

```

<Files ~ "^.*\.([Hh][Tt][Aa])">
    order allow,deny
    deny from all
    satisfy all
</Files>

<Files ~ "\.(html|html|css|js|php)$">
    AddDefaultCharset UTF-8
    DefaultLanguage en-US
</Files>

<IfModule mod_rewrite.c>
    DirectoryIndex index.php
    RewriteEngine on
    RewriteCond $1 !^(index\.php|images|css|js|robots\.txt|favicon\.ico)
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule ^(.*)$ ./index.php/$1 [L,QSA]
</IfModule>

```

Apache Configuration File

(!) Using '.htaccess' files slows down Apache, therefore, if you have access to the main server config file (usually called 'httpd.conf'), you should add this logic there: <http://httpd.apache.org/docs/current/howto/htaccess.html>.

```

# #####
# # CROSS-ORIGIN RESOURCE SHARING (CORS) #
# #####

# -----
# | Cross-domain AJAX requests |
# -----

# Enable cross-origin AJAX requests.
# http://code.google.com/p/html5security/wiki/CrossOriginRequestSecurity
# http://enable-cors.org/

# <IfModule mod_headers.c>
#     Header set Access-Control-Allow-Origin "*"
# </IfModule>

```

```

# </IfModule>

# -----
# | CORS-enabled images |
# -----

# Send the CORS header for images when browsers request it.
# https://developer.mozilla.org/en/CORS_Enabled_Image
# http://blog.chromium.org/2011/07/using-cross-domain-images-in-webgl-and.html
# http://hacks.mozilla.org/2011/11/using-cors-to-load-webgl-textures-from-cross-domain-images/

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

# -----
# | Web fonts access |
# -----

# Allow access from all domains for web fonts

<IfModule mod_headers.c>
    <FilesMatch "\.(eot|font.css|otf|ttf|woff)$">
        Header set Access-Control-Allow-Origin "*"
    </FilesMatch>
</IfModule>

# #####
# # ERRORS #
# #####

# -----
# | 404 error prevention for non-existing redirected folders |
# -----

# Prevent Apache from returning a 404 error for a rewrite if a directory
# exists and is not a directory

```

```
# with the same name does not exist.  
# http://httpd.apache.org/docs/current/content-negotiation.html#multiviews  
# http://www.webmasterworld.com/apache/3808792.htm
```

Options -MultiViews

```
# -----  
# | Custom error messages / pages |  
# -----  
  
# You can customize what Apache returns to the client in case of an error (see  
# http://httpd.apache.org/docs/current/mod/core.html#errordocument), e.g.:
```

ErrorDocument 404 /404.html

```
# #####  
# # INTERNET EXPLORER #  
# #####  
  
# -----  
# | Better website experience |  
# -----  
  
# Force IE to render pages in the highest available mode in the various  
# cases when it may not: http://hsivonen.iki.fi/doctype/ie-mode.pdf.  
# Use, if installed, Google Chrome Frame.
```

```
<IfModule mod_headers.c>
```

```
Header set X-UA-Compatible "IE=edge,chrome=1"
```

```
# `mod_headers` can't match based on the content-type, however, we only  
# want to send this header for HTML pages and not for the other resources
```

```
<FilesMatch "\.(appache|crx|css|eot|gif|html|ico|jpe?
```

```
gl|js|m4a|m4v|manifest|mp4|oex|ogal|oggl|ogv|otf|pdf|png|safariextz|svg|svgz|ttf|vcf|webapp|wet
```

```
Header unset X-UA-Compatible
```

```
</FilesMatch>
```

```
</IfModule>
```

```
# -----  
# | Cookie setting from iframes |  
# -----
```

```

# Allow cookies to be set from iframes in IE.

# <IfModule mod_headers.c>
#   Header set P3P "policyref=\"/w3c/p3p.xml\"", CP=\"IDC DSP COR ADM DEVi TAIi PSA PSD IVAi
CNT\"""
# </IfModule>

# -----
# | Screen flicker |
# -----

# Stop screen flicker in IE on CSS rollovers (this only works in
# combination with the `ExpiresByType` directives for images from below).

# BrowserMatch "MSIE" brokenvary=1
# BrowserMatch "Mozilla/4.[0-9]{2}" brokenvary=1
# BrowserMatch "Opera" !brokenvary
# SetEnvIf brokenvary 1 force-no-vary

# #####
# # MIME TYPES AND ENCODING #
# #####

# -----
# | Proper MIME types for all files |
# -----

<IfModule mod_mime.c>

# Audio
AddType audio/mp4 m4a f4a f4b
AddType audio/ogg oga ogg

# JavaScript
# Normalize to standard type (it's sniffed in IE anyways):
# http://tools.ietf.org/html/rfc4329#section-7.2
AddType application/javascript js jsonp
AddType application/json json

# Video
AddType video/mp4 mp4 m4v f4v f4p

```

```

AddType video/ogg                ogv
AddType video/webm               webm
AddType video/x-flv              flv

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

# Browsers usually ignore the font MIME types and sniff the content,
# however, Chrome shows a warning if other MIME types are used for the
# following fonts.
AddType application/x-font-ttf   ttc ttf
AddType font/opentype            otf

# Make SVGZ fonts work on iPad:
# https://twitter.com/FontSquirrel/status/14855840545
AddType      image/svg+xml       svg svgz
AddEncoding  gzip                svgz

# Other
AddType application/octet-stream safariextz
AddType application/x-chrome-extension crx
AddType application/x-opera-extension oex
AddType application/x-shockwave-flash swf
AddType application/x-web-app-manifest+json webapp
AddType application/x-xpinstall xpi
AddType application/xml          atom rdf rss xml
AddType image/webp               webp
AddType image/x-icon             ico
AddType text/cache-manifest      appcache manifest
AddType text/vtt                 vtt
AddType text/x-component         htc
AddType text/x-vcard             vcf

</IfModule>

# -----
# | UTF-8 encoding |
# -----

# Use UTF-8 encoding for anything served as `text/html` or `text/plain`.

```

```
AddDefaultCharset utf-8
```

```
# Force UTF-8 for certain file formats.
```

```
<IfModule mod_mime.c>
```

```
    AddCharset utf-8 .atom .css .js .json .rss .vtt .webapp .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
```

```
<IfModule mod_rewrite.c>
```

```
    Options +FollowSymLinks
```

```
    # Options +SymLinksIfOwnerMatch
```

```
    RewriteEngine On
```

```
    # RewriteBase /
```

```
</IfModule>
```

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

```
# | Suppressing / Forcing the "www." 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.
```

```
# By default option 1 (no "www.") is activated:
# http://no-www.org/faq.php?q=class_b

# If you'd prefer to use option 2, just comment out all the lines from option 1
# and uncomment the ones from option 2.

# IMPORTANT: NEVER USE BOTH RULES AT THE SAME TIME!

# -----

# Option 1: rewrite www.example.com → example.com

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

# -----

# Option 2: rewrite example.com → www.example.com

# Be aware that the following might not be a good idea if you use "real"
# subdomains for certain parts of your website.

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

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

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

# You can mitigate the risk of cross-site scripting and other content-injection
# attacks by setting a Content Security Policy which whitelists trusted sources
```

of content for your site.

*# The example header below allows ONLY scripts that are loaded from the current
site's origin (no inline scripts, no CDN, etc). This almost certainly won't
work as-is for your site!*

*# To get all the details you'll need to craft a reasonable policy for your site,
read: <http://html5rocks.com/en/tutorials/security/content-security-policy> (or
see the specification: <http://w3.org/TR/CSP>).*

```
# <IfModule mod_headers.c>
#   Header set Content-Security-Policy "script-src 'self'; object-src 'self'"
#   <FilesMatch "\.(appcache|crx|css|eot|gif|html|ico|jpe?
gl|js|m4a|m4v|manifest|mp4|oex|ogal|ogg|ogv|otf|pdf|png|safariextz|svg|svgz|ttf|vcf|webapp|wel
#       Header unset Content-Security-Policy
#   </FilesMatch>
# </IfModule>
```

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

*# 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 hidden files and directories.
This includes directories used by version control systems such as Git and SVN.*

```
<IfModule mod_rewrite.c>
    RewriteCond %{SCRIPT_FILENAME} -d [OR]
    RewriteCond %{SCRIPT_FILENAME} -f
    RewriteRule "(^|/)\." - [F]
</IfModule>
```


.....

*# Block access to backup and source files.
These files may be left by some text editors and can pose a great security
danger when anyone has access to them.*

```
<FilesMatch "(^#.#|\. (bak| config| dist| fl| incl| ini| log| psd| sh| sql| sw[op])| (~)$">  
    Order allow,deny  
    Deny from all  
    Satisfy All  
</FilesMatch>
```

| Secure Sockets Layer (SSL) |

*# Rewrite secure requests properly to prevent SSL certificate warnings, e.g.:
prevent `https://www.example.com` when your certificate only allows
`https://secure.example.com`.*

```
# <IfModule mod_rewrite.c>  
#     RewriteCond %{SERVER_PORT} !^443  
#     RewriteRule ^ https://example-domain-please-change-me.com%{REQUEST_URI} [R=301,L]  
# </IfModule>
```

Force client-side SSL redirection.

*# If a user types "example.com" in his browser, the above rule will redirect him
to the secure version of the site. That still leaves a window of opportunity
(the initial HTTP connection) for an attacker to downgrade or redirect the
request. The following header ensures that browser will ONLY connect to your
server via HTTPS, regardless of what the users type in the address bar.
<http://www.html5rocks.com/en/tutorials/security/transport-layer-security/>*

```
# <IfModule mod_headers.c>  
#     Header set Strict-Transport-Security max-age=16070400;  
# </IfModule>
```

```

# -----
# | Server software information |
# -----

# Avoid displaying the exact Apache version number, the description of the
# generic OS-type and the information about Apache's compiled-in modules.

# ADD THIS DIRECTIVE IN THE `httpd.conf` AS IT WILL NOT WORK IN THE `.htaccess`!

# ServerTokens Prod

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

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

<IfModule mod_deflate.c>

    # Force compression for mangled headers.
    # http://developer.yahoo.com/blogs/ymdn/posts/2010/12/pushing-beyond-gzipping
    <IfModule mod_setenvif.c>
        <IfModule mod_headers.c>
            SetEnvIfNoCase ^(\Accept-Encoding|X-cept-Encoding|X(15)|^(15)|-(15))$ ^(gzip|def
(4,13)$ HAVE_Accept-Encoding
            RequestHeader append Accept-Encoding "gzip, deflate" env=HAVE_Accept-Encoding
        </IfModule>
    </IfModule>

    # Compress all output labeled with one of the following MIME-types
    # (for Apache versions below 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).
    <IfModule mod_filter.c>
        AddOutputFilterByType DEFLATE application/atom+xml \
            application/javascript \
            application/json \
            application/rss+xml \
            application/vnd.ms-fontobject \

```

```
application/vnd.ms-fontobject \
application/x-font-ttf \
application/x-web-app-manifest+json \
application/xhtml+xml \
application/xml \
font/opentype \
image/svg+xml \
image/x-icon \
text/css \
text/html \
text/plain \
text/x-component \
text/xml
```

```
</IfModule>
```

```
</IfModule>
```

```
# -----
# | Content transformations |
# -----
```

```
# Prevent some of the mobile network providers from modifying the content of
# your site: http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9.5.
```

```
# <IfModule mod_headers.c>
#   Header set Cache-Control "no-transform"
# </IfModule>
```

```
# -----
# | ETag removal |
# -----
```

```
# Since we're sending far-future expires headers (see below), ETags can
# be removed: http://developer.yahoo.com/performance/rules.html#etags.
```

```
# `FileETag None` is not enough for every server.
```

```
<IfModule mod_headers.c>
```

```
    Header unset ETag
```

```
</IfModule>
```

```
FileETag None
```

```

# -----
# | Expires headers (for better cache control) |
# -----

# The following expires headers are set pretty far in the future. If you don't
# control versioning with filename-based cache busting, consider lowering the
# cache time for resources like CSS and JS to something like 1 week.

<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!)
    ExpiresByType image/x-icon        "access plus 1 week"

    # HTML components (HTCs)
    ExpiresByType text/x-component    "access plus 1 month"

    # HTML
    ExpiresByType text/html            "access plus 0 seconds"

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

    # Manifest files
    ExpiresByType application/x-web-app-manifest+json    "access plus 0 seconds"
    ExpiresByType text/cache-manifest    "access plus 0 seconds"

    # Media
    ExpiresByType audio/ogg            "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 video/mp4	"access plus 1 month"
ExpiresByType video/ogg	"access plus 1 month"
ExpiresByType video/webm	"access plus 1 month"

Web feeds

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

Web fonts

ExpiresByType application/font-woff	"access plus 1 month"
ExpiresByType application/vnd.ms-fontobject	"access plus 1 month"
ExpiresByType application/x-font-ttf	"access plus 1 month"
ExpiresByType font/opentype	"access plus 1 month"
ExpiresByType image/svg+xml	"access plus 1 month"

</IfModule>

```
# -----  
# | Filename-based cache busting |  
# -----
```

*# If you're not using a build process to manage your filename version revving,
you might want to consider enabling the following directives to route all
requests such as `/css/style.12345.css` to `/css/style.css`.*

*# To understand why this is important and a better idea than `\*.css?v231`, read:
<http://stevesouders.com/blog/2008/08/23/revving-filenames-dont-use-querystring>*

```
# <IfModule mod_rewrite.c>  
#     RewriteCond %{REQUEST_FILENAME} !-f  
#     RewriteCond %{REQUEST_FILENAME} !-d  
#     RewriteRule ^(.+)%\.(\d+)\. (js|css|png|jpg|gif)$ $1. $3 [L]  
# </IfModule>
```

```
# -----  
# | File concatenation |  
# -----
```

Allow concatenation from within specific CSS and JS files, e.g.:

```

# Inside of `script.combined.js` you could have
#   <!--#include file="libs/jquery.js" -->
#   <!--#include file="plugins/jquery.idletimer.js" -->
# and they would be included into this single file.

# <IfModule mod_include.c>
#   <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>
# </IfModule>

# -----
# | Persistent connections |
# -----

# Allow multiple requests to be sent over the same TCP connection:
# http://httpd.apache.org/docs/current/en/mod/core.html#keepalive.

# Enable if you serve a lot of static content but, be aware of the
# possible disadvantages!

# <IfModule mod_headers.c>
#   Header set Connection Keep-Alive
# </IfModule>

```