# Apache configuration file
# httpd.apache.org/docs/2.2/mod/quickreference.html

# Note .htaccess files are an overhead, this logic should be in your Apache config if possil
# httpd.apache.org/docs/2.2/howto/htaccess.html

# Techniques in here adapted from all over, including:
# Kroc Camen: camendesign.com/.htaccess
# perishablepress.com/press/2006/01/10/stupid-htaccess-tricks/
# Sample .htaccess file of CMS MODx: modxcms.com

###
### If you run a webserver other than Apache, consider:
### github.com/h5bp/server-configs
###

# Better website experience for IE users
# -----------------------------------------------

# Force the latest IE version, in various cases when it may fall back to IE7 mode
# github.com/rails/rails/commit/123eb25#commitcomment-118920
# Use ChromeFrame if it's installed for a better experience for the poor IE folk

<IfModule mod_headers.c>
  Header set X-UA-Compatible "IE=Edge,chrome=1"
  # mod_headers can't match by content-type, but we don't want to send this header on *ever*
  <FilesMatch "\.js\.css\.gif\.png\.jpe?\n  gl\ pdf\ xml\.og\ og\ og\ og\ og\ og\ mp4\.m4v\ webm\.svg\.svg\.eot\.ttf\.ttf\.woff\.woff\.ic\.eb\.app\.cach\.mancstl\>

  Header unset X-UA-Compatible
</FilesMatch>
</IfModule>

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

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

# Serve cross-domain Ajax requests, disabled by default.
# enable-cors.org
# code.google.com/p/http-equiv-mod-mod_headers/Cross-Origin-Request-Security
# Code: google.com/p/RRIm/security/wiki/CORSSiteRequestsSecurity

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

#  

# CORS-enabled images (@crossorigin)
#  

# Send CORS headers if browsers request them; enabled by default for images.
# developer.mozilla.org/en/CORS_Edible_Images
# wiki.mozilla.org/Security/Reviews/crossoriginAttribute

<IfModule mod_setenvif.c>
  <IfModule mod_headers.c>
    # mod_headers, you no match by Content-Type?!
    <FilesMatch \.(gif|png|jpeg|webp)$>
      SetEnvIf Origin ":" IS_CORS
      Header set Access-Control-Allow-Origin "*" env=IS_CORS
    </FilesMatch>
  </IfModule>
</IfModule>

#  

# Webfont access
#  

# Allow access from all domains for webfonts.
# Alternatively you could only whitelist your
# subdomains like "subdomain.example.com".

<IfModule mod_headers.c>
  <FilesMatch \.(ttf|ttc|otf|eot|woff|woff2|font.css)>
    Header set Access-Control-Allow-Origin "*"
  </FilesMatch>
</IfModule>

#  
# Proper MIME type for all files
# --------------------------------------

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

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

# Video
AddType video/ogg ogv
AddType video/mp4 mp4 m4v
AddType video/webm webm

# SVG
# Required for svg webfonts on iPad
# twitter.com/FontSquirrel/status/14855840545
AddType image/svg+xml svg svgz
AddEncoding gzip svgz

# Webfonts
AddType application/vnd.ms-fontobject eot
AddType application/x-font-otf ttf ttc
AddType font/opentype otf
AddType application/x-font-woff woff

# Assorted types
AddType image/x-icon ico
AddType image/webp webp
AddType text/cache-manifest appcache manifest
AddType text/x-component htc
AddType application/x-chrome-extension crx
AddType application/x-opera-extension oex
AddType application/x-xpinstall xpi
AddType application/octet-stream safarientz
AddType application/x-web-app-manifest+json webapp
AddType text/x-vcard vcf
# Allow concatenation from within specific js and css files
# -------------------------------

# e.g. Inside of script.combined.js you could have
# !--#include file="libs/jquery-1.5.0.min.js" -->
# !--#include file="plugins/jquery.idletimer.js" -->
# and they would be included into this single file.

# This is not in use in the boilerplate as it stands. You may
# choose to name your files in this way for this advantage or
# concatenate and minify them manually.
# Disabled by default.

.GetFilesMatch "\..\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>

# -----------------------------------------------
# Gzip compression
# -----------------------------------------------

<IfModule mod_deflate.c>

  # Force deflate for mangled headers developer.yahoo.com/blogs/ydn/posts/2010/12/pushing-bx
  <IfModule mod_setenvif.c>
    <IfModule mod_headers.c>
      SetEnvIfNoCase "(Accept-EncodXngl X-cept-Encoding) X{15}i "{15}l -{15})$ (^((gzip| deflate)\n      HAVE_Accept-Encoding
        RequestHeader append Accept-Encoding "gzip, deflate" env=HAVE_Accept-Encoding
      </IfModule>
    </IfModule>
  </IfModule>
</IfModule>

# HTML, TXT, CSS, JavaScript, JSON, XML, HTC:
<IfModule filter_module>
    FilterDeclare COMPRESS
    FilterProvider COMPRESS DEFLATE resp=Content-Type $text/html
    FilterProvider COMPRESS DEFLATE resp=Content-Type $text/css
    FilterProvider COMPRESS DEFLATE resp=Content-Type $text/plain
    FilterProvider COMPRESS DEFLATE resp=Content-Type $text/xml
    FilterProvider COMPRESS DEFLATE resp=Content-Type $text/x-component
    FilterProvider COMPRESS DEFLATE resp=Content-Type $application/javascript
    FilterProvider COMPRESS DEFLATE resp=Content-Type $application/json
    FilterProvider COMPRESS DEFLATE resp=Content-Type $application/xml
    FilterProvider COMPRESS DEFLATE resp=Content-Type $application/xhtml+xml
    FilterProvider COMPRESS DEFLATE resp=Content-Type $application/rss+xml
    FilterProvider COMPRESS DEFLATE resp=Content-Type $application/atom+xml
    FilterProvider COMPRESS DEFLATE resp=Content-Type $application/vnd.ms-fontobject
    FilterProvider COMPRESS DEFLATE resp=Content-Type $image/svg+xml
    FilterProvider COMPRESS DEFLATE resp=Content-Type $image/x-icon
    FilterProvider COMPRESS DEFLATE resp=Content-Type $font/opentype
    FilterChain COMPRESS
    FilterProtocol COMPRESS DEFLATE change=yes;byteranges=no
</IfModule>

<IfModule !mod_filter.c>
    # Legacy versions of Apache
    AddOutputFilterByType DEFLATE text/html text/plain text/css application/json
    AddOutputFilterByType DEFLATE application/javascript application/x-javascript
    AddOutputFilterByType DEFLATE text/xml application/xml text/x-component
    AddOutputFilterByType DEFLATE application/xhtml+xml application/rss+xml application/atom+xml
    AddOutputFilterByType DEFLATE image/x-icon image/svg+xml application/vnd.ms-fontobject $font/opentype
</IfModule>

</IfModule>

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

# These are pretty far-future expires headers.
They assume you control versioning with cachebusting query params like

<script src="application.js?20100608"></script>

Additionally, consider that outdated proxies may miscache
don't use querystring/

If you don't use filenames to version, lower the CSS and JS to something like
"access plus 1 week" or so.

<IfModule mod_expires.c>
  ExpiresActive on

# Perhaps better to whitelist expires rules? Perhaps.
  ExpiresDefault "access plus 1 month"

# cache, appcache needs re-requests in FF 3.6 (thanks Remy "Introducing HTML5")
  ExpiresByType text/cache-manifest "access plus 0 seconds"

# Your document html
  ExpiresByType text/html "access plus 0 seconds"

# Data
  ExpiresByType text/xml "access plus 0 seconds"
  ExpiresByType application/xml "access plus 0 seconds"
  ExpiresByType application/json "access plus 1 year"

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

# Favicon (cannot be renamed)
  ExpiresByType image/x-icon "access plus 1 week"

# Media: images, video, audio
  ExpiresByType image/gif "access plus 1 month"
  ExpiresByType image/png "access plus 1 month"
  ExpiresByType image/jpg "access plus 1 month"
  ExpiresByType image/jpeg "access plus 1 month"
  ExpiresByType video/ogg "access plus 1 month"
  ExpiresByType audio/ogg "access plus 1 month"
  ExpiresByType video/mp4 "access plus 1 month"
  ExpiresByType video/webm "access plus 1 month"
# HTC files (css3pie)
ExpiresByType text/x-component "access plus 1 month"

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

# CSS and JavaScript
ExpiresByType text/css "access plus 1 year"
ExpiresByType application/javascript "access plus 1 year"

</IfModule>

# -----------------------------------------------
# ETag removal
# -----------------------------------------------

# FileETag None is not enough for every server.
<IfModule mod_headers.c>
    Header unset ETag
</IfModule>

# Since we're sending far-future expires, we don't need ETags for
# static content.
# developer.yahoo.com/performance/rules.html#etags
FileETag None

# -----------------------------------------------
# Stop screen flicker in IE on CSS rollovers
# -----------------------------------------------

# The following directives stop screen flicker in IE on CSS rollovers – in
# combination with the "ExpiresByType" rules for images (see above). If
# needed, un-comment the following rules.

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

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

# Allow cookies to be set from iframes (for IE only)
# If needed, uncomment and specify a path or regex in the Location directive

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

# -----------------------------------------------
# Start rewrite engine
# -----------------------------------------------

# Turning on the rewrite engine is necessary for the following rules and features,
# FollowSymLinks must be enabled for this to work.

<IfModule mod_rewrite.c>
  Options +FollowSymLinks
  RewriteEngine On
</IfModule>

# -----------------------------------------------
# Suppress or force the "www." at the beginning of URLs
# -----------------------------------------------

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

# By default option 1 (no "www.") is activated. Remember: Shorter URLs are sexier.
# no-www.org/faq.php?q=class_b

# If you rather want to use option 2, just comment out all option 1 lines
# and uncomment 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://%{REQUEST_URI} [R=301,L]
# RewriteCond %{HTTPS} !=on
# RewriteRule . (*) https://%{HTTP_HOST}%{REQUEST_URI}

# For OpenShift
#RewriteCond %{HTTP: X-Forwarded-Proto} !https
#RewriteRule .* https://%{HTTP_HOST}%{REQUEST_URI} [R,L]

</IfModule>

# Option 2:
# To rewrite "example.com -> www.example.com" uncomment the following lines.
# Be aware that the following rule 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>

# Remove index.php from URL for CodeIgniter
# <IfModule mod_rewrite.c>
# RewriteCond %{REQUEST_FILENAME} !-f
# RewriteCond %{REQUEST_FILENAME} !-d
# RewriteRule . index.php/$0 [PT,L]
# </IfModule>
<IfModule mod_rewrite.c>

    RewriteEngine On

    # my url is http://localhost/any
    #RewriteBase /any/
    # set your own
    # RewriteBase /startit/

    # 000webhost settings
    # php_value date.timezone Asia/Kolkata
    # php_flag zlib.output_compression On
    # php_value zlib.output_compression_level 5
    # RewriteBase /
    RewriteBase /webstart/

    #Removes access to the system folder by users.
    #Additionally this will allow you to create a System.php controller,
    #previously this would not have been possible.
    # 'system' can be replaced if you have renamed your system folder.
    RewriteCond %{REQUEST_URI} ^system.*
    RewriteRule ^(.*)$ /index.php/$1 [L]

    #When your application folder isn't in the system folder
    #This snippet prevents user access to the application folder
    #Submitted by: Fabdrol
    #Rename 'application' to your applications folder name.
    RewriteCond %{REQUEST_URI} ^application.*
    RewriteRule ^(.*)$ /index.php/$1 [L]

    #Checks to see if the user is attempting to access a valid file,
    #such as an image or css document, if this isn't true it sends the
    #request to index.php
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule ^(.*)$ index.php/$1 [L]
</IfModule>

# -----------------------------------------------------------------------------------------------------------
# Built-In filename-based cache busting
"
If you're not using the build script to manage your filename version revving, you might want to consider enabling this, which will route requests for `/css/style.20110203.css` to `/css/style.css`

To understand why this is important and a better idea than all.css?v1231, read: github.com/h5bp/html5-boilerplate/wiki/Version-Control-with-Cachebusting

Uncomment to enable.

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

Prevent SSL cert warnings

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

Uncomment the following lines to use this feature.

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

Prevent 404 errors for non-existing redirected folders

without -MultiViews, Apache will give a 404 for a rewrite if a folder of the same name does, e.g. /blog/hello: webmasterworld.com/apache/3003792.htm

Options -MultiViews

Custom 404 page
You can add custom pages to handle 500 or 403 pretty easily, if you like.

ErrorDocument 404 /404.html

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

# Use UTF-8 encoding for anything served text/plain or text/html
AddDefaultCharset utf-8

# Force UTF-8 for a number of file formats
AddCharset utf-8 .css .js .xml .json .rss .atom

# A little more security
# -----------------------------------------------------------------------------

# Do we want to advertise the exact version number of Apache we're running?
# Probably not.
## This can only be enabled if used in httpd.conf – It will not work in .htaccess
# ServerTokens Prod

# "-Indexes" will have Apache block users from browsing folders without a default document
# Usually you should leave this activated, because you shouldn’t allow everybody to surf the
# every folder on your server (which includes rather private places like CMS system folders:
<IfModule mod_autoindex.c>
  Options -Indexes
</IfModule>

# Block access to "hidden" directories whose names begin with a period. This
# includes directories used by version control systems such as Subversion or Git.
<IfModule mod_rewrite.c>
  RewriteCond %{SCRIPT_FILENAME} -d
  RewriteCond %{SCRIPT_FILENAME} -f
  RewriteRule "(^\//\." - [F]
</IfModule>

# Block access to backup and source files
# This files may be left by some text/html editors and
# pose a great security danger, when someone can access them
<FilesMatch "/(\bsql|sql|sql\.|sql\:\W|sql\:\:\W|sql\:\:\:\W|sql\:\:\:\:\W|sql\:\:\:\:\:\W|sql\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\:\:\:\W|sql\:\:\:\:\\...)">
    Order allow,deny
    Deny from all
    Satisfy All
</FilesMatch>

# If your server is not already configured as such, the following directive
# should be uncommented in order to set PHP's register_globals option to OFF.
# This closes a major security hole that is abused by most XSS (cross-site
# scripting) attacks. For more information: http://php.net/register_globals
#
# IF REGISTER_GLOBALS DIRECTIVE CAUSES 500 INTERNAL SERVER ERRORS:
#
# Your server does not allow PHP directives to be set via .htaccess. In that
# case you must make this change in your php.ini file instead. If you are
# using a commercial web host, contact the administrators for assistance in
# doing this. Not all servers allow local php.ini files, and they should
# include all PHP configurations (not just this one), or you will effectively
# reset everything to PHP defaults. Consult www.php.net for more detailed
# information about setting PHP directives.

# php_flag register_globals Off

# Rename session cookie to something else, than PHPSESSID
# php_value session.name sid

# Do not show you are using PHP
# Note: Move this line to php.ini since it won't work in .htaccess
# php_flag expose_php Off

# Level of log detail - log all errors
# php_value error_reporting -1

# Write errors to log file
# php_flag log_errors On

# Do not display errors in browser (production - Off, development - On)
# php_flag display_errors Off

# Do not display startup errors (production - Off, development - On)
# php_flag display_startup_errors Off
# Format errors in plain text
# Note: Leave this setting 'On' for xdebug's var_dump() output
# php_flag html_errors Off

# Show multiple occurrence of error
# php_flag ignore_repeated_errors Off

# Show same errors from different sources
# php_flag ignore_repeated_source Off

# Size limit for error messages
# php_value log_errors_max_len 1024

# Don't precede error with string (doesn't accept empty string, use whitespace if you need)
# php_value error_prepend_string " "

# Don't prepend to error (doesn't accept empty string, use whitespace if you need)
# php_value error_append_string " "

# Increase cookie security
<IfModule php5_module>
   php_value session.cookie_httponly true
</IfModule>