عنوان: **مثال - نمایش بلادرنگ تعداد کاربران آنلاین توسط SignalR**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۴ ۱۴:۲۳
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, jQuery, SignalR

راه حل‌های زیادی برای محاسبه و نمایش تعداد کاربران آنلاین یک برنامه وب وجود دارند و عموما مبتنی بر کار با متغیرهای سشن یا Application و امثال آن هستند. این روش‌ها عموما دقیق نبوده و خصوصا قسمت قطع اتصال کاربر را نمی‌توانند دقیقا تشخیص دهند. به همین جهت نیاز به یک تایمر دارند که مثلا اگر در 5 دقیقه قبل، کاربری درخواست مشاهده آدرسی را به سرور ارسال نکرده بود، از لیست کاربران آنلاین حذف شود.
در ادامه بجای این روش‌ها، از SignalR برای محاسبه تعداد کاربران آنلاین و همچنین به روز رسانی بلادرنگ این عدد در سمت کاربر، استفاده خواهیم کرد.

تشخیص اتصال و قطع اتصال کاربران در SignalR

زیر ساخت‌های کلاس Hub موجود در SignalR، دارای متدهای ردیابی اتصال (OnConnected)، قطع اتصال (OnDisconnected) و یا برقراری مجدد اتصال کاربران (OnReconnected) هستند. با بازنویسی این متدها می‌توان به تخمین بسیار دقیقی از تعداد کاربران آنلاین یک سایت رسید.

پیشنیازهای بحث

پیشنیازهای این بحث با مطلب « مثال - نمایش درصد پیشرفت عملیات توسط SignalR » یکی است. برای مثال نحوه دریافت وابستگی‌ها، تنظیمات فایل global.asax و افزودن اسکریپت‌ها، تفاوتی با مثال یاد شده ندارند.

تعریف هاب کاربران آنلاین برنامه

```
using System.Collections.Concurrent;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNet.SignalR;

namespace SignalR05.Common
{
    public class OnlineUsersHub : Hub
    {
        public static readonly ConcurrentDictionary<string, string> OnlineUsers = new
ConcurrentDictionary<string, string>();

        public void UpdateUsersOnlineCount()
        {
            // آی پی معرف یک کاربر است
            // اما کانکشن آی دی معرف یک برگه جدید در مرورگر او است
            // هر کاربر می‌تواند چندین برگه را به یک سایت گشوده یا ببندد
            var ipsCount = OnlineUsers.Select(x => x.Value).Distinct().Count();
            this.Clients.All.updateUsersOnlineCount(ipsCount);
        }

        /// <summary>
        /// اگر کاربران اعتبار سنجی شده‌اند بهتر است از
        /// this.Context.User.Identity.Name
        /// بجای آی پی استفاده شود
        /// </summary>
        protected string GetUserIpAddress()
        {
            object environment;
            if (!Context.Request.Items.TryGetValue("owin.environment", out environment))
                return null;

            object serverRemoteIpAddress;
            if (!((IDictionary<string, object>)environment).TryGetValue("server.RemoteIpAddress", out
serverRemoteIpAddress))
                return null;
```

```
            return serverRemoteIpAddress.ToString();
        }

        public override Task OnConnected()
        {
            var ip = GetUserIpAddress();
            OnlineUsers.TryAdd(this.Context.ConnectionId, ip);
            UpdateUsersOnlineCount();

            return base.OnConnected();
        }

        public override Task OnReconnected()
        {
            var ip = GetUserIpAddress();
            OnlineUsers.TryAdd(this.Context.ConnectionId, ip);
            UpdateUsersOnlineCount();

            return base.OnReconnected();
        }

        public override Task OnDisconnected()
        {
            // در این حالت ممکن است مرورگر کاملا بسته شده باشد
            // یا حتی صرفا یک برگه مرورگر از چندین برگه متصل به سایت بسته شده باشند
            string ip;
            OnlineUsers.TryRemove(this.Context.ConnectionId, out ip);
            UpdateUsersOnlineCount();

            return base.OnDisconnected();
        }
    }
}
```

کدهای کامل هاب شمارش کاربران آنلاین را در اینجا ملاحظه می‌کنید؛ به همراه نکته‌ی نحوه‌ی دریافت IP کاربر متصل شده به سایت، در یک هاب. کار افزودن یا حذف این کاربران به ConcurrentDictionary تعریف شده، در روال‌های بازنویسی شده اتصال، قطع اتصال و اتصال مجدد یک کاربر، انجام شده است.

در اینجا، هم به IP کاربر و هم به ConnectionId او نیاز است. از این جهت که هر ConnectionId، معرف یک برگه جدید باز شده در مرورگر کاربر است. اگر صرفا IPها را پردازش کنیم، با بسته شدن یکی از چندین برگه مرورگر او که اکنون به سایت متصل هستند، آمار او را از دست خواهیم داد. این کاربر هنوز چندین برگه باز دیگر را دارد که با سایت در ارتباط هستند، اما چون IP او را از لیست حذف کرده‌ایم (در نتیجه بسته شدن یکی از برگه‌ها)، آمار کلی شخص را نیز از دست خواهیم داد. بنابراین هر دوی IP و ConnectionIdها باید پردازش شوند.

اگر برنامه شما دارای اعتبارسنجی است (یک صفحه لاگین دارد)، بهتر است بجای IP از this.Context.User.Identity.Name استفاده کنید.

کدهای سمت کلاینت نمایش آمار کاربران

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <script src="Scripts/jquery-1.6.4.min.js" type="text/javascript"></script>
    <script src="Scripts/jquery.signalR-1.1.3.min.js" type="text/javascript"></script>
    <script type="text/javascript" src='<%= ResolveClientUrl("~/signalr/hubs") %>'></script>
</head>
<body>
    <form id="form1" runat="server">
    online users count: <span id="usersCount"></span>
    </form>
    <script type="text/javascript">
        $(function () {
            $.connection.hub.logging = true;
            var onlineUsersHub = $.connection.onlineUsersHub;
            onlineUsersHub.client.updateUsersOnlineCount = function (count) {
                $('#usersCount').text(count);
            };
            $.connection.hub.start();
        });
    </script>
```

```
</body>
</html>
```

با توجه به اینکه در هاب تعریف شده، متد پویای updateUsersOnlineCount، آمار تعداد کاربران متصل را (تعداد آی پی‌های منحصربفرد متصل را) به کلاینت‌ها ارسال می‌کند، بنابراین در سمت کلاینت نیز با تعریف callback ایی به همین نام، می‌توان این آمار دریافتی را به کاربران سایت نمایش داد. آماری که به صورت خودکار با کم و زیاد شدن کاربران به روز شده و نیازی نیست کاربر به صورت دستی، صفحه را به روز کند.

کدهای کامل این مثال را از اینجا نیز می‌توانید دریافت کنید:

SignalR05.zip

نظرات خوانندگان

نویسنده: احمد اقامحمدی
تاریخ: ۱۳۹۳/۱۱/۰۳ ۲۲:۱۷

این تابع در ورژن 2 بایستی به صورت زیر تغییر کنه:

```
protected string GetUserIpAddress()
{
    object serverRemoteIpAddress;
    if (!Context.Request.Environment.TryGetValue("server.RemoteIpAddress", out serverRemoteIpAddress))
                return null;

    return serverRemoteIpAddress.ToString();
}
```