

```

# Apache Server Configs v2.2.0 | MIT License
# https://github.com/h5bp/server-configs-apache

# (!) Using `.htaccess` files slows down Apache, therefore, if you have access
# to the main server 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 |
# -----

# Allow 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>

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

# Send the CORS header for images when browsers request it.
# https://developer.mozilla.org/en-US/docs/HTML/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 "\.(curl|gif|icol|jpe?g|png|svgz?|webp)$">
      SetEnvIf Origin ":" IS_CORS
      Header set Access-Control-Allow-Origin "*" env=IS_CORS
    </FilesMatch>
  </IfModule>
</IfModule>

```

```
# -----  
# / Web fonts access /  
# -----
```

```
# Allow access to web fonts from all domains.
```

```
<IfModule mod_headers.c>  
  <FilesMatch "\.(eot|otf|tt[cf]|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 as the result of a rewrite  
# when the 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 /  
# -----
```

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

ErrorDocument 404 /404.html

```
# #####  
# # INTERNET EXPLORER #  
# #####
```

```
# -----
# | Better website experience |
# -----

# Force Internet Explorer to render pages in the highest available mode
# in the various cases when it may not.
# http://hsivonen.iki.fi/doctype/ie-mode.pdf

<IfModule mod_headers.c>
    Header set X-UA-Compatible "IE=edge"
    # `mod_headers` cannot match based on the content-type, however, this
    # header should be send only for HTML pages and not for the other resources
    <FilesMatch "\.(appcache|atom|crx|css|curl|eot|f4[abpv]|flv|gif|htc|icol|jpe?g|jst|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 X-UA-Compatible
    </FilesMatch>
</IfModule>

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

# Allow cookies to be set from iframes in Internet Explorer.
# http://msdn.microsoft.com/en-us/library/ms537343.aspx
# 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 DEVI TAIi PSA PSD IVAi
CONi HIS OUR IND CNT\""
# </IfModule>

# #####
# # 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 opus

# Data interchange
AddType application/json          json map
AddType application/ld+json      jsonld

# JavaScript
# Normalize to standard type.
# http://tools.ietf.org/html/rfc4329#section-7.2
AddType application/javascript    js

# Video
AddType video/mp4                 f4v f4p m4v mp4
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 simply sniff the bytes
# to figure out the font type.
# http://mimesniff.spec.whatwg.org/#matching-a-font-type-pattern

# Chrome however, shows a warning if any 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 the iPad.
# https://twitter.com/FontSquirrel/status/14855840545
AddType image/svg+xml            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-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 cur
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 .jsonld .rss .vtt .webapp .xml
```

```
</IfModule>
```

```
# #####
# # URL REWRITES #
# #####
```

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

```
# Turn on the rewrite engine and enable the `FollowSymLinks` option (this is
# necessary in order 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
```

```
<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), and 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 would 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]

RewriteCond %{SERVER_ADDR} !=127.0.0.1

RewriteCond %{SERVER_ADDR} !=::1

RewriteRule ^ http://www.%{HTTP_HOST}%{REQUEST_URI} [R=301,L]

</IfModule>

*#####
SECURITY #
#####*

*-----
| Clickjacking |
-----*

Protect website against clickjacking.

*# The example below sends the `X-Frame-Options` response header with the value
`DENY`, informing browsers not to display the web page content in any frame.*

*# This might not be the best setting for everyone. You should read about the
other two possible values for `X-Frame-Options`: `SAMEORIGIN` & `ALLOW-FROM`.
<http://tools.ietf.org/html/rfc7034#section-2.1>*

*# Keep in mind that while you could send the `X-Frame-Options` header for all
of your site's pages, this has the potential downside that it forbids even
non-malicious framing of your content (e.g.: when users visit your site using
a Google Image Search results page).*

*# Nonetheless, you should ensure that you send the `X-Frame-Options` header for
all pages that allow a user to make a state changing operation (e.g: pages
that contain one-click purchase links, checkout or bank-transfer confirmation
pages, pages that make permanent configuration changes, etc.).*

Sending the `X-Frame-Options` header can also protect your website against

```
# more than just clickjacking attacks: https://cure53.de/xfo-clickjacking.pdf.

# http://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|crx|css|curl|eot|f4[abpv]|flv|gif|html|icol|jpe?g|jsl|json|j|
|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 X-Frame-Options
#     </FilesMatch>
# </IfModule>

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

# 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
# site's origin (no inline scripts, no CDN, etc). This almost certainly won't
# work as-is for your site!

# For more details on how to craft a reasonable policy for your site, read:
# http://html5rocks.com/en/tutorials/security/content-security-policy (or the
# specification: http://w3.org/TR/CSP). Also, to make things easier, you can
# use an online CSP header generator such as: http://cspisawesome.com/.

# <IfModule mod_headers.c>
#     Header set Content-Security-Policy "script-src 'self'; object-src 'self'"
#     <FilesMatch "\.(appcache|atom|crx|css|curl|eot|f4[abpv]|flv|gif|html|icol|jpe?g|jsl|json|j|
|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.
# You should leave the following uncommented, as you shouldn't allow anyone to
# surf through every directory on your server (which may includes rather private
# places such as 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 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/cmsexploit/

# IMPORTANT: 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|info|lib|local|sd|sh|sql|src|on|)\.($)">

```



```
# 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 tell browsers to prevent the rendering of the page altogether, instead
# of attempting to modify it.
#
# http://hackademix.net/2009/11/21/ies-xss-filter-creates-xss-vulnerabilities
#
# IMPORTANT: 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 site'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-;
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|crx|css|curl|eot|f4[abpv]|flv|gif|html|icol|jpe?g|js|json|j
|m4[av]|manifest|map|mp4|oex|og[agv]|opus|otf|pdf|png|rdf|rssl|safari-extend|svgz?
|swf|tt[cf]|vcf|vtt|webapp|web[mp]|woff|xml|xpi)$">
#         Header unset X-XSS-Protection
#     </FilesMatch>
# </IfModule>

# -----
# | Secure Sockets Layer (SSL) |
# -----
```

```
# Rewrite secure requests properly in order 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>

# -----
# | HTTP Strict Transport Security (HSTS) |
# -----

# 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://tools.ietf.org/html/draft-ietf-websec-strict-transport-sec-14#section-6.1
# http://www.html5rocks.com/en/tutorials/security/transport-layer-security/

# IMPORTANT: Remove the `includeSubDomains` optional directive if the subdomains
# are not using HTTPS.

# <IfModule mod_headers.c>
#   Header set Strict-Transport-Security "max-age=16070400; includeSubDomains"
# </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/ydn/posts/2010/12/pushing-beyond-gzipping
```

```
    <IfModule mod_setenvif.c>
```

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

```
            SetEnvIfNoCase ^(\Accept-EncodXngl X-cept-Encodingl X(15)l ~(15)l -(15))$
```

```
^((gzip|deflate)\s*, ?\s*)+ [X"-]{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/ld+json \
```

```
        application/rss+xml \
```

```
        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 mobile network providers from modifying the website's content.  
# http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9.5.
```

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

```
# -----  
# | ETags |  
# -----
```

```
# Remove `ETags` as resources are sent with far-future expires headers.  
# http://developer.yahoo.com/performance/rules.html#etags.
```

```
# `FileETag None` doesn't work in all cases.
```

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

FileETag None

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

```
# 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 such as style sheets and JavaScript
# files to something like one 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/ld+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 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 revwing,
# 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/revwing-filenames-dont-use-querystring

```

```

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

```

```

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

```

```
# Allow concatenation from within specific style sheets and JavaScript files.
```

```

# e.g.:
#
# If you have the following content in a file
#
#   <!--#include file="libs/jquery.js" -->

```



```

#      <!--#include file="plugins/jquery.timer.js" -->
#
#   Apache will replace it with the content from the specified files.

# <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>

# #####
# # MOBILE SPECIFIC #
# #####

# Proper MIME types

<IfModule mod_mime.c>

# Blackberry
# http://docs.blackberry.com/en/developers/deliverables/18169/
AddType application/x-bb-appworld          bbaw
AddType text/vnd.rim.location.xloc         xloc

# Nokia
#
# http://www.developer.nokia.com/Community/Wiki/Apache\_configuration\_for\_mobile\_application\_development
#
# http://wiki.forum.nokia.com/index.php/How\_to\_enable\_OTTA\_\(Over\_The\_Air\)\_SIS\_install\_from\_your\_device
AddType application/octet-stream          six
AddType application/vnd.symbian.install   sis

</IfModule>

# -----

```

```
..

# Prevent mobile transcoding

# <FilesMatch "\.(php|cgi|pl)$">
#     <IfModule mod_headers.c>
#         Header append Cache-Control "no-transform"
#         Header append Vary "User-Agent, Accept"
#     </IfModule>
# </FilesMatch>

# -----

# Mobile Redirection Script is used to detect if user is viewing the site from
# mobile device. If the script detects the user is viewing from mobile phone,
# they will be redirected to the mobile version of the site. One thing to note
# is that if you want to allow the user on the mobile version of your site to
# have the option to switch to desktop version, you may consider using other
# methods like JavaScript or PHP at http://detectmobilebrowser.com/.
#
# To use the script, first, uncomment the lines below, and second, change
# 'http://www.example.com/mobile' to the URL of your mobile site.

# <IfModule mod_rewrite.c>
#     RewriteEngine On
#     RewriteBase /
#     RewriteCond %{HTTP_USER_AGENT}
# android|avantgo|blackberry|blazer|compal|elaine|fennec|hiptop|iemobile|ip(hone|od)|iris|kin
# |maemo|midp|mmp|opera\ m(ob|in)i|palm(\ os)?|phone|p(ixi|re)\v|plucker|pocket|psp|symbian|t(
# |browser|link)|vodafone|wap|windows\ (cel|phone)|xda|xiino [NC,OR]
# RewriteCond %{HTTP_USER_AGENT} ^(1207|6310|6590|3gso|4thp|50[1-6]i|770s|802s|a\
# wal|aba|ac(er|ool s\
# )|ai(kol rn)|al(avl cal co)|amoil an(exl nyl yw)|aptul ar(chl go)|as(tel us)|attw|au(di| \-ml r\ | s\
# )|avant|be(ckl |ll nq)|bi(1bl rd)|bl(ack az)|br(el v)w|bumbl bu\-(nl u)|c55\| capil ccual cdm\
# |cel|l|chtml|cl dcl cmd\|-|co(mpl nd)|crawl|da(it|ll|ng)|dbtel dc\|-|devil|di cal dmobl do(cl p)|ds(12|
# d)|e(49l ai)|em(12l ul)|er(i cl k0)|es|8l ez|[4-7]0|osl wal ze)|fetcl f1y(\|-_|)l g1\ |ul g560| genel gf
# mol go(\. wl od)|gr(adl un)|hai el hci ti hd\-(ml pl t)|hei\|-|hi(ptl ta)|hp(\ il ip)|hs\|-cl ht(cc\|-|
# |_)l al gl pl sl t)|tp|huc(awl tc)|i\-(20l gol ma)|i230|iac(\ | \-
# | \)|ibrol ideal ig0|l ikoml im1kl innol ipaq|iris|ja(cti v)al jbrol jemul jigs|kddil keji|kgt(\ | \)|k(
# | kuc\|-|kyo(cl k)|le(nol xi)|lg(\ gl \/(kl ll u)|50|54l e\|-e\| \-[a-w])|libul lynxl m1\
# |ul m3gal m50\| ma(tel uil xo)|mc(0|1|2| ca)|m\-crl me(dil rcl ri)|mi(o8l oal ts)|mnefl mo(0|1|02| bil del
# | cl v)|zz)|mt(50l nll v\ |)mubol mual n10f 0-011 020f 0-311 030f 0| 0y| 050f 0| 0l 5y| 070f 0| 1y| 10y| n0f(c|
```

```
l or\y2l 227l m6\30l p4l y\y 2l mw0p1 mgw0r n000-23l n2002-33l n3000l 27l n3000l 2l 32l n1\0\0l 127l 107l n10\0l  
l onl tfl wfl wgl wt)l nok(6l i)l nzphi o2iml op(til wv)l oranl owgll p800l pan(al dl t)l pdxgl pg(13l \-[1-  
8]l c)l phi1l p1rel p1(ayl uc)l pn\ -2l po(ckl rtl se)l proxl psi ol pt\ -gl qa\ -al qc(07l 12l 21l 32l 60l \-[2-  
)l qtek1 r380l r600l raksl rim9l ro(vel zo)l s55\ /l sa(ge1 mal mml msl nyl va)l sc(01l h\ -l ool p\ -)l sdk\ /l se  
l 0l 1)l 47l mcl ndl ri)l sgh\ -l sharl sie(\ -l m)l sk\ -0l sl(45l id)l sm(all arl b3l itl t5)l so(ftl ny)l sp(01l )l  
l v\ )l sy(01l mb)l t2(18l 50)l t6(00l 10l 18)l ta(gt1 k)l tc1\ -l tdg\ -l tel(il m)l tim\ -l t\ -  
mol to(p1l sh)l ts(70l m\ -l m3l m5)l tx\ -9l up(\. bl gl1 si)l utstl v400l v750l veril vi(rgl te)l vk(40l 5[0-3.  
v)l vm40l vod1 cl vx(52l 53l 60l 61l 70l 80l 81l 83l 85l 98)l w3c(\ -l \ )l webcl whi tl wi(g\  
l ncl nw)l wm1bl wonul x700l xda(\ -l 2l g)l yas\ -l yourl zetol zte\ -) [NC]  
# RewriteRule ^$ http://www.example.com/mobile [R,L]  
# </IfModule>
```