

جهت بهینه سازی روش ارائه شده در مقاله "[بارگذاری یک یوزر کنترل با استفاده از جی کوئری](#)" ، می توان مبحث فشرده سازی را نیز به آن افزود.

برای این منظور نیاز است تا بتوان response حاصل را کاملا کنترل کرد و این مورد از طریق یک http module به خوبی قابل انجام است. مبحث http compression و پیاده سازی آن را احتمالا بارها در سایت های مختلف نیز دیده اید:

```
using System;
using System.IO;
using System.IO.Compression;
using System.Globalization;
using System.Web;

public class JsonCompressionModule : IHttpModule
{
    public JsonCompressionModule()
    {
    }

    public void Dispose()
    {
    }

    public void Init(HttpApplication app)
    {
        app.PreRequestHandlerExecute += new EventHandler(Compress);
    }

    private void Compress(object sender, EventArgs e)
    {
        HttpApplication app = (HttpApplication)sender;
        HttpRequest request = app.Request;
        HttpResponse response = app.Response;

        if (request.ContentType.ToLower(CultureInfo.InvariantCulture).StartsWith("application/json"))
        {
            if (!((request.Browser.IsBrowser("IE")) && (request.Browser.MajorVersion <= 6)))
            {
                string acceptEncoding = request.Headers["Accept-Encoding"];

                if (!string.IsNullOrEmpty(acceptEncoding))
                {
                    acceptEncoding = acceptEncoding.ToLower(CultureInfo.InvariantCulture);

                    if (acceptEncoding.Contains("gzip"))
                    {
                        response.Filter = new GZipStream(response.Filter, CompressionMode.Compress);
                        response.AddHeader("Content-encoding", "gzip");
                    }
                    else if (acceptEncoding.Contains("deflate"))
                    {
                        response.Filter = new DeflateStream(response.Filter, CompressionMode.Compress);
                        response.AddHeader("Content-encoding", "deflate");
                    }
                }
            }
        }
    }
}
```

در این ماژول تنها درخواست هایی از نوع application/json بررسی خواهند شد. هر چند این فشرده سازی را بر روی خروجی هر نوع WebMethod ایی نیز می توان اعمال کرد. در این حالت، سطر بررسی json را حذف کرده و آن را به صورت زیر تغییر دهید:

```
if ( !request.Url.PathAndQuery.ToLower().Contains( ".asmx" ) )
```

```
return;
```

مرورگر IE6 و پایین تر نیز از این فشرده سازی معاف شده اند (چون یا پشتیبانی کاملی را ارائه نمی دهند یا باید منتظر کرش مرورگر بود).

جهت اعمال این مازول به برنامه ASP.Net خود، کافی است سطر زیر را به قسمت httpModules وب کانفیگ افزود:

```
<httpModules>  
  <add name="JsonCompressionModule" type="JsonCompressionModule"/>  
</httpModules>
```

روش آزمایش مازول تهیه شده:

متاسفانه افزونه‌ی فایرباگ فایرفاکس اندازه‌ی نهایی response را نمایش می‌دهد و در گزارش آن حتی خبری از Content-encoding اضافه شده نیز نخواهد بود. بنابراین برای بررسی این روش مناسب نیست. ابزار دیگری که اساساً برای این نوع آزمایشات طراحی شده است، برنامه معروف [فیدلر](#) می‌باشد (که توسط مدیر پروژه تیم IE برنامه نویسی شده است). برای استفاده از فیدلر جهت دیباگ درخواست‌های local باید یک نکته‌ی کوچک را رعایت کرد:

/http://localhost :25413

همانطور که در URL فوق مشاهده می‌کنید یک نقطه پس از localhost اضافه شده است تا خروجی محلی مربوطه قابل بررسی شود.

Fiddler - HTTP Debugging Proxy

File Edit Rules Tools View Extensions neXpert Help

Comment Reissue Remove Resume All Streaming Find Save Launch IE Clear Cache

Web Sessions

| # | Result | Protocol | Host | URL |
|----|--------|----------|-----------|----------------------------|
| 11 | 200 | HTTP | localh... | /testajax/Ajax.aspx/Render |

neXpert FiddlerScript Statistics Inspectors

Headers TextView WebForms HexView Auth

No Proxy-Authorization Header is present.
No Authorization Header is present.

Response is encoded and may need to be decoded.

Transformer Headers TextView ImageView XML

Response Headers

HTTP/1.1 200 OK

- Cache**
 - Cache-Control: private, max-age=0
 - Date: Mon, 13 Jul 2009 16:48:02 GMT
- Entity**
 - Content-Length: 1196
 - Content-Type: application/json; charset=utf-8
- Miscellaneous**
 - Server: Microsoft-IIS/5.1
 - X-AspNet-Version: 2.0.50727
 - X-Powered-By: ASP.NET
- Transport**
 - Content-Encoding: gzip

Capturing All Processes Latency: Off 1 / 1 http://localhost./testajax/Ajax.aspx/Render

مطابق تصویر فوق، هم content-encoding اضافه شده مشخص است و هم حجم پاسخ دریافتی از 40 کیلوبایت (بر اساس یک تست معمولی روی صفحه ای مشخص) به نزدیک یک کیلوبایت و اندکی کاهش یافته است.