عنوان: **استفاده از Awesomium.NET در برنامه‌های وب**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۵ ۱۵:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Awesomium

برای تهیه تصاویر سایت‌های معرفی شده در قسمت اشتراک‌های سایت ، پیشتر از کنترل WebBrowser دات نت که در پشت صحنه از امکانات IE کمک می‌گیرد، استفاده می‌کردم. بسیار ناپایدار است؛ به روز رسانی مشکلی داشته و وابسته است به سیستم عامل جاری سیستم. برای مثال مرتبا برای تهیه تصاویر بند انگشتی (Thumbnails) سایت‌های تهیه شده با بوت استرپ کرش می‌کرد و این کرش چون از نوع unmaged code است، عملا پروسه IIS وب سایت را از کار می‌انداخت و در این حالت سایت تا ری‌استارت بعدی IIS در دسترس نبود. برای حل این مشکل و بهبود کیفیت تصاویر تهیه شده، از پروژه Awesomium که در حقیقت مرورگر کروم را جهت استفاده در انواع و اقسام زبان‌های برنامه نویسی محصور می‌کند، کمک گرفته شد؛ که شرح آن‌را در ادامه ملاحظه خواهید کرد.

دریافت و نصب Awesomium

پروژه Awesomium دارای یک SDK است که از اینجا قابل دریافت می‌باشد. بعد از نصب آن در مسیر Awesomium SDK\1.7.3.0\wrappers\Awesomium.NET\Assemblies\Packed می‌توانید محصور کننده‌ی دات نتی آن‌را مشاهده کنید. منظور از Packed در اینجا، استفاده از DLLهای فشرده شده‌ی native آن است که در مسیر Awesomium SDK\1.7.3.0\build\bin\packed کپی شده‌اند. بنابراین برای توزیع این نوع برنامه‌ها نیاز است اسمبلی دات نتی Awesomium.Core.dll به همراه دو فایل بومی icudt.dll و awesomium.dll ارائه شوند.

تهیه تصاویر سایت‌ها به کمک Awesomium.NET

پس از نصب Awesomium اگر به مسیر Documents\Awesomium SDK Samples\1.7.3.0\Awesomium.NET\Samples\Core\CSharp\BasicSample مراجعه کنید، مثالی را در مورد تهیه تصاویر سایت‌ها به کمک Awesomium.NET، مشاهده خواهید کرد. خلاصه‌ی آن چند سطر ذیل است:

```
try
            {
                using (WebSession mywebsession = WebCore.CreateWebSession(
new WebPreferences() { CustomCSS = "::-webkit-scrollbar { visibility: hidden; }" }))
                {
                    using (var view = WebCore.CreateWebView(1240, 1000, mywebsession))
                    {
                        view.Source = new Uri("https://site.com/");

                        bool finishedLoading = false;
                        view.LoadingFrameComplete += (s, e) =>
                        {
                            if (e.IsMainFrame)
                                finishedLoading = true;
                        };

                        while (!finishedLoading)
                        {
                            Thread.Sleep(100);
                            WebCore.Update();
                        }

                        using (var surface = (BitmapSurface)view.Surface)
                        {
                            surface.SaveToJPEG("result.jpg");
                        }
                    }
                }
            }
            finally
            {
                WebCore.Shutdown();
            }
```

کار با ایجاد یک WebSession شروع می‌شود. سپس با مقدار دهی CustomCSS، اسکرول بار صفحات را جهت تهیه تصاویری بهتر مخفی می‌کنیم. سپس یک WebView آغاز شده و منبع آن به Url مدنظر تنظیم می‌شود. در ادامه باید اندکی صبر کنیم تا بارگذاری سایت خاتمه یافته و نهایتا می‌توانیم سطح این View را به صورت یک تصویر ذخیره کنیم.

مشکل! این روش در برنامه‌های ASP.NET کار نمی‌کند!

مثال همراه آن یک مثال کنسول ویندوزی است و به خوبی کار می‌کند؛ اما در برنامه‌های وب پس از چند روز سعی و خطا مشخص شد که:
الف) WebCore.Shutdown فقط باید در پایان کار یک برنامه فراخوانی شود. یعنی اصلا نیازی نیست تا در برنامه‌های وب فراخوانی شود.

```
System.InvalidOperationException: You are attempting to re-initialize the WebCore.
The WebCore must only be initialized once per process and must be shut down only when the process
exits.
```

ب) Awesomium فقط در یک ترد کار می‌کند. به این معنا که اگر کدهای فوق را در یک صفحه‌ی وب فراخوانی کنید، بار اول کار خواهد کرد. بار دوم برنامه کرش می‌کند؛ با این پیغام خطا:

```
System.AccessViolationException: Attempted to read or write protected memory.
This is often an indication that other memory is corrupt. at
Awesomium.Core.NativeMethods.WebCore_CreateWebView_1(HandleRef jarg1, Int32 jarg2, Int32 jarg3,
HandleRef jarg4)
```

چون هر صفحه‌ی وب در یک ترد مجزا اجرا می‌شود، عملا استفاده‌ی مستقیم از Awesomium در آن ممکن نیست.
خطای فوق هم از آن نوع خطاهایی است که پروسه‌ی IIS را درجا خاموش می‌کند.

استفاده از Awesomium در یک ترد پس زمینه

راه حلی که نهایتا پاسخ داد و به خوبی و پایدار کار می‌کند، شامل ایجاد یک ترد مجزای Awesomium در زمان آغاز برنامه‌ی وب و زنده نگه داشتن آن تا زمان پایان کار برنامه است.

```
using System;
using System.Collections.Concurrent;
using System.IO;
using System.Threading;
using System.Web;
using Awesomium.Core;

namespace AwesomiumWebModule
{
    public class AwesomiumModule : IHttpModule
    {
        private static readonly Thread WorkerThread = new Thread(awesomiumWorker);
        private static readonly ConcurrentQueue<AwesomiumRequest> TaskQueue = new
ConcurrentQueue<AwesomiumRequest>();
        private static bool _isRunning = true;

        static AwesomiumModule()
        {
            WorkerThread.Start();
        }

        private static void awesomiumWorker()
        {
            while (_isRunning)
            {
                if (TaskQueue.Count != 0)
                {
                    AwesomiumRequest outRequest;
                    if (TaskQueue.TryDequeue(out outRequest))
                    {
                        var img = AwesomiumThumbnail.FetchWebPageThumbnail(outRequest);
```

```
                        File.WriteAllBytes(outRequest.SavePath, img);
                        Thread.Sleep(500);
                    }
                }
                Thread.Sleep(5);
            }
        }

        public void Dispose()
        {
            _isRunning = false;
            WebCore.Shutdown();
        }

        public void Init(HttpApplication context)
        {
            context.EndRequest += endRequest;
        }

        static void endRequest(object sender, EventArgs e)
        {
            var url = HttpContext.Current.Items[Constants.AwesomiumRequest] as AwesomiumRequest;
            if (url!=null)
            {
                TaskQueue.Enqueue(url);
            }
        }
    }
}
```

در اینجا اگر در برنامه‌های وب فرم، از طریق HttpContext.Current.Items.Add و یا در برنامه‌های MVC به کمک this.HttpContext.Items.Add یک آیتم جدید، با کلید Constants.AwesomiumRequest و از نوع کلاس AwesomiumRequest دریافت گردد، مقدار آن به یک ConcurrentQueue اضافه خواهد شد. این صف در یک ترد مجزا مدام در حال بررسی است. اگر مقداری به آن اضافه شده‌است، از صف خارج شده و پردازش خواهد شد.
نمونه‌ی استفاده از آن، در سمت یک برنامه‌ی وب نیز به صورت زیر است. ابتدا ماژول تهیه شده باید در برنامه ثبت شود:

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
    <httpModules>
      <add name="AwesomiumWebModule" type="AwesomiumWebModule.AwesomiumModule"/>
    </httpModules>
  </system.web>

  <system.webServer>
    <validation validateIntegratedModeConfiguration="false"/>
    <modules>
      <add name="AwesomiumWebModule" type="AwesomiumWebModule.AwesomiumModule"/>
    </modules>
  </system.webServer>
</configuration>
```

سپس باید تنها مدیریت HttpContext.Current.Items در سمت برنامه صورت گیرد:

```
protected void btnStart_Click(object sender, EventArgs e)
        {
            var host = new Uri(txtUrl.Text).Host;
            HttpContext.Current.Items.Add(Constants.AwesomiumRequest, new AwesomiumRequest
            {
                Url = txtUrl.Text,
                SavePath = Path.Combine(HttpRuntime.AppDomainAppPath, "App_Data\\Thumbnails\\" + host +
".jpg"),
                TempDir = Path.Combine(HttpRuntime.AppDomainAppPath, "App_Data\\Temp")
            });
            lblInfo.Text = "Please wait. Your request will be served shortly.";
        }
```

در اینجا کاربر درخواست خود را در صف قرار می‌دهد. پس از مدتی کار آن در WorkerThread ماژول تهیه شده انجام گردیده و

تصویر نهایی تهیه می‌شود.
Url، آدرس وب سایتی است که می‌خواهید تصویر آن تهیه شود. SavePath مسیر کامل فایل jpg نهایی است که قرار است دریافت و ذخیره گردد. TempDir محل ذخیره سازی فایل‌های موقتی Awesomium است. Awesomium یک سری کوکی، تصاویر و فایل‌های هر سایت را به این ترتیب کش کرده و در دفعات بعدی سریعتر عمل می‌کند.

پروژه‌ی کامل آن‌را از اینجا می‌توانید دریافت کنید:

AwesomiumWebApplication_V1.0.zip

نظرات خوانندگان

نویسنده: سعید شیرزادیان
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۹:۱۳

پروژه فوق در ویژوال استودیو 2012 باز نمی‌شود پیغام عدم سازگاری می‌دهد. آیا راهنمایی می‌توانید بکنید
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۹:۲۳

فایل‌های csproj و sln آن‌را حذف کنید. بعد دو پروژه وب فرم و MVC خالی درست کنید و فایل‌های موجود را به آن‌ها اضافه کنید.
ماژول آن هم یک class library ساده است.

نویسنده: Raees
تاریخ: ۱۳۹۳/۰۱/۰۵ ۲۰:۳

```
Hi Vahid
```

```
You have done an excellent job. But I have one query after generating one image the second one doesnot generate i mean do i need to refresh the page or do something special. The files in the  folder gets hold by the worker process and hence the subsequent request are not processed can you help us in this
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۵ ۲۰:۵۷

```
Remove the AwesomiumModule->Dispose method's body. It's not necessary
```