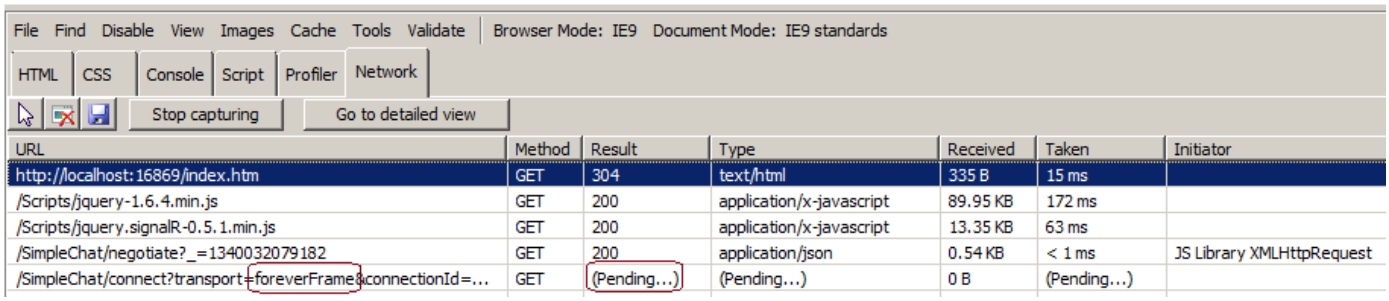
عنوان: **SignalR - قسمت دوم**
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۳۱
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, SignalR

در قسمت اول بحث‌های مقدماتی درباره وب زمان واقعی (real time web) و معرفی کتابخونه SignalR به همراه یک مثال ساده رو با هم دیدیم. در ادامه به جزئیات ریزی از کتابخونه SignalR که توسط آقای David Fowler توسعه داده میشه میپردازم.

همونطور که قبلا هم اشاره شد قلب این کتابخونه در سمت سرور دو کلاس پایه PersistentConnection و Hub هستن که اولی سطحی پایینتر داره یعنی به تنظیمات و کدنویسی (بسیار) بیشتری برای پیاده‌سازی نیاز داره اما در عوض امکانات سطح پایینتری هم در اختیار برنامه نویس قرار میده که در برخی موارد موردنیاز هستن. در مورد بخشهای مختلف این دو کلاس و نحوه پیاده‌سازی هر دو کلاس فوق، تو آدرسی که قبلا اشاره کردم (آدرس پروژه متن باز این کتابخونه در github) راهنمایی‌های نسبتا مفصلی ارائه شده و نیازی به تکرار این مطالب در اینجا نیست. پیشنهاد میکنم که این مطالب رو با دقت مطالعه کنین.

SignalR به صورت توکار از 4 روشی که در قسمت قبل به اون اشاره شد برای برقراری ارتباط استفاده میکنه. WebSocket که به بسترهای جدیدی نیاز داره. Server-sent Events که تنها در مرورگرهایی که پشتیبانی کاملی از html5 دارند قابل استفاده است (بنابراین ie9 نمیتونه از این روش استفاده کنه). forever frame داده‌ها رو به‌صورت chunked (بخشی از استاندارد http 1.1) دریافت میکنه و روش آخر که Long-polling نام داره از هموش روش قدیمی اِیجکس (Ajax) بهره میبره و با استفاده از آبجکت معروف XmlHttpRequest کار ارتباط و تبادل داده‌ها رو انجام میده.

در اینجا برای بررسی این ارتباطات ابتدا برنامه چت ساده قسمت قبل رو در اینترنت اکسپلورر اجرا کنین و با استفاده developer tool (کلید F12) به درخواستهای مختلف ارسال شده به سرور نگاه کنین. پس از اجرای برنامه و قبل ارسال هرگونه داده به سرور در تب Network این ابزار چیزی شبیه به شکل زیر مشاهده خواهید کرد (البته باید قبل از ورود به صفحه برنامه چت روی دکمه Start capturing کلیک کنین):



با توجه به تصویر بالا SignalR در شرایط موجود بهترین روش برای برقراری ارتباط با سرور رو forever frame تشخیص داده و مشاهده میشه که این ارتباط دائمیه و فعلا نتیجه‌ای از سمت سرور دریافت نکرده و ارتباط کاملا زنده است. البته اگر در این ابزار درباره درخواستهای ارسالی به سرور بیشتر جستجو بکنین اطلاعات بیشتری نصیبتون میشه که آوردنش اینجا بحث رو طولانی میکنه.

حالا برنامه رو در یه مرورگر دیگه که از html5 پشتیبانی میکنه اجرا کنین. مثلا نتیجه در گوگل کروم و ابزار توسعه اون به شکل زیره:

index.htm	GET	200 OK	text/html	Other	1.10KB 1.02KB	5ms 5ms
jquery-1.6.4.min.js /Scripts	GET	200 OK	application/x-javascript	index.htm:5 Parser	40.40KB 89.52KB	166ms 166ms
jquery.signalR-0.5.1.min.js /Scripts	GET	200 OK	application/x-javascript	index.htm:5 Parser	5.84KB 12.92KB	81ms 81ms
negotiate /SimpleChat	GET	200 OK	application/json	jquery-1.6.4.min.js:4 Script	557B 147B	12ms 2ms
connect /SimpleChat	GET	200 OK	text/event-stream	jquery.signalR-0.5.1.min.js Script	477B 29B	33.62s 13ms

http://localhost:16869/SimpleChat/connect?transport=serverSentEvents&connectionId=921a3946-b983-4b3a-8d23-96aae5e7f53e

همونطور که میبینین در اینجا روش استفاده شده Server Sent Events هست.

در فایرفاکس هم با استفاده از ابزار محبوب firebug نتیجه مشابه کروم بدست میاد:

Console HTML CSS Script DOM Net

Clear Persist All HTML CSS JS XHR Images Flash Media

URL	Status	Domain	Size	Remote IP	Timeline
Net panel activated. Any requests while the net panel is inactive are not shown.					
GET index.htm	200 OK	localhost:16869	658 B	::1:16869	45ms
GET jquery-1.6.4.min.js	200 OK	localhost:16869	39.9 KB	::1:16869	
GET jquery.signalR-0.5.1.min.js	200 OK	localhost:16869	5.3 KB	::1:16869	
GET negotiate?_=1340033212593	200 OK	localhost:16869	147 B	::1:16869	
http://localhost:16869/SimpleChat/connect?transport=serverSentEvents&connectionId=4d8b307a-a94f-44a2-a471-5685ed7e7eff					
5 requests			46 KB		

البته اگر علاقه زیادی به کندوکاو در جزئیات این درخواستها دارین (مثل خود من) چیزی بهتر از fiddler2 پیدا نمیشه. میتونین پس از ارسال یک متن دوباره این درخواستها رو مورد بررسی قرار بدین و ببینین که چیجوری کانالهای ارتباط پس از ارسال و دریافت دیتا قطع و برقرار میشه.

این نکته رو هم باید یادآور بشم که هرچند که این کتابخونه بهترین روش رو میتونه انتخاب کنه اما به برنامه نویس امکان تعیین صریح روش ارتباط رو هم میده. اگر به راهنماهای این کتابخونه سر بزنین میبینین که امکانات زیادی بهش اضافه شده و امکانات زیادی هم در آینده به اون اضافه میشه. امکاناتی از قبیل ارسال داده‌ها به یک کلاینت خاص و یا به گروهی خاص از کلاینتها، خصوصی‌سازی آدرس سرور و همچنین پشتیبانی از Cross Domain در آخرین نسخه، امکان استفاده از Reactive Extension (بلاگ)، بحث Self Hosting که امکان خیلی جالبیه و میتونه خیلی جاها یه عنوان یک راه‌حل سبک و سریع به کار بیاد، قابلیت فوق العاده در بایندینگ داده‌ها در سمت سرور و مخصوصا کلاینت، امکان تشخیص برقراری یا قطع ارتباط کلاینتها در سمت سرور، استفاده از امکانات این کتابخونه برای برقراری ارتباط با کلاینتها در خارج از فضای کلاسهای مشتق شده از دو کلاس پایه (Hub و PersistentConnection) و چند مورد دیگه تا نسخه جاری اضافه شدند.

درحال حاضر دارم روی یه برنامه چت با امکانات بیشتر کار میکنم که پس از آماده شدن ارائه میدمش. یکی از پروژه‌های متن بازی که با استفاده از این کتابخونه توسعه داده شده jabbr.net است. یه اتاق گفتگوی کامل با امکانات جالبه که میتونین به اون هم یه سری بزنین.

در آخر هم یه لینک جالب برای مطالعه معرفی میکنم: Highest voted Signalr Questions - stackoverflow