

Compressão de Respostas em APIs ASP.Net Core

Quando estamos desenvolvendo nossas APIs é muito importante nos preocuparmos com o tráfego de mensagens que ela irá gerar, pois largura de banda é um recurso limitado e não podemos esperar que nossos clientes irão poder contar com conexões de banda larga via fibra ótica à todo momento, ou ainda, conexões 4G ultra velozes sem limites de dados. Infelizmente essa não é a nossa realidade ainda... =/

O que podemos fazer é reduzir o tamanho das respostas em nossas APIs, e isso tem um impacto muito grande tanto financeiro, quanto de performance. Lembre-se que a grande maioria dos players de nuvem, ou todos eles, cobram por tráfego de dados, e além de tentar reduzir o custo devemos concordar que quanto menor for o tamanho dos dados, mais rápida será a resposta para quem estiver consumindo nossas APIs e assim melhorando a performance de nosso sistema de uma forma geral.

Uma maneira de diminuir o tamanho das mensagens trafegadas é implementando um mecanismo de compressão de respostas.

A compressão de respostas pode ser feita diretamente nos servidores web como o IIS e Nginx, só para citar alguns. Já o Kestrel e HTTP.sys não possuem esse recurso nativamente até o momento em que escrevo esse artigo.

Middleware de Compressão de Respostas do ASP.Net Core

O ASP.Net Core conta com alguns middlewares muito úteis, e um deles serve justamente para comprimir as respostas de nossas APIs.

Para isso basta adicionarmos uma referência ao pacote nuget [Microsoft.AspNetCore.ResponseCompression](https://www.nuget.org/packages/Microsoft.AspNetCore.ResponseCompression) e seguir os trechos de código abaixo:

```
public class Startup
{
    public Startup(IConfiguration configuration)
    {
        Configuration = configuration;
    }

    public IConfiguration Configuration { get; }

    public void ConfigureServices(IServiceCollection services)
    {

```

```
services.AddResponseCompression();  
services.AddMvc();  
}  
  
public void Configure(IApplicationBuilder app, IHostingEnvironment env)  
{  
    if (env.IsDevelopment())  
    {  
        app.UseDeveloperExceptionPage();  
    }  
  
    app.UseResponseCompression();  
    app.UseMvc();  
}
```

[view raw Startup.cs](#) hosted with ❤ by [GitHub](#)

O que fizemos foi a chamada de dois métodos bem específicos, nas linhas 12 e 23. Esses métodos de extensão são disponibilizados quando referenciamos o componente mencionado acima.

Linha 12 → foi adicionado o método “`services.AddResponseCompression()`”, que é responsável por adicionar o serviço de compressão de respostas no injetor de dependências nativo do ASP.Net Core.

Linha 23 → foi adicionado o método “`app.UseResponseCompression()`”, que adiciona o middleware de compressão de respostas no pipeline do ASP.Net Core.

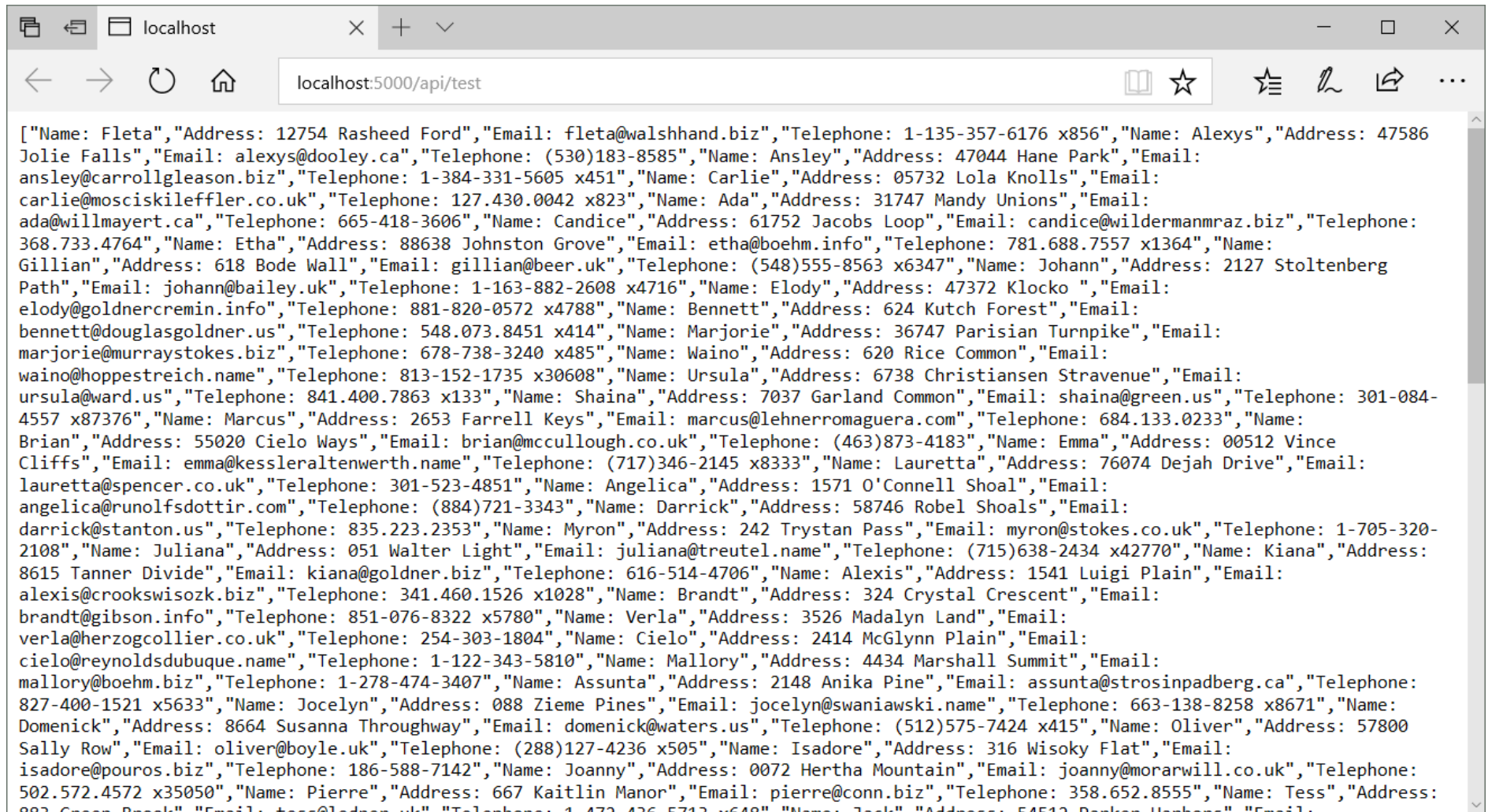
Com apenas duas linhas de código a compressão de resposta já irá funcionar, mas se quiser ir além, poderá customizar alguns aspectos como otimizar ainda mais a compressão, ou ainda, criar seu próprio algoritmo conforme podemos ver abaixo — os trechos de código de customização foram extraídos da documentação oficial que você encontra no fim deste artigo.

Use a compressão de respostas nativa do servidor web sempre que possível, pois ela tende a ter melhor performance que o middleware de compressão.

Como funciona?

Esse middleware irá interceptar todos os retornos das suas controllers e usar um algoritmo de compressão para diminuir o tamanho da resposta. Por padrão ele irá usar o algoritmo Gzip. Apenas para conhecimento, existe outro algoritmo de compressão também muito famoso chamado Deflate.

Para ver a compressão funcionando na prática podemos fazer uma chamada à nossa API de testes com dados fake através do navegador mesmo, conforme você pode ver abaixo:



```
[{"Name": "Fleta", "Address": "12754 Rasheed Ford", "Email": "fleta@walshhand.biz", "Telephone": "1-135-357-6176 x856"}, {"Name": "Alexys", "Address": "47586 Jolie Falls", "Email": "alexys@dooley.ca", "Telephone": "(530)183-8585"}, {"Name": "Ansley", "Address": "47044 Hane Park", "Email": "ansley@carrollgleason.biz", "Telephone": "1-384-331-5605 x451"}, {"Name": "Carlie", "Address": "05732 Lola Knolls", "Email": "carlie@mosciskileffler.co.uk", "Telephone": "127.430.0042 x823"}, {"Name": "Ada", "Address": "31747 Mandy Unions", "Email": "ada@willmayert.ca", "Telephone": "665-418-3606"}, {"Name": "Candice", "Address": "61752 Jacobs Loop", "Email": "candice@wildermanmraz.biz", "Telephone": "368.733.4764"}, {"Name": "Etha", "Address": "88638 Johnston Grove", "Email": "etha@boehm.info", "Telephone": "781.688.7557 x1364"}, {"Name": "Gillian", "Address": "618 Bode Wall", "Email": "gillian@beer.uk", "Telephone": "(548)555-8563 x6347"}, {"Name": "Johann", "Address": "2127 Stoltenberg Path", "Email": "johann@bailey.uk", "Telephone": "1-163-882-2608 x4716"}, {"Name": "Elody", "Address": "47372 Klocko ", "Email": "elody@goldnercremin.info", "Telephone": "881-820-0572 x4788"}, {"Name": "Bennett", "Address": "624 Kutch Forest", "Email": "bennett@douglasgoldner.us", "Telephone": "548.073.8451 x414"}, {"Name": "Marjorie", "Address": "36747 Parisian Turnpike", "Email": "marjorie@murraystokes.biz", "Telephone": "678-738-3240 x485"}, {"Name": "Waino", "Address": "620 Rice Common", "Email": "waino@hoppestreich.name", "Telephone": "813-152-1735 x30608"}, {"Name": "Ursula", "Address": "6738 Christiansen Stravenue", "Email": "ursula@ward.us", "Telephone": "841.400.7863 x133"}, {"Name": "Shaina", "Address": "7037 Garland Common", "Email": "shaina@green.us", "Telephone": "301-084-4557 x87376"}, {"Name": "Marcus", "Address": "2653 Farrell Keys", "Email": "marcus@lehnerromaguera.com", "Telephone": "684.133.0233"}, {"Name": "Brian", "Address": "55020 Cielo Ways", "Email": "brian@mccullough.co.uk", "Telephone": "(463)873-4183"}, {"Name": "Emma", "Address": "00512 Vince Cliffs", "Email": "emma@kessleraltenwerth.name", "Telephone": "(717)346-2145 x8333"}, {"Name": "Lauretta", "Address": "76074 Dejah Drive", "Email": "lauretta@spencer.co.uk", "Telephone": "301-523-4851"}, {"Name": "Angelica", "Address": "1571 O'Connell Shoal", "Email": "angelica@runolfsdottir.com", "Telephone": "(884)721-3343"}, {"Name": "Darrick", "Address": "58746 Robel Shoals", "Email": "darrick@stanton.us", "Telephone": "835.223.2353"}, {"Name": "Myron", "Address": "242 Trystan Pass", "Email": "myron@stokes.co.uk", "Telephone": "1-705-320-2108"}, {"Name": "Juliana", "Address": "051 Walter Light", "Email": "juliana@treutel.name", "Telephone": "(715)638-2434 x42770"}, {"Name": "Kiana", "Address": "8615 Tanner Divide", "Email": "kiana@goldner.biz", "Telephone": "616-514-4706"}, {"Name": "Alexis", "Address": "1541 Luigi Plain", "Email": "alexis@crookswisozk.biz", "Telephone": "341.460.1526 x1028"}, {"Name": "Brandt", "Address": "324 Crystal Crescent", "Email": "brandt@gibson.info", "Telephone": "851-076-8322 x5780"}, {"Name": "Verla", "Address": "3526 Madalyn Land", "Email": "verla@herzogcollier.co.uk", "Telephone": "254-303-1804"}, {"Name": "Cielo", "Address": "2414 McGlynn Plain", "Email": "cielo@reynoldsdubuque.name", "Telephone": "1-122-343-5810"}, {"Name": "Mallory", "Address": "4434 Marshall Summit", "Email": "mallory@boehm.biz", "Telephone": "1-278-474-3407"}, {"Name": "Assunta", "Address": "2148 Anika Pine", "Email": "assunta@strosinpadberg.ca", "Telephone": "827-400-1521 x5633"}, {"Name": "Jocelyn", "Address": "088 Zieme Pines", "Email": "jocelyn@swaniawski.name", "Telephone": "663-138-8258 x8671"}, {"Name": "Domenick", "Address": "8664 Susanna Throughway", "Email": "domenick@waters.us", "Telephone": "(512)575-7424 x415"}, {"Name": "Oliver", "Address": "57800 Sally Row", "Email": "oliver@boyle.uk", "Telephone": "(288)127-4236 x505"}, {"Name": "Isadore", "Address": "316 Wisoky Flat", "Email": "isadore@pouros.biz", "Telephone": "186-588-7142"}, {"Name": "Joanny", "Address": "0072 Hertha Mountain", "Email": "joanny@morarwill.co.uk", "Telephone": "502.572.4572 x35050"}, {"Name": "Pierre", "Address": "667 Kaitlin Manor", "Email": "pierre@conn.biz", "Telephone": "358.652.8555"}, {"Name": "Tess", "Address": "882 Green Brook", "Email": "tess@ledner.uk", "Telephone": "1-472-426-5712 x648"}, {"Name": "Jack", "Address": "54512 Barker Harbor", "Email": "jack@ledner.uk", "Telephone": "1-472-426-5712 x648"}]
```

E então através do [Fiddler](#) podemos conferir o tamanho da resposta, que nesse exemplo foi de **1.143 bytes**, além disso foi adicionado o header **Content-Encoding: gzip**, que indica que essa resposta foi compactada utilizando o algoritmo Gzip conforme o esperado.

Progress Telerik Fiddler Web Debugger

File Edit Rules Tools View Help

WinConfig Replay Go Stream Decode

Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff

Statistics Inspectors AutoResponder Composer Fiddler Orchestra Beta FiddlerScript Log Filters Timeline

#	Result	Protocol	Host	URL	Body	Caching
1	200	HTTP	localhost:5000	/api/test	1.143	

Request Headers

GET /api/test HTTP/1.1

Cache

Pragma: no-cache

Client

Accept: text/html, application/xhtml+xml, image/jxr, */*

Accept-Encoding: gzip, deflate

Accept-Language: pt-BR

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36 Edge/15.15063

Transport

Connection: Keep-Alive

Host: localhost:5000

Response body is encoded. Click to decode.

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies Raw JSON XML

Response Headers

HTTP/1.1 200 OK

Cache

Date: Sat, 17 Mar 2018 06:58:34 GMT

Vary: Accept-Encoding

Entity

Content-Encoding: gzip

Content-Type: application/json; charset=utf-8

Miscellaneous

Server: Kestrel

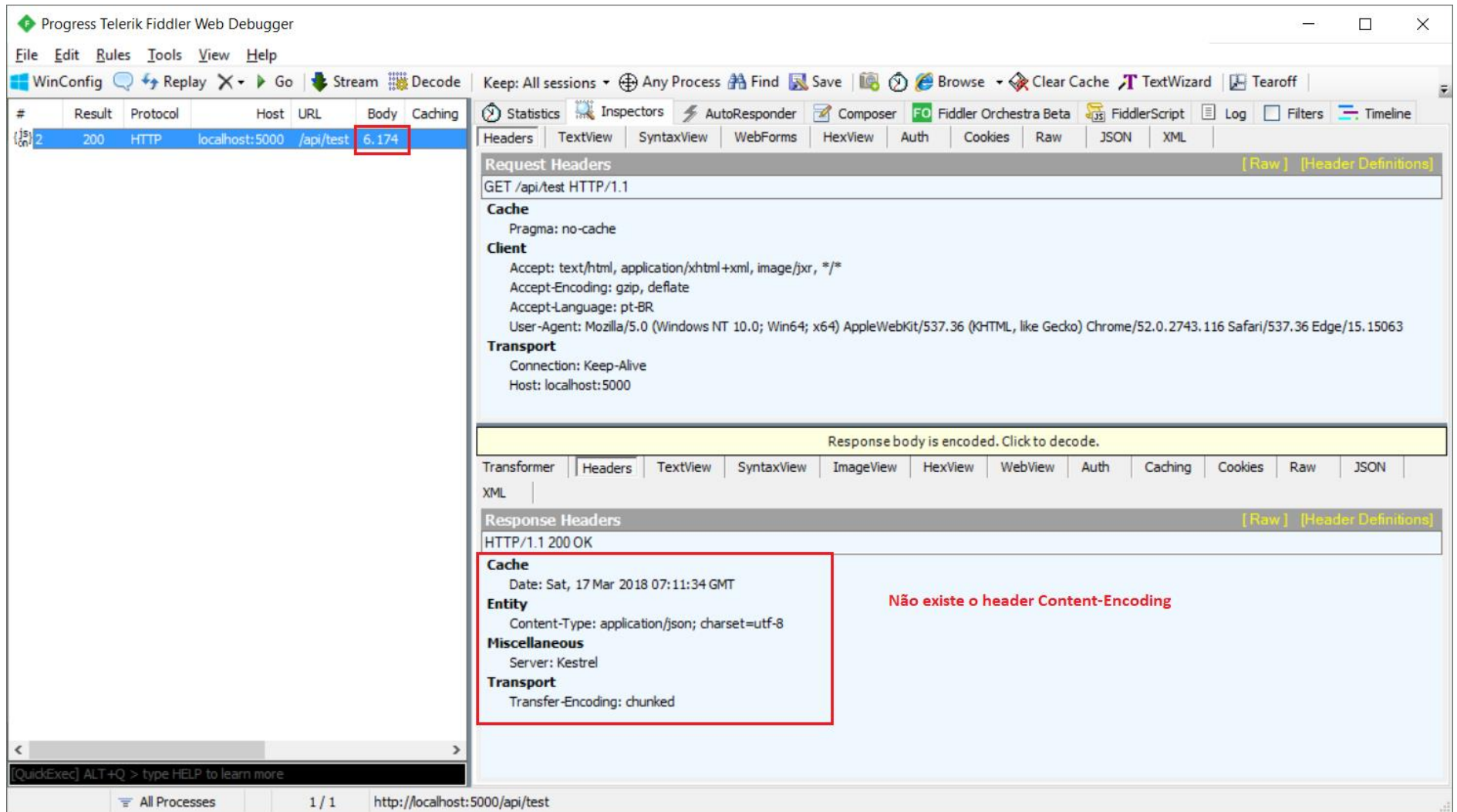
Transport

Transfer-Encoding: chunked

[QuickExec] ALT+Q > type HELP to learn more

All Processes 1 / 1 http://localhost:5000/api/test

Em uma requisição sem a compactação, podemos ver o tamanho da resposta em **6.174 bytes**, e não existe o header **Content-Encoding** indicando o algoritmo de compressão.



Verificando a documentação oficial, você irá observar uma nota de atenção para não tentar comprimir arquivos menores que 1.000 bytes, pois a compressão não será eficiente, o que pode inclusive resultar em uma resposta com tamanho maior ao original sem a compressão, além do tempo de processamento que foi gasto.

Assim podemos perceber que com uma simples adição ao nosso pipeline do ASP.Net conseguimos comprimir a resposta de nossa API, e dessa forma reduzir o tráfego de dados de nosso serviço.

É isso pessoal, espero que tenham gostado e se ficou qualquer dúvida ou tenham sugestões e críticas, fico à disposição para bater um papo.

Abrços!

Referências

- [Response Compression Middleware for ASP.NET Core](#)
- [Microsoft.AspNetCore.ResponseCompression Nuget Package](#)



[Wellington Nascimento](#)

.Net Developer