


```
# enable expirations,
ExpiresActive On

# Cache all files for 2 weeks after access (A).
ExpiresDefault A1209600

<FilesMatch \.php$>
  # Do not allow PHP scripts to be cached unless they explicitly send cache
  # headers themselves. Otherwise all scripts would have to overwrite the
  # headers set by mod_expires if they want another caching behavior. This may
  # fail if an error occurs early in the bootstrap process, and it may cause
  # problems if a non-Drupal PHP file is installed in a subdirectory.
  ExpiresActive Off
</FilesMatch>
</IfModule>

# Various rewrite rules.
<IfModule mod_rewrite.c>
  RewriteEngine on

  # Set "protossl" to "s" if we were accessed via https://. This is used later
  # if you enable "www." stripping or enforcement, in order to ensure that
  # you don't bounce between http and https.
  RewriteRule ^ - [E=protossl]
  RewriteCond %{HTTPS} on
  RewriteRule ^ - [E=protossl:s]

  # Block access to "hidden" directories whose names begin with a period. This
  # includes directories used by version control systems such as Subversion or
  # Git to store control files. Files whose names begin with a period, as well
  # as the control files used by CVS, are protected by the FilesMatch directive
  # above.
  #
  # NOTE: This only works when mod_rewrite is loaded. Without mod_rewrite, it is
  # not possible to block access to entire directories from .htaccess, because
  # <DirectoryMatch> is not allowed here.
  #
  # If you do not have mod_rewrite installed, you should remove these
  # directories from your webroot or otherwise protect them from being
  # downloaded.
  RewriteRule "(^/)\." - [F]
```

```

# If your site can be accessed both with and without the 'www.' prefix, you
# can use one of the following settings to redirect users to your preferred
# URL, either WITH or WITHOUT the 'www.' prefix. Choose ONLY one option:
#
# To redirect all users to access the site WITH the 'www.' prefix,
# (http://example.com/... will be redirected to http://www.example.com/...)
# uncomment the following:
# RewriteCond %{HTTP_HOST} .
# RewriteCond %{HTTP_HOST} !^www\. [NC]
# RewriteRule ^ http%{ENV: protoss1}: //www. %{HTTP_HOST}%{REQUEST_URI} [L, R=301]
#
# To redirect all users to access the site WITHOUT the 'www.' prefix,
# (http://www.example.com/... will be redirected to http://example.com/...)
# uncomment the following:
# RewriteCond %{HTTP_HOST} ^www\. (. +)$ [NC]
# RewriteRule ^ http%{ENV: protoss1}: //%1 %{REQUEST_URI} [L, R=301]

# Modify the RewriteBase if you are using Drupal in a subdirectory or in a
# VirtualDocumentRoot and the rewrite rules are not working properly.
# For example if your site is at http://example.com/drupal uncomment and
# modify the following line:
# RewriteBase /drupal
#
# If your site is running in a VirtualDocumentRoot at http://example.com/,
# uncomment the following line:
# RewriteBase /
### BOOST START ###

# Allow for alt paths to be set via htaccess rules; allows for cached variants (future mod
support)
RewriteRule .* - [E=boostpath:normal]

# Caching for anonymous users
# Skip boost IF not get request OR uri has wrong dir OR cookie is set OR request came from
this server OR https request
RewriteCond %{REQUEST_METHOD} !^(GET|HEAD)$ [OR]
RewriteCond %{REQUEST_URI} (^/equipment-
inventory/(admin|cache|misc|modules|sites|system|openid|themes|node/add|comment/reply))|
(/(edit|user|user/(login|password|register)))$) [OR]
RewriteCond %{HTTPS} on [OR]

```

```

RewriteCond %{HTTP_COOKIE} DRUPAL_UID [OR]
RewriteCond %{ENV:REDIRECT_STATUS} 200
RewriteRule .* - [S=3]

# GZIP
RewriteCond %{HTTP:Accept-encoding} !gzip
RewriteRule .* - [S=1]
RewriteCond %{DOCUMENT_ROOT}/equipment-inventory/cache/%{ENV:boostpath}/%{HTTP_HOST}%
{REQUEST_URI}_%{QUERY_STRING}\.html -s
RewriteRule .* cache/%{ENV:boostpath}/%{HTTP_HOST}%{REQUEST_URI}_%{QUERY_STRING}\.html
[L,T=text/html,E=no-gzip:1]

# NORMAL
RewriteCond %{DOCUMENT_ROOT}/equipment-inventory/cache/%{ENV:boostpath}/%{HTTP_HOST}%
{REQUEST_URI}_%{QUERY_STRING}\.html -s
RewriteRule .* cache/%{ENV:boostpath}/%{HTTP_HOST}%{REQUEST_URI}_%{QUERY_STRING}\.html
[L,T=text/html]

### BOOST END ###

# Rewrite URLs of the form 'x' to the form 'index.php?q=x'.
# Pass all requests not referring directly to files in the filesystem to
# index.php. Clean URLs are handled in drupal_environment_initialize().
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_URI} !=/favicon.ico
RewriteRule ^ index.php [L]

# Pass all requests not referring directly to files in the filesystem to
# index.php. Clean URLs are handled in drupal_environment_initialize().
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_URI} !=/favicon.ico
RewriteRule ^ index.php [L]

# Rules to correctly serve gzip compressed CSS and JS files.
# Requires both mod_rewrite and mod_headers to be enabled.
<IfModule mod_headers.c>
  # Serve gzip compressed CSS files if they exist and the client accepts gzip.
  RewriteCond %{HTTP:Accept-encoding} gzip
  RewriteCond %{REQUEST_FILENAME}\.gz -s

```

```
RewriteRule ^(.*)\.css$ $1\.css\.gz [QSA]

# Serve gzip compressed JS files if they exist and the client accepts gzip.
RewriteCond %{HTTP:Accept-encoding} gzip
RewriteCond %{REQUEST_FILENAME}\.gz -s
RewriteRule ^(.*)\.js$ $1\.js\.gz [QSA]

# Serve correct content types, and prevent mod_deflate double gzip.
RewriteRule \.css\.gz$ - [T=text/css,E=no-gzip:1]
RewriteRule \.js\.gz$ - [T=text/javascript,E=no-gzip:1]

<FilesMatch "(\.js\.gz|\.css\.gz)$">
  # Serve correct encoding type.
  Header set Content-Encoding gzip
  # Force proxies to cache gzipped & non-gzipped css/js files separately.
  Header append Vary Accept-Encoding
</FilesMatch>
</IfModule>
</IfModule>

RewriteEngine On
RewriteCond %{REQUEST_URI} ^/sites/default/files/(private/.*)$
RewriteRule ^.*$ /system/files/%1 [L]
```