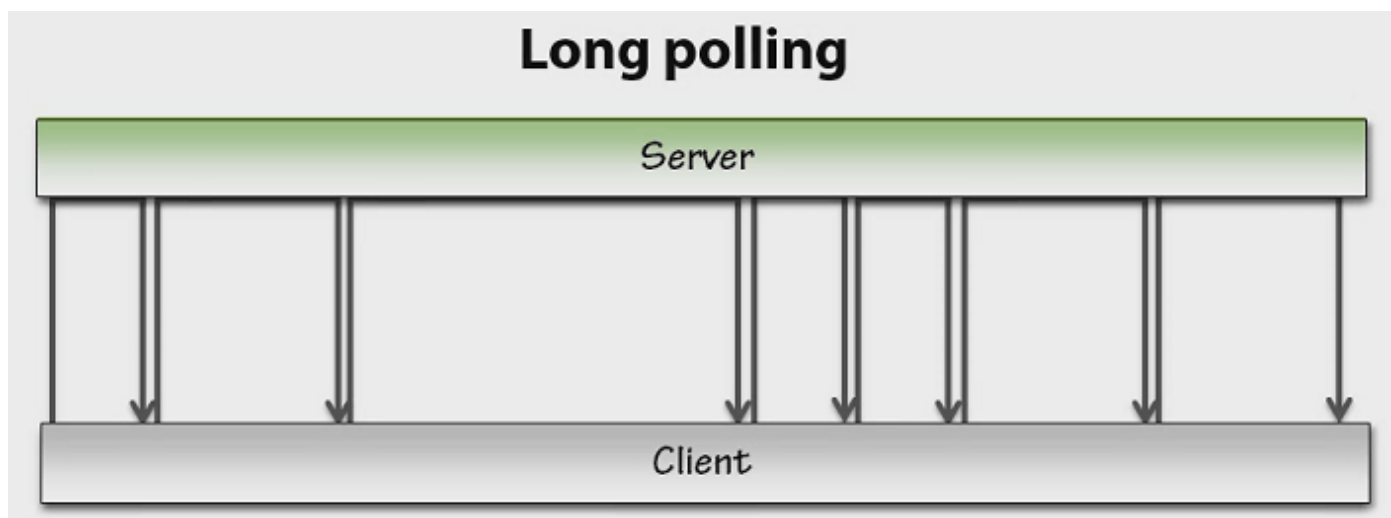
function getInfo() {
    $.ajax("url", function (newInfo){
        if ( newInfo != null) {
            // do something with the data
        }
    });
    // poll again after 20 seconds
    setTimeout(getInfo,20000);
}
// start the polling loop
getInfo();
  
```



Long polling (2)

به آن HTTP Streaming یا Comet هم گفته می‌شود. این روش نسبتاً هوشمند بوده و کلاینت اتصالی را به سرور برقرار خواهد کرد. سرور در این حالت تا زمانیکه اطلاعاتی را در دسترس نداشته باشد، پاسخی نخواهد داد. برای نمونه:

```
function getNewInfo(){
    $.ajax("url", function (newinfo) {
        // do something with the data
        // start the new request
        getNewInfo();
    });
}
// start the polling loop
getNewInfo();
```



این روش نسبت به حالت Periodic polling بهینه‌تر است اما نیاز به اتصالات زیادی داشته و همچنین تردهای بسیاری را در سمت سرور به خود مشغول خواهد کرد.

Forever frame (3)

فقط در IE پشتیبانی می‌شود. در این روش یک Iframe مخفی توسط مرورگر تشکیل شده و از طریق آن درخواستی به سرور ارسال می‌شود. سپس سرور متناوباً با تزریق اسکریپت‌هایی به این Iframe سبب فراخوانی مجدد وضعیت خود می‌گردد. در این روش نیز به ازای هر درخواست و پاسخ، ارتباطات گشوده و بسته خواهند شد.

SSE یا Server Sent Events (4)

این مورد جزو استاندارد HTML5 است. در اینجا اتصالی برقرار شده و داده‌ها از طریق اتصالات HTTP منتقل می‌شوند.

```
var eventSrc = new EventSource("url");
// register event handler for the message
eventSrc.addEventListener( "message",function (evt) {
    //process the data
});
```

این روش نیز بسیار شبیه به حالت long polling است. سرور تا زمانیکه اطلاعاتی را برای پاسخ دهی فراهم نداشته باشد، اتصال را باز نگه می‌دارد. به این ترتیب از لحاظ مقیاس پذیری گزینه بهتری است (نسبت به حالتیکه مدام اتصال برقرار و قطع می‌شود). اکثر مرورگرها منهای نگارش‌های قدیمی IE از این روش پشتیبانی می‌کنند. تنها تفاوت آن با حالت long polling در این است که پس از ارائه پاسخ به کلاینت، اتصال را قطع نمی‌کند. Long polling نیز

اتصال را باز نگه می‌دارد، اما این اتصال را بلافاصله پس از ارائه پاسخ، می‌بندد.

Web sockets (5)

Web sockets در سکوی کاری ویندوز، تنها در ویندوزهای 8، ویندوز سرور 2012 و دات نت 4 و نیم پشتیبانی می‌شود. هرچند این روش در حال حاضر به عنوان بهترین روش Push مطرح است اما به دلیل محدودیتی که یاد شد، مدتی طول خواهد کشید تا استفاده گسترده‌ای پیدا کند.

```
var socket = new WebSocket("url");
socket.onmessage = function (msg) {
    var newInfo = msg.data;
    // do something with the data
}
// client can also send request to server
socket.send(.... )
```

با این اوصاف آیا راه حل بهتر و میانه‌تری وجود دارد؟
بلی. اگر به وضعیت فعلی سکوی کاری ASP.NET نگاه کنیم:

ASP.NET platform

MVC

Web
Pages

Web
Forms

Web
API

Signal

Sites

Services

ASP.NET

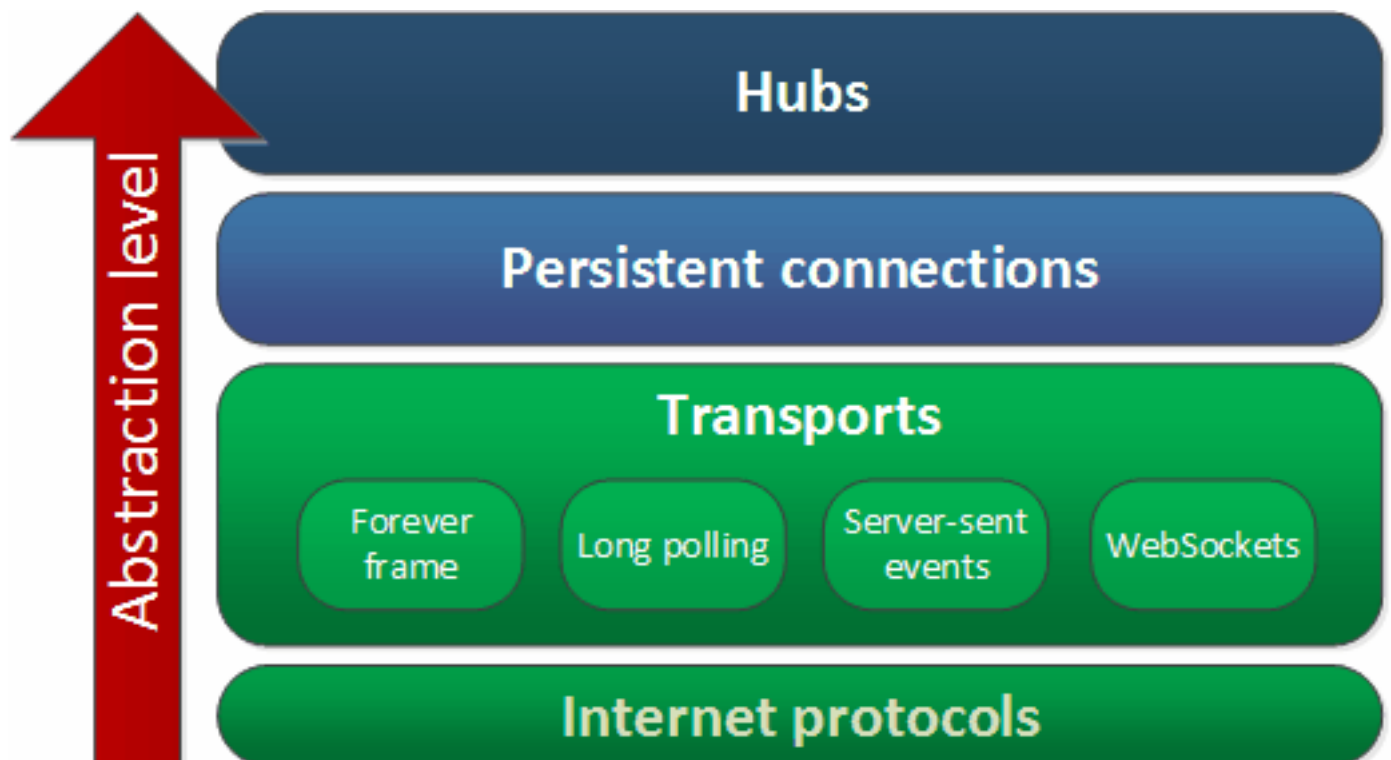
SignalR را می‌توان مشاهده کرد که در گروه ساخت سرویس‌های آن قرار گرفته است. همانطور که ملاحظه می‌کنید، این سرویس جدید آنچنان وابستگی به سایر اجزای آن نداشته و می‌تواند خارج از ASP.NET نیز مورد استفاده قرار گیرد.

SignalR چیست؟

SignalR راه حلی است سمت سرور برای نوشتن push services. همچنین به همراه کتابخانه‌های سمت کاربری است که ارتباطات push services را در انواع و اقسام سکوها‌ی کاری میسر می‌سازد. SignalR [سورس باز بوده](#) و برای اعمال غیرهمزمان (asynchronous) بهینه سازی شده است.

SignalR بر اساس مدل ذهنی اتصالات ماندگار (persistent connections) طراحی شده است. اتصالات ماندگار را باید به عنوان اتصالاتی سریع و غیرطولانی در نظر گرفت. در اینجا Signal یک اتصال است که اطلاعاتی به آن ارسال می‌گردد و هدف، انتقال قطعات کوچکی از اطلاعات است و هدف، ارسال حجم عظیمی از اطلاعات نیست. برای مثال اطلاع رسانی سریعی صورت گیرد که تغییری رخ داده است و سپس ادامه کار و دریافت اطلاعات واقعی توسط فرآیندهای متداول مثلاً HTTP GET انجام شود. البته باید دقت داشت SignalR نیز نهایتاً از یکی از 5 روش push بررسی شده در این قسمت استفاده می‌کند. اما بر اساس توانایی‌های کلاینت و سرور، به صورت هوشمند بهترین و بهینه‌ترین انتخاب را به کاربر ارائه می‌دهد.

اتصالات ماندگار قسمت سطح پایین SignalR را تشکیل می‌دهند. سطح بالاتر آن که این مفاهیم را به شکلی کپسوله شده ارائه می‌دهد، Hubs نام دارد که پایه اصلی دوره جاری را تشکیل خواهد داد.



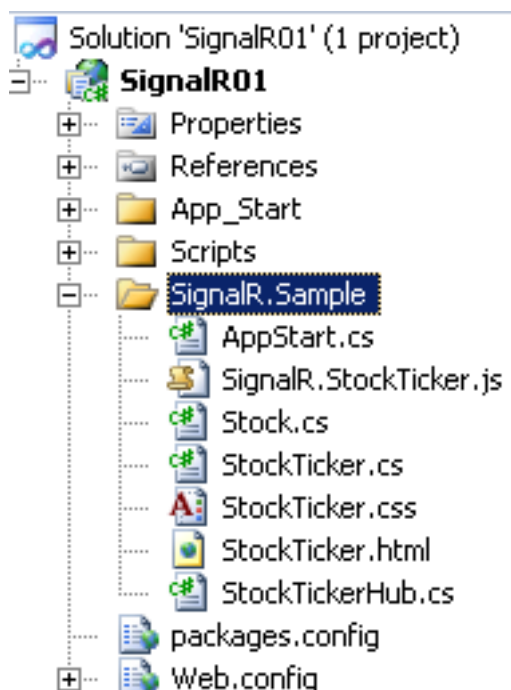
همانطور که عنوان شد، SignalR سورس باز بوده و دارای مخزن کدی عمومی در [GitHub](#) است. همچنین بسته‌های تشکیل دهنده‌ی آن از طریق NuGet نیز قابل دریافت هستند. این بسته‌ها شامل [هسته SignalR](#) و کلاینت‌های آن مانند کلاینت‌های WinRT، سیلورلایت، jQuery، ویندوز فون 8 و امثال آن هستند.

شروع کار با SignalR

تیم SignalR [مثالی مقدماتی](#) از نحوه کار با SignalR را به صورت یک بسته NuGet ارائه داده‌اند که از طریق آدرس و فرمان ذیل قابل دریافت است:

```
PM> Install-Package Microsoft.AspNet.SignalR.Sample
```

قبل از اینکه این مثال را دریافت کنید نیاز است ابتدا یک برنامه ASP.NET جدید را آغاز نمایید (تفاوتی نمی‌کند که MVC باشد یا Web forms). سپس دستور فوق را فراخوانی کنید.



پس از دریافت مثال، یکبار پروژه را کامپایل کرده و سپس بر روی فایل StockTicker.html آن کلیک راست نموده و گزینه مشاهده در مرورگر را انتخاب کنید. همچنین برای اینکه این مثال را بهتر مشاهده کنید، بهتر است دو وهله از مرورگر را باز کرده و آدرس باز شده را در آن بررسی کنید تا اعمال تغییرات همزمان به کلاینت‌های متفاوت را بهتر بتوان بررسی و مشاهده کرد.

ASP.NET SignalR Stock Ticker Sample

Open Market

Close Market

Reset

Live Stock Table

Symbol	Price	Open	High	Low	Change	%
GOOG	570.80	570.3	570.84	569.68	▲ 0.5	0.09%
MSFT	30.39	30.31	30.39	30.31	▲ 0.08	0.26%
APPL	576.81	578.18	578.18	576.81	▼ -1.37	-0.24%

Live Stock Ticker

OG 570.80 ▲ 0.5 (0.09%)
 MSFT 30.39 ▲ 0.08 (0.26%)
 APPL 57

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱:۲

مهندس مثل همیشه عالی بود.
میشه بفرمایین تفاوتش با برنامه نویسی سوکت چی می‌تونه باشه برای برنامه‌های چت؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۲ ۸:۴۸

در اینجا

- در سمت کلاینت فایروال مانعی نخواهد بود چون ارتباطات از طریق مرورگر (هم می‌تواند) انجام می‌شود.
- باز هم نهایتاً از سوکت‌ها استفاده خواهد شد اما در سطحی بالاتر و بدون درگیری با جزئیات آن‌ها. اینبار یک فریم ورک آماده، تست شده و تهیه شده بفرز سوکت‌های دات نت و ویندوز در اختیار شما است. به علاوه این فریم ورک فراتر است از صرفاً برقراری ارتباط و ارسال داده، بلکه حالت امکان اجرای متدهای خاصی در سمت کلاینت یا سرور را هم دارا است (بحث قسمت بعد).
- تنوع کلاینت‌ها. محدود به یک برنامه ویندوزی نخواهید بود. مثلاً امکان استفاده از یک کلاینت jQuery، که برای اجرا، نیازی به سطح دسترسی خاصی ندارد، یا حتی یک کلاینت سیلورلایت یا اندروید و غیره هم برای آن تهیه کرده‌اند.
- امکان استفاده از IIS به عنوان سرور. همین مساله یعنی درگیر نشدن با مسایلی مانند مقیاس پذیری، مدیریت تعداد کانکشن‌های بالا و امثال آن.
- امکان یکپارچه کردن یک برنامه سرویس دهنده هاب با یک برنامه ASP.NET در کنار هم در یک پروژه.

و ...

نویسنده: Alireza Godazchian
تاریخ: ۱۳۹۲/۰۱/۱۳ ۱۰:۵۳

خیلی عالی توضیح می‌دهید
از شما واقعا سپاسگذاریم....

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۲/۰۱/۱۳ ۱۶:۳۵

سلام آقای نصیری حرف R در SignalR به چه معناست

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۳ ۱۶:۴۳

real time یا بلادرنگ

نویسنده: مرتضی
تاریخ: ۱۳۹۲/۰۱/۱۳ ۲۳:۰۰

سلام-

بخاطر آموزشاتون واقعا سپاسگذارم-

لطفاً همیشه یه مثال با web sockets بذارید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۳ ۲۳:۲۷

لازم هست تمام قسمت‌ها را یکبار مطالعه کنید. در قسمت دوم (نگاهی به SignalR Hubs) در مورد نحوه انتخاب لایه transport به صورت خودکار بحث شده. در قسمت سوم (نگاهی به SignalR Clients) در طی یک نکته ویژه عنوان شده که همین مثال مورد بحث رو به چه صورتی و تنها در کجا می‌تونید بر اساس webSocket اجرا کنید.

نویسنده: مرتضی
تاریخ: ۱۳۹۲/۰۱/۱۴ ۰:۱

مهندس کامل مطالعش کردم-

منظورم یه سرفصل جدا با web sockets بود -

چون با web sokcets که کار کردم به یه مشکل کوچیک برخوردم-

مرسی

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۴ ۰:۱۷

- نیازی به سرفصل جدا ندارد با توجه به خودکار بودن انتخاب لایه انتقال بر اساس توانایی سکوی کاری مورد استفاده (در حین کار با SignalR، وب سوکت فقط در ویندوز 8، IIS8 به همراه پروژه‌ای مبتنی بر دات نت 4 و نیم پشتیبانی می‌شود). سایر بحث‌ها و نکات یکی است و تفاوتی نمی‌کند. زمانیکه با Hub کار می‌کنید در لایه‌ای قرار دارید که این جزئیات از شما مخفی می‌شود و کار انتخاب خودکار است (تصویر abstraction level مطلب جاری).
+ دوره‌ها در سایت قسمتی دارند جهت [پرسش و پاسخ](#) اختصاصی که می‌شود مشکلات و سؤالات مرتبط به دوره را در آنجا ارسال کرد با توضیح بیشتر.

نویسنده: اردلان شاه قلی
تاریخ: ۱۳۹۲/۱۰/۲۹ ۱۸:۱۳

سلام من بعد از اضافه کردن سمپل نام برده و مشاهده‌ی صفحه ی StockTicker.html ، آنچه را شما در بالا نمایش داده اید نمی‌بینم.؟ تصویر چیزی که من می‌بینم به این صورت می‌باشد.

ASP.NET SignalR Stock Ticker Sample

Open Market

Close Market

Reset

Live Stock Table

Symbol	Price	Open	High	Low	Change	%
loading...						

Live Stock Ticker

loading...

نویسنده: وحید نصیری

تاریخ: ۱۸:۳۷ ۱۳۹۲/۱۰/۲۹

به نظر اسکریپت‌های آن بارگذاری نشده‌اند. در کروم روی دکمه F12 کلیک کنید تا کنسول آن ظاهر شود. بعد بررسی کنید آیا خطایی در برگه network آن گزارش شده یا حتی در کنسول لاگ‌های آن که خطاهای جاوا اسکریپتی را نمایش می‌دهد. [با فایرباگ هم می‌شود](#) این نوع برنامه‌ها را دیباگ کرد.

اطلاعات بیشتر: « [عیب‌یابی و دیباگ برنامه‌های SignalR](#) »همچنین این مثال‌ها را از اینجا نیز می‌توانید دریافت کنید: [SignalRSamples.zip](#)

نویسنده: vici

تاریخ: ۲۱:۱۰ ۱۳۹۲/۱۰/۳۰

سلام

آقای نصیری برای کارهای ویندوزی مثل datagrid چه روشی رو پیشنهاد می‌کنید؟

نویسنده: وحید نصیری

تاریخ: ۲۱:۱۷ ۱۳۹۲/۱۰/۳۰

در چهارمین قسمت این سری « [نگاهی به گزینه‌های مختلف مهبای جهت میزبانی SignalR](#) » شده‌است.

نویسنده: ح مراداف

تاریخ: ۱۸:۳۲ ۱۳۹۲/۱۱/۱۹

فوق العاده بود.

چند وقت پیش برای کار مصاحبه دادم و توی اون مصاحبه مطالعه در زمینه SignalR و angularJS بهم پیشنهاد شد. خیلی وقت بود فرصت نمی‌کردم توی نت درباره این مورد سرچ کنم (مطلب انگلیسی خوشم نمیاد).

تا اینکه به دنبال سرچ درباره MVVM به این سایت عالی برخورد کردم.

واقعا کارتون یکه

تازه فهمیدم که اطلاعاتم خیلی قدیمیه و باید سریع بروزش کنم.

با تشکر از مقاله عالیتون.

خدا قوت

یا حق

نویسنده: ساده

تاریخ: ۱۴:۴۲ ۱۳۹۳/۰۳/۱۱

سلام

GET http://localhost:7186/signalr/hubs 500 (Internal Server Error)

نشان می‌ده.

من دایرکتوری پروژه رو چک کردم

اما اصلا چنین شاخه ای نیست تو پروژه که!

نویسنده: وحید نصیری

تاریخ: ۱۴:۴۹ ۱۳۹۳/۰۳/۱۱

- در مورد مسیر پویای signalr/hubs [در قسمت بعدی](#) بیشتر بحث شده‌است.

- برای خطایابی نیاز به توضیحات بیشتری هست. [اطلاعات بیشتر](#)

- در کنسول لاگ‌های آن خطاهای جاوا اسکریپتی را نمایش می‌دهد. [با فایرباگ هم می‌شود](#) این نوع برنامه‌ها را دیباگ کرد.

اطلاعات بیشتر: « [عیب یابی و دیباگ برنامه‌های SignalR](#) »

نویسنده: وحید نصیری

تاریخ: ۱۳:۳۳ ۱۳۹۳/۱۱/۱۳

جهت اطلاع

کلیه مثال‌های این دوره برای استفاده از SignalR 2.x به روز شده و در مخزن کد ذیل قرار گرفتند:

[SignalR-Samples](#)