

مقاله ای درباره کاربران Online در ASP.NET

نویسنده: Saeed Taghvaei

از ابتدا تا کنون روش های مختلفی برای بدست آوردن لیست کاربران آنلاین در سایت ها استفاده شده که مهم ترین آن ها استفاده از دیتا بیس و یا متغیر های Application بود که همگی نیاز اساسی به فایل Global.asax داشتند. این در حالی است که در بسیاری از هاستینگ های مجانی استفاده از این فایل پشتیبانی نمیشود و باعث ایجاد مشکلات فراوانی میشود .

از طرفی در روش استفاده از دیتا بیس نیز دو اشکال عمده وجود داشت که

- 1- فشار زیادی رو دیتا بیس وارد میشد که این اصلا به صلاح کیفیت یک سایت نیست.
- 2- سرعت لود صفحات با افزایش کاربران نیز به طبع پایین تر می آمد.

اما در روش زیر شما به هیچ فایلی حتی Web.config برای تنظیم Timeout سشن ها و ... نیاز ندارید و میتوانید با بهترین خروجی و کارکرد منظم یک آمار بسیار دقیق از کاربران سایت داشته باشید .

در این روش از Cache به جای DataBase و همچنین از شی DataTable به جای جداول فیزیکی استفاده میشود و باعث میشود تا تغییر سرعت در هر وضعیتی از تعداد کاربران نداشته باشید و هیچ فشاری نیز بر روی دیتا بیس شما نباشد .

در پروژه خود یک فایل Class ایجاد کنید و آن را از System.Web.UI.Page مشتق کنید. و در کلاس خود یک فیلد عمومی به نام minut از نوع Int تعریف کنید و همچنین یک متد برای زیر روال لود صفحه تعریف کرده و آن را در InitializeComponent() ثبت نمایید. در حال حاضر کلاس شما همچین شکلی خواهد داشت .

کد:

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Web.Caching;

namespace WebSite5
{
    public class Pages : System.Web.UI.Page
    {
        public int minut=5;
        private void Page_Load(object sender, System.EventArgs e)
        {
        }
        #region Web Form Designer generated code
        override protected void OnInit(EventArgs e)
        {
            InitializeComponent();
            base.OnInit(e);
        }
        private void InitializeComponent()
        {
            this.Load += new System.EventHandler(this.Page_Load);
        }
        #endregion
    }
}
```

همانطور که حتما فهمیدی قراره از این کلاس به جای System.Web.UI.Page در ارث بری صفحات استفاده شود و همگی صفحات از آن مشتق شوند به عبارت دیگر هر روتینی که در روال لود صفحه نوشته شود در هنگام لود همه صفحاتی که از آن مشتق میشوند نیز استفاده میشود .

در نتیجه این مکان قسمت مناسبی برای جایگزینی ساب روتین های مورد نیاز در فایل های Global.asax

میباشد.

در حال حاضر ما میخواهیم در صورتی که کش (Cache) مورد نظر در سرور وجود نداشت یک کش ساده و خالی ایجاد کنیم و در آن اطلاعات کاربران را ثبت کنیم. همچنین در این قسمت میبایستی DataTable مورد نظر را نیز در Cache خود Add کنیم تا در صفحات دیگر از آن استفاده کنیم. بدین منظور یک متد بدون بازگشتی با نام loadCaching() در فایل خود با محتویات زیر وارد میکنیم:

کد:

```
public void loadCaching()
{
    DataTable dt=new DataTable("UsersOnline");
    DataColumn dcA=new DataColumn("UserId",typeof(String));
    DataColumn dcB=new DataColumn("UserIp",typeof(String));
    DataColumn dcC=new DataColumn("UserLastVisit",typeof(System.DateTime));
    dt.Columns.Add(dcA);
    dt.Columns.Add(dcB);
    dt.Columns.Add(dcC);
    System.Web.Caching.CacheItemRemovedCallback CloseUser;
    CloseUser=new CacheItemRemovedCallback(Pages_CloseUser);
    Cache.Add("UserOnlineList",dt,null,DateTime.Now.AddMinutes(2),
        System.Web.Caching.Cache.NoSlidingExpiration,
        CacheItemPriority.Default,CloseUser);
}
```

همانطور که میبینید در متد بالا ابتدا پس از ایجاد یک DataTable سه Column به آن اضافه میکنیم که در یکی از این ها IP کاربر ذخیره میشود، در یکی از آن های نام کاربری کاربر ذخیره میشود و در آخرین فیلد نیز آخرین دسترسی کاربر به یکی از صفحات تعریف میشود . سپس یک delegate از نوع System.Web.Caching.CacheItemRemovedCallback تعریف میکنیم و آن را با متد Pages_CloseUser که بعدا متن آن را میگویم) ثبت میکنیم. اما مهم ترین قسمت این سیستم آخرین دستور آن می باشد. پارامتر های Cache.Add به ترتیب از قرار زیر خواهند بود :

اولین پارامتر : نام مقداری که در دیتا کش سرور ذخیره میشود .

دومین پارامتر: مقداری که باید در آن قسمت از دیتا کش سرور ذخیره شود(همان جدول خالی)ک سومین پارامتر: یک CacheDependency بار DataCache مورد نظر که چون اینجا لازم نیست میتوان مقدار null به آن پاس داد.

چهارمین پارامتر: یک AbsoluteExpiration برای تعیین مدت زمان عمر این متغیر که می توان آن را با نوع DateTime تنظیم نمود ، این عمل در صورتی که دقیقا به تاریخ مورد نظر رسید دیتا کش را پاک میکند .

پنجمین پارامتر: یک SlidingExpiration میباشد که کار پارامتر با لا با نوع TimeSpan انجام میدهد و پس از گذشت زمانی که توسط این مقدار تنظیم میشود دیتا کش را پاک میکند.

توجه : ماکرو سافت در MSDN خود اشاره داشته که همیشه از یک نوع از پارامتر اول یا دوم استفاده کنید و پارامتر بعدی را با یکی از انواع System.Web.Caching.Cache.NoSlidingExpiration در صورت استفاده از پارامتر چهارم) و یا System.Web.Caching.Cache.NoAbsoluteExpiration در صورت استفاده از پارامتر پنجم توصیه میکنند) که همانطور که میبینید در بالا استفاده از DateTime ترجیح داده شده.

پارامتر ششم: یک نوع از انوم System.Web.Caching.CacheItemPriority(enum) میباشد که ترجیحا آن را با مقدار System.Web.Caching.CacheItemPriority.Default تنظیم کنید .

آخرین پارامتر :

اصلی ترین پارامتر این متد یک دلگیت میباشد که آن را هنگام از بین بردن DataCache و فرا رسیدن Expiration مورد نظر که در پارامتر های قبلی تنظیم کردم ، ارجا میکند. به عبارتی تمامی متد هایی که در این دلگیت ذخیره و پاس میشوند زمانی که عمر دیتا کش مورد نظر پایان میرسد اجرا میشود. اما بخش بعد و مهم این سیستم متدی است که به آن دلگیت پاس داده بودیم (Pages_CloseUser) که متن آن در زیر آمده :

کد:

```
private void Pages_CloseUser(string key, object Value,
                             CacheItemRemovedReason reason)
{
    if(Value.GetType() is System.Data.DataTable && key=="UserOnlineList")
    {
        DataTable dt=(DataTable)Value;
        for(int i=0;i!=dt.Rows.Count;i++)
        {
            if(((DateTime)dt.Rows[i]["UserLastVisit"]).Minute-minut<
                DateTime.Now.Minute-minut)
            {
                dt.Rows.RemoveAt(i);
            }// end if
        }//end for
    }
    System.Web.Caching.CacheItemRemovedCallback CloseUser;
    CloseUser=new CacheItemRemovedCallback(Pages_CloseUser);
    Cache.Add("UserOnlineList",dt,null,DateTime.Now.AddMinutes(2)
             ,System.Web.Caching.Cache.NoSlidingExpiration
             ,CacheItemPriority.Default,CloseUser);
}
```

همانطور که میبینید ابتدا چک میشود که واقعا آیتم در حال نابود شدن همان جدول مورد نظر بوده یا خیر و حال در صورتی که آن جدول بود ابتدا آن را با استفاده از TypeCasting بازیابی میکنیم. سپس تمامی سطرها ی جدول که توسط قسمتی دیگر از این کد پاس میشود رو چک میکنیم . در صورتی که سطر مورد نظر (که همون کاربر مورد نظر میباشد) چیزی بیشتر از زمانی که ما تعیین کردیم در سایت حاضر نبود آن را از لیست خود حذف میکنیم . در انتها دوباره جدول را در دیتا کش به همان روش بالا ذخیره می کنیم . عمدتا این الگوریتم تا بی نهایت تکرار میشود و هر وقت که زمان آخرین بازدید کاربر مورد نظر از زمان تعیین شده بیشتر شد آن کاربر را از لیست اعضا حذف میکنیم. و دوباره لیست را ذخیره کرده و 2 دقیقه بعد آن را چک میکنیم . (به همین سادگی!) اما بریم سراغ روال لود صفحه : آنجا ابتدا چک میکنیم که اگر دیتا کش مورد نظر ست نشده بود ، آن متدی که در بالا نوشتیم را اجرا میکنیم تا یک جدول خالی در دیتا کش ذخیره شود.

کد:

```
private void Page_Load(object sender, EventArgs e)
{
    if(Cache["UserOnlineList"]==null)
    {
        this.loadCaching();
    }
}
```

اما اصلی ترین قسمت این کد زمانی است که میخواهیم کاربران را به لیست اضافه کنیم . روال دیگری در کد خود به نام Check_Session() ایجاد کنید و متن زیر را در آن وارد کنید:

:کد

```
public void Check_Session()
{
    String dd=Session["User"].ToString();
    DataTable dt=(DataTable)Cache["UserOnlineList"];
    string filter="UserIp='"+Request.UserHostAddress+"'";
    int cc=dt.Select(filter).Length;
    filter="UserId='"+dd+"'";
    int vv=dt.Select(filter).Length;
    if(vv==0 && cc==0)
    {
        DataRow dr=dt.NewRow();
        dr["UserIp"]=Request.UserHostAddress;
        dr["UserId"]=dd;
        dr["UserLastVisit"]=DateTime.Now;
        dt.Rows.Add(dr);
    }
    else
    {
        for(int i=0;i!=dt.Rows.Count;i++)
        {
            if(dt.Rows[i]["UserId"].ToString()==dd &&
                dt.Rows[i]["UserIp"].ToString()==Request.UserHostAddress)
            {
                dt.Rows[i]["UserLastVisit"]=DateTime.Now;
            } // end if
        } // end for
    }
}
```

در این روال فرض شده شما نام کاربر اعضای سایت خود را در سشن User ذخیر می کنید.) ابتدا نام کاربری را از سشن میگیریم. سپس جدول را از دیتا کش بازیابی میکنیم. در جدول موجود در دیتا کش با استفاده از متد Select جست و جو میکنیم که آیا کاربری با نام و Ip در لیست وجود دارد یا خیر. اگر کاربر موجود نبود در جدول یک سطر جدید باز کرده و در آن نام و زمان و ای پی کاربر را مینویسیم. اما اگر همچنین کاربری در لیست موجود بود زمان آخرین بازدیدش را به زمان حال تغییر میدهیم تا در دیتا کش نام کاربریش اش از بین نرود.

در نهایت روال لود صفحه بدین شکل خواهد بود:

:کد

```
private void Page_Load(object sender, System.EventArgs e)
{
    if(Cache["UserOnlineList"]==null)
    {
        this.loadCaching();
    }
    else
    {
        if(Session["User"]!=null)
        {
            Check_Session();
        }
    }
}
```

خب فایل مورد نظر که تمامی صفحات در آن نوشته میشوند به شکل زیر میشود :

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Web.Caching;

namespace WebSite5
{
    public class Pages : System.Web.UI.Page
    {
        public int minut=5;
        private void Page_Load(object sender, System.EventArgs e)
        {
            if(Cache["UserOnlineList"]==null)
            {
                this.loadCaching();
            }
            else
            {
                if(Session["User"]!=null)
                {
                    Check_Session();
                }
            }
        } // end Page_Load
        public void Check_Session()
        {
            String dd=Session["User"].ToString();
            DataTable dt=(DataTable)Cache["UserOnlineList"];
            string filter="UserId='"+dd+"'";
            int cc=dt.Select(filter).Length;
            filter="UserIp='"+Request.UserHostAddress+"'";
            int vv=dt.Select(filter).Length;
            if(vv==0)
            {
                DataRow dr=dt.NewRow();
                dr["UserIp"]=Request.UserHostAddress;
                dr["UserId"]=dd;
                dr["UserLastVisit"]=DateTime.Now;
                dt.Rows.Add(dr);
            }
            else
            {
                for(int i=0;i!=dt.Rows.Count;i++)
                {
                    if(dt.Rows[i]["UserId"].ToString()==dd &&
                        dt.Rows[i]["UserIp"].ToString()==Request.UserHostAddress)
                    {
                        dt.Rows[i]["UserLastVisit"]=DateTime.Now;
                    } // end if
                } // end for
            }
        }
    }
}
```

```

    }
} // end check_Session
public void loadCaching()
{
    DataTable dt=new DataTable("UsersOnline");
    DataColumn dcA=new DataColumn("UserId",typeof(String));
    DataColumn dcB=new DataColumn("UserIp",typeof(String));
    DataColumn dcC=new DataColumn("UserLastVisit",typeof(System.DateTime));
    dt.Columns.Add(dcA);
    dt.Columns.Add(dcB);
    dt.Columns.Add(dcC);
    System.Web.Caching.CacheItemRemovedCallback CloseUser;
    CloseUser=new CacheItemRemovedCallback(Pages_CloseUser);
    Cache.Add("UserOnlineList",dt,null,DateTime.Now.AddMinutes(2)
        ,System.Web.Caching.Cache.NoSlidingExpiration
        ,CacheItemPriority.Default,CloseUser);
} // end loadCaching

#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    InitializeComponent();
    base.OnInit(e);
}
private void InitializeComponent()
{
    this.Load += new System.EventHandler(this.Page_Load);
}
#endregion

private void Pages_CloseUser(string key, object Value,
    CacheItemRemovedReason reason)
{
    if(Value.GetType() is System.Data.DataTable && key=="UserOnlineList")
    {
        DataTable dt=(DataTable)Value;
        for(int i=0;i!=dt.Rows.Count;i++)
        {
            if(((DateTime)dt.Rows[i]["UserLastVisit"]).Minute-minut<
                DateTime.Now.Minute-minut)
            {
                dt.Rows.RemoveAt(i);
            }
        } // end for
        System.Web.Caching.CacheItemRemovedCallback CloseUser;
        CloseUser=new CacheItemRemovedCallback(Pages_CloseUser);
        Cache.Add("UserOnlineList",dt,null,DateTime.Now.AddMinutes(2)
            ,System.Web.Caching.Cache.NoSlidingExpiration
            ,CacheItemPriority.Default,CloseUser);
    } // end if
} // end Pages_CloseUser

} // end class

} //end namespace

```

خب، حالا هر صفحه ای رو که میخواهید میتونید براحتی دیتا کش را با TypeCasting به دیتا تیبل تبدیل کرده و داده های خود را نمایش دهید .

فقط یادتون باشه که وقتی میخواید سشن یوزرتون رو ست کنید بعد از ست کردن روال `Check_Session()` را بعدش اجرا نمایید که لیست همیشه آپ دیت بمونه .