



# Performance Guida d'uso

Modulo PrestaShop · Compatibile con PrestaShop 1.7, 8 e 9

## Un negozio più veloce **un interruttore alla volta**

Performance velocizza il tuo shop con la consegna WebP (fallback sicuro all'originale), lazy loading, minificazione e combinazione di CSS/JS, header di cache del browser, suggerimenti preload/preconnect e manutenzione del database non distruttiva. Ogni ottimizzazione ha il suo interruttore, così attivi esattamente ciò che il tuo tema supporta e lasci spento il resto. Nessun codice, nessuna modifica al tema.

### 1. Installazione

1. Nel back office vai su **Moduli** → **Gestione moduli** → **Carica un modulo** e seleziona lo ZIP del modulo.
2. A installazione completata, clicca **Configura** per aprire la procedura guidata.

**Gli aggiornamenti sono automatici.** Sono legati al tuo dominio — nessuna chiave di licenza da inserire. I domini di sviluppo ( `.local` , `.test` , `localhost` ) sono sempre ammessi. Quando esce una nuova versione viene segnalata dentro al modulo.

### 2. Configurazione — passo per passo

La schermata Configura è una pagina unica: un pannello di stato, le impostazioni per ogni ottimizzazione, i pulsanti azione e uno strumento di pulizia del database.

#### Stato

Un pannello di sola lettura mostrato in cima alla pagina.

- **Interruttore generale** — se le ottimizzazioni sono attualmente ON o OFF.
- **Motore WebP (GD/Imagick)** — se il server può generare WebP. Se non disponibile, la riscrittura WebP viene semplicemente saltata.
- **Cartella cache WebP scrivibile** — se i file WebP generati possono essere salvati.
- **Blocco .htaccess cache del browser** — se le regole di cache sono attualmente scritte.

Master switch: **ON**  
WebP engine (GD/Imagick): **available**  
WebP cache directory writable: **ON**  
Browser-cache .htaccess block: **OFF**

 **General & Images**

Stato — capacità del server a colpo d'occhio

## Generale e Immagini interruttore generale

L'interruttore generale e tutte le ottimizzazioni delle immagini.

- **Abilita ottimizzazioni** — interruttore generale. Off = il modulo non fa nulla sul front office.
- **Servi WebP** — incapsula le immagini in `<picture>` con una sorgente WebP e l'originale come fallback.
- **Lazy-load immagini / Lazy-load iframe/video** — aggiungono `loading="lazy"` dove manca.
- **Fallback IntersectionObserver** — piccolo script per browser vecchi senza lazy loading nativo.
- **Aggiungi dimensioni mancanti** — width/height best-effort per ridurre il layout shift (più pesante).
- **Qualità WebP (1-100)** — la qualità di compressione usata per generare WebP.

<input checked="" type="checkbox"/> <b>Enable optimizations</b> Master switch. Off = the module does nothing on the front office.	<input checked="" type="checkbox"/> <b>Serve WebP</b> Wrap images in <code>&lt;picture&gt;</code> with a WebP source and the original as fallback.	<input checked="" type="checkbox"/> <b>Lazy-load images</b> Add <code>loading="lazy"</code> to images that do not declare it.
<input checked="" type="checkbox"/> <b>Lazy-load iframes/videos</b> Add <code>loading="lazy"</code> to iframes (maps, embeds, videos).	<input checked="" type="checkbox"/> <b>IntersectionObserver fallback</b> Tiny script for old browsers without native lazy loading.	<input type="checkbox"/> <b>Add missing dimensions</b> Best-effort width/height to reduce layout shift (heavier).
<b>WebP quality (1-100)</b> <input type="text" value="82"/>		
<code>&lt;/&gt; CSS / JS / HTML</code>		

Generale e Immagini — interruttore generale, WebP e lazy loading

## CSS / JS / HTML

Minifica, combina e ritarda i tuoi asset. Le opzioni "invasive" sono spente di default — prova sul tuo tema.

- **Minifica CSS / Minifica JS** — usano la cache nativa per-file di PrestaShop (sicure).
- **Combina CSS (invasiva) / Combina JS (invasiva)** — spente di default; possono rompere alcuni temi, prova prima di attivare.
- **Defer JS** — aggiunge `defer` agli script non critici.
- **Sposta JS nel footer** — opt-in, dipende dal tema.
- **Minifica HTML (invasiva)** — comprime gli spazi; preserva `pre/textarea/script/style`.
- **Esclusioni** — un frammento per riga; ogni URL di asset o src immagine che contiene uno di questi viene lasciato intatto.

**Minify CSS**  
Uses PrestaShop native per-file CSS cache (safe).

**Minify JS**  
Uses PrestaShop native per-file JS cache (safe).

**Combine CSS (invasive)**  
Off by default. Can break some themes - test before enabling.

**Combine JS (invasive)**  
Off by default. Can break some themes - test before enabling.

**Defer JS**  
Opt-in. Adds defer to non-critical scripts.

**Move JS to footer**  
Opt-in, theme dependent.

**Minify HTML (invasive)**  
Off by default. Collapses whitespace; preserves `pre/textarea/script/style`.

**Exclusions (one fragment per line)**  
Any asset URL or image src containing one of these fragments is left untouched.

**⚡ Preload / Preconnect**

CSS / JS / HTML — minifica, combina, defer ed esclusioni

## Preload / Preconnect

Aggiungi i resource hint per velocizzare il primo render.

- **Preload** — un URL per riga per font/asset critici. L'attributo `as` viene rilevato dall'estensione.
- **Preconnect** — un'origine per riga, es. `https://fonts.gstatic.com`.

**Preload (one URL per line)**  
Fonts/critical assets. The "as" attribute is detected from the extension.

**Preconnect (one origin per line)**  
e.g. `https://fonts.gstatic.com`

**🔧 Critical CSS (advanced)**

Preload / Preconnect — resource hint per font e origini

## Critical CSS (avanzato) opzionale

Incorpora il CSS above-the-fold nel `<head>` della pagina. Avanzato e spento di default.

- **Inline critical CSS (invasiva)** — spenta di default; incorpora il CSS sottostante nell'head.
- **CSS above-the-fold** — il CSS da incorporare. Lascia vuoto se non sai cosa estrarre.

**Inline critical CSS (invasive)**  
Off by default. Inlines the CSS below into the `<head>`.

**Above-the-fold CSS**

---

**Database maintenance thresholds**

Critical CSS — incorpora gli stili above-the-fold

## Soglie di manutenzione database

Quanto devono essere vecchi i dati transitori prima che lo strumento di pulizia li tocchi.

- **Carrelli abbandonati più vecchi di (giorni)**
- **Connessioni/ospiti più vecchi di (giorni)**
- **Righe di log più vecchie di (giorni)**

Ogni valore è compreso tra 7 e 3650 giorni. Queste soglie guidano lo strumento di pulizia più in basso nella pagina.

<b>Abandoned carts older than (days)</b> <input style="width: 80%;" type="text" value="30"/>	<b>Connections/guests older than (days)</b> <input style="width: 80%;" type="text" value="30"/>	<b>Log rows older than (days)</b> <input style="width: 80%;" type="text" value="30"/>
---	--	--

Save settings

Soglie di manutenzione database — limiti d'età per i dati transitori

## Azioni

Pulsanti di manutenzione con un clic.

- **Svuota cache WebP** — rimuove i file WebP generati.
- **Rigenera WebP (batch 500)** — pre-genera i WebP per un massimo di 500 immagini alla volta.
- **Scrivi / Rimuovi regole cache del browser** — scrive le regole di cache nel `.htaccess` dentro un blocco reversibile `# BEGIN/END cll_perf` (viene creato un backup una tantum `.htaccess.cll_perf-bak`). GZIP/Brotli vengono attivati dove il modulo Apache è presente.

Clear WebP cacheRegenerate WebP (batch 500)Write browser-cache rules to .htaccess

The .htaccess rules are written in a reversible `"# BEGIN/END cll_perf"` block and a one-time backup (`.htaccess.cll_perf-bak`) is created. GZIP/Brotli are enabled where the Apache module is present.

---

**Database cleanup (maintenance)**

Azioni — cache WebP e regole .htaccess di cache del browser

## Pulizia database (manutenzione)

Una pulizia non distruttiva dei dati transitori. Questa **non** è una "modalità manutenzione" del negozio: il front office resta online e i visitatori non vengono mai bloccati.

- **Esegui (altrimenti dry-run)** — lascia deselezionato per l'anteprima; nulla viene eliminato finché non lo selezioni.
- **Esegui anche OPTIMIZE TABLE** — recupera spazio nelle tabelle dopo la pulizia.
- **Avvia manutenzione** — vengono toccati solo i dati transitori di ospiti/log più vecchi delle tue soglie; ordini e clienti validi non vengono mai toccati.

*This is a non-destructive database cleanup. It is NOT a shop "maintenance mode": your front office stays online and visitors are never blocked.*

Always run a dry-run first. Nothing is deleted unless you tick "Execute". Only transient guest/log data older than the configured thresholds is affected; valid orders and customers are never touched.

Execute (otherwise dry-run)  Also OPTIMIZE TABLE

▶ Run maintenance

Pulizia database — prima sempre dry-run, poi esegui

## 3. Verifica che funzioni

- Apri il pannello *Stato*: l'interruttore generale dovrebbe essere ON e il motore WebP "disponibile".
- Con "Servi WebP" attivo, guarda il sorgente di una scheda prodotto — le immagini dovrebbero essere incapsulate in `<picture>` con una sorgente `.webp`.
- Esegui prima la pulizia database in *dry-run*: riporta quanti carrelli, connessioni e righe di log *verrebbero* rimossi prima di eseguire davvero.
- Dopo aver scritto le regole di cache del browser, controlla nel tuo `.htaccess` il blocco `# BEGIN cll_perf`.

## 4. Domande frequenti

### Un'ottimizzazione può rompere il tema?

Le opzioni "invasive" (combina CSS/JS, minifica HTML, critical CSS) sono spente di default. Attivale una alla volta e prova; usa la lista Esclusioni per saltare qualsiasi asset problematico.

### E se il mio server non può creare WebP?

Il pannello Stato te lo dice. Se né GD né Imagick sono disponibili, la riscrittura WebP viene semplicemente saltata e vengono servite le immagini originali.

### La pulizia database è sicura?

Sì. È non distruttiva, offre sempre un dry-run e rimuove solo dati transitori di ospiti/log più vecchi delle tue soglie — mai ordini o clienti validi.

### Devo modificare il tema?

No. Performance funziona tramite hook standard ed è compatibile con qualsiasi tema.