COESIONE ITALIA 21-27 PIEMONTE
Cofinanziato dall'Unione europea
REGIONE PIEMONTE

.htaccess

Web development

Docente: Shadi Lahham

Intervento realizzato da
ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

.htaccess

additional .htaccess directives

1. General directives

<Files>
targets specific files by name

<Directory>
applies rules to specific directories and their subdirectories

<FilesMatch>
uses regular expressions to match file names, similar to Perl-compatible regular expressions (PCRE)

RewriteEngine On
enables the mod_rewrite module, necessary for URL rewriting

RewriteCond
sets conditions for the following RewriteRule, with multiple RewriteCond directives used together to create AND logic

RewriteRule
defines how URLs should be rewritten, with the Pattern being a regular expression and the Substitution potentially including captured groups from the Pattern

```
# Noindex a specific file
<Files "private-page.html">
    Header set X-Robots-Tag "noindex, nofollow"
</Files>
```

```
# Noindex an entire directory
<Directory "/path/to/private-directory">
    Header set X-Robots-Tag "noindex, nofollow"
</Directory>
```

```
# Noindex files with specific extensions
<FilesMatch "\.(txt|pdf)$">
```

```
    Header set X-Robots-Tag "noindex, nofollow"
</FilesMatch>
```

```
# Noindex files matching a pattern
<FilesMatch "private-*">
    Header set X-Robots-Tag "noindex, nofollow"
</FilesMatch>
```

2. Parameters commonly used in .htaccess

%{HTTPS}
used to check if the current request is using HTTPS, with common values being **"on"** for HTTPS and **"off"** for HTTP, useful for redirecting HTTP traffic to HTTPS

%{HTTP_HOST}
represents the domain name used in the request, including subdomains if present, useful for managing WWW vs. non-WWW redirects or handling multiple domains on the same server

%{REQUEST_URI}
captures the path and query string of the requested URL, including everything after the domain name, useful for redirecting old URLs to new ones or handling specific paths differently

```
# Enable the rewrite engine
RewriteEngine On
```

```
# Basic HTTP to HTTPS redirect
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

```
# Redirect to HTTPS while preserving query string
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301,QSA]
```

```
# Redirect specific domain to HTTPS
```

```
RewriteCond %{HTTP_HOST} ^example\.com [NC]
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://example.com%{REQUEST_URI} [L,R=301]
```

```
# Redirect to HTTPS except for certain directories
RewriteCond %{HTTPS} off
RewriteCond %{REQUEST_URI} !^/unsecure/
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

```
# Force HTTPS for specific directories
RewriteCond %{HTTPS} off
RewriteCond %{REQUEST_URI} ^/secure/
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

3. www/non-www redirect examples

```
# Redirect WWW to Non-WWW
RewriteCond %{HTTP_HOST} ^www\.(.+)$ [NC]
RewriteRule ^ http://%1%{REQUEST_URI} [L,R=301]
```

```
# Redirect Non-WWW to WWW
RewriteCond %{HTTP_HOST} !^www\. [NC]
RewriteRule ^ http://www.%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

```
# Redirect WWW to Non-WWW for specific domain
RewriteCond %{HTTP_HOST} ^www\.example\.com$ [NC]
RewriteRule ^(.*)$ http://example.com/$1 [L,R=301]
```

```
# Redirect both HTTP and HTTPS WWW to Non-WWW
RewriteCond %{HTTP_HOST} ^www\.(.+)$ [NC]
RewriteRule ^ https://%1%{REQUEST_URI} [L,R=301]
```

```
# Redirect subdomain to WWW
RewriteCond %{HTTP_HOST} ^subdomain\.example\.com$ [NC]
RewriteRule ^(.*)$ http://www.example.com/subdomain/$1 [L,R=301]
```

```
# Redirect HTTP to HTTPS and non-WWW to WWW in a single rule
RewriteEngine On
RewriteCond %{HTTPS} off [OR]
RewriteCond %{HTTP_HOST} !^www\. [NC]
RewriteCond %{HTTP_HOST} ^(?:www\.)?(.+)$ [NC]
RewriteRule ^ https://www.%1%{REQUEST_URI} [L,NE,R=301]
```

```
# Redirect HTTP to HTTPS and WWW to non-WWW in a single rule
RewriteEngine On
RewriteCond %{HTTPS} off [OR]
RewriteCond %{HTTP_HOST} ^www\. [NC]
RewriteCond %{HTTP_HOST} ^(?:www\.)?(.+)$ [NC]
RewriteRule ^ https://%1%{REQUEST_URI} [L,NE,R=301]
```

```
# [NE] stands for or "No Encoding"
# [OR] flag means if either of the conditions is true
# (?:www\.)? is a non-capturing group that optionally matches www.
# The ? makes it optional, and ?: prevents it from being captured or
stored
```

4. Additional usage examples

```
# 1. <Files> directive
# Applies rules to files matching the specified name
<Files "example.php">
    # Rules here apply only to example.php
    Deny from all
</Files>
```

```
# 2. <Directory> directive
# Applies rules to a specific directory and its subdirectories
<Directory "/var/www/html/private">
    # Rules here apply to the /var/www/html/private directory
    Require all denied
</Directory>
```

```
# 3. <FilesMatch> directive
# Uses regular expressions to match file names
<FilesMatch "\.(?i:php|phtml)$">
    # Rules here apply to all .php and .phtml files (case-insensitive)
    php_flag display_errors off
</FilesMatch>
```

```
# 4. RewriteEngine On
# Enables the rewrite engine, allowing the use of RewriteRule
directives
RewriteEngine On
```

```
# 5. RewriteCond (RewriteCondition)
# Sets a condition for the following RewriteRule
# Format: RewriteCond TestString CondPattern [Flags]

# Example: Check if the request is not for an existing file
RewriteCond %{REQUEST_FILENAME} !-f
# %{REQUEST_FILENAME}: The full path to the requested file
# !-f: Not an existing file

# Example: Check if the request is not for an existing directory
RewriteCond %{REQUEST_FILENAME} !-d
# !-d: Not an existing directory

# Example: Check if the request is for the home page
RewriteCond %{REQUEST_URI} ^/$
# ^/$: Matches the root URL
```

```
# 6. RewriteRule
# Defines a rule for rewriting URLs
# Format: RewriteRule Pattern Substitution [Flags]

# Example: Redirect all requests to index.php
```

```
RewriteRule ^(.*)$ index.php [L]
# ^(.*)$: Matches the entire URL path
# index.php: The target file
# [L]: Last rule to process

# Example: Redirect old URL to new URL
RewriteRule ^old-page\.html$ new-page.php [R=301,L]
# ^old-page\.html$: Matches exactly "old-page.html"
# new-page.php: The new URL
# [R=301,L]: 301 (permanent) redirect, Last rule

# Example: Rewrite URL to include .php extension
RewriteRule ^([^/]+)$ $1.php [L]
# ^([^/]+)$: Matches any string without slashes
# $1.php: Appends .php to the matched string
# [L]: Last rule
```

```
# 7. Gzip Compression
# Purpose: Compresses website files to reduce load time
<IfModule mod_deflate.c>
    # AddOutputFilterByType compresses specified file types
    AddOutputFilterByType DEFLATE text/html text/plain text/xml
text/css application/javascript
</IfModule>
```

```
# 8. Setting Caching Headers
# Purpose: Improves page load speed and reduces server load
<IfModule mod_expires.c>
    # Enable expirations
    ExpiresActive On
    # Set default expiry times
    ExpiresByType image/jpg "access plus 1 year"
    ExpiresByType image/jpeg "access plus 1 year"
    ExpiresByType image/gif "access plus 1 year"
    ExpiresByType image/png "access plus 1 year"
    ExpiresByType text/css "access plus 1 month"
    ExpiresByType application/pdf "access plus 1 month"
    ExpiresByType text/x-javascript "access plus 1 month"
    ExpiresByType application/javascript "access plus 1 month"
    ExpiresByType application/x-shockwave-flash "access plus 1 month"
    ExpiresByType image/x-icon "access plus 1 year"
    ExpiresDefault "access plus 2 days"
</IfModule>
```

```
# 9. Preventing Directory Browsing
# Purpose: Enhances security and prevents content duplication issues
Options -Indexes
```

```
# 10. Setting Character Encoding
# Purpose: Ensures proper rendering of content
AddDefaultCharset UTF-8
```

5. Access control

Purpose: control resource visibility based on various aspects

```
# Deny access to a specific directory
RewriteEngine On
RewriteCond %{REQUEST_URI} ^/admin/
RewriteRule ^ - [F]
```

RewriteEngine On: This directive activates the URL rewriting engine, allowing you to modify URLs before they are processed by the web server.
RewriteCond %{REQUEST_URI} ^/admin/: This condition checks if the requested URI starts with `/admin/`. If it does, the rewriting rule will be applied.
RewriteRule ^ - [F]:

- **^**: This matches the beginning of the URL.
- **-**: This indicates that no substitution should be made.
- **[F]**: This flag causes the web server to return a forbidden status code (403), effectively denying access to the requested resource.

```
# Allow access only to logged-in users
RewriteEngine On
```

```
RewriteCond %{HTTP_COOKIE} !^logged_in=yes$
RewriteRule ^ - [R=401,L]
```

RewriteEngine On: This directive activates the URL rewriting engine, allowing you to modify URLs before they are processed by the web server.
RewriteCond %{HTTP_COOKIE} !^logged_in=yes$: This condition checks if the logged_in cookie is not set to yes. If it's not, the user is considered not logged in.
RewriteRule ^ - [R=401,L]:

- **^**: This matches the beginning of the URL.
- **-**: This indicates that no substitution should be made.
- **[R=401,L]**:
  - **R=401**: This flag redirects the user to the same URL with a 401 Unauthorized status code.
  - **L**: This flag indicates that if a match is found, no further rewriting rules should be applied.