

Rails 3

Fast by Default

Convention

over

Configuration



Fast isn't a Feature.
Fast is a Requirement.

Grade **A** Overall performance score 99 Ruleset applied: YSlow(V2) URL: http://search.yahoo.com/web?...

ALL (22) FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(5\)](#)

- A **Make fewer HTTP requests**
- A Use a Content Delivery Network (CDN)
- A 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
- A Remove duplicate JavaScript and CSS
- A Configure entity tags (ETags)
- A Make AJAX cacheable
- A Use GET for AJAX requests
- A Reduce the number of DOM elements
- A Avoid HTTP 404 (Not Found) error
- B Reduce cookie size
- A Use cookie-free domains
- A Avoid AlphaImageLoader filter
- A Do not scale images in HTML
- A Make favicon small and cacheable

Grade A on Make fewer HTTP requests

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 © 2009 Yahoo! Inc. All rights reserved.

YSlow!

YSlow Best Practices

Rails Conventions

**Far Future
Expires**

Expires: Thu, 13 Feb 2014 00:00:00 GMT

Updates

```

```

```

```

```
def rewrite_asset_path(source)
  asset_id = rails_asset_id(source)
  if asset_id.blank?
    source
  else
    source + "?#{asset_id}"
  end
end
```

**Minimize
HTTP
Requests**

```
javascript_include_tag :all
```

```
javascript_include_tag :all,  
                       :cache => true
```

New Default

Little Details

Development Mode

CSS and JS

**Reduce DNS
Lookups**

**Cookie Free
Domains**

**Split
Components
Across
Domains**



*My guideline is to split these components across **at least two** but **no more than four** hostnames*

```
config.action_controller.asset_host =  
  "assets%d.myapp.com"
```

```
config.action_controller.asset_host =  
  "assets%d.myapp.com"
```

assets0.myapp.com
assets1.myapp.com
assets2.myapp.com
assets3.myapp.com

Combo Move

```
image_tag("velocity.png")
```

```

```

```

```

```

```

```
javascript_include_tag "jquery",  
                        "application"
```

```
<script  
  src="http://assets1.myapp.com/  
jascripts/all.js?1232285206">
```

```
<script  
  src="http://assets1.myapp.com/  
jascripts/all.js?1232285206">
```

```
<script  
  src="http://assets1.myapp.com/  
javascripts/all.js?1232285206">
```

```
<script  
  src="http://assets1.myapp.com/  
javascripts/all.js?1232285206">
```

Development Mode

All Assets

Conventional

Integrated

ETags for Dynamic Content

Hard Part?

Generate ETag
Find If-None-Match
Compare

Get Time
Find Last-Modified
Compare

```
def show
  @article = Article.find(params[:id])

  if stale?(:etag => @article)
    @statistics = @article.really_expensive_call
    respond_to do |format|
      # all the supported formats
    end
  end
end
end
```

```
def show
  @article = Article.find(params[:id])

  if stale?(:etag => @article)
    @statistics = @article.really_expensive_call
    respond_to do |format|
      # all the supported formats
    end
  end
end
end
```

```
def show
  @article = Article.find(params[:id])

  if stale?(:etag => @article, :public => true)
    @statistics = @article.really_expensive_call
    respond_to do |format|
      # all the supported formats
    end
  end
end
end
```

```
def show
  @article = Article.find(params[:id])

  if stale?(:last_modified => @article.created_at)
    @statistics = @article.really_expensive_call
    respond_to do |format|
      # all the supported formats
    end
  end
end
end
```

**Less Tedious ==
More Usage**

Thanks