.htaccess

E’ un **file di configurazione utilizzato nei server Apache** per gestire diversi aspetti del sito web e migliorare l’SEO, permettendo di **configurare URL amichevoli per i motori di ricerca**, **gestire i reindirizzamenti** e **impostare pagine di errore personalizzate**.   
Questo file si trova generalmente nella directory principale del sito e consente di controllare molte funzioni del server senza dover modificare le impostazioni globali.

# Utilizzi base

## Rewrite URLs

**Obiettivo**: Trasformare URL complessi e poco leggibili in versioni più semplici e ottimizzate per i motori di ricerca.

**Esempio**: Convertire un URL come https://www.example.com/index.php?page=about in https://www.example.com/about.

**Comando**: Si usa il modulo mod\_rewrite di Apache, con una sintassi come:

Immagine che contiene testo, Carattere, schermata, Elementi grafici

Descrizione generata automaticamente

## Redirect URLs

**Obiettivo**: Reindirizzare vecchi URL a nuovi, utile quando si cambiano i nomi delle pagine, si eliminano contenuti, o si migra a una nuova struttura del sito.   
Questo previene errori e preserva il traffico e il posizionamento SEO.

**Tipi comuni di Redirect:**

* **301 (Permanente):** segnala che la pagina è stata spostata definitivamente.
* **302 (Temporaneo):** segnala che la pagina è temporaneamente in un’altra posizione.

**Esempio**: Per un redirect 301, si usa:



## Definire pagine di errore personalizzate

**Obiettivo**: Configurare pagine di errore personalizzate per migliorare l’esperienza utente e mantenere il visitatore sul sito, riducendo l’abbandono.

Immagine che contiene Carattere, testo, schermata, Elementi grafici

Descrizione generata automaticamente**Comandi**: Le pagine di errore comuni includono:

**Uso**: Mostrare pagine di errore personalizzate come “404 Not Found” o “500 Server Error” con un messaggio o un design che si integra con il sito.

# **Rewrite**

Qui vengono spiegati due principali esempi di regole di riscrittura che consentono di nascondere l'estensione dei file e riscrivere URL complessi in modo semplice e descrittivo.

Immagine che contiene testo, Carattere, schermata, design

Descrizione generata automaticamentePrima di tutto, il blocco assicura che le regole di riscrittura vengano applicate solo se il modulo mod\_rewrite è attivo nel server Apache.

### Rimuovere l'Estensione .php dalle URL

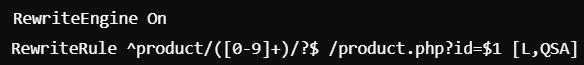
Immagine che contiene testo, Carattere, schermata, tipografia

Descrizione generata automaticamenteQuesta regola trasforma URL dinamici (come example.com/page.php) in URL più "puliti" (come example.com/page), senza mostrare l'estensione del file .php.

* **RewriteEngine On**: attiva la riscrittura delle URL.
* **RewriteCond %{REQUEST\_FILENAME} !-f**: verifica che il file richiesto non esista già fisicamente sul server. Se il file non esiste, passa alla regola successiva.
* **RewriteRule ^([^\.]+)$ $1.php [NC,L]:** riscrive le URL che non contengono un punto finale (quindi senza estensione) in modo da aggiungere .php. I dettagli:
  + **^([^\.]+)$**: seleziona tutto il testo prima di un punto (se presente), quindi corrisponde solo alle URL senza estensione.
  + **$1.php**: riscrive l'URL inserendo .php alla fine.
  + **[NC,L]:** [NC] rende la regola "case-insensitive" (ignora maiuscole e minuscole), [L] segnala che è l’ultima regola da applicare in caso di match, bloccando altre riscritture.

### 2. Riscrivere URL per i Prodotti con ID Dinamici

Questa regola riscrive URL come example.com/product/123 in modo che puntino a product.php?id=123, dove "123" è un ID prodotto dinamico.



* **RewriteEngine On**: attiva la riscrittura delle URL.
* **RewriteRule ^product/([0-9]+)/?$ /product.php?id=$1 [L,QSA]:** gestisce le URL che iniziano con /product/ seguite da una serie di numeri (l'ID del prodotto) e un eventuale slash finale.
  + **^product/([0-9]+)/?$:** intercetta URL come /product/123 o /product/123/, dove [0-9]+ cattura solo numeri consecutivi.
  + **/product.php?id=$1**: riscrive l’URL per inviare l’ID (catturato come $1) al parametro id di product.php.
  + **[L,QSA]:** [L] indica che è l’ultima regola applicabile se c’è un match, [QSA] preserva i parametri aggiuntivi di query string, aggiungendoli alla fine della riscrittura.

# **Redirect**

Il codice di Redirect in .htaccess **permette di reindirizzare i visitatori da una pagina a un’altra**, utilizzando una serie di opzioni flessibili per redirezionamenti permanenti e temporanei.   
Un **reindirizzamento permanente (301**) è importante per l’ottimizzazione SEO, perché segnala ai motori di ricerca che la vecchia pagina è stata spostata definitivamente, aggiornando i loro indici di conseguenza.

## 1. Reindirizzamento con Redirect 301

**Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente**Usare Redirect 301 è una scelta diretta e semplice per spostare utenti e motori di ricerca da una vecchia pagina a una nuova.   
Vediamo due esempi:

* **Redirect 301 /old-page.html /new-page.html**: reindirizza gli utenti da /old-page.html a /new-page.html nello stesso dominio.   
  Questo è utile per aggiornamenti interni.
* **Redirect 301 /old-page.html https://www.example.com/new-page.html**: reindirizza a una pagina su un dominio diverso.   
  Questo è utile per cambi di dominio, anche se i motori di ricerca possono impiegare più tempo per riconoscere la modifica.

### 2. Reindirizzamento con **RewriteRule** e **RewriteCond**

La direttiva RewriteRule offre un maggiore controllo rispetto a Redirect, consentendo anche condizioni di applicazione.   
Con RewriteRule, si possono aggiungere condizioni come il reindirizzamento basato sull'indirizzo IP del visitatore.

#### Esempio base

Immagine che contiene testo, Carattere, schermata, nero

Descrizione generata automaticamente

* **RewriteEngine On**: attiva il modulo di riscrittura delle URL.
* **RewriteRule ^old-page\.html$ /new-page.html [L,R=301]:** reindirizza in modo permanente (R=301) le richieste di /old-page.html a /new-page.html.
* **[L]:** specifica che questa è l'ultima regola applicata se c’è un match, evitando ulteriori riscritture.

#### Aggiunta di Condizioni (RewriteCond) per Reindirizzare per IP

Immagine che contiene testo, Carattere, schermata, nero

Descrizione generata automaticamenteSi può usare RewriteCond per applicare la regola di reindirizzamento solo se l’IP del visitatore soddisfa certe condizioni.

* **RewriteCond %{REMOTE\_ADDR} ^192\.168\.1\.1$:** verifica che l'indirizzo IP del visitatore sia 192.168.1.1.
* **RewriteRule**: applica il reindirizzamento solo per quell'IP specifico, reindirizzando /old-page.html a /new-page.html.

# **Custom error pages**

Le **pagine di errore personalizzate** in .htaccess migliorano l’esperienza utente (UX) **mostrando pagine d'errore che sono informative e graficamente in linea con il sito**, al posto dei messaggi di errore generici del server Apache.

Immagine che contiene testo, schermata, Carattere, nero

Descrizione generata automaticamentePer configurare pagine d'errore specifiche in .htaccess, si utilizza la direttiva ErrorDocument per definire la pagina di errore per ogni codice di stato HTTP.   
Ecco due esempi comuni:

* **ErrorDocument 404 /404.html**: quando un visitatore cerca una pagina inesistente, Apache reindirizza automaticamente a /404.html. Questo può includere un messaggio come "Pagina non trovata" e suggerimenti per proseguire.
* **ErrorDocument 500 /500.html**: quando il server incontra un problema interno (come errori di configurazione), mostra /500.html al posto di un messaggio di errore generico.

È possibile rinominare le pagine d’errore con nomi più descrittivi, ad esempio:

Immagine che contiene testo, Carattere, schermata, Elementi grafici

Descrizione generata automaticamente

# **1.Managing index/noindex**

Gestire l’**index/noindex** con .htaccess consente di specificare, direttamente a livello di server, quali pagine devono essere indicizzate dai motori di ricerca e quali no.   
Questo è essenziale per l’ottimizzazione SEO perché **permette di indirizzare i motori di ricerca verso i contenuti più importanti del sito**, evitando che contenuti di basso valore come pagine di amministrazione, di accesso o pagine temporanee vengano indicizzate e appaiano nei risultati di ricerca.

### Noindex per il file specifico

Immagine che contiene testo, Carattere, schermata, nero

Descrizione generata automaticamenteQuesto impedisce l’indicizzazione del file pagina-privata.html e dice ai motori di ricerca di non seguire i link al suo interno.

### Noindex per un’intera directory

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamenteCon questa configurazione, tutte le pagine all’interno della directory directory-privata sono escluse dall’indicizzazione.

### Noindex per file con estensioni specifiche

Immagine che contiene testo, Carattere, schermata, nero

Descrizione generata automaticamenteQuesto impedisce l’indicizzazione per tutti i file .txt e .pdf, il che è utile per contenuti scaricabili che non devono comparire nelle ricerche.

### Noindex per file che seguono un certo pattern

Immagine che contiene testo, Carattere, schermata, nero

Descrizione generata automaticamenteQuesta regola noindexerà tutti i file che iniziano con "privato-", come privato-123.html.

# Differenza tra **Noindex** e **robots.txt**

robots.txt  
permette di controllare solo l’accesso dei crawler, ma non impedisce l’indicizzazione; **se una pagina è bloccata in robots.txt ma è linkata da altre pagine, potrebbe comunque essere indicizzata**.

X-Robots-Tag con Noindex  
può essere applicato tramite .htaccess per dare indicazioni di noindex più granulari su file, directory e pattern specifici. Questo **impedisce effettivamente l’indicizzazione**, indipendentemente da eventuali link a tali pagine.

# **Managing HTTP/HTTPS**

Gestire i redirect da HTTP a HTTPS tramite .htaccess è fondamentale per garantire che tutto il traffico venga servito tramite una connessione sicura, il che **è importante sia per la sicurezza che per l’ottimizzazione SEO**.   
HTTPS, infatti, protegge i dati degli utenti e, essendo un fattore di ranking per Google, può migliorare il posizionamento nei risultati di ricerca.

Immagine che contiene testo, Carattere, schermata, nero

Descrizione generata automaticamente

* **RewriteEngine On**: attiva il motore di riscrittura delle URL, permettendo l’esecuzione dei redirect.
* **RewriteCond %{HTTPS} off:** verifica se la connessione corrente non è HTTPS (off indica una richiesta HTTP non sicura). Se è vero, passa alla regola successiva.
* **RewriteRule ^(.\*)$ https://%{HTTP\_HOST}%{REQUEST\_URI} [L,R=301]:** reindirizza tutte le richieste HTTP a HTTPS mantenendo sia il dominio sia il percorso della pagina richiesta (%{HTTP\_HOST} e %{REQUEST\_URI}). Il codice [L,R=301] indica un reindirizzamento permanente (301), informando i motori di ricerca del cambio definitivo.

# **Managing www and non-www – HTTPS**

Gestire la versione **www** e **non-www** di un sito web **è importante per consolidare l’autorità del dominio** ed evitare problemi di contenuti duplicati.   
I motori di ricerca considerano infatti le versioni www e non-www come URL separati, il che può diluire gli sforzi SEO.   
Uniformare tutte le versioni sotto un unico URL migliora sia l’esperienza utente sia il branding, creando un unico punto di riferimento per i link in ingresso.

### Quale Versione Scegliere?

In termini SEO non c'è vantaggio tra www e non-www, quindi la scelta dipende da branding e preferenze:

* **E-commerce globale** - "[www.shop.com](http://www.shop.com)" potrebbe risultare più familiare e autorevole.
* **Business locale** - "localbakery.com" può sembrare più intimo e diretto.
* **Startup tecnologiche** - usare una versione senza www, ad esempio "app.io", può dare un aspetto più moderno.

## Redirect da WWW a Non-WWW

## Redirect da Non-WWW a WWW

Immagine che contiene testo, Carattere, schermata, nero

Descrizione generata automaticamenteImmagine che contiene testo, Carattere, schermata, nero

Descrizione generata automaticamente