

فرض کنید می‌خواهیم وضعیت یک سایت را از لحاظ قابلیت دسترسی مونیتور کنیم، آیا Up است، Down است و امثال آن. یک سری از وب سرورها ping را بسته‌اند ([ICMP Replies](#)). بنابراین الزاما با استفاده از این روش ساده نمی‌توان به مقصود رسید. خوشبختانه انجام این کار با استفاده از فضای نام استاندارد System.Net و کلاس HttpRequest، بدون نیاز به هیچگونه کلاس یا کامپوننت خارجی، به سادگی قابل انجام است. کلاس زیر به همین منظور تهیه شده است:

```
using System;
using System.Net;

public class CSiteMonitor
{
    public struct UrlHeaderInfo
    {
        public DateTime _lastModified;
        public string _statusCode;
        public string _errorMessage;
    }

    /// <summary>
    /// آیا آدرس اینترنتی وارد شده معتبر است؟
    /// </summary>
    /// <param name="url">بررسی</param>
    /// <returns></returns>
    public static bool IsValidURL(string url)
    {
        try
        {
            Uri uri = new Uri(url);
            return (uri.Scheme == Uri.UriSchemeHttp) || (uri.Scheme == Uri.UriSchemeHttps);
        }
        catch { return false; }
    }

    /// <summary>
    /// آدرس اینترنتی جهت بررسی
    /// </summary>
    /// <param name="url"></param>
    /// <returns></returns>
    /// <exception cref="ArgumentException">آدرس اینترنتی وارد شده معتبر نیست</exception>
    public static UrlHeaderInfo GetSiteHeaderInfo(string url)
    {
        if (!IsValidURL(url))
            throw new ArgumentException("آدرس اینترنتی وارد شده معتبر نیست", "url");

        UrlHeaderInfo hhi = new UrlHeaderInfo { _lastModified = DateTime.Now, _statusCode = "NOK!",
        _errorMessage = string.Empty };

        HttpRequest request = (HttpRequest)WebRequest.Create(url);
        //request.Proxy
        request.Method = "HEAD";
        request.AllowAutoRedirect = true;
        request.UserAgent = "Mozilla/5.0 (Windows; U; Windows NT 5.0; ; rv:1.8.0.7) Gecko/20060917
Firefox/1.9.0.1";
        request.Timeout = 1000 * 300;
        request.AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;

        try
        {
            using (HttpWebResponse response = (HttpWebResponse) request.GetResponse())
            {
                hhi._statusCode = response.StatusCode.ToString();
                hhi._lastModified = response.LastModified;
            }
        }
        catch (Exception ex)
        {
            hhi._errorMessage = ex.Message;
        }
    }
}
```

```
        return hhi;
    }
}
```

توضیحات:

- در متد GetSiteHeaderInfo نیاز بود تا از یک تابع بیش از یک خروجی داشته باشیم. راههای زیاد برای انجام این کار هست. برای مثال:

الف) ارائه خروجیها به صورت یک آرایه. زیاد جالب نیست، چون اگر شخصی دقیقا مستندات متد شما را مطالعه نکند نمی داند که ترتیب خروجیها چگونه است و هر کدام چه معنایی دارند.

ب) ارائه خروجیها با استفاده از آرگومانهایی از نوع out یا ref. در دنیای شیء گرایی این نوع روشها را باید منسوخ شده در نظر گرفت و صرف سازگاری با زبانهایی مانند C که این روش در آنها رواج دارد (استفاده از آرگومانهایی از نوع اشاره گر) باید به آن نگاه کرد و نه بیشتر.

ج) خروجیها را به صورت یک کلاس یا struct در نظر گرفت تا استفاده کننده دقیقا بداند که فیلد خروجی چه معنایی دارد و هم چنین دقیقا چه تعداد خروجی مد نظر است.

- حتما باید از try/finally جهت اطمینان حاصل نمودن از بسته شدن response استفاده شود، در غیر اینصورت پس از دو خطای متوالی حاصل شده عملا دیگر نمیتوان از شیء response استفاده کرد. البته همانطور که [پیش تر](#) نیز ذکر شد، عبارت using توسط کامپایلر به try/finally بست داده میشود، بنابراین جهت خوانایی بیشتر کد بهتر است از این روش استفاده شود.

- جهت بلاک نشدن درخواست بهتر است از یک UserAgent کمک گرفته شود.

- جهت بررسی اعتبار یک آدرس اینترنتی یا میتوان از Regular expressions استفاده کرد یا از شیء Uri که روش آنرا ملاحظه میکنید.

- اگر در شبکه داخلی خود از پروکسی استفاده میشود، میتوان قسمت request.Proxy را با شیء پروکسی تنظیم شده مطابق مشخصات پروکسی سرور خود، بکار برد.

- در این مثال بیشتر هدف پیاده سازی کلاس دریافت اطلاعات هدر سایت بود و از ارائه کدهای مربوط به تایمر یا یک ترد جهت بررسی متوالی وضعیت سایت صرفنظر شد.

مثالی در مورد نحوه ی استفاده از کلاس فوق:

```
CSiteMonitor.UrlHeaderInfo info = CSiteMonitor.GetSiteHeaderInfo("http://www.google.com");
MessageBox.Show(info._statusCode);
```

نظرات خوانندگان

نویسنده: Anonymous
تاریخ: ۰۰:۰۹:۰۰ ۱۳۸۷/۱۱/۱۸

بسیار عالی
ممنون