عنوان: **SignalR**
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۳/۲۸ ۲۲:۱۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, SignalR

چند وقتی هست که در کنار بدنه اصلی دات‌نت فریم‌ورک چندین کتابخونه به صورت متن‌باز در حال توسعه هستند. این مورد در ASP.NET بیشتر فعاله و مثلا دو کتابخونه **SignalR** و **WebApi** توسط خود مایکروسافت توسعه داده میشه.
SignalR همونطور که در سایت بسیار خلاصه و مفید یک صفحه‌ای! خودش توضیح داده شده (^) یک کتابخونه برای توسعه برنامه‌های وب «زمان واقعی»! (**real-time web**) است:

```
Async library for .NET to help build real-time, multi-user interactive web applications.
```

برنامه‌های زمان واقعی به صورت خلاصه و ساده به‌صورت زیر تعریف میشن (^):

```
The real-time web is a set of technologies and practices that enable users to receive information as soon as it is published by its authors, rather than requiring that they or their software check a source periodically for updates.
```

یعنی کاربر سیستم ما بدون نیاز به ارسال درخواستی صریح! برای دریافت آخرین اطلاعات به روز شده در سرور، در برنامه کلاینتش از این تغییرات آگاه بشه. مثلا برنامه‌هایی که برای نمایش نمودارهای آماری داده‌ها استفاده میشه (بورس، قیمت ارز و طلا و ...) و یا مهمترین مثالش میتونه برنامه «چت» باشه. متاسفانه پروتوکل HTTP مورد استفاده در وب محدودیت‌هایی برای پیاده‌سازی این گونه برنامه‌ها داره. روش‌های گوناگونی برای پیاده‌سازی برنامه‌های زمان واقعی در وب وجود داره که کتابخونه SignalR فعلا از موارد زیر استفاده میکنه:
تکنولوژی جدید **WebSocket** (^) که خوشبختانه پشتیبانی کاملی از اون در دات نت 4.5 (چهار نقطه پنج! نه چهار و نیم!) وجود داره. اما تمام مرورگرها و تمام وب سرورها از این تکنولوژی پشتیبانی نمیکنند و تنها برخی نسخه‌های جدید قابلیت استفاده از آخرین ورژن WebSocket رو دارند که میشه به کروم 16 به بالا و فایرفاکس 11 به بالا و اینترنت اکسپلورر 10 اشاره کرد (برای استفاده از این تکنولوژی در ویندوز نیاز به **IIS 8.0** است که متاسفانه فقط در ویندوز 8.0 موجوده):

Chrome 16, Firefox 11 and Internet Explorer 10 are currently the only browsers supporting the latest specification (RFC 6455).

یه روش دیگه **Server-sent Events** نام داره که داده‌های جدید رو به فرم رویدادهای DOM به سمت کلاینت میفرسته(^).

روش دیگه‌ای که موجوده به **Forever Frame** معروفه که در این روش یک iframe مخفی درون کد html مسئول تبادل داده‌هاست. این iframe مخفی به‌صورت یک بلاک Chunked (^) به سمت کلاینت فرستاده میشه. این iframe که مسئول رندر داده‌های جدید در سمت کلاینت هست ارتباط خودش رو با سرور تا ابد! (برای همین بهش forever میگن) حفظ میکنه. هر وقت رویدادی سمت سرور رخ میده با استفاده از این روش داده‌ها به‌صورت تگ‌های script به این فریم مخفی فرستاده می‌شوند و چون مرورگرها محتوای html رو به صورت افزایشی (incrementally) رندر میکنن بنابراین این اسکریپتها به‌ترتیب زمان دریافت اجرا می‌شوند. (البته ظاهرا عبارت forever frame در صنعت عکاسی! معروف‌تره بنابراین در جستجو در زمینه این روش ممکنه کمی مشکل داشته باشین) (^).
روش آخر که در کتابخونه SignalR ازش استفاده میشه **long-polling** نام داره. در روش polling معمولی پس از ارسال درخواست توسط کلاینت، سرور بلافاصله نتیجه حاصله رو به سمت کلاینت میفرسته و ارتباط قطع میشه . بنابراین برای داده‌های جدید درخواست جدیدی باید به سمت سرور فرستاده بشه که تکرار این روش باعث افزایش شدید بار بر روی سرور و کاهش کارآمدی اون می‌شه. اما در روش long-polling پس از برقراری ارتباط کلاینت با سرور این ارتباط تا مدت زمان معینی (که توسط یه مقدار تایم اوت مشخص میشه و مقدار پیش‌فرضش 2 دقیقه است) برقرار میمونه . بنابراین کلاینت میتونه بدون ایجاد مشکلی در کارایی، داده‌های جدید رو از سرور دریافت کنه. به این روش در برنامه‌نویسی وب اصطلاحا برنامه‌نویسی کامت (**Comet Programming**) میگن (^ ^).

(البته روش‌های دیگری هم برای پیاده‌سازی برنامه‌های زمان اجرا وجود داره مثل کتابخونه **node.js** که جستجوی بیشتر به خوانندگان واگذار میشه)

SignalR برای برقراری ارتباط ابتدا بررسی میکنه که آیا هر دو سمت سرور و کلاینت قابلیت پشتیبانی از WebSocket رو دارند. در غیراینصورت سراغ روش Server-sent Events میره. اگر باز هم موفق نشد سعی به برقراری ارتباط با روش forever frame میکنه و اگر باز هم موفق نشد در آخر سراغ long-polling میره.
با استفاده از SignalR شما میتونین از سرور، متدهایی رو در سمت کلاینت فراخونی کنین. یعنی درواقع با استفاده از کدهای سی شارپ میشه متدهای جاوااسکریپت سمت کلاینت رو صدا زد!
بطور خلاصه در این کتابخونه دو کلاس پایه وجود داره:
کلاس سطح پایین **PersistentConnection**

کلاس سطح بالای **Hub**

علت این نامگذاری به این دلیله که کلاس سطح پایین پیاده‌سازی پیچیده‌تر و تنظیمات بیشتری نیاز داره اما امکانات بیشتری هم در اختیار برنامه‌نویس قرار می‌ده.
خوب پس از این مقدمه نسبتا طولانی برای دیدن یک مثال ساده میتونین با استفاده از نوگت (Nuget) مثال زیر رو نصب و اجرا کنین (اگه تا حالا از نوگت استفاده نکردین قویا پیشنهاد میکنم که کار رو با دریافتش از اینجا آغاز کنین) :

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

پس از کامل شدن نصب این مثال اون رو اجرا کنین. این یک مثال فرضی ساده از برنامه نمایش ارزش آنلاین سهام برخی شرکتهاست. میتونین این برنامه رو همزمان در چند مرورگر اجرا کنین و نتیجه رو مشاهده کنین.
حالا میریم سراغ یک مثال ساده. میخوایم یک برنامه چت ساده بنویسیم. ابتدا یک برنامه وب اپلیکیشن خالی رو ایجاد کرده و با استفاده از دستور زیر در خط فرمان نوگت، کتابخونه SignalR رو نصب کنین:

```
PM> Install-Package SignalR
```

پس از کامل شدن نصب این کتابخونه، ریفرنس‌های زیر به برنامه اضافه میشن:

```
Microsoft.Web.Infrastructure
Newtonsoft.Json
SignalR
SignalR.Hosting.AspNet
SignalR.Hosting.Common
```

برای کسب اطلاعات مختصر و مفید از تمام اجزای این کتابخونه به اینجا مراجعه کنین.
همچنین اسکریپت‌های زیر به پوشه Scripts اضافه میشن (این نسخه‌ها مربوط به زمان نگارش این مطلب است):

```
jquery-1.6.4.js
jquery.signalR-0.5.1.js
```

بعد یک کلاس با نام SimpleChat به برنامه اضافه و محتوای زیر رو در اون وارد کنین:

```
using SignalR.Hubs;
namespace SimpleChatWithSignalR
{
  public class SimpleChat : Hub
  {
    public void SendMessage(string message)
    {
      Clients.reciveMessage(message);
    }
```

```
  }
}
```

دقت کنین که این کلاس از کلاس Hub مشتق شده و همچنین خاصیت **Clients** از نوع **dynamic** است. (در مورد جزئیات این کتابخونه در قسمت‌های بعدی توضیحات مفصل‌تری داده میشه)
سپس یک فرم به برنامه اضافه کرده و محتوای زیر رو در اون اضافه کنین:

```
<input type="text" id="msg" />
<input type="button" value="Send" id="send" /><br />
<textarea id='messages' readonly="true" style="height: 200px; width: 200px;"></textarea>
<script src="Scripts/jquery-1.6.4.min.js" type="text/javascript"></script>
<script src="Scripts/jquery.signalR-0.5.1.min.js" type="text/javascript"></script>
<script src="signalr/hubs" type="text/javascript"></script>
<script type="text/javascript">
  var chat = $.connection.simpleChat;
  chat.reciveMessage = function (msg) {
   $('#messages').val($('#messages').val() + "-" + msg + "\r\n");
  };
  $.connection.hub.start();
  $('#send').click(function () {
    chat.sendMessage($('#msg').val());
  });
</script>
```

همونطور که میبینین برنامه چت ما آماده شد! حالا برنامه رو اجرا کنین و با استفاده از دو مرورگر مختلف نتیجه رو مشاهده کنین.
نکته کلیدی کار SignalR در خط زیر نهفته است:

```
<script src="signalr/hubs" type="text/javascript"></script>
```

اگر محتوای آدرس فوق رو دریافت کنین می‌بینین که موتور این کتابخانه تمامی متدهای موردنیاز در سمت کلاینت رو با استفاده از کدهای جاوااسکریپت تولید کرده. البته در این کد تولیدی از نامگذاری camel Casing استفاده میشه، بنابراین متد SendMessage در سمت سرور به‌صورت sendMessage در سمت کلاینت در دسترسه.
امیدوارم تا اینجا تونسته باشم علاقه شما به استفاده از این کتابخونه رو جلب کرده باشم. در قسمت‌های بعد موارد پیشرفته‌تر این کتابخونه معرفی میشه.
اگه علاقه‌مند باشین میتونین از این ویکی اطلاعات بیشتری بدست بیارین.

به روز رسانی

در دوره‌ای به نام SignalR در سایت ، به روز شده‌ای این مباحث را می‌توانید مطالعه کنید.

نظرات خوانندگان

نویسنده: روح الله
تاریخ: ۱۳۹۱/۰۳/۲۹ ۰:۸

جالب بود. اتفاقا امروز با دوستان صحبت از همچین نیازمندی بود. ممنون از زحمات شما.

نویسنده: افشار محبی
تاریخ: ۱۳۹۱/۰۳/۲۹ ۹:۱۷

به نظر فریمورک جذاب و مفیدی میاد. همینجا به آقای نصیری و همکارانش به خاطر راه اندازی وبلاگ جدید تبریک می‌گویم.

نویسنده: علیرضا جهانشاهلو
تاریخ: ۱۳۹۱/۰۳/۲۹ ۹:۳۹

خیلی عالی بود! اگه بدونید این کتابخونه پایه اساس خیلی از ایده‌های تجاریه!؟ که من در حال حاضر مشغول به کار بر روی یکی از همین ایده‌ها هستم.

نویسنده: احمد
تاریخ: ۱۳۹۱/۰۳/۲۹ ۱۹:۵۶

سلام و تبریک برای سایت جدید
از اینکه مطلب بسیار مفید و خوبی گذاشتید بسیار سپاسگزارم.اما اسکریپت signalr/hubs کجا قرار داره؟

نویسنده: Mona
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۰:۳۵

با سلام خدمت دوستان عزیز
ممنون از مطالب دقیق و کاربردی شما
لطفاً اگر امکان دارد مقایسه ای بین SignalR و Nod.js انجام دهید، زیرا در هرکجای وب که قدم میگذارید بحث از Nod.js است.
ممنون

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۴۶

این مسیر درواقع آدرس پیش‌فرض signalr برای کلاس پایه Hub هست که البته میتونه عوض هم بشه. این آدرس درواقع به یه هندلر مپ شده و این هندلر به محض دریافت یه درخواست از سمت کلاینت شروع به گشتن کل اسمبلی برای یافتن کلاسهای مشتق شده از Hub میگرده و برای تمام اجزای public اونا جهت در دسترس بودن در سمت کلاینت متدهای جاوااسکریپت متناظر تولید کرده و در متن پاسخ ارسالی مینویسه (البته یکسری کد دیگه هم برای کار با کلاس Hub تولید میکنه).
برای دریافت این اسکریپت تولیدی، شما میتونین این مسیر رو در نوار آدرس مرورگرتون وارد کنید و کد جاوااسکریپت تولیدی رو مشاهده کنین. حتی میتونین این کد تولیدشده رو در یک فایل ذخیره کرده و به جای ریفرنس اصلی استفاده کنین. البته به شرطی که در اجزای پابلیک کلاسهاتون بعدا تغییری ایجاد نشه.

نویسنده: ramin
تاریخ: ۱۳۹۱/۰۳/۳۰ ۱۰:۵۹

سلام
یعنی مسیر و محتوای signalr/hubs در هنگام اجرا تولید میشه؟
من مثال شما رو دنبال کردم ولی خروجی مورد نظر رو نگرفتم

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۳/۳۰ ۱۱:۱۰

من خودم تخصصی در زمینه node.js ندارم. البته ازش خوشم هم نمیاد :)
اینجوری که خودشون میگن این یه پلتفرمه که از Chrome's JavaScript runtime استفاده میکنه و شونصدتا ویژگی دیگه هم داره! برای استفاده ازش باید برنامشو رو که بهش node میگن دریافت و نصب کنین (حدود 3 مگابایت نسخه 0.6.19). بعد کد سمت سرور کاملا با استفاده از زبان جاوااسکریپت نوشته میشه (عیب) بنابراین کاملا cross platform هستش (مزیت). یعنی با استفاده از جاوااسکریپت میتونین مثلا یه وب سرور بسیار سبک (مزیت) ایجاد کنین که به یکسری درخواستا پاسخ میده. حالا یکسری اومدن برای این پلتفرم ماژولهایی نوشتن مثل socket.io و nowJS که برقراری ارتباطی راحت و آسان با سرور و تبادل داده‌ها رو به عهده میگیره (مزیت).
در مقابل SignalR یه کتابخونه غیرهمزمان (Async) هستش که برا دات نت نوشته شده (مزیت)، مخصوص وب زمان واقعی چندکاربره و شونصدتا ویژگی دیگه هم نداره (مزیت) :). بنابراین سمت سرور از مزایای بیشماری برخورداره، مثل یک IDE قدرتمند (VS) (مزیت2x) و یک کتابخونه کامل (.NetFx) (مزیت2x)
درضمن یه فریمورک دیگه هم برای ASP.NET وجود داره: PokeIn که نسخه تجاری (پولی) هم داره و از WebSocket استفاده میکنه. جایی خوندم که کاراییش خوبه در حد 10 تا 50 هزار کلاینت همزمان(^).

برای SignalR یه اپلیکیشن توسط خود نویسندگانش نوشته شده (SignalR-Crank) که برای آزمایش بار روی سرور استفاده میشه. طبق گفته خودشون تا 100k (صد هزار!) کلاینت رو تونستن بدون هیچ مشکلی راه بندازن (البته با مصرف 5 گیگ رم). نمیدونم این تست چی بوده ولی این عدد خیلی زیاده.

در مورد بحث کارآیی خودم دارم یه تستایی انجام میدم. نمیدونم بتونم آزمایش مشابهی رو بقیه فریمورکها انجام بدم یا نه. اگه نتیجه‌ای حاصل شد اینجا میزارم.

با تشکر

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۳/۳۰ ۱۱:۱۴

اگه برنامه چت داره درست اجرا میشه، مسیر مورد نظر برای دریافت این کد جاوااسکریپت مثلا رو سیستم من اینه:
http://localhost:16869/signalr/hubs

آره در زمان اجرا تولید میشه و بار کوچیکی روی سرور میزاره. برا همین پیشنهاد میشه تو نسخه ریلیز برنامه‌ها این کد تولیدی تو یه فایل ذخیره بشه و به جای اون مسیر ریفرنس داده بشه.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۷/۲۹ ۱:۲

سلام
ممنونم بابت مطلب خوب کاربردیتون
می تونم بگم یکی از سئوال‌های ذهن من پاسخ داده شد و این امکان راه حل مشکل چند هفته ای من بود
تشکر

نویسنده: hossein
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۲:۲۶

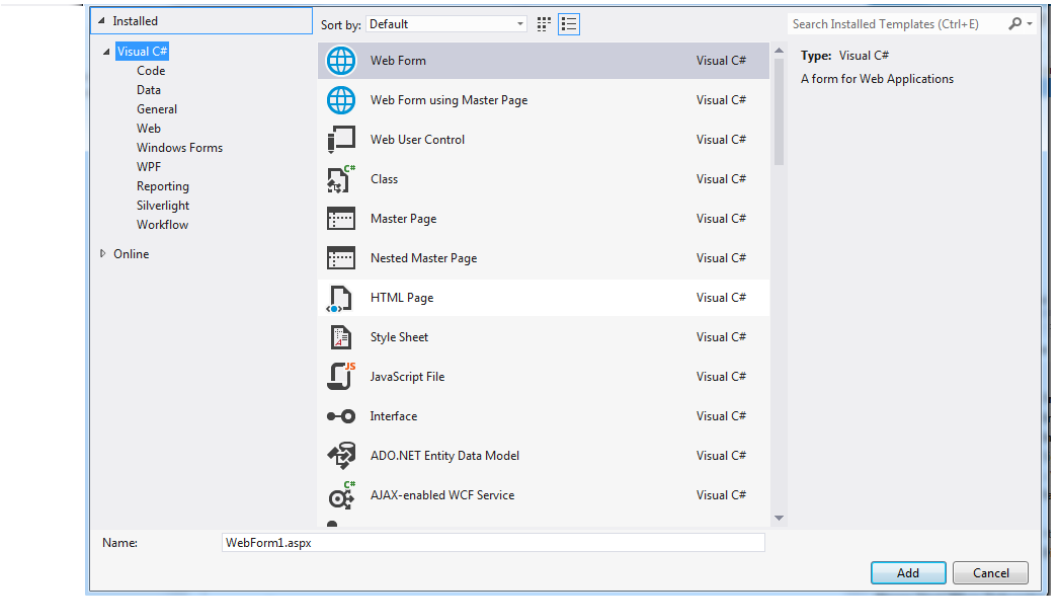
با سلام میخواستم بپرسم ایا این روش (signallr) برای لود کردن مثلا 40-50 تا عکس اسلاید شو کارایی بالاتری داره یا روش lazy loading پلاگینی برای jquery اون وجود داره
با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۲:۳۳

سؤالت ربطی به بحث نداره. کار SiganlR به زبان ساده فرستادن پیغام از طرف سرور به کلاینت‌ها هست. مثلا 2 تا ایمیل جدید داری. رکورد جدیدی به صفحه اضافه شده. یا برای نمونه درصد پیشرفت عملیات سمت سرور مثلا 50 درصد بوده و ارسالش به تمام کلاینت‌های متصل. هدف تبادل اطلاعات همزمان و بلادرنگ و بدون معطلی هست؛ خصوصا از طرف سرور به کلاینت.

نویسنده: ابوالفضل روشن ضمیر
تاریخ: ۱۳۹۲/۰۵/۳۰ ۱۶:۲۵

سلام
من پکیج Jquery و SignalR را در برنامه نصب کردم ... ولی وقتی روی نام پروژه راست کلیک می‌کنم تا کلاس SignalR Hub Class
را اضافه کنم وجود نداره ....
ممنون



نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۳۰ ۱۷:۵۹

- آپدیت سوم VS 2012 را نصب کنید .
+ هیچ نیازی به این پیشنیازها نیست. با *VS 2010* کل مطالب SignalR قابل پیاده سازی هستند. فقط باید بتونید با نیوگت کار کنید و بسته‌های لازم رو اضافه کنید. بعد از آن کلاس هاب ، یک کلاس ساده است که امضایی مشخص داره و از کلاسی به نام Hub مشتق می‌شود و همچنین ویژگی HubName هم می‌تواند داشته باشد.

نویسنده: mostafa
تاریخ: ۱۳۹۲/۰۹/۳۰ ۲۳:۱۶

سلام
میشه این رو برای website معمولی visual studio استفاده کرد؟ یعنی غیر از وب اپلیکشن
ممنون

نویسنده: bahman
تاریخ: ۱۳۹۲/۱۰/۰۹ ۱۱:۵۴

سلام.
آیا این امکان وجود داره که SingnalR رو با RestFull سرویس در سمت سرور ترکیب کنم و از سرویس در سمت کلاینت ، داخل یک برنامه WPF استفاده کتم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۰۹ ۱۳:۴

نگاهی به گزینه‌های مختلف مهیای جهت میزبانی SignalR

نویسنده: عرفان
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۱:۲۲

(Clients.reciveMessage(message
خطا میده. همچین کلاسی وجود نداره حتی اگه غلط املاییش رو هم درست کنیم!

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۱:۳۶

شما که به غلط املاییش دقت کردی، به انتهای بحث که نوشته شده این مباحث به روز شده‌اش در دوره SignalR سایت ارائه شدند، دقت نکردی؟ اون متد dynamic هست؛ یعنی اصلا نیازی نیست وجود خارجی داشته باشه. فقط کمی در نگارش‌های جدید، Refactoring انجام دادن، بعدش باید مشخص کنی به All یا به گروه خاصی این پیام‌ها ارسال بشه. کلیاتش یکی هست. فقط کمی تعاریف اولیه رو Refactor کردن. در بحث معرفی hubs دوره‌ای که نام برده شد این‌ها هست.