

در این مطلب می‌خواهم شما را با نحوه بار گذاری ساعت و تاریخ سیستم سرور با استفاده از JQuery Ajax آشنا کنم. در بعضی از سایتها با استفاده از جاوا اسکریپت تاریخ و ساعت جاری سیستم کلاینت به او نشان داده می‌شود. این روش یک مزیت دارد: اول اینکه این کدها سمت کلاینت اجرا می‌شوند و برای سرور بار اضافی ایجاد نمی‌کنند. و یک عیب هم دارد: در صورتی که ساعت و تاریخ روی سیستم کلاینت تنظیم نباشد، همین ساعت و تاریخ نادرست برای او نمایش داده می‌شود. همین عیب می‌تواند باعث افت کیفیت وب سایت شود.

اما راهی هست که تاریخ و ساعت سیستم سرور برای کاربر نشان داده شود و آن هم استفاده از JQuery Ajax هست. به صورتی که هر ثانیه درخواستی برای یک handler فرستاده می‌شود و آن handler نیز ساعت و تاریخ روی سرور را باز می‌گرداند و این مقدار بازگشته شده را می‌توان در تگی از صفحه وب نمایش داد.

مثال: ابتدا یک صفحه aspx می‌سازیم و تگ زیر را در آن قرار می‌دهیم:

```
<p id="datetime"></p>
```

ساعت و تاریخ بار شده از سرور در این تگ باید نشان داده شود.

سپس کدهای اسکریپت زیر را می‌نویسیم:

```
var auto_referesh = setInterval
(
    function()
    {
        $.post
        (
            "GetDateTime.ashx",
            function (result)
            {
                $('#datetime').html(result);
            }
        );
    }, 1000
);
```

با نوشتن این کدها هر ثانیه یک بار، بوسیله Ajax درخواستی برای یک handler به اسم GetDateTime.ashx فرستاده می‌شود. وظیفه این handler برگرداندن تاریخ و ساعت فعلی سیستم سرور است. بعد از دریافت مقدار این مقدار از این handler، آنرا در تگ با شناسه datetime قرار می‌دهیم.

کد استفاده شده در handler هم به این صورت است:

```
<%@ WebHandler Language="C#" Class="GetDateTime" %>
using System;
using System.Web;

public class GetDateTime : IHttpHandler {

    public void ProcessRequest (HttpContext context) {
        context.Response.ContentType = "text/plain";
        context.Response.Write(DateTime.Now.ToString());
    }

    public bool IsReusable {
```

```
        get {  
            return false;  
        }  
    }  
}
```

در انتها فایل ضمیمه این مثال را از این لینک دریافت کنید:

[AjaxDateTime.zip](#)

نظرات خوانندگان

نویسنده: صابر فتح اللهی
تاریخ: ۱۰:۰۰ ۱۳۹۱/۰۶/۲۵

سلام
روش شما خوبه اما عیبی که داره اینه که هر ثانیه ساعت از سرور میگیره و باعث سربار میشه
به نظر من روش بهتر اینه از همون کد طرف کلاینت استفاده بشه اما زمان اولیه از سرور خونده بشه بعد طرف کلاینت این ساعت
خودمون اضافه کنیم و ساعت شبیه سازی کنیم
در صورتی هم که کاربر به صفحه دیگری بره یا صفحه رفرش کنه دوباره ساعت از سرور خونده میشه و الگوریتم ساعت طرف
کلاینت به کار می افته
به نظر من این روش بهینه تر هست حالا باز ببینیم دوستان چه نظری دارن
موفق و موید باشید

نویسنده: پژمان پارسائی
تاریخ: ۱۰:۳۷ ۱۳۹۱/۰۶/۲۵

سلام
بله این روش سربار اضافی داره. روشی که گفتید خیلی بهتر هست.
ولی خوب مسئله اینه که چطور میشه داده نوع DateTime در ASP.Net رو با jQuery خوند؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۰:۵۵ ۱۳۹۱/۰۶/۲۵

روش دوم: خود وب سرور هم با درخواست های Head [تاریخ رو](#) ارسال می کنه. به این صورت هم قابل خواندن است:

```
<script src="Scripts/jquery.min.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function () {
        try {
            var result = $.ajax({ 'type': 'HEAD', 'url': '/' }).success(function () {
                var date1 = new Date(result.getResponseHeader('Date'));
                alert(date1);
            });
        } catch (err) {
            //...
        }
    });
</script>
```

و مهم ترین مزیتش این است که با تمام وب سرورهای استاندارد کار می کنه و فرقی نمی کنه کد شما PHP است یا ASP.NET.
+ این رو هم باید در نظر داشت که حین پردازش تاریخ دریافتی از وب سرور باید مسایل GMT را هم لحاظ کرد تا تاریخ و زمان
دریافتی با زمان ایران تطابق پیدا کند.

نویسنده: احمد احمدی
تاریخ: ۱۲:۰۵ ۱۳۹۱/۰۶/۲۵

سلام
همونطور که جناب فتح اللهی فرمودند، بهتر هست که یک بار زمان را از سرور دریافت کنید و سپس بروزرسانی را بوسیله جاوا
اسکریپت انجام دهید.
بنده در ارتباط با این بحث، یک کلاس ساده نوشتم که می تونه عملیات بروزرسانی را انجام بده.

```
var MyTime = function () {
    var date;
    var tag;

    var init = function (hour, minute, seconds, tagId) {
        var constructor = getConstructorString(hour, minute, seconds);
        date = new Date(constructor);
        tag = document.getElementById(tagId);
        //console.log('MyTime : Init(%s, %s, %s, %s)', hour, minute, seconds, tagId);
    };

    var run = function () {
        update();
        window.setInterval(update, 1000);
        //console.log('MyTime : Run');
    };

    var update = function updateClock() {
        var h = date.getHours();
        var m = date.getMinutes();
        var s = date.getSeconds();

        s++;
        if (s == 60) { m++; s = 0; };
        if (m == 60) { h++; m = 0; };
        if (h == 13) h = 1;

        var constructor = getConstructorString(h, m, s);
        date = new Date(constructor);

        h = (h < 10) ? ("0" + h) : h;
        m = (m < 10) ? ("0" + m) : m;
        s = (s < 10) ? ("0" + s) : s;

        tag.innerHTML = h + ":" + m + ":" + s;
        //console.log('MyTime : update');
    };

    var getConstructorString = function (hour, minute, seconds) {
        //console.log('MyTime : getConstructorString');
        return '01/01/2000 ' + hour + ':' + minute + ':' + seconds;
    };

    return {
        Init: init,
        Run: run
    };
}
```

روش کار به این صورت هست که شما یک بار متد Init را به همراه پارامترهای (ساعت، دقیقه، ثانیه، آی دی تگ) فراخوانی می‌کنید، در نهایت برای اجرای ساعت، یک بار هم متد Run را فراخوانی می‌کنید.

مثال :

```
myTime = new MyTime();
myTime.Init(12, 59, 50, 'clock');
myTime.Run();
```

امیدوارم مفید بوده باشه .