# 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.
#

# # CROSS-ORIGIN
# # CROSS-ORIGIN

# ! Cross-origin requests
# ! Cross-origin requests

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

# ! Cross-origin images
# ! Cross-origin images

<IfModule mod_setenvif.c>
  <IfModule mod_headers.c>
    <FilesMatch "\.(b64|bmp|curl|gif|ico|jpeg|jpg|png|svg|webp)"/>
      SetEnvIf Origin ":. IS_CORS
      Header set Access-Control-Allow-Origin "*" env=IS_CORS
    </FilesMatch>
  </IfModule>
</IfModule>

# ! Cross-origin web fonts
# ! Cross-origin web fonts

<IfModule mod_headers.c>
  <FilesMatch "\.(eot|otf|ttf|woff|woff2)">
    Header set Access-Control-Allow-Origin "*"
  </FilesMatch>
</IfModule>
# Cross-origin resource timing

# Header set Timing-Allow-Origin: "*

# 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.
# 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.
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>
</IfModule>

# IIframes cookies
# Allow cookies to be set from iframes in Internet Explorer.
# <IfModule mod_headers.c>
#   HEADER set P3P "policyref=""/w3c/p3p.xml"", CP="IDC DSP COR ADM DEV TAIi PSA PSD IVAI IVDi CONi HIS OUR IND CHT"
# </IfModule>

# MEDIA TYPES AND CHARACTER ENCODINGS
# # Serve resources with the proper media types (f.k.a. MIME types).

<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.
AddType application/javascript js

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

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

AddType image/x-icon cur ico

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

AddType application/x-font-ttf ttc ttf
AddType font/opentype otf

AddType application/octet-stream safarientz
AddType application/x-bb-appworld bbaaw
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>

# ###########################################################################
# # 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
#
# (3) If your web host doesn't allow the `FollowSymLinks` option,
# you need to comment it out or remove it, and then uncomment
# the `Options +SymLinksIfOwnerMatch` line (4), but be aware
# of the performance impact.
#
# https://httpd.apache.org/docs/current/misc/perf-tuning.html#symlinks
#
# (4) Some cloud hosting services will require you set `RewriteBase`.
#
# https://www.rackspace.com/knowledge_center/frequently-asked-question/why-is-
# modrewrite-not-working-on-my-site
# https://httpd.apache.org/docs/current/mod/mod_rewrite.html#rewritestem
#
# (5) Depending on how your server is set up, you may also need to
# use the `RewriteOptions` directive to enable some options for
# the rewrite engine.
#
# https://httpd.apache.org/docs/current/mod/mod_rewrite.html#rewrittenoptions
#
# (6) Set `%{ENV:PROTO}` variable, to allow rewrites to redirect with the
# appropriate schema automatically (http or https).

<IfModule mod_rewrite.c>

# SECTION BEGIN Redcms
RewriteEngine On
RewriteCond %{HTTP_HOST} ^redcms\.
RewriteCond %{REQUEST_URI} !^/redcms/
RewriteCond %{REQUEST_URI} /$
#RewriteCond %{DOCUMENT_ROOT}/%1 -d
RewriteRule ^(.*)$ redcms/index.php?q=$1 [L]
# SECTION END SubDomainOnFolder

# SECTION BEGIN SubDomainOnFolder
RewriteCond %{HTTP_HOST} ^\{album\|arthur\|redcms\|mariage\|walogram\|schlempf\|test3\|mouloul\|test1\|test2\|testl\|tikiwiki}\.
RewriteCond %{REQUEST_URI} !^\{album\|arthur\|redcms\|mariage\|walogram\|schlempf\|test3\|mouloul\|test1\|test2\|testl\|tikiwiki}\/
RewriteCond %{DOCUMENT_ROOT}/%1 -d
RewriteRule ^(.*)$ %1/$1 [L]
# SECTION END SubDomainOnFolder

RewriteEngine On
Options +FollowSymlinks
# Options +SymLinksIfOwnerMatch
RewriteBase /RedAgent/

# RewriteOptions <options>
RewriteCond %{HTTPS} =on
RewriteRule ^ - [env=proto:https]
RewriteCond %{HTTPS} !=on
RewriteRule ^ - [env=proto:http]

######################## Houto map /*user to /user ###########################
#RewriteEngine On
#RewriteCond %{REQUEST_URI} ^/([^/]*)$
#RewriteCond %{DOCUMENT_ROOT}/%1 !-d
#RewriteCond /home/sites/%{SERVER_NAME}/users/%1 -d
#RewriteRule ^([^/]*) http://%{HTTP_HOST}/%1/ [R,L]
#RewriteCond %{REQUEST_URI} ^/([^/]*)$
#RewriteCond %{DOCUMENT_ROOT}/%1 !-d
#RewriteCond /home/sites/%{SERVER_NAME}/users/%1 -d
#RewriteRule ^([^/]*)(.*) /%1/$2 [L]

######################## red-agent.com -> www.red-agent.com
RewriteCond %{HTTP_HOST} ^www.red-agent.com [nocase]
RewriteRule ^(.*) http://red-agent.com/$1 [L,R=301]

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^/(.*$) imgp.php?src=$1 [QSA,NC,L]
RewriteRule ^imgpp/(.*)$ imgp.php?src=$1 [QSA,NC,L]

RewriteRule ^User+Experience/$ projects.html
RewriteRule ^Resume/$ http://red-agent.com/contact.html [R=301,L]
RewriteRule ^Contact/$ http://red-agent.com/contact.html [R=301,L]
RewriteRule ^User+Experience/$ http://red-agent.com/projects.html [R=301,L]
RewriteRule ^Games/$ http://red-agent.com/projects.html [R=301,L]

RewriteRule ^contact\.html$ index.php?page=contact [L]
RewriteRule ^projects\.html$ index.php?page=projects [L]
RewriteRule ^blog\.html$ index.php?page=blog [L]
RewriteRule ^post~(.*)\.html$ index.php?page=post&post=$1 [L]
</IfModule>

# --------------------------------------------------------------------------------
# ! Forcing 'https://' /
# --------------------------------------------------------------------------------

# <IfModule mod_rewrite.c>
#   RewriteEngine On
#   RewriteCond %{HTTPS} !=on
#   RewriteRule ^(.*)$ https://%(HTTP_HOST)/$1 [R=301,L]
# </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`.
# #
# (!!) NEVER USE BOTH RULES AT THE SAME TIME!
# Option 1: rewrite www.example.com → example.com
#<IfModule mod_rewrite.c>
#  RewriteEngine On
#  RewriteCond %{HTTPS} !=on
#  RewriteCond %{HTTP_HOST} ~www\.([\-\w]+)$ [NC]
#  RewriteRule ^ %{ENV:PROTO}://%{HTTP_HOST} %{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>
#  RewriteEngine On
#  RewriteCond %{HTTPS} !=on
#  RewriteCond %{HTTP_HOST} ~\www\. [NC]
#  RewriteCond %{SERVER_ADDR} !=127.0.0.1
#  RewriteCond %{SERVER_ADDR} !=:1
#  RewriteRule ^ %{ENV:PROTO}://www. %{HTTP_HOST} %{REQUEST_URI} [R=301,L]
</IfModule>

### SECURITY

---

# Clickjacking

---

Protect website against clickjacking.
#<IfModule mod_headers.c>
#  Header set X-Frame-Options "DENY"
#  `mod_headers` cannot match based on the content-type, however,
#  the `X-Frame-Options` response header should be send only for
#  HTML documents and not for the other resources.
#  <FilesMatch ".(appcachel atmll bwbw bmpl crxI cssl curl eotl [f4[AAbpv]) flvl geojsonl gifl htcl icol jpe?gl jsl json(1d)?
#  [m4[Cav1] manifestl mapl mp4I oexl og[lagv]) opusl offl pdfl pngl rdfl rssi safariextzl svgz?
#  [swfl topojsonl tt[cf3] txtl vcardl vcf1 vttl webappvl web[mp]l webmanifestl woff2?! x1ocl xmll 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 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/
# https://w3c.github.io/webappsec-csp/

# <IfModule mod_headers.c>

# Header set Content-Security-Policy "script-src 'self'; object-src 'self''

# `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 "\.\n(appcachef atoml bbawl bmapl crxl cssl curl eotl f4[abpv]J f1vl geo.jsoml gifl htcl icol jpe?gl jssl json(1d)?
| m4[av]J manifestl mapl mp4l oexl og[av]J opusl otfpl pdfpl pngl rdfpl rssl safair extzl svgz?
| swfl topoj sonl tttl cfl ttxtl vcardl vcfpl vttl webapppl web[mp]l webmanifestl woff2?! x1ocl xmll 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 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.mnoet.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/cmsgploit/
#
# (!) 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| flai| ini| .c| .dll| .log| .psd| .sh| .sql| .sw[op])")">  

# Apache < 2.3
<br> <IfModule ! mod_authz_core.c>
   <br> Order allow,deny
   <br> Deny from all
   <br> Satisfy All
</IfModule>

# Apache ≥ 2.3
<br> <IfModule mod_authz_core.c>
   <br> Require all denied
</IfModule>

</FilesMatch>

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

# Force client-side SSL redirection.
#
# If a user types `example.com` in their browser, even if the server
# redirects them to the secure version of the website, 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 browser's
# address bar.
# (!) Remove the `includeSubDomains` optional directive if the website's subdomains are not using HTTPS.

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

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

# Prevent some browsers from MIME-sniffing the response.
# This reduces exposure to drive-by download attacks and cross-origin data leaks, and should be left uncommented, especially if the server is serving user-uploaded content or content that could potentially be treated as executable by the browser.
# http://www.slideshare.net/hasegawayosuke/owasp-hasegawa
# https://mimesniff.spec.whatwg.org/

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

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

# #     (1) (2)
# Header set X-XSS-Protection "1; mode=block"
# `mod_headers` cannot match based on the content-type, however,
# the `X-XSS-Protection` response header should be send only for
# HTML documents and not for the other resources.

# <FilesMatch \\.
(appcachef atoml bbawl bmpl crxl cssl curl eotl f4[abpv]] f1vl geojsonl gifl html icol jpe?gl jsl json(1d)?
| m4[lav] manifest mapl mp4l oexl og[agv]| opusl otfl pdfl pngl rdfl rssl safariextzl svgz?
| swfl topojsonl tt[cf]| txtl vcardl vcf| vttl webappl web[mp]| webmanifest| woff2| xml| xml|xpi)">
# Header unset X-XSS-Protection
# </FilesMatch>

#</IfModule>

# ---------------------------------------------------------------------------
# I Server-side technology information
# # ---------------------------------------------------------------------------

# Remove the `X-Powered-By` response header that:
#
# * is set by some frameworks and server-side languages
# (e.g.: ASP.NET, PHP), and its value contains information
# about them (e.g.: their name, version number)
#
# * doesn't provide any value to users, contributes to header
# bloat, and in some cases, the information it provides can
# expose vulnerabilities
#
# (!) If you can, you should disable the `X-Powered-By` header from the
# language / framework level (e.g.: for PHP, you can do that by setting
# `expose_php = off` in `php.ini`)
#

<IfModule mod_headers.c>
  Header unset X-Powered-By
</IfModule>

# ---------------------------------------------------------------------------
# I Server software information
# # ---------------------------------------------------------------------------
ServerSignature Off

ServerTokens Prod

# WEB PERFORMANCE

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

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

<IfModule mod_deflate.c>

# Force compression for mangled `Accept-Encoding` request headers

<IfModule mod_setenvif.c>

<IfModule mod_headers.c>

SetEnvIfNoCase "^Accept-EncodXng\ X-cept-Encod\ng\[\s\s\]+[X\-]{4,13}$ HAVE_Accept_Encodng

RequestHeader append Accept-Encoding "gzip, deflate" env=HAVE_Accept_Encodng

</IfModule>
<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" \
"text/javascript" \
"text/plain" \
"text/vcard" \
</IfModule>
"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 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 # the response, and thus, wouldn't be able to understand the # content.
#
# https://httpd.apache.org/docs/current/mod/mod_mimie.html#addencoding

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

</IfModule>

# ---

# // Content transformation
# ---

# Prevent intermediate caches or proxies (e.g.: such as the ones
# used by mobile network providers) from modifying the website's
# content.
#
#
# (!) If you are using `mod_pagespeed`, please note that setting
# the `Cache-Control: no-transform` response header will prevent
`PageSpeed` from rewriting `HTML` files, and, if the
`ModPagespeedDisableRewriteOnNoTransform` directive isn't set
to `off`, also from re-writing other resources.

# https://developers.google.com/speed/pagespeed/module/configuration#notransform

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

# Remove `ETags` as resources are sent with far-future expires headers.
# `FileETag None` doesn't work in all cases.
`<IfModule mod_headers.c>`
  `  Header unset ETag`
`</IfModule>`

FileETag None

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

# Web fonts

# Embedded OpenType (EOT)
ExpiresByType application/vnd.ms-fontobject "access plus 1 year"
ExpiresByType font/eot "access plus 1 year"

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

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

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

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

# Other

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

</IfModule>
# Allow concatenation from within specific files.
# e.g.:
# If you have the following lines in a file called, for example, `main.combined.js`:
# ```
# <!--include file="js/jquery.js" -->
# <!--include file="js/jquery.timer.js" -->
# 
# Apache will replace those lines with the content of the specified files.
#
```<IfModule mod_include.c>
  <FilesMatch "\.combined\..js$">
    Options +Includes
    AddOutputFilterByType INCLUDES application/javascript \ application/x-javascript \ text/javascript
    SetOutputFilter INCLUDES
  </FilesMatch>
  <FilesMatch "\.combined\..css$">
    Options +Includes
    AddOutputFilterByType INCLUDES text/css
    SetOutputFilter INCLUDES
  </FilesMatch>
</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 `*/style.12345.css` to `*/style.css`.
#
# To understand why this is important and even a better solution than using something like `*.css?v=234`, please see:
# <IfModule mod_rewrite.c>
#    RewriteEngine On
#    RewriteCond %{REQUEST_FILENAME} !-f
#    RewriteRule ^(.+)/(\d+)/.(b|c|s|l|url|gif|ico|jpe?g|j|s|l|png|svgz?|webp|webmanifest)$ $1 $3 [L]
# </IfModule>