

Tối ưu hóa ứng dụng ASP.NET MVC

- Nguyễn Thanh Tùng
- CTO - MISA

Nội dung

1. Tối ưu phía client
2. Tối ưu phía server



Tối ưu phía client



Tool sử dụng

- YSlow (Yahoo slow)
- Cài đặt Chrome/Firefox extension

The screenshot displays the YSlow Chrome extension interface. The top navigation bar includes 'Home', 'Grade', 'Components', and 'Statistics'. The 'Grade' tab is active, showing a 'Grade B' overall performance score. A list of 23 items is shown, with the first item being 'A Make fewer HTTP requests'. The right sidebar provides detailed advice for this item, stating that the page has 4 external JavaScript scripts and suggesting to combine them into one. The bottom of the interface shows a copyright notice for 2016 Yahoo! Inc.

chrome-extension://ninejcohidippngpapiilnmkgllmakh/yslow.html#138

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help

Grade your web pages with YSlow

YSlow gives you:

- Grade based on the performance of the page (you can define your own ruleset)
- Summary of the page components
- Chart with statistics

☐ Check here if the current page prevents itself from being embedded/iframed. A simpler post onload detection will be used instead.

Run Test

[»Learn more about YSlow and the Yahoo! Developer Network](#)

Copyright © 2016 Yahoo! Inc. All rights reserved.

chrome-extension://ninejcohidippngpapiilnmkgllmakh/yslow.html#138

Home Grade Components Statistics Rulesets YSlow(V2) Edit

Grade B Overall performance score 87 Ruleset applied: YSlow(V2) URL: https://localhost:44300/#=_

ALL (23) FILTER BY: CONTENT (6) COOKIE (2) CSS (6) IMAGES (2) JAVASCRIPT (4) SERVER (6) [Tweet](#)

A	Make fewer HTTP requests
A	Use a Content Delivery Network (CDN)
A	Avoid empty src or href
F	Add Expires headers
A	Compress components with gzip
A	Put CSS at top
A	Put JavaScript at bottom
A	Avoid CSS expressions
N/A	Make JavaScript and CSS external
A	Reduce DNS lookups
A	Minify JavaScript and CSS
A	Avoid URL redirects

Grade A on Make fewer HTTP requests

This page has 4 external Javascript scripts. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[»Read More](#)

Copyright © 2016 Yahoo! Inc. All rights reserved.

Một số nguyên tắc tối ưu client

- Giảm số lượng request
 - Sử dụng asp.net mvc bundles
 - Sử dụng sprite images để giảm số lượng request ảnh

<http://css.spritegen.com/>

<http://spritegen.website-performance.org/>

1 reference | 1 author, 3 changes

```
public static void RegisterBundles(BundleCollection bundles)
{
    bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
        "~/Scripts/jquery-{version}.js",
        "~/Scripts/jquery.backstretch.js",
        "~/Scripts/jquery-ui-1.11.4.js",
        "~/Scripts/jquery.unobtrusive-ajax.js"
    ));

    bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
        "~/Scripts/jquery.validate*"));

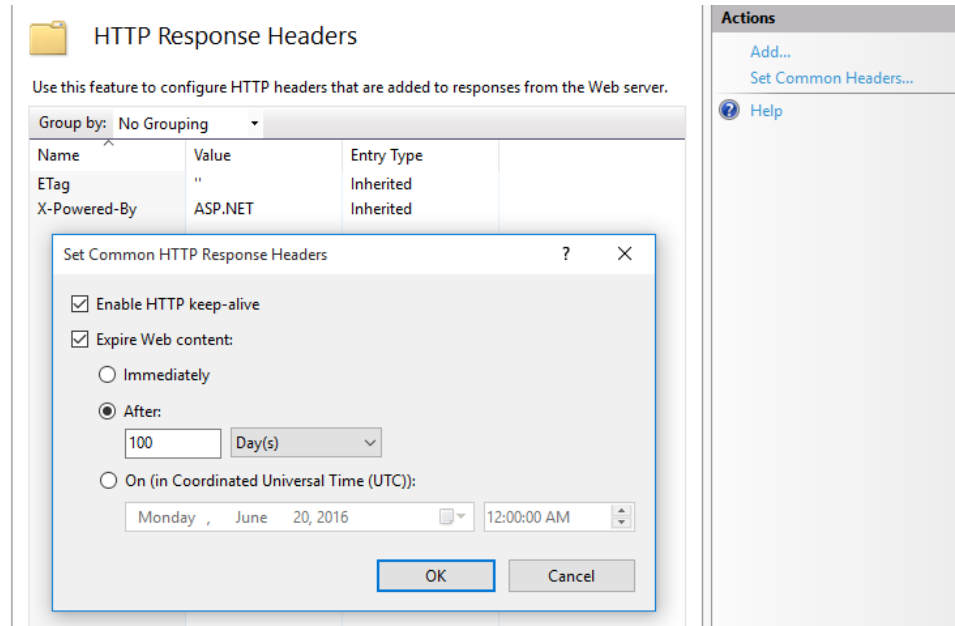
    bundles.Add(new ScriptBundle("~/bundles/moments").Include(
        "~/Scripts/jquery.signalR-2.1.2.min.js",
        "~/Scripts/app/infiniteScroll.js",
        "~/Scripts/app/tm.moment.feed.js"));
}
```

Một số nguyên tắc tối ưu client

- Sử dụng CDN
 - Tăng số request đồng thời
 - Mỗi 1 host trình duyệt chỉ cho phép 6 request đồng thời
 - Vì vậy cần sử dụng CDN để tăng số lượng request đồng thời
 - CDN thường lưu các file tĩnh như ảnh, css, javascript

Một số nguyên tắc tối ưu client

- Tăng cache cho các resource không thay đổi thường xuyên
 - Sử dụng Expired header cho các thư mục trong IIS



Một số nguyên tắc tối ưu client

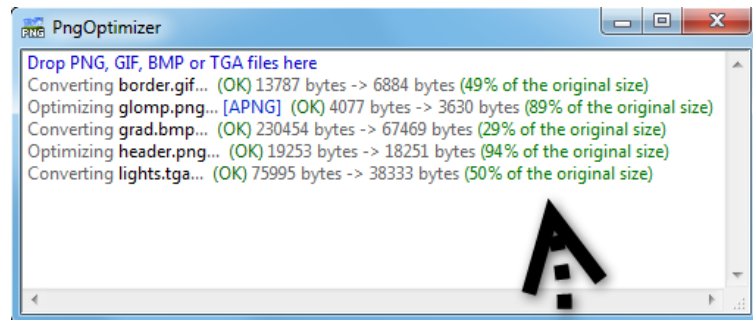
- Giảm dung lượng request
 - Sử dụng ASP.NET MVC minification
 - Sử dụng nén Gzip dữ liệu tĩnh và động (xml, json...) trong IIS
 - Nén các ảnh png
 - Tool sử dụng png-optimizer (<http://psydk.org/pngoptimizer>) hoặc các tool online
 - Sử dụng ảnh resize



Compression

Use this feature to configure settings for compression of responses. This can improve the perceived performance of a website greatly and reduce bandwidth-related charges.

- ☒ Enable dynamic content compression
- ☒ Enable static content compression



Một số nguyên tắc tối ưu client

- Đặt **css** trong thẻ head
 - Giúp trang render nhanh hơn do không cần phải đợi css để render lại
- Đặt **javascript** ở cuối trang html
 - Javascript block download các file khác khi nó đang được download vì vậy cần để dưới cùng

Một số nguyên tắc tối ưu client

- Sử dụng ajax để tối ưu ứng dụng
 - Không load cả trang mà load từng phần
 - Dữ liệu trả về nhẹ chỉ là json

Tối ưu phía server



ASP.NET

Tool sử dụng

- Sử dụng **Glimpse** để profile xem ứng dụng bị chậm ở phần nào



“NguyễnThanhTùng” https://localhost:44300/

?

⌵

🗄

🔌

Cache

Configuration

Environment

Execution

Metadata

Model Binding

Request

Routes

SQL

Server

Session

Timeline

Trace

Views

Ajax

History

Ordinal	Controller	Action	Is Child	Category	Category	Type.Method	Time Elapsed
0	Home	Index	false	Authorization	--	HomeController.OnAuthorization()	0.18 ms
1	Home	Index	false	Authorization	--	AuthorizeAttribute.OnAuthorization()	0.03 ms
2	Home	Index	false	Action	Executing	HomeController.OnActionExecuting()	0.15 ms
3	Home	Index	false	--	--	HomeController.Index()	1259.55 ms
4	Home	Index	false	Action	Executed	HomeController.OnActionExecuted()	0.24 ms
5	Home	Index	false	Result	Executing	HomeController.OnResultExecuting()	0.23 ms
6	Home	Index	false	--	--	ViewResult.ExecuteResult()	219.3 ms
7	Home	Index	false	Result	Executed	HomeController.OnResultExecuted()	0.22 ms

Một số nguyên tắc tối ưu server

- Tối ưu code .NET
 - Sử dụng `async/await` action
 - Do thread pool trong IIS là giới hạn - 5000 thread, mỗi 1 request sẽ chiếm 1M bộ nhớ stack nên nếu nhiều request đồng thời sẽ **chiếm 5G RAM** và báo lỗi **503 service unavailable**
 - Tăng khả năng phục vụ request của IIS vì không chiếm dụng thread pool
 - Áp dụng cho các action long-running phụ thuộc vào I/O hoặc network như gọi web service

Một số nguyên tắc tối ưu server

- Sử dụng Cache
 - Sử dụng outputCache nếu là action tĩnh
 - Sử dụng .net cache/redis cache nếu cần lưu giá trị mức toàn bộ ứng dụng

```
// GET: Profile
[OutputCache(Duration = 3600, VaryByParam = "userName")]
2 references | 0/2 passing | tungnt185, 26 days ago | 1 author, 3 changes
public ActionResult Detail(string userName)
{
    string loginUserId = UserHelper.GetCurrentUserIdFromIdentity(User);
    string userId;
    if (string.IsNullOrEmpty(userName))
    {
        userId = loginUserId;
    }
    else
    {
        userId = UserManager.FindByName(userName).Id;
    }
    UserProfileViewModel userProfile = UserService.GetUserProfileByUserId(userId,
        loginUserId != userId ? loginUserId : null);
    return View(userProfile);
}
```

Một số nguyên tắc tối ưu server

- Sử dụng **Store Procedure** với EF
 - Ưu điểm cần gì lấy đấy kiểm soát tốt performance
 - Nhược điểm phải bảo trì cả code .NET và code T-SQL

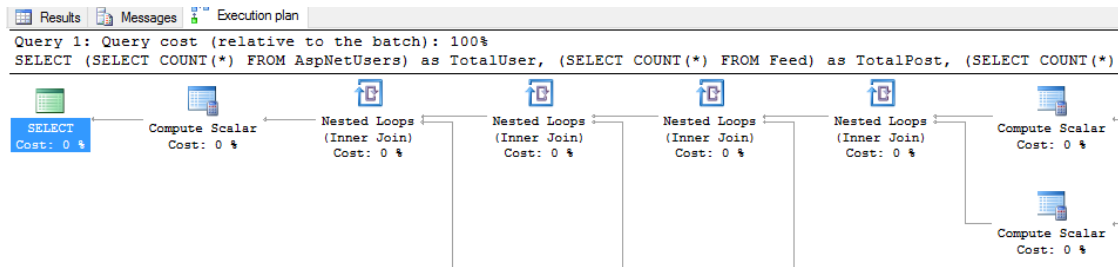
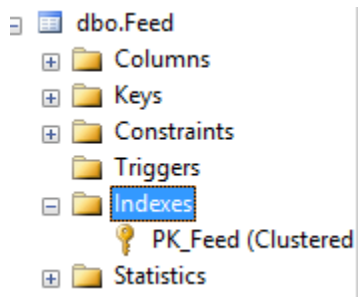
```
public object GetStatistic()  
{  
    TMMomentEntities DbContext = new TMMomentEntities();  
    var result = DbContext.Proc_GetStatistic();  
    return result;  
}
```

- Sử dụng **Dapper**
 - Ưu điểm performance cao hơn rất nhiều EF
 - Nhược điểm phải viết Command String hoặc Store Procedure
 - <https://github.com/StackExchange/dapper-dot-net>

```
public IEnumerable<UserStatistic> GetUserStatistic()  
{  
    IDbConnection _db = new SqlConnection(ConfigurationManager.ConnectionStrings  
        ["DefaultConnection"].ConnectionString);  
    IEnumerable<UserStatistic> result = new List<UserStatistic>();  
    result = _db.Query<UserStatistic>("SELECT TOP 8 * FROM [User] Order by CreatedDate  
        desc").ToList();  
    return result;  
}
```

Một số nguyên tắc tối ưu server

- SQL Server
 - Sử dụng và tối ưu Index dùng Execution Plan



Thanks for your listening

Q&A