Let’s do a bunch of simple stuff to make our websites faster.
Let’s do a bunch of **simple stuff** to make our websites **faster**
Let’s do a bunch of simple stuff to make our websites faster.
LOW EFFORT BIG IMPACT

1. “Have a good to awesome affect on performance.”
2. “A single person can do without coordinating with anyone else.”
3. “Don’t require complex or mature processes.”
4. “Require minimal or no ongoing maintenance efforts.”
THE PERFORMANCE GOLDEN RULE

80 20
The Performance Golden Rule

On average, 80% of the end-user response time is spent on the front end.
“20%”
“80%”
117 Requests
7.81 MB
4.59s Total*
0.16s Back End
4.43s Front End
3% / 97%

* on Madison broadband
24%/76%  Top 10
8%/92%  10 ~10,000
16%/84%  Startups
13%/87%  Top 50,000
“Have a good to awesome affect on performance.”

“A single person can do without coordinating with anyone else.”

“Don’t require complex or mature processes.”

“Require minimal or no ongoing maintenance efforts.”

Focus more on the front end.
THE PLAN

1. Turn on HTTP Compression! i.e. (GZip)

2. Cache stuff! server and client side

3. Losslessly optimize images! and sprite if you can

4. Combine CSS and JS! three, two, or one per page
1. Turn on HTTP compression.
The Internet

SERVER

file

“The Internet”

file

imperceptibly fast squishing

imperceptibly fast un-squishing

BROWSER

file
gzipwtf.com is the easiest f***ing way to check for gzip and more. The purpose of this site is to aide web designers & developers in speeding up sites by pinpointing which resources are not being gzipped by the server, which resources are slow and which resources are causing 404s. Our mission is to increase awareness of site speed as an important part of the web’s future. George Washington approves.

Read more...

http://css-tricks.com
html5boilerplate.com

HTML5 ★ BOILERPLATE

A rock-solid default for HTML5 awesome.
▶ I'm new here: plz explain why it's good, first.

DOWNLOAD BOILERPLATE 3.0.2 UPDATED FEB 19TH
KEEP THE HINTS AND LINKS

OR

DOWNLOAD BOILERPLATE 'STRIPPED'
NO COMMENTS, JUST THE BIZNIZ.

OR

CUSTOMIZE BOILERPLATE
100% HIPSTER.

READ THE Docs ★ CONTRIBUTE ON GITHUB ★ FOLLOW ON TWITTER

HTML5 BOILERPLATE IS 3.0!
MONDAY, FEB 6TH, 2012

The key feature of this update is making Boilerplate smaller. Most of the changes have been working towards that. We have significant work done to our build script, so much so that we thought it deserved its own repository. Meanwhile, here is what's new.

CHANGelog

3.0 gets some major updates Here are all the commits since last release.
Should you upgrade existing sites? Short answer: nah, you're good.
A professional front-end template that helps you build fast, robust, adaptable, and future-proof websites. — Read more http://html5boilerplate.com/

Clone in Mac | ZIP | HTTP Git Read-Only https://github.com/h5bp/html5-boilerplate.git Read-Only access

Latest commit to the master branch

.htaccess: Serve WebVTT files as 'text/vtt' with UTF-8 encoding

airra authored 14 days ago

mathiasbynens committed 14 days ago

Commit 83f4f28186

.html5-boilerplate /

<table>
<thead>
<tr>
<th>name</th>
<th>age</th>
<th>message</th>
</tr>
</thead>
<tbody>
<tr>
<td>css</td>
<td>1 month</td>
<td>Rename style.css to main.css [necolas]</td>
</tr>
<tr>
<td>img</td>
<td>8 months</td>
<td>git ignore files do not need to ignore ignoring themselves [AD7six]</td>
</tr>
<tr>
<td>js</td>
<td>a month</td>
<td>Fixing two bugs introduced in the multiline reformat [nchaves]</td>
</tr>
<tr>
<td>.gitattributes</td>
<td>11 months</td>
<td>Simplify .gitattributes. Thanks, @jeffbyrnes! See <a href="https://github.com/">https://github.com/</a>... [mathiasbynens]</td>
</tr>
<tr>
<td>.gitignore</td>
<td>a month</td>
<td>Add a comment and link to more .gitignore info [drublic]</td>
</tr>
<tr>
<td>.htaccess</td>
<td>14 days</td>
<td>.htaccess: Serve WebVTT files as 'text/vtt' with UTF-8 encoding [airra]</td>
</tr>
<tr>
<td>404.html</td>
<td>a month</td>
<td>Formatting and consistency improvements [necolas]</td>
</tr>
<tr>
<td>apple-touch-icon-114x114-precomposed.png</td>
<td>4 months</td>
<td>Further optimize the new touch icons [mathiasbynens]</td>
</tr>
<tr>
<td>apple-touch-icon-144x144-precomposed.png</td>
<td>4 months</td>
<td>Further optimize the new touch icons [mathiasbynens]</td>
</tr>
<tr>
<td>apple-touch-icon-57x57-precomposed.png</td>
<td>4 months</td>
<td>Redesigned touch icons and favicon [richardcomish]</td>
</tr>
<tr>
<td>apple-touch-icon-72x72-precomposed.png</td>
<td>4 months</td>
<td>Redesigned touch icons and favicon [richardcomish]</td>
</tr>
<tr>
<td>apple-touch-icon-precomposed.png</td>
<td>4 months</td>
<td>Redesigned touch icons and favicon [richardcomish]</td>
</tr>
</tbody>
</table>
<FilesMatch ".\combined\.js">
  # Options +Includes
  # AddOutputFilterByType INCLUDES application/javascript application/json
  # SetOutputFilter INCLUDES
</FilesMatch>

<FilesMatch ".\combined\.css">
  # Options +Includes
  # AddOutputFilterByType INCLUDES text/css
  # SetOutputFilter INCLUDES
</FilesMatch>

# Gzip compression

<IfModule mod_deflate.c>
  # Force deflate for mangled headers developer.yahoo.com/blogs/ydn/posts/2010/12/pushing-beyond-gzipping/
  # IfModule mod_setenvif.c>
  # SetEnvIfNoCase ^Accept-Encoding: (X-cept-Encoding)\(X-\{15\}-\{15\}\)$ ^\((gzip|deflate)\s*,\s*\?)+$ [X-\{4,13\}]$ HAVE_Accept-Encoding
  # RequestHeader append Accept-Encoding "gzip, deflate" env=HAVE_ACCEPT-Encoding
</IfModule>

# HTML, TXT, CSS, JavaScript, JSON, XML, HTC:
<IfModule filter_module>
  FilterDeclare COMPRESS
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/html
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/css
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/plain
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/x-component
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/javascript
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/json
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/xml
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/xhtml+xml
  FilterProvider COMPRESS DEFLATE resp=Content-Type $image/svg+xml
  FilterProvider COMPRESS DEFLATE resp=Content-Type $image/x-icon
  FilterProvider COMPRESS DEFLATE resp=Content-Type $font/opentype
  FilterProvider COMPRESS DEFLATE resp=Content-Type $vnd.ms-fontobject
  FilterProtocol COMPRESS DEFLATE change=yes;byteranges=no
</IfModule>

<IfModule !mod_filter.c>
  # Legacy versions of Apache
  AddOutputFilterByType DEFLATE text/html text/plain text/css application/json
  AddOutputFilterByType DEFLATE application/javascript
  AddOutputFilterByType DEFLATE application/x-component
  AddOutputFilterByType DEFLATE application/xhtml+xml application/rss+xml
  AddOutputFilterByType DEFLATE application/atom+xml
  AddOutputFilterByType DEFLATE application/vnd.ms-fontobject application/x-font-ttf
  AddOutputFilterByType DEFLATE image/x-icon image/svg+xml
  AddOutputFilterByType DEFLATE application/vnd.ms-fontobject application/x-font-ttf font/opentype
</IfModule>
# Gzip compression

```
<IfModule mod_deflate.c>
  # Force deflate for mangled headers developer.yahoo.com/blogs/ydn/posts/2010/12/pushing-beyond-gzipping/
  <IfModule mod_setenvif.c>
    <IfModule mod_deflate.c>
      SetEnvIfNoCase ^{(Accept-Encoding|x-encoding|encoding|x-15|~x-15)15}$ ^{(gzip|deflate)\*|\*\*}+[x-\*]4,13}$ HAVE_Accept-Encoding
      RequestHeader append Accept-Encoding "gzip, deflate" env=HAVE_Accept-Encoding
    </IfModule>
  </IfModule>
</IfModule>

# HTML, TXT, CSS, JavaScript, JSON, XML, HTC:
<IfModule filter_module.c>
  FilterDeclare COMPRESS
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/html
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/css
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/plain
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/xml
  FilterProvider COMPRESS DEFLATE resp=Content-Type $text/x-component
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/javascript
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/json
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/x-xl
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/x-x+1ml
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/x-x+1ml
  FilterProvider COMPRESS DEFLATE resp=Content-Type $image/svg+xml
  FilterProvider COMPRESS DEFLATE resp=Content-Type $image/x-icon
  FilterProvider COMPRESS DEFLATE resp=Content-Type $application/x-font-ttf
  FilterProvider COMPRESS DEFLATE resp=Content-Type $font/opentype
  FilterChain COMPRESS DEFLATE change=yes;byteranges=no
</IfModule>

<IfModule !mod_filter.c>
  # Legacy versions of Apache
  AddOutputFilterByType DEFLATE text/html text/plain text/css application/json
  AddOutputFilterByType DEFLATE application/javascript
  AddOutputFilterByType DEFLATE text/xml application/xml text/x-component
  AddOutputFilterByType DEFLATE application/xhtml+xml application/rss+xml
  AddOutputFilterByType DEFLATE application/atom+xml
  AddOutputFilterByType DEFLATE image/x-icon image/svg+xml application/vnd.ms-fontobject
  AddOutputFilterByType DEFLATE application/x-font-ttf font/opentype
</IfModule>
```
2 CACHE STUFF
Hey, you’re going to need these files.

I already got a couple of those right here.
A rock-solid default for HTML5 awesome.

I'm new here: plz explain why it's good, first.

DOWNLOAD BOILERPLATE 3.0.2 UPDATED FEB 19TH
KEEP THE HINTS AND LINKS

DOWNLOAD BOILERPLATE 'STRIPPED'
NO COMMENTS, JUST THE BIZNIS.

CUSTOMIZE BOILERPLATE
100% HIPSTER.

READ THE Docs ★ CONTRIBUTE ON GHUB ★ FOLLOW ON TWITTER

HTML5 BOILERPLATE IS 3.0!
MONDAY, FEB 6TH, 2012

The key feature of this update is making Boilerplate smaller. Most of the changes have been working towards that. We have significant work done to our build script, so much so that we thought it deserved its own repository. Meanwhile, here is what’s new.

CHANGELOG

3.0 gets some major updates. Here are all the commits since last release.

Should you upgrade existing sites? Short answer: nah, you’re good.
# Expires headers (for better cache control)

# These are pretty far-future expires headers.
# They assume you control versioning with filename-based cache busting
# Additionally, consider that outdated proxies may miscache
# www.stevensouders.com/blog/2008/08/23/revving-filenames-dont-use-queryString/
# If you don't use filenames to version, lower the CSS and JS to something like
# "access plus 1 week" or so.

<IfModule mod_expires.c>
ExpiresActive on

# Perhaps better to whitelist expires rules? Perhaps.
ExpiresDefault "access plus 1 month"

# cache.appCache needs re-requests in FF 3.6 (thanks Remy - Introducing HTML5)
ExpiresByType text/cache-manifest "access plus 0 seconds"

# Your document html
ExpiresByType text/html "access plus 0 seconds"

# Data
ExpiresByType text/xml "access plus 0 seconds"
ExpiresByType application/xml "access plus 0 seconds"
ExpiresByType application/json "access plus 0 seconds"

# Feed
ExpiresByType application/rss+xml "access plus 1 hour"
ExpiresByType application/atom+xml "access plus 1 hour"

# Favicon (cannot be renamed)
ExpiresByType image/x-icon "access plus 1 hour"

# Media: images, video, audio
ExpiresByType image/gif "access plus 1 month"
ExpiresByType image/png "access plus 1 month"
ExpiresByType image/jpg "access plus 1 month"
ExpiresByType video/ogg "access plus 1 month"
ExpiresByType audio/ogg "access plus 1 month"
ExpiresByType video/mp4 "access plus 1 month"
ExpiresByType video/webm "access plus 1 month"

# HTC files (css3pie)
ExpiresByType text/x-component "access plus 1 month"
YO BROWSER, you can hold on to a bunch of these files. You might need them later.

Sweet.
The fastest HTTP request is the one not made.

http://www.stevesouders.com/blog/2012/03/22/cache-them-if-you-can/
It’s up to you to break cache.

style.v1.css
style.v2.css
style.v3.css
style.v4.css
...

3 LOSSLESSLY OPTIMIZE IMAGES
Americana Fans Happy with No Depression Showcase

SXSW News

Americana Fans Happy with No Depression Showcase

Posted by kevin on February 17, 2012

Among the hundreds of SXSW showcases announced this week, the No Depression lineup includes some of the brightest lights in Americana, kicking off at 8pm on Wednesday 3/14 at the Continental Club with the Trishas, a quartet of women who have quickly risen to headliner status here in Austin and whose
It’s like steroids for web developers.
ImageOptim optimizes images — so they take up less disk space and load faster — by finding best compression parameters and by removing unnecessary comments and color profiles. It handles PNG, JPEG and GIF animations.

ImageOptim seamlessly integrates various optimisation tools: PNGOUT, AdvPNG, Pngcrush, extended OptiPNG, JpegOptim, jpegrescan, jpegtran, and Gifsicle.

It’s excellent for publishing images on the web (easily shrinks images “Saved for Web” in Photoshop) and also useful for making Mac and iPhone/iPad applications smaller (see Xcode warning).

How to use it
Simply drag’n’drop images or folders into the window! You can also drop files on ImageOptim’s Dock icon or launch it from shell scripts with: open -a ImageOptim.app "*.png".

If you install this system service you can run ImageOptim from Finder’s context menu.

More about PNG
My article “PNG that works” explains why ImageOptim removes gamma information and how to get even smaller PNG files with transparency that works in ancient versions of IE.

Download

PNGGauntlet
Smash PNGs for faster sites

Download for Windows

- Combines PNGOUT, OptiPNG, and DeflOpt to create the smallest PNGs
- No image quality is lost — only file size
- Converts JPG, GIF, TIFF, and BMP files to PNG
4 COMBINE CSS AND JS & Compress
http://css-tricks.com/one-two-three/
Why did you use jQuery?
rabble rabble rabble rabble.
• 31K minified and gzipped
• On the world’s fastest CDN
• Probably cached
Why did you use jQuery?

rabble rabble rabble.
GROUP
PINKY SWEAR
I PROMISE TO...

1. Turn on HTTP Compression!
   i.e. (GZip)

2. Cache stuff!
   server and client side

3. Losslessly optimize images!
   and sprite if you can

4. Combine CSS and JS!
   three, two, or one per page
THANKS

@chriscoyier