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

# ######################################################################
# # 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#errordocument

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#multiviews

Options -MultiViews

# ######################################################################
# # INTERNET EXPLORER                                                  #
# ######################################################################

# ----------------------------------------------------------------------
# | Document modes                                                     |
# ----------------------------------------------------------------------

# 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
```

```
# https://nsivonen.fi/doctype/#ieo
#
# (!) Starting with Internet Explorer 11, document modes are deprecated.
#
# https://msdn.microsoft.com/en-us/library/ie/bg182625.aspx#docmode
# http://blogs.msdn.com/b/ie/archive/2014/04/02/stay-up-to-date-with-enterprise-mode-for-
internet-explorer-11.aspx

<IfModule mod_headers.c>

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

    <FilesMatch "\.
(appcache|atom|bbaw|bmp|crx|css|curl|eot|f4[abpv]|flv|geojson|gif|htc|ico|jpe?
g|js|json(ld)?
|m4[av]|manifest|map|mp4|oex|og[agv]|opus|otf|pdf|png|rdf|rss|safariextz|svgz?
|swf|topojson|tt[cf]|txt|vcard|vcf|vtt|webapp|web[mp]|woff2?|xloc|xml|xpi)$">
        Header unset X-UA-Compatible
    </FilesMatch>

</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/json                            json map
    AddType application/xml                             atom rdf rss xml
```

```
  # JavaScript
    AddType application/javascript                  js

  # Media files
    AddType image/svg+xml                           svg svgz
    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

</IfModule>

# ----------------------------------------------------------------------
# | Character encodings                                                |
# ----------------------------------------------------------------------

# Serve all resources labeled as `text/html` or `text/plain`
# with the media type `charset` parameter set to `UTF-8`.
#
# https://httpd.apache.org/docs/current/mod/core.html#adddefaultcharset

AddDefaultCharset utf-8

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

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

```
# https://httpd.apache.org/docs/current/mod/mod_mime.html#addcharset

<IfModule mod_mime.c>
    AddCharset utf-8 .atom \
                     .css \
                     .js \
                     .json \
                     .rdf \
                     .rss \
                     .xml
</IfModule>

# ######################################################################
# # REWRITES                                                           #
# ######################################################################

# ----------------------------------------------------------------------
# | Rewrite engine                                                     |
# ----------------------------------------------------------------------

# (1) Turn on the rewrite engine (this is necessary in order for
#     the `RewriteRule` directives to work).
#     https://httpd.apache.org/docs/current/mod/mod_rewrite.html#RewriteEngine
#
# (2) Enable the `FollowSymLinks` option if it isn't already.
#     https://httpd.apache.org/docs/current/mod/core.html#options

<IfModule mod_rewrite.c>

    # (1)
    RewriteEngine On

    # (2)
    Options +FollowSymlinks

</IfModule>

# ----------------------------------------------------------------------
# | Suppressing the `www.` at the beginning of URLs                    |
# ----------------------------------------------------------------------
```

```
# Rewrite `www.example.com` → `example.com`

<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteCond %{HTTPS} !=on
    RewriteCond %{HTTP_HOST} ^www\.(.+)$ [NC]
    RewriteRule ^ http://%1%{REQUEST_URI} [R=301,L]
</IfModule>

# ######################################################################
# # SECURITY                                                           #
# ######################################################################

# ----------------------------------------------------------------------
# | Clickjacking                                                       |
# ----------------------------------------------------------------------

# Protect website against clickjacking and other types of attacks by
# informing browsers not to display the web page content in any frame.
#
# https://cure53.de/xfo-clickjacking.pdf.
# https://tools.ietf.org/html/rfc7034
# http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-
frame-options.aspx
# https://www.owasp.org/index.php/Clickjacking

<IfModule mod_headers.c>

    Header set X-Frame-Options "DENY"

    <FilesMatch "\.
(appcache|atom|bbaw|bmp|crx|css|curl|eot|f4[abpv]|flv|geojson|gif|htc|ico|jpe?
g|js|json(ld)?
|m4[av]|manifest|map|mp4|oex|og[agv]|opus|otf|pdf|png|rdf|rss|safariextz|svgz?
|swf|topojson|tt[cf]|txt|vcard|vcf|vtt|webapp|web[mp]|woff2?|xloc|xml|xpi)$">
        Header unset X-Frame-Options
    </FilesMatch>

</IfModule>

# ----------------------------------------------------------------------
```

```
# | Content Security Policy (CSP)                                        |
# ------------------------------------------------------------------------

# Mitigate the risk of cross-site scripting and other content-injection
# attacks.
#
# http://www.html5rocks.com/en/tutorials/security/content-security-policy/
# http://www.w3.org/TR/CSP11/

<IfModule mod_headers.c>

    # All HTML pages
    Header set Content-Security-Policy "\
default-src 'none';\
font-src fonts.gstatic.com;\
frame-src platform.twitter.com;\
img-src data: i.ytimg.com www.google-analytics.com;\
script-src 'unsafe-inline' www.google-analytics.com ajax.googleapis.com
platform.twitter.com;\
style-src 'unsafe-inline' fonts.googleapis.com"

    # The 404 page
    <FilesMatch "404.html">
        # Replace previous set header
        Header set Content-Security-Policy "\
default-src 'none';\
style-src 'unsafe-inline'"
    </FilesMatch>

    <FilesMatch "\.(appcache|atom|crx|css|curl|eot|f4[abpv]|flv|gif|htc|ico|jpe?
g|js|json(ld)?
|m4[av]|manifest|map|mp4|oex|og[agv]|opus|otf|pdf|png|rdf|rss|safariextz|svgz?
|swf|tt[cf]|vcf|vtt|webapp|web[mp]|woff|xml|xpi)$">
        Header unset Content-Security-Policy
    </FilesMatch>

</IfModule>

# ------------------------------------------------------------------------
# | File access                                                          |
# ------------------------------------------------------------------------
```

```
# Block access to directories without a default document.

<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.
#
# (!) 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|fla|in[ci]|log|psd|sh|sql|sw[op])|~)$">
    <IfModule !mod_authz_core.c>
        Order allow,deny
        Deny from all
        Satisfy All
    </IfModule>
</FilesMatch>

# ----------------------------------------------------------------------
# | HTTP Strict Transport Security (HSTS)                              |
# ----------------------------------------------------------------------
#
# Force client-side SSL redirection.
#
# http://www.html5rocks.com/en/tutorials/security/transport-layer-security/
# https://tools.ietf.org/html/draft-ietf-websec-strict-transport-sec-14#section-6.1
# http://blogs.msdn.com/b/ieinternals/archive/2014/08/18/hsts-strict-transport-security-
attacks-mitigations-deployment-https.aspx

<IfModule mod_headers.c>
    Header always set Strict-Transport-Security "max-age=16070400"
</IfModule>

# ----------------------------------------------------------------------
# | Reducing MIME type security risks                                  |
# ----------------------------------------------------------------------

# Prevent some browsers from MIME-sniffing the response.
#
# http://www.slideshare.net/hasegawayosuke/owasp-hasegawa
# http://blogs.msdn.com/b/ie/archive/2008/07/02/ie8-security-part-v-comprehensive-
protection.aspx
# https://msdn.microsoft.com/en-us/library/ie/gg622941.aspx
# https://mimesniff.spec.whatwg.org/

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

# ----------------------------------------------------------------------
```

```
# ----------------------------------------------------------------------
# | Reflected Cross-Site Scripting (XSS) attacks                       |
# ----------------------------------------------------------------------

# (1) Try to re-enable the cross-site scripting (XSS) filter built
#     into most web browsers.
#
#     The filter is usually enabled by default, but in some cases it
#     may be disabled by the user. However, in Internet Explorer for
#     example, it can be re-enabled just by sending the
#     `X-XSS-Protection` header with the value of `1`.
#
# (2) Prevent web browsers from rendering the web page if a potential
#     reflected (a.k.a non-persistent) XSS attack is detected by the
#     filter.
#
#     By default, if the filter is enabled and browsers detect a
#     reflected XSS attack, they will attempt to block the attack
#     by making the smallest possible modifications to the returned
#     web page.
#
#     Unfortunately, in some browsers (e.g.: Internet Explorer),
#     this default behavior may allow the XSS filter to be exploited,
#     thereby, it's better to inform browsers to prevent the rendering
#     of the page altogether, instead of attempting to modify it.
#
#     https://hackademix.net/2009/11/21/ies-xss-filter-creates-xss-vulnerabilities
#
# (!) Do not rely on the XSS filter to prevent XSS attacks! Ensure that
#     you are taking all possible measures to prevent XSS attacks, the
#     most obvious being: validating and sanitizing your website's inputs.
#
# http://blogs.msdn.com/b/ie/archive/2008/07/02/ie8-security-part-iv-the-xss-filter.aspx
# http://blogs.msdn.com/b/ieinternals/archive/2011/01/31/controlling-the-internet-
explorer-xss-filter-with-the-x-xss-protection-http-header.aspx
# https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29

<IfModule mod_headers.c>

    #                            (1)    (2)
    Header set X-XSS-Protection "1; mode=block"
```

```
    <FilesMatch "\.
(appcache|atom|bbaw|bmp|crx|css|curl|eot|f4[abpv]|flv|geojson|gif|htc|ico|jpe?
g|js|json(ld)?
|m4[av]|manifest|map|mp4|oex|og[agv]|opus|otf|pdf|png|rdf|rss|safariextz|svgz?
|swf|topojson|tt[cf]|txt|vcard|vcf|vtt|webapp|web[mp]|woff2?|xloc|xml|xpi)$">
        Header unset X-XSS-Protection
    </FilesMatch>

</IfModule>

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

# ----------------------------------------------------------------------
# | Compression                                                        |
# ----------------------------------------------------------------------

<IfModule mod_deflate.c>

    # Force compression for mangled `Accept-Encoding` request headers
    # https://developer.yahoo.com/blogs/ydn/pushing-beyond-gzipping-25601.html

    <IfModule mod_setenvif.c>
        <IfModule mod_headers.c>
            SetEnvIfNoCase ^(Accept-EncodXng|X-cept-Encoding|X{15}|~{15}|-{15})$
^((gzip|deflate)\s*,?\s*)+|[X~-]{4,13}$ HAVE_Accept-Encoding
            RequestHeader append Accept-Encoding "gzip,deflate" env=HAVE_Accept-Encoding
        </IfModule>
    </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 files types would be served without an appropriate
    # `Content-Enable` response header, client applications (e.g.:
    # browsers) wouldn't know that they first need to uncompress
```

```
# 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              gz
    AddEncoding gzip              svgz
</IfModule>

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

# Make Apache serve the Zopfli compressed version of the requested
# file if it exists on the server and the browser supports `gzip`
# compression

<IfModule mod_rewrite.c>
    RewriteEngine On
    ReWriteCond %{HTTP:accept-encoding} gzip
    ReWriteCond %{REQUEST_FILENAME} !\.gz$
    RewriteCond %{REQUEST_FILENAME}.gz -f
    RewriteRule ^(.+)\.(css|html|ico|js|svg|txt|xml)$ $1.$2.gz [L]
</IfModule>

<FilesMatch "\.css\.gz$">
    ForceType text/css
</FilesMatch>

<FilesMatch "\.html\.gz$">
    ForceType text/html
</FilesMatch>

<FilesMatch "\.ico\.gz$">
    ForceType image/x-icon
</FilesMatch>

<FilesMatch "\.js\.gz$">
    ForceType application/javascript
</FilesMatch>
```

```
    <FilesMatch "\.svg\.gz$">
        ForceType image/svg+xml
    </FilesMatch>

    <FilesMatch "\.txt\.gz$">
        ForceType text/plain
    </FilesMatch>

    <FilesMatch "\.xml\.gz$">
        ForceType application/xml
    </FilesMatch>

    <IfModule mod_headers.c>
        <FilesMatch "\.(css|html|ico|js|svg|txt|xml)\.gz$">
            Header merge Vary "Accept-Encoding"
        </FilesMatch>
    </IfModule>

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

    # Compress all output labeled with one of the following media types.
    # https://httpd.apache.org/docs/current/mod/mod_filter.html#addoutputfilterbytype

    AddOutputFilterByType DEFLATE "application/javascript" \
                                  "application/json" \
                                  "application/manifest+json" \
                                  "application/vnd.ms-fontobject" \
                                  "application/x-font-ttf" \
                                  "application/xml" \
                                  "font/opentype" \
                                  "image/svg+xml" \
                                  "image/vnd.microsoft.icon" \
                                  "text/css" \
                                  "text/html" \
                                  "text/plain"

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

FileETag None

# ----------------------------------------------------------------------
# | Expires headers                                                    |
# ----------------------------------------------------------------------

# Serve resources with far-future expires headers.
# 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/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/x-icon                    "access plus 1 week"

  # HTML
    ExpiresByType text/html                       "access plus 1 hour"

  # JavaScript
    ExpiresByType application/javascript          "access plus 1 year"
```

```
</IfModule>
```