Inspired by

Apache Server Configs v2.14.0 | MIT License
https://github.com/h5bp/server-configs-apache

(!) Using `.htaccess` files slows down Apache, therefore, if you have
access to the main server configuration file (which is usually called
`.httpd.conf`), you should add this logic there.

https://httpd.apache.org/docs/current/howto/htaccess.html

# CROSS-ORIGIN RESOURCE SHARING (CORS)

# Allow cross-origin requests.

http://enable-cors.org/
http://www.w3.org/TR/cors/

#ifModule mod_headers.c
# Header set Access-Control-Allow-Origin "*
#endif

# Send the CORS header for images when browsers request it.

<IfModule mod_setenvif.c>
  <FilesMatch "\.\(bmp|cur|gif|ico|jpe?g|png|svgz?|webp\)\$">
    SetEnvIf Origin \." IS_CORS
    Header set Access-Control-Allow-Origin "*" env=IS_CORS
  </FilesMatch>
</IfModule>

# Cross-origin web fonts
# Allow cross-origin access to web fonts.

<IfModule mod_headers.c>
  <FilesMatch "\.\(eot|otf|tt[cf]|woff2\)\$">
    Header set Access-Control-Allow-Origin "*
  </FilesMatch>
</IfModule>

# Cross-origin resource timing
# Allow cross-origin access to the timing information for all resources.
# If a resource isn't served with a `Timing-Allow-Origin` header that
# would allow its timing information to be shared with the document,
# some of the attributes of the `PerformanceResourceTiming` object will
# be set to zero.
# http://www.w3.org/TR/resource-timing/
# http://www.stevensouders.com/blog/2014/08/21/resource-timing-practical-tips/

# <IfModule mod_headers.c>
#  Header set Timing-Allow-Origin: "*"
# </IfModule>

# .................................................................
# ERRORS
# .................................................................
# Custom error messages/pages

# Customize what Apache returns to the client in case of an error.
# https://httpd.apache.org/docs/current/mod/core.html#error_document

ErrorDocument 404 /404.html

# Error prevention

# Disable the pattern matching based on filenames.
# This setting prevents Apache from returning a 404 error as the result of a rewrite when the directory with the same name does not exist.
# https://httpd.apache.org/docs/current/content-negotiation.html#multi_views

Options MultiViews

# Force Internet Explorer 8/9/10 to render pages in the highest mode available in the various cases when it may not.
# https://hsivonen.fi/doctype/#ie8

# (!) Starting with Internet Explorer 11, document modes are deprecated.
# If your business still relies on older web apps and services that were designed for older versions of Internet Explorer, you might want to consider enabling `Enterprise Mode` throughout your company.
<IfModule mod_headers.c>

    Header set X-UA-Compatible "IE=edge"

    # `mod_headers` cannot match based on the content-type, however,
    # the `X-UA-Compatible` response header should be send only for
    # HTML documents and not for the other resources.

    <FilesMatch "\.
     (appcachel atomi bbwl bmp1 crxl cssl curl eotl f4[abpv]l flvl geojsonl gifl html col jpe?
     gl jsl jsonl(ld)?
     lm4[lav]| manifest| mapl mp4| oexl og[agv]| opusl otfl pdfl pngl rdfl rssl safariextzl svgz?
     lwfl topojsonl tt[cfl] ttxl vcardl vclf vttl webappl web[lmp]l webmanifestl woff2?[l xlocl xmlll xpi]$">
        Header unset X-UA-Compatible
    </FilesMatch>

</IfModule>

# ------------------------------
# Iframes cookies
# ------------------------------

# Allow cookies to be set from iframes in Internet Explorer.
#
# http://www.w3.org/TR/2000/CR-P3P-20001215/

# <IfModule mod_headers.c>
#    Header set P3P "policyref="/\w3c/p3p.xml"", CP="/IDC DSP COR ADM DEV\i TAi PSA PSD
# IVAi VDI CONi HIS OUR IND CNT""
# </IfModule>

# ##################################################################
# # MEDIA TYPES AND CHARACTER ENCODINGS
# ##################################################################
# Media types

Serve resources with the proper media types (f.k.a. MIME types).

https://www.iana.org/assignments/media-types/media-types.xhtml
https://httpd.apache.org/docs/current/mod/mod_mime.html#addtype

IfModule mod_mime.c

# Data interchange

AddType application/atom+xml atom
AddType application/json json map topojson
AddType application/ld+json jsonld
AddType application/rss+xml rss
AddType application/vnd.geo+json geojson
AddType application/xml rdf xml

# JavaScript

Normalize to standard type.
https://tools.ietf.org/html/rfc4329#section-7.2

AddType application/javascript js

# Manifest files

AddType application/manifest+json webmanifest
AddType application/x-web-app-manifest+json webapp
AddType text/cache-manifest appcache

# Media files

AddType audio/mp4 f4a f4b m4a
AddType audio/ogg oga ogg opus
AddType image/bmp bmp
AddType image/svg+xml svg svgz
AddType image/webp webp
AddType video/mp4 f4v f4p m4v mp4
AddType video/ogg ogv
AddType video/webm webm
AddType video/x-flv flv

# Serving `.ico` image files with a different media type
# prevents Internet Explorer from displaying them as images:
# https://github.com/h5bp/html5-boilerplate/commit/37b5fec090d00f38de64b591bcddcb205aaf8ee
AddType image/x-icon cur.ico

# Web fonts
AddType application/font-woff woff
AddType application/font-woff2 woff2
AddType application/vnd.ms-fontobject eot

# Browsers usually ignore the font media types and simply sniff
# the bytes to figure out the font type.
# https://mimesniff.spec.whatwg.org/#matching-a-font-type-pattern
#
# However, Blink and WebKit based browsers will show a warning
# in the console if the following font types are served with any
# other media types.
AddType application/x-font-ttf ttc ttf
AddType font/opentype otf

# Other
AddType application/octet-stream safariextz
AddType application/x-bb-appworld bbaw
AddType application/x-chrome-extension crx
AddType application/x-opera-extension oex
AddType application/x-xpinstall xpi
AddType text/vcard vcard vcf
AddType text/vnd.rim.location.xloc xloc
AddType text/vtt vtt
AddType text/x-component htc

</IfModule>
# Character encodings

Serve all resources labeled as `text/html` or `text/plain` with the media type `charset` parameter set to `UTF-8`.

AddDefaultCharset utf-8

Serve the following file types with the media type `charset` parameter set to `UTF-8`.

<IfModule mod_mime.c>
  AddCharset utf-8 .atom \
    .bbaw \
    .css \
    .geojson \
    .js \
    .json \
    .jsonld \
    .manifest \
    .rdf \
    .rss \
    .topojson \
    .vtt \
    .webapp \
    .webmanifest \
    .xloc \
    .xml
</IfModule>

# URL REWRITES
### Rewrite engine

# Turning on the rewrite engine and enabling the `FollowSymLinks` option is necessary for the following directives to work.

# If your web host doesn't allow the `FollowSymLinks` option, you may need to comment it out and use `Options +SymLinksIfOwnerMatch` but, be aware of the performance impact: http://httpd.apache.org/docs/current/misc/perf-tuning.html#symlinks

# Also, some cloud hosting services require `RewriteBase` to be set: http://www.rackspace.com/knowledge_center/frequently-asked-question/why-is-mod-rewrite-not-working-on-my-site

```html
<IfModule mod_rewrite.c>
  Options +FollowSymLinks
  # Options +SymLinksIfOwnerMatch
  RewriteEngine On
  RewriteBase /
</IfModule>
```

### Cleaning the domain at the beginning of URLs

# The same content should never be available under two different URLs especially not with and without "www." at the beginning. This can cause SEO problems (duplicate content), therefore, you should choose one of the alternatives and redirect the other one.

# Remove www.

RewriteCond %{HTTP_HOST} ^www\. [NC]
RewriteRule ^(.*) https://nicolas-hoizey.com/$1 [QSA,R=301,L]

# Redirect from .fr to .com

RewriteCond %{HTTP_HOST} \.fr$ [NC]
RewriteRule ^(.*) https://nicolas-hoizey.com/$1 [QSA,R=301,L]
# Redirect from domain without dash
RewriteCond %{HTTP_HOST} nicolas-hoiyey\..com$ [NC]
RewriteRule ^(.*) https://nicolas-hoiyey.com/$1 [QSA,R=301,L]

# Don't redirect Let's Encrypt ACME Challenge
# https://github.com/oncletom/letsencrypt-alwaysdata#http-redirects
RewriteCond %{REQUEST_URI} ^/.well-known/acme-challenge [NC]
RewriteRule .* - [L]

# Redirect from http to https
# RewriteCond specific to AlwaysData proxy
RewriteCond %{HTTP:X-Forwarded-Proto} !https
RewriteRule ^(.*) https://nicolas-hoiyey.com/$1 [QSA,R=301,L]

# Block access to directories without a default document.
# Usually you should leave this uncommented because you shouldn't allow anyone
# to surf through every directory on your server (which may includes rather
# private places like the CMS's directories).

<IfModule mod_autoindex.c>
  Options -Indexes
</IfModule>

# Block access to all hidden files and directories with the exception of
# the visible content from within the `/.well-known/` hidden directory.

# These types of files usually contain user preferences or the preserved
# state of an utility, and can include rather private places like, for
# example the `git` or `svn` directories
# The `./well-known/` directory represents the standard (RFC 5785) path
# prefix for "well-known locations" (e.g.: `./well-known/manifest.json`,
# `./well-known/keybase.txt`), and therefore, access to its visible
# content should not be blocked.
#
# https://www.mnot.net/blog/2010/04/07/well-known
# https://tools.ietf.org/html/rfc5785

<IfModule mod_rewrite.c>

RewriteEngine On
RewriteCond %{REQUEST_URI} "!(^/\.well-known/([^/]+/\.*))" [NC]
RewriteCond %{SCRIPT_FILENAME} -d [OR]
RewriteCond %{SCRIPT_FILENAME} -f
RewriteRule "(^/\.*") - [F]
</IfModule>

# Block access to files that can expose sensitive information.
#
# By default, block access to backup and source files that may be
# left by some text editors and can pose a security risk when anyone
# has access to them.
#
# http://feross.org/cmsploit/
#
# (!) Update the `<FilesMatch>` regular expression from below to
# include any files that might end up on your production server and
# can expose sensitive information about your website. These files may
# include: configuration files, files that contain metadata about the
# project (e.g.: project dependencies), build scripts, etc.

<FilesMatch "(^[#.]*#|\.bak|conf|dist|fail|ini|cli|log|psd|sh|sql|sw[op]|\")">  

# Apache < 2.3
<IfModule !mod_authz_core.c>

Order allow,deny
Deny from all
Satisfy All
</IfModule>
# Apache ≥ 2.3
<IfModule mod_authz_core.c>
    Require all denied
</IfModule>

</FilesMatch>

# -------------------
# Block access from libwww-perl bots

<IfModule mod_rewrite.c>
    RewriteCond %{HTTP_USER_AGENT} libwww-perl.*
    RewriteRule .* - [F,L]
</IfModule>

# -------------------
# Content Security Policy (CSP)
# from https://github.com/h5bp/server-configs-apache/blob/master/src/security/content-security-policy.conf
# Mitigate the risk of cross-site scripting and other content-injection attacks.
# This can be done by setting a `Content Security Policy` which whitelists trusted sources of content for your website.
# The example header below allows ONLY scripts that are loaded from the current website's origin (no inline scripts, no CDN, etc).
# That almost certainly won't work as-is for your website!
# To make things easier, you can use an online CSP header generator such as: http://cspisawesome.com/.
# http://content-security-policy.com/
# http://www.w3.org/TR/CSP11/).
<IfModule mod_headers.c>

https://links.services.disqus.com wss://realtime.services.disqus.com *.algolia.net
*.algolianet.com; report-uri https://nhocom.report-uri.io/r/default/csp/enforce"

    # `mod_headers` cannot match based on the content-type, however,
    # the `Content-Security-Policy` response header should be send
    # only for HTML documents and not for the other resources.

    <FilesMatch "\.
(appcachef atoml bbawl bml crxl cssl curl eot f4[abpy]l f1vl geojsonl gifl hlci icol jpe?
gl js json(1d)?
1m4[lav]l manifestl map1 mp4l oexl og[agv]l opusl otfl pdfl pngl rdfl rssl safariextzl svgz?
1swfl topojsonl tt[cf]l txtl vcardl vcf l vttl webappl web[mp]l webmanifestl woff2?! xloc1 xml1 xpi)"">
        Header unset Content-Security-Policy
    </FilesMatch>

</IfModule>

# -----------------------------------------------
# / X-Frame-Options
# -----------------------------------------------

# from https://developer.mozilla.org/en-US/docs/Web/HTTP/X-Frame-Options
# The X-Frame-Options HTTP response header can be used to indicate whether or
# not a browser should be allowed to render a page in a <frame>, <iframe> or
# <object>. Sites can use this to avoid clickjacking attacks, by ensuring that
# their content is not embedded into other sites.

<IfModule mod_headers.c>
    Header always append X-Frame-Options SAMEORIGIN
</IfModule>

# -----------------------------------------------
# / X-Xss-Protection
# -----------------------------------------------
# Used to configure the built in reflective XSS protection found in Internet
# Explorer, Chrome and Safari (Webkit). Valid settings for the header are 0,
# which disables the protection, 1 which enables the protection and
# 1; mode=block which tells the browser to block the response if it detects an
# attack rather than sanitising the script.

<IfModule mod_headers.c>
    Header always set X-Xss-Protection "1; mode=block"
</IfModule>

# ---------------------------------------------------------------------------
# ! X-Content-Type-Options
# ---------------------------------------------------------------------------

# Prevents Google Chrome and Internet Explorer from trying to mime-sniff the
# content-type of a response away from the one being declared by the server.
# Reduces exposure to drive-by downloads and the risks of user uploaded content
# that, with clever naming, could be treated as a different content-type,
# like an executable.

<IfModule mod_headers.c>
    Header always set X-Content-Type-Options "nosniff"
</IfModule>

# ##################################################################
# # WEB PERFORMANCE   #
# ##################################################################

# ---------------------------------------------------------------------------
# ! Compression
# ---------------------------------------------------------------------------

<IfModule mod_deflate.c>
    # Force compression for mangled `Accept-Encoding` request headers
</IfModule>

<IfModule mod_setenvif.c>
    <IfModule mod_headers.c>
        SetEnvIfNoCase ^((Accept-Encoding)X-cept-Encoding)X{15}I{15}I-{15})$
compress all output labeled with one of the following media types.

# For Apache versions below version 2.3.7 you don't need to
# enable `mod_filter` and can remove the `<IfModule mod_filter.c>`
# and `</IfModule>` lines as `AddOutputFilterByType` is still in
# the core directives.

# https://httpd.apache.org/docs/current/mod/mod_filter.html#addoutputfilterbytype

<IfModule mod_filter.c>
  AddOutputFilterByType DEFLATE "application/atom+xml" \
  "application/javascript" \
  "application/json" \
  "application/ld+json" \
  "application/manifest+json" \
  "application/rdf+xml" \
  "application/rss+xml" \
  "application/schema+json" \
  "application/vnd.geo+json" \
  "application/vnd.ms-fontobject" \
  "application/x-font-ttf" \
  "application/x-javascript" \
  "application/x-web-app-manifest+json" \
  "application/xhtml+xml" \
  "application/xml" \
  "font/eot" \
  "font/opentype" \
  "image/bmp" \
  "image/svg+xml" \
  "image/vnd.microsoft.icon" \
  "image/x-icon" \
  "text/cache-manifest" \
  "text/css" \
  "text/html" \
</IfModule>
"text/javascript" \
"text/plain" \
"text/vcard" \
"text/vnd.rim.location.xloc" \
"text/vtt" \
"text/x-component" \
"text/x-cross-domain-policy" \
"text/xml"

</IfModule>

# Map the following filename extensions to the specified 
# encoding type in order to make Apache serve the file types
# with the appropriate `Content-Encoding` response header
# (do note that this will NOT make Apache compress them!).
# # If these file types would be served without an appropriate
# `Content-Encode` response header, client applications (e.g.: 
# browsers) wouldn't know that they first need to uncompress
# the response, and thus, wouldn't be able to understand the
# content.
# # https://httpd.apache.org/docs/current/mod/mod_mime.html#addencoding

<IfModule mod_mime.c>
    AddEncoding gzip svgz
</IfModule>

</IfModule>

# ---------------------------
# / ETags                        /
# ---------------------------

# Remove `ETags` as resources are sent with far-future expires headers.
# # https://developer.yahoo.com/performance/rules.html#etags
# https://tools.ietf.org/html/rfc7232#section-2.3
# `FileETag None` doesn't work in all cases.

```xml
<IfModule mod_headers.c>
  Header unset ETag
</IfModule>
```

FileETag None

```bash
# ┌───────────────────────────────────────────────────────────────────────────────────────────
# | / Expires headers
# └───────────────────────────────────────────────────────────────────────────────────────────

# Serve resources with far-future expires headers.
#
# (!) If you don't control versioning with filename-based
# cache busting, you should consider lowering the cache times
# to something like one week.
#
# https://httpd.apache.org/docs/current/mod/mod_expires.html

<IfModule mod_expires.c>

  ExpiresActive on
  ExpiresDefault         "access plus 1 month"

# CSS

  ExpiresByType text/css  "access plus 1 year"

# Data interchange

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

  ExpiresByType application/json          "access plus 0 seconds"
  ExpiresByType application/ld+json       "access plus 0 seconds"
  ExpiresByType application/schema+json   "access plus 0 seconds"
  ExpiresByType application/vnd.geo+json  "access plus 0 seconds"
  ExpiresByType application/xml           "access plus 0 seconds"
  ExpiresByType text/xml                 "access plus 0 seconds"
```
# Favicon (cannot be renamed!) and cursor images

ExpiresByType image/vnd.microsoft.icon  "access plus 1 week"
ExpiresByType image/x-icon  "access plus 1 week"

# HTML

ExpiresByType text/html  "access plus 0 seconds"

# JavaScript

ExpiresByType application/javascript  "access plus 1 year"
ExpiresByType application/x-javascript  "access plus 1 year"
ExpiresByType text/javascript  "access plus 1 year"

# Manifest files

ExpiresByType application/manifest+json  "access plus 1 week"
ExpiresByType application/x-web-app-manifest+json  "access plus 0 seconds"
ExpiresByType text/cache-manifest  "access plus 0 seconds"

# Media files

ExpiresByType audio/ogg  "access plus 1 month"
ExpiresByType image/bmp  "access plus 1 month"
ExpiresByType image/gif  "access plus 1 month"
ExpiresByType image/jpeg  "access plus 1 month"
ExpiresByType image/png  "access plus 1 month"
ExpiresByType image/svg+xml  "access plus 1 month"
ExpiresByType image/webp  "access plus 1 month"
ExpiresByType video/mp4  "access plus 1 month"
ExpiresByType video/ogg  "access plus 1 month"
ExpiresByType video/webm  "access plus 1 month"

# Web fonts

# Embedded OpenType (EOT)

ExpiresByType application/vnd.ms-fontobject  "access plus 1 month"
ExpiresByType font/eot  "access plus 1 month"

# OpenType
ExpiresByType font/opentype "access plus 1 month"

# TrueType
ExpiresByType application/x-font-ttf "access plus 1 month"

# Web Open Font Format (WOFF) 1.0
ExpiresByType application/font-woff "access plus 1 month"
ExpiresByType application/x-font-woff "access plus 1 month"
ExpiresByType font/woff "access plus 1 month"

# Web Open Font Format (WOFF) 2.0
ExpiresByType application/font-woff2 "access plus 1 month"

# Other
ExpiresByType text/x-cross-domain-policy "access plus 1 week"

</IfModule>

# ##############################################################################
# # REDIRECTS                     
# ##############################################################################

# / Redirecting from old to current pagination pages
# / Redirect old paginations
RewriteRule ^([0-9]{1,2})/?$ /page-$1/ [QSA,R=301,L]

# Remove index.html
# -----------------------------------------------

# If it's a request to index(.html)
RewriteCond %{THE_REQUEST} \/(.+)/index\(.html\)?\(\?\.*\)? [NC]
# Remove it.
RewriteRule ^(.+)?index\(.html\)?$ /%1 [R=301,L]

# -----------------------------------------------
# Remove trailing double slashes //


# Remove trailing double slashes

# -----------------------------------------------------------------------------

RewriteCond %{REQUEST_URI} ^(.*)//(/.*)$
RewriteRule . %1/%2 [R=301,L]

# -----------------------------------------------------------------------------

# Put well known resources in the .well-known folder
# -----------------------------------------------------------------------------

RewriteCond %{REQUEST_FILENAME} !-f

# Some very common, then less common "well-known" patterns
RewriteCond %{REQUEST_URI} =/favicon.jpg [OR]
RewriteCond %{REQUEST_URI} =/favicon.ico [OR]
RewriteCond %{REQUEST_URI} =/robots.txt [OR]
RewriteCond %{REQUEST_URI} =/crossdomain.xml [OR]
RewriteCond %{REQUEST_URI} =/apple-touch-icon.png [OR]
RewriteCond %{REQUEST_URI} =/apple-touch-icon-precomposed.png [OR]
RewriteCond %{REQUEST_URI} =/foaf.rdf [OR]
RewriteCond %{REQUEST_URI} =/w3c/p3p.xml [OR]
RewriteCond %{REQUEST_URI} ^=/google\[da-f]{16}\..html$ [OR]
RewriteCond %{REQUEST_URI} ^=/y_key_[\da-f]{16}\..html$ [OR]
RewriteCond %{REQUEST_URI} =/BingSiteAuth.xml$ [OR]
RewriteCond %{REQUEST_URI} =/myopenid-hosted-verification.html [OR]
RewriteCond %{REQUEST_URI} =/humans.txt

# To further limit possible issues (e.g. if you mess up the conditions above),
# we only match files at the root, not folders or files in subfolders
RewriteRule ^([a-zA-Z0-9-\-\.]+)$ /.well-known/$1 [L,NC]

# -----------------------------------------------------------------------------

# Renamed or moved articles and pages
# -----------------------------------------------------------------------------


RewriteRule ^a-propos.html /a-propos/de-moi.html [QSA,L,R=301]
RewriteRule ^a-propos/\?/? /a-propos/de-moi.html [QSA,L,R=301]