SIGNALR

مترجم: سید احمد حسینی

asm_3010@yahoo.com

webography.ir

به نام خدا

مقدمه

نرم افزارهای زمان حقیقی از جمله چت روم ها، بازی های آنلاین، نرم افزارهای نظارتی، نرم افزارهای همکاری بر خط و ... همه جز برنامه های کاربردی و پر کاربرد تحت وب دنیای امروز هستند.

با توجه به اهمیت برنامه های زمان حقیقی توانایی ایجاد و به کارگیری اینگونه برنامه ها در حال حاضر ضروری به نظر می رسد.از طرفی باید این فعالیت به گونه ایی انجام گردد تا زمان توسعه محصول کاهش یابد به همین دلیل استفاده از کتابخانه ها و کدهای ایجاد شده برای این منظور راه حل منطقی به نظر می رسد.به همین منظور متن جاری که معرفی کتابخانه ایی جهت ایجاد چنین برنامه های در چارچوب .NET است تهیه شده است.

متن اصلی این مقاله با عنوان "Introduction to SignalR" نوشته "Patrick Fletcher" در آدرس زیر موجود می باشد:

http://www.asp.net/signalr/overview/signalr-20/getting-started-with-signalr-20/introduction-to-signalr

سید احمد حسینی

asm_3010@yahoo.com

webography.ir

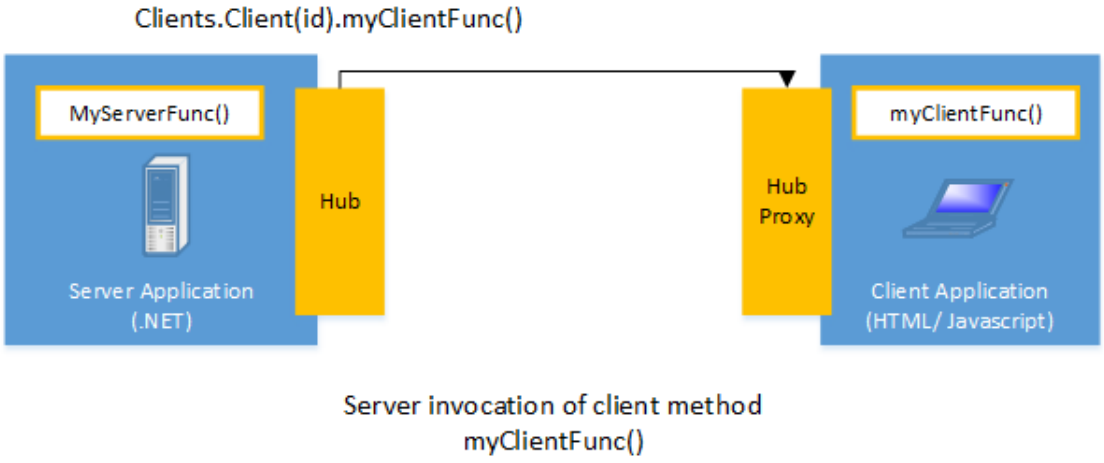
SignalR چیست؟

SignalR یک کتابخانه برای ASP.NET است که فرآیند افزودن قابلیت های زمان حقیقی تحت وب به برنامه را ساده می نماید.منظور از قابلیت های زمان حقیقی تحت وب توانایی ارسال محتوا به کاربر به محض اینکه وی قابل دسترس است می باشد. این قابلیت بر خلاف روش گذشته یعنی انتظار سرور برای درخواست داده جدید از سوی کاربر عمل می کند.

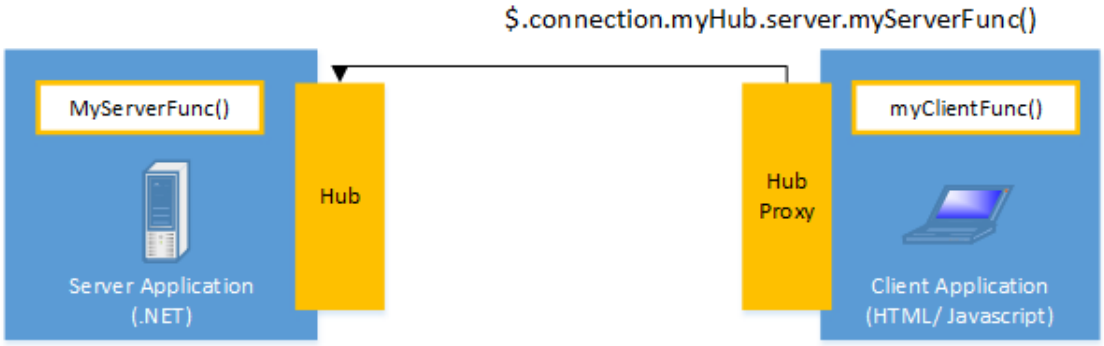
SignalR می تواند برای افزودن هر گونه فعالیت زمان حقیقی به برنامه شما اضافه گردد.به عنوان نمونه ایجاد یک سیستم چت. مواردی که کاربر نیاز به تازه سازی صفحه برای مشاهده داده های جدید دارد از موارد کاربرد این کتابخانه است. مثال هایی از این دست عبارتند از: نرم افزارهای نظارتی،نرم افزارهای کارگروهی و همکاری برخط و فرم های زمان حقیقی.

SignalR قابلیت ایجاد نسل جدیدی از نرم افزارها که نیاز به نرخ بروز رسانی بالایی دارند را نیز دارا می باشد برای مثال بازی های برخط. برای یک مثال مناسب می توانید بازی ShootR را ملاحظه نمایید.

SignalR یک API ساده به منظور ایجاد قابلیت فراخوانی متدهای راه دور (RPC) از سمت سرور به کاربر را ایجاد می کند که این API توابع JavaScript را در مرورگر کاربر از طریق کدهای سمت سرور فراخوانی میکند.SignalR همچنین API هایی جهت مدیریت و گروه بندی ارتباطات دارا می باشد.



Server invocation of client method myClientFunc()



Client invocation of server method MyServerFunc()

SignalR مدیریت ارتباطات را به طور خودکار رسیدگی می کند و به شما اجازه می دهد که یک پیام را به تمام کاربران متصل مانند یک چت روم ارسال نمایید. شما همچنین می توانید این کار را برای کاربر خاصی نیز انجام دهید.ارتباط بین کاربر و سرور بر خلاف ارتباط سنتی HTML به صورت پایدار است.

SignalR از قابلیت "Server Push" پشتیبانی می نماید. توسط این قابلیت کدهای سمت کاربر می توانند از طریق فراخوانی متدهای راه دور (RPC) کدهای سمت سرور را فراخوانی نمایند.این قابلیت بر خلاف مدل معمول درخواست – پاسخ کنونی است.

برنامه های SignalR می توانند بین هزاران کاربری که از SQL Server،Service Bus یا Redis استفاده می کنند توزیع گردند.

SignalR متن باز است و از طریق github قابل دسترسی است.

SignalR و WebSocket

SignalR از حامل های WebSocket زمانی که در دسترس باشند استفاده می کند و در غیر اینصورت از حامل های قدیمی تر استفاده می کند.اگرچه شما می توانید برنامه خود را مستقیما از طریق WebSocket ایجاد نمایید، اما با استفاده از ابزارهای SignalR شما می توانید از قابلیت های فراوانی که قبلا در این کتابخانه گنجانده شده است استفاده نمایید.این به معنی است که شما می توانید برنامه خود را با استفاده از مزایای WebSocket ایجاد نمایید بدون اینکه نگران ایجاد کدهای مجزا برای کاربران قدیمی باشید.SignalR نیاز به بروزرسانی ندارد،چون خود برای پشتیبانی از حامل های مورد استفاده بروز می گردد،که این مورد یک رابط ثابت برای برنامه شما در استفاده از WebSocket مهیا می کند.

در حالی که شما می توانید برنامه خود را تنها با استفاده از WebSocket ایجاد نمایید،اما SignalR بسیاری از عملکردهای مورد نیاز شما را ایجاد نموده است.

حامل ها و تنزل ها

SignalR یک انتزاع برای تمام حامل هایی است که برای انجام کارهای زمان حقیقی بین کاربر و سرور مورد نیاز هستند.یک ارتباط با استفاده از SignalR در ابتدا به عنوان یک ارتباط HTTP آغاز می گردد،سپس اگر ارتباط از طریق WebSocket در دسترس باشد ارتباط ارتقا می یابد.WebSocket حامل ایده آل برای SignalR است زیرا استفاده بهینه از حافظه سرور را موجب می گردد،کمترین زمان تاخیر را دارا می باشد و خصوصیات دیگری چون ارتباط دو طرفه کامل بین کاربر و سرور را به همراه دارد.در کنار تمام این مزایا WebSocket پیش نیازهای سختی هم دارد؛WebSocket نیاز به سروری دارد که از Windows Server 2012 یا Windows 8 استفاده و از NET Framework 4.5. پشتیبانی نماید.اگر این پیش نیازها مهیا نباشند SignalR از حامل های دیگری به منظور ایجاد ارتباطات استفاده می نماید.

حامل های HTML5

حامل های به کار رفته در ارتباط بستگی به پشتیبانی از HTML5 دارد.اگر مرورگر کاربر از استاندارد HTML5 پشتیبانی نکند،حامل های قدیمی تر برای ایجاد ارتباط استفاده خواهند شد.

- WebSocket: این حامل تنها حاملی است که یک پایداری حقیقی و ارتباط دو طرفه کامل بین کاربر و سرور را ایجاد می کند.WebSocket پیش نیازهای سختی دارد؛ این حامل تنها در آخرین نسخه مرورگرهای IE، Chrome و Firefox به طور کامل پشتیبانی شده است و پیاده سازی جزئی از آن در مرورگرهای دیگر مانند Opera و Safari اجرا شده است.
- Server Sent Events: این حامل با نام Event Source نیز شناخته می شود. (همه مرورگرها به جز IE از آن پشتیبانی می کنند.)

حامل های دنباله دار

این حامل ها بر اساس مدل برنامه های تحت وب دنباله دار بنا شده اند که در آن مرورگر یک درخواست HTTP با طول عمر بالا را جهت ایجاد ارتباط نگه داری می کند که سرور می تواند از آن به منظور ارسال داده به کاربر بدون اینکه کاربر درخواست مشخصی را ارسال نماید استفاده کند.

- Forever Frame: تنها در IE پشتیبانی می گردد.این حامل یک IFrame پنهان ایجاد می کند که یک درخواست برای نقطه پایانی بر روی سرور که کامل نیست بوجود می آورد.سپس سرور بطور پیوسته اسکریپت هایی را به سمت کاربر ارسال می کند که این اسکریپت ها بی درنگ اجرا می گردند و یک ارتباط زمان حقیقی یک طرفه از سمت سرور به کاربر را ایجاد می کند.ارتباط از سمت کاربر به سرور از ارتباط جداگانه ایی استفاده می کند و مشابه هر درخواست استاندارد HTML ، برای هر بخش از داده که نیاز به ارسال دارد یک ارتباط جدید ایجاد می گردد.
- Ajax long polling: این حامل یک ارتباط پایدار ایجاد نمی کند، در مقابل به سرور و از طریق یک درخواست یادآوری می کند که تا زمان پاسخگویی در حالت باز باقی بماند. این درخواست در نقطه ایی که ارتباط در حال بسته شدن است ایجاد می گردد و یک ارتباط جدید بی درنگ درخواست می گردد.این روش زمانی که ارتباط راه اندازی مجدد می گردد موجب تاخیر می گردد.

برای اطلاعات بیشتر در مورد نحوه پشتیبانی از حامل ها در پیکربندی های مختلف به Supported Platforms مراجعه نمایید.

فرآیند انتخاب حامل

فهرست زیر مراحل انتخاب حامل توسط SignalR را نمایش می دهد.

۱. اگر مرورگر IE 8 یا نسخه های قدیمی تر است، از Long Polling استفاده می شود.
۲. اگر JSONP پیکربندی شده است از Long Polling استفاده می شود.
۳. اگر ارتباط میان دامنه ایی(cross-domain connection) ایجاد شده است، اگر شرایط زیر برقرار باشند از SignalR استفاده می شود:

- کاربر از Cross-Origin Resource Sharing(CROS) پشتیبانی نماید.برای اینکه نحوه پشتیبانی کاربر از این مورد را ملاحظه نمایید به آدرس www.caniuse.com/CORS مراجعه نمایید.
- کاربر از WebSocket پشتیبانی نماید.
- سرور از WebSocket پشتیبانی نماید.

اگر هر یک از این سه شرط برقرار نباشند، از حامل Long Polling استفاده خواهد شد.برای اطلاعات بیشتر در مورد ارتباط بین دامنه ایی (cross-domain connection) به مقاله How to establish a cross-domain connection مراجعه فرمایید.

۴. اگر JSONP پیکربندی نشده باشد و ارتباط از نوع میان دامنه ایی برقرار نباشد، WebSocket در صورت پشتیبانی توسط کاربر و سرور استفاده خواهد شد.
۵. اگر کاربر یا سرور از WebSocket پشتیبانی نکنند، Server Sent Events در صورت در دسترس بودن استفاده خواهد شد.
۶. اگر Server Sent Events در دسترس نباشد، Forever Frame استفاده خواهد شد.
۷. اگر Forever Frame با شکست مواجه گردد، از Long Polling استفاده خواهد شد.

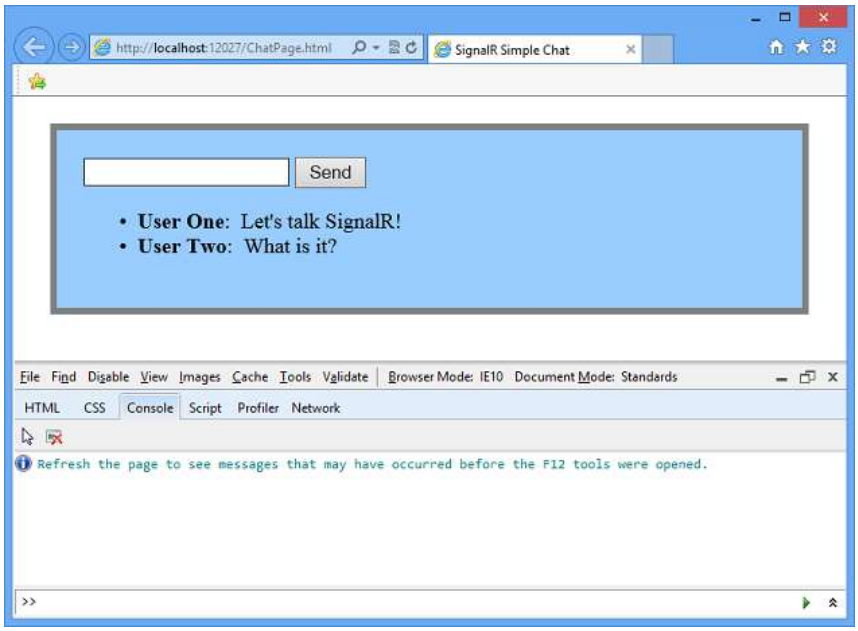
ردیابی حامل ها

شما می توانید نوع حاملی که برنامه شما از آن استفاده می کند را از طریق فعال سازی ثبت رخدادها در hub و باز کردن پنجره console مرورگر خود تعیین نمایید.

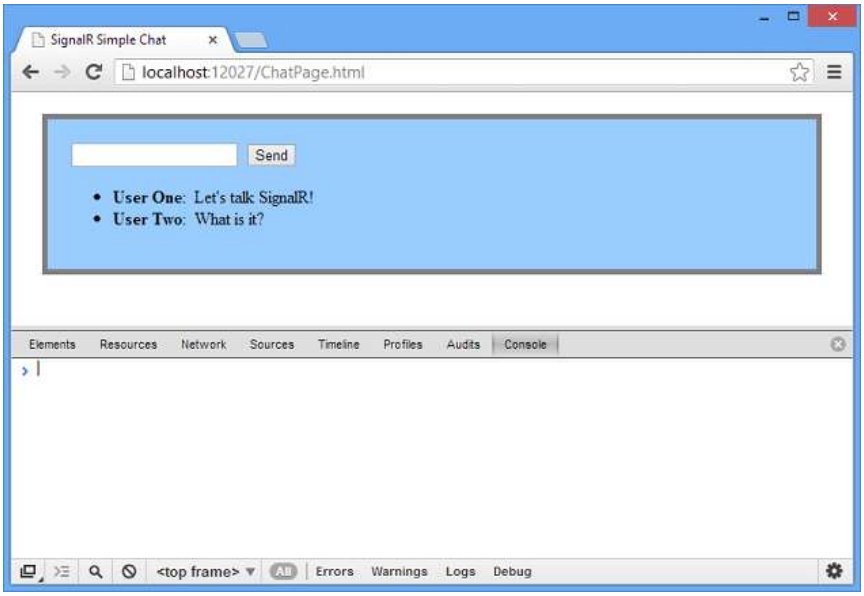
به منظور ثبت رخدادها ی hub خود در مرورگر،دستورات زیر را به برنامه خود اضافه نمایید.

```
$.connection.hub.logging = true;
```

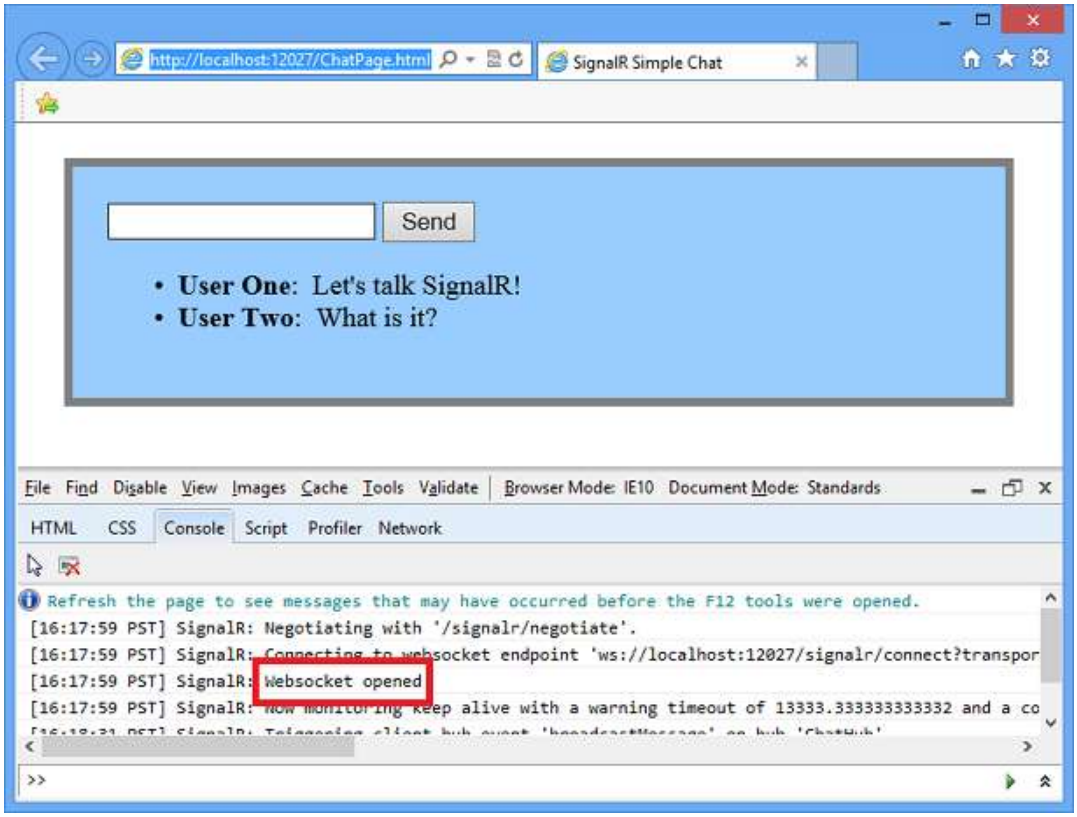
- در IE، developer tools را با فشردن کلید F12 باز نمایید و سپس بر وری console کلیک نمایید.



- در Chrome با فشردن کلیدهای Ctrl+Shift+J پنجره console را باز نمایید.



با فعال بودن ثبت رخداد hub و پنجره console شما می توانید حامل مورد استفاده SignalR را مشاهده نمایید.



مشخص کردن یک حامل

ارتباط یک حامل مقدار مشخصی از زمان و منابع کاربر و سرور را مصرف می کند.اگر ظرفیت های کاربر مشخص باشند می توان در زمان آغاز ارتباط یک نوع حامل خاص را مشخص کرد.کد زیر مشخص می کند که در آغاز ارتباط از حامل Ajax Long Polling استفاده شود.در صورتی که کاربر از هیچ قرارداد دیگری نیز پشتیبانی نکند از این قرارداد استفاده خواهد شد.

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

شما می توانید یک تربیت برای استفاده از حامل ها مشخص نمایید.این تنظیمات در شرایطی استفاده می گردد که شما می خواهید کاربر از یک ترتیب مشخص از حامل ها استفاده نماید.کد زیر تعیین می نماید که اگر تلاش برای استفاده از WebSocket با شکست مواجه گردد مستقیما از Long Polling استفاده گردد.

```
connection.start({ transport: ['webSockets','longPolling'] });
```

کلمات مورد استفاده برای مشخص کردن حامل ها به این صورت می باشد:

- `webSockets`
- `foreverFrame`
- `serverSentEvents`
- `longPolling`

Connections و Hubs

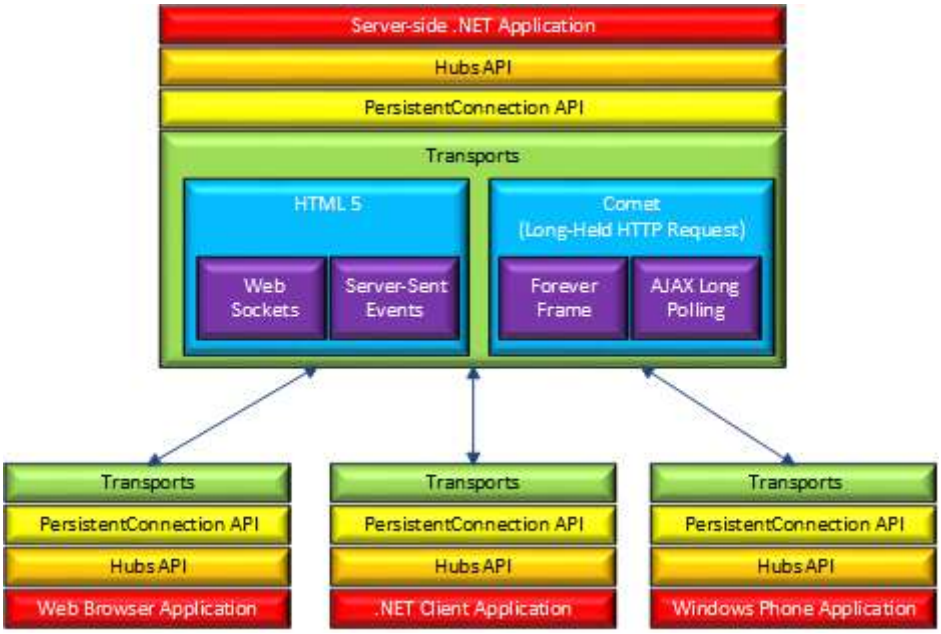
SignalR از دو مدل ارتباطی بین کاربر و سرور استفاده می کند: Persistent Connections و Hubs.

یک Connection تنها یک نقطه پایانی برای یک گیرنده منفرد، گروه یا پیام های فراگیر را نمایش می دهد.API ارتباط پایدار(که توسط کلاس PersistenteConnection در .NET. ارائه گردید.) یک دسترسی مستقیم به پروتکل های مکالمه سطح پایین که توسط SignalR مورد استفاده قرار می گیرد را در اختیار برنامه نویسان قرار می دهد.استفاده از مدل مکالمه ایی Connection برای برنامه نویسانی که از API های پایه ایی ارتباطی چون Windows Communication Foundation استفاده کرده اند آشنا خواهد بود.

یک Hub یک خط لوله سطح بالا است که بر روی Connection API ایجاد گردیده است که به کاربر و سرور شما اجازه می دهد توابعی را بر روی یکدیگر به طور مستقیم فراخوانی نمایند.SignalR ارسال بین محدوده ماشین ها را خود رسیدگی می کند و به کاربران اجازه می دهد که توابع سمت سرور را به آسانی توابع محلی فراخوانی نمایند و بالعکس.استفاده از مدل مکالمه ایی Hub برای برنامه نویسانی که از API های فراخوانی راه دور مثل .NET Remoting. استفاده کرده اند آشنا خواهد بود.استفاده از Hub به شما اجازه می دهد که داده های پیچیده را به عنوان پارامتر به توابع ارسال نمایید.

معماری

شکل زیر ارتباط بین هاب ها ، ارتباطات پایدار و تکنولوژی های مورد استفاده برای حامل ها را نمایش می دهد.



هاب ها چگونه کار می کنند.

زمانی که سرور یک تابع از سمت کاربر را فراخوانی می کند،یک بسته از طریق حامل فعال که شامل نام و پارامترهای تابع است ارسال می گردد.(زمانی که یک شی به عنوان پارامتر تابع ارسال می گردد،با استفاده از JSON به داده سریال تبدیل می گردد.) کاربر سپس نام تابع را با نام های تعریف شده در کد سمت کاربر مقایسه می کند اگر تابعی با نام مورد نظر وجود داشته باشد،تابع کاربر با استفاده از پارامتر مورد نظر اجرا می گردد.

تابع فراخوانی شده می تواند توسط ابزار هایی مانند fiddler مشاهده شود.تصویر زیر یک فراخوانی تابع از سرور SignalR به مرورگر کاربر را در fiddler نمایش می دهد.فراخوانی تابع از یک هاب به نام MoveShapeHub ارسال می گردد و تابع مورد درخواست UpdateShape نام دارد.

```
10:38:03:1298 Session846.WebSocket'WebSocket #846' - Pushing 104 bytes from server WebSocket
81 66 7B 22 43 22 3A 22 42 2C 31 35 7C 43 2C 30     f{"C":"B,15|C,0
7C 44 2C 30 7C 45 2C 30 22 2C 22 4D 22 3A 5B 7B  |D,0|E,0","M":[{
22 48 22 3A 22 4D 6F 76 65 53 68 61 70 65 48 75  "H":"MoveShapeHu
62 22 2C 22 4D 22 3A 22 75 70 64 61 74 65 53 68  b","M":"updateSh
61 70 65 22 2C 22 41 22 3A 5B 7B 22 6C 65 66 74  ape","A":[{"left
22 3A 35 30 31 2E 30 2C 22 74 6F 70 22 3A 33 30  ":501.0,"top":30
32 2E 30 7D 5D 7D 5D 7D                          2.0}]}]}
```

در این مثال نام هاب توسط پارامتر H ، نام تابع از طریق پارامتر M و داده های ارسالی از طریق پارامتر A مشخص شده اند.

انتخاب مدل مکالمه

اکثر برنامه باید از هاب استفاده نمایند.Connections API باید در محیط های زیر استفاده شوند.

- قالب واقعی پیام ارسالی نیاز به مشخص بودن دارد.
- برنامه نویسانی که ترجیح می دهند از مدل ارسال فراگیر به جای مدل فراخوانی از راه دور استفاده نمایند.
- یک برنامه موجود که از مدل پیام رسانی استفاده می کند که برای استفاده از SignalR از یک پورت مشخص استفاده می کند.