

# Modulo generatore WebP



versione 1.0.4  
di PrestaChamps

<b>Requisiti</b>	<b>2</b>
<b>Panoramica modulo</b>	<b>3</b>
Configurazione modulo	4
EWWW Cloud Convert	6
Impostazioni conversione CWEBP	7
<b>Rigenerare WebP</b>	<b>8</b>
<b>Supporto browser WEBP</b>	<b>10</b>
<b>Compatibilità Prestashop 1.6</b>	<b>11</b>

## Requisiti

Assicurarsi che la versione di PHP sia almeno di 5.6 altrimenti il modulo non funzionerà (come menzionato nella scheda dei requisiti del prodotto su Addons PrestaShop)

Sul server deve essere configurato uno dei prossimi moduli PHP:

**Cwebp** => funzione php exec abilitata

**Imagick** => Modulo php Imagick con webp compilato

**Gmagick** => modulo Gmagick php con webp compilato

**Gd** => Gd modulo php con imagewebp, imagecreatefrompng e imagecreatefromjpeg functions

**Ewww** => sottoscrizione ewww.io e php curl Service



## Panoramica

modulo Questo modulo è un elemento essenziale se si desidera un sito web veloce.

Il modulo WebP Generator, creato per i siti Prestashop, consentirà agli utenti di rigenerare in modo efficiente le loro immagini di prodotti, categorie e produttori, aumentando così la velocità del sito.

Questa nuova tecnologia offre una compressione senza perdita e perdita per le immagini sul tuo sito web. Dopo aver iniziato a generare immagini attraverso il nostro modulo, si libererà spazio sul tuo server. Inoltre, la qualità delle tue immagini non sarà influenzata.

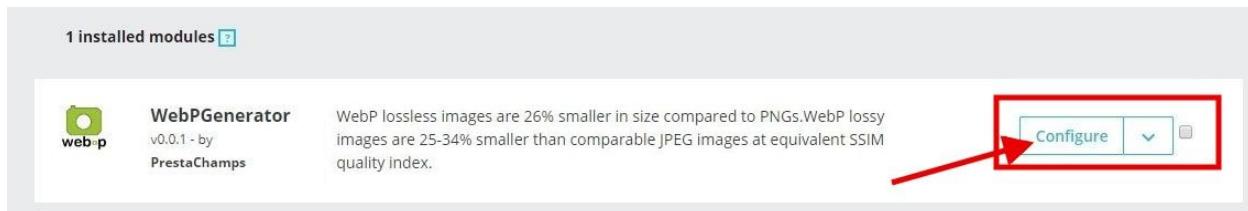
Le immagini generate tramite il **modulo generatore WebP** hanno dimensioni inferiori del 26% rispetto ai PNG. Le immagini con perdita di dati WebP sono inferiori del 25-34% rispetto alle immagini JPEG comparabili con indice di qualità SSIM equivalente.

**Informazioni:** - il modulo può essere installato dal back office del sito web, proprio come tutti gli altri moduli.

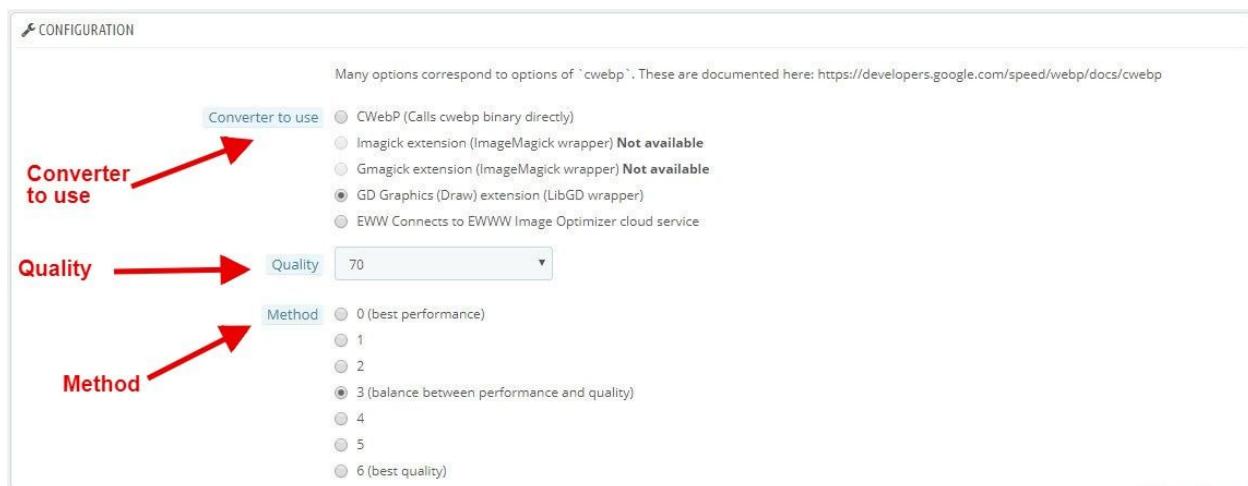


## Configurazione del modulo

: dopo l'installazione è possibile accedere al modulo facendo clic sul **Configura** pulsante.



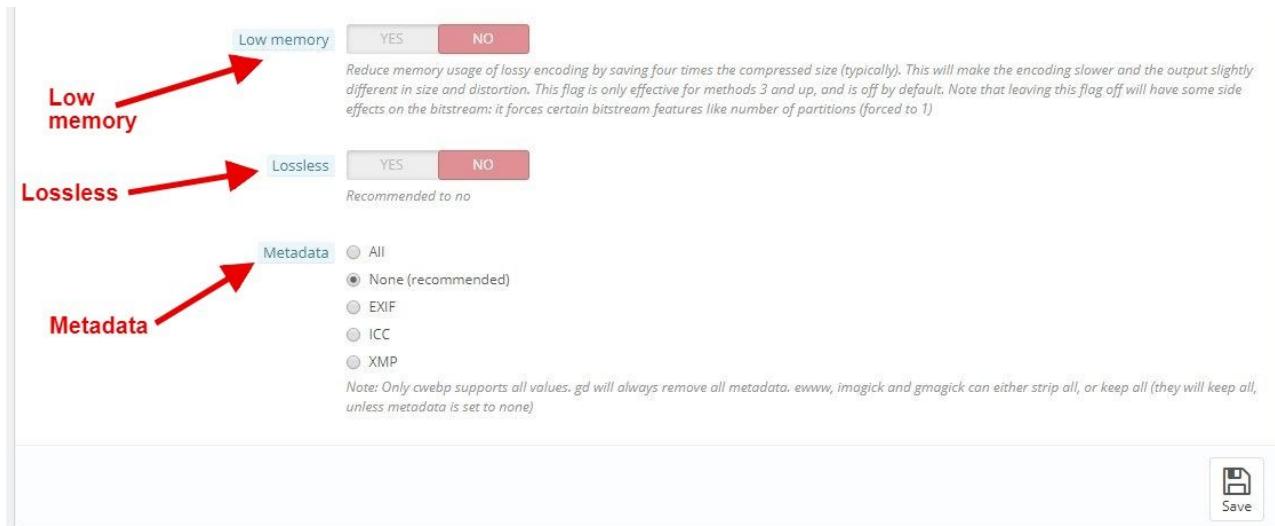
- **Convertitore da utilizzare** - Impostare i metodi di conversione da utilizzare in base alla configurazione del server
- **Qualità** - Specificare il fattore di compressione per i canali RGB tra 0 e 100
- **Metodo:** questo parametro controlla il compromesso tra velocità di codifica e dimensioni e qualità del file compresso. I valori possibili vanno da 0 a 6. 0 è il più veloce. 6 risultati nella migliore qualità.



- **Memoria**- insufficiente Riduce l'utilizzo della memoria della codifica lossy al costo del ~ 30% di tempo di codifica più lungo e delle dimensioni dell'output leggermente più grandi.
- **perdita di dati**: Senza codifica l'immagine senza alcuna perdita. L'opzione viene ignorata per i(forzati) PNG

# Modulo generatore WebP v1.0.4

- **metadati**- Metadati da copiare dall'input all'output se presente.



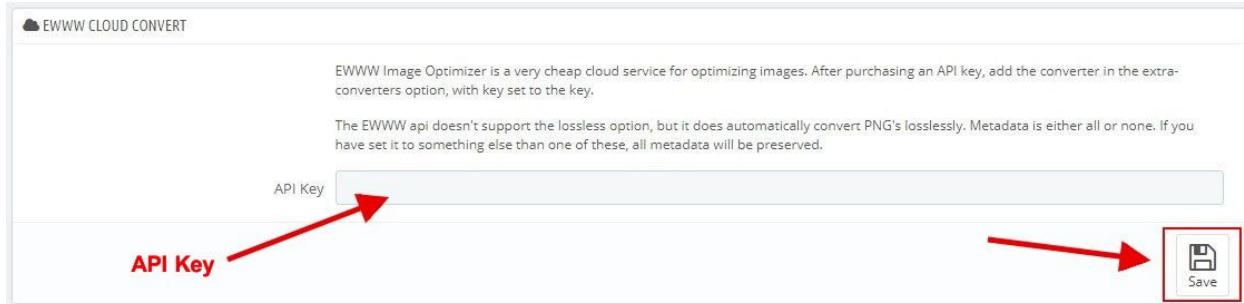
- **Modalità demo**: abilitare la "modalità demo" durante il processo di generazione dell'immagine. Questo è raccomandato per evitare 404 errori di immagini che non sono ancora state generate. Una volta che il processo è completo, puoi disabilitare questa opzione.



## EWWW Cloud Convert

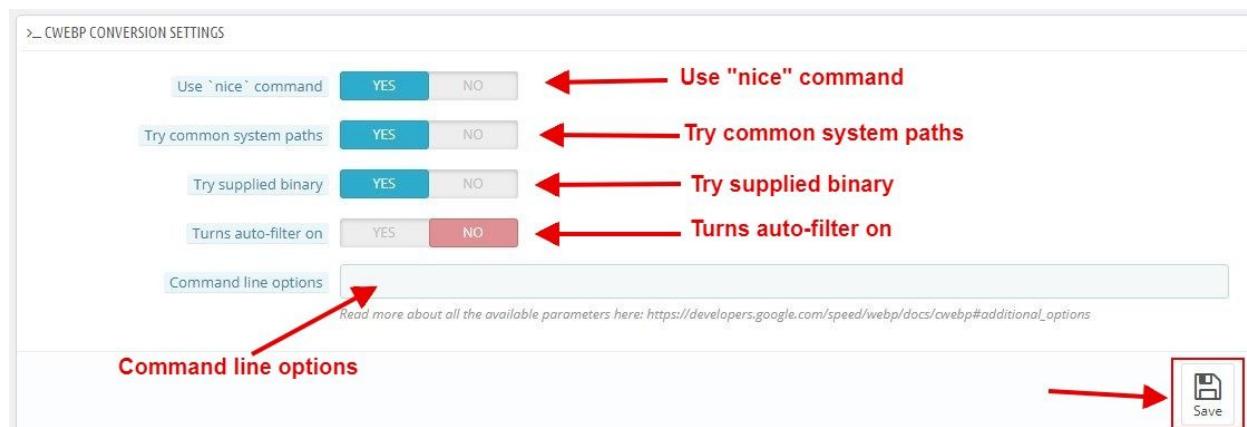
EWWW Image Optimizer è un servizio cloud molto economico per l'ottimizzazione delle immagini. Dopo aver acquistato una chiave API, aggiungi il convertitore nell'opzione extra-convertitori, con il campo chiave contenente la chiave API.

L'API EWWW non supporta l'opzione lossless ma converte automaticamente i PNG senza perdita di dati. I metadati sono o tutti o nessuno. Se lo hai impostato su qualcosa di diverso da uno di questi, tutti i metadati verranno mantenuti.



## Impostazioni di conversione CWEBP

- **Utilizza il comando `nice`** - Se il comando ` nice` viene trovato sull'host, binary viene eseguito con priorità bassa per salvare risorse di sistema
- **Prova percorsi di sistema comuni** - Questa opzione verifica se cwebp è disponibile in un percorso di sistema comune (ad es. / usr / bin / cwebp ..)
- **Prova binario fornito** - Se CWebP non è installato sul server, il binario fornito viene selezionato da Convertitori / Binari (secondo OS) - dopo aver validato il checksum
- **Attiva il filtro automatico** - Questo algoritmo spenderà tempo aggiuntivo per ottimizzare la forza filtrante per raggiungere una qualità ben bilanciata. Sfortunatamente, è estremamente costoso in termini di calcolo. Occorrono circa 5-10 volte di più per eseguire una conversione. Un'immagine da 1 MB che in genere richiede circa 2 secondi per la conversione, richiede circa 15 secondi per la conversione con filtro automatico. Quindi nella maggior parte dei casi, vorrai lasciare questo al suo valore predefinito, che è spento.
- **Opzioni della riga di comando:** consente di impostare qualsiasi parametro disponibile per cwebp nello stesso modo in cui si eseguirà l'esecuzione di cwebp. Ad esempio, potresti impostarlo su " -sharpness 5 -mt -crop 10 10 40 "



Ulteriori informazioni su tutti i parametri disponibili qui:

[https://developers.google.com/speed/webp/docs/cwebp#additional\\_options](https://developers.google.com/speed/webp/docs/cwebp#additional_options)

## Rigenerare WebP

Dopo aver configurato il modulo, vai su "**Parametri avanzati**" nel menu a sinistra. Scegli la "**Rigenera WebP**" scheda.

The screenshot shows the 'Advanced Parameters' configuration interface. On the left, there is a sidebar with various tabs: Information, Performance, Administration, E-mail, Import, Team, Database, Logs, Webservice, Multistore, and Regenerate WebP. The 'Regenerate WebP' tab is highlighted with a red border and has a red arrow pointing to it from the text above. The main area is titled 'CWEBP CONVERSION SETTINGS' and contains several configuration options with 'YES' and 'NO' buttons: 'Use `nice` command', 'Try common system paths', 'Try supplied binary', and 'Turns auto-filter on'. There is also a 'Command line options' input field and a link to 'Read more about all the available parameters'.

È possibile scegliere le immagini che si desidera generare

- Immagini prodotto
- Categoria Immagini
- Fornitore Immagini
- Archivio Immagini
- Produttore Immagini

Una volta scelto, è possibile iniziare a generare. Puoi vedere quante immagini ci sono per ogni tipo e la barra di elaborazione mostrerà la percentuale delle immagini generate.

# Modulo generatore WebP v1.0.4

 WebP is a modern image format that provides superior **lossless and lossy** compression for images on the web. Using WebP, webmasters and web developers can create smaller, richer images that make the web faster.

WebP lossless images are 26% smaller in size compared to PNGs. WebP lossy images are 25-34% smaller than comparable JPEG images at equivalent SSIM quality index.

Lossless WebP **supports transparency** (also known as alpha channel) at a cost of just 22% additional bytes. For cases when lossy RGB compression is acceptable, **lossy WebP also supports transparency**, typically providing 3x smaller file sizes compared to PNG.

**REGENARATE WEBP IMAGES**

You can regenerate all your images safely.

 PRODUCT IMAGES	0/20588	0%
 CATEGORY IMAGES	0/0	0%
 SUPPLIER IMAGES	0/1	0%
 STORE IMAGES	0/5	0%
 MANUFACTURER IMAGES	0/6	0%

**IMPORTANTE:** la finestra o la scheda in cui è stato avviato il processo di rigenerazione **devono rimanere aperte durante l'intero processo**. La chiusura interromperà il processo e dovrà essere reiniziato.



Presta**Champs**

## Supporto browser WEBP

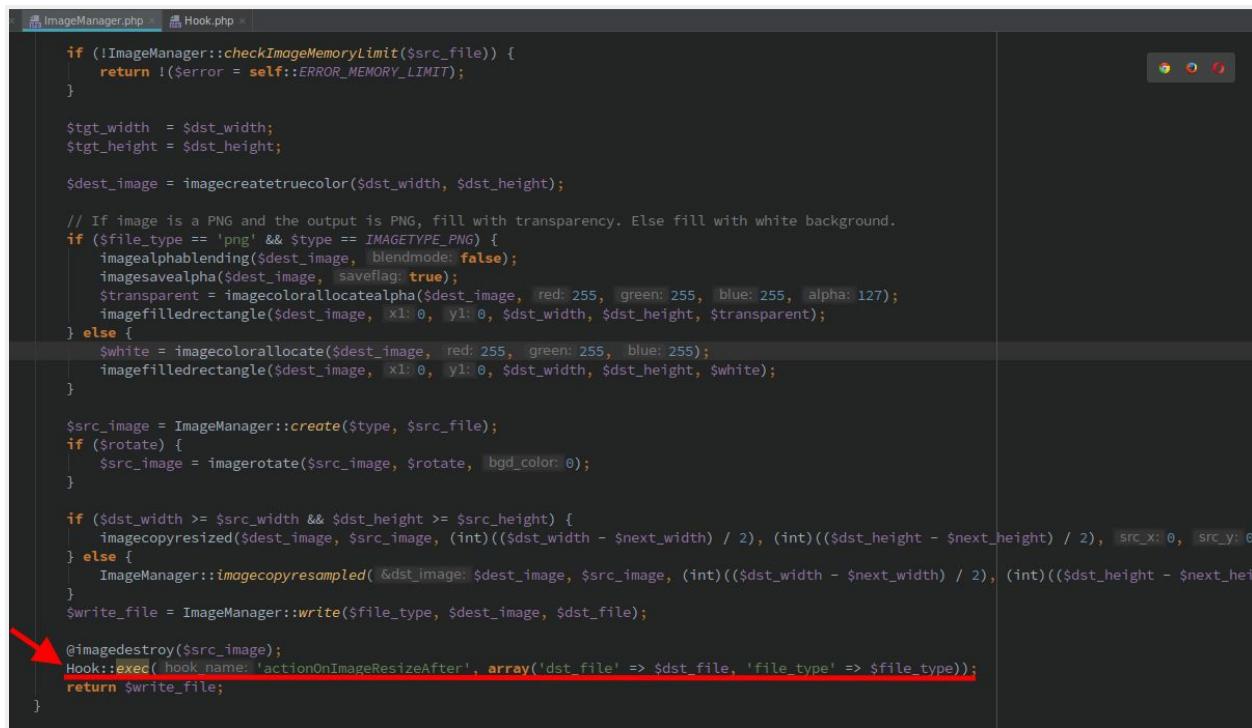
Attualmente solo Google Chrome e Opera supportano le immagini WebP. Sebbene altri browser, come Firefox, Safari o Internet Explorer non supportino attualmente il formato dell'immagine in modo nativo, non devi preoccuparti. Le immagini del tuo sito non verranno interrotte nemmeno per i client che utilizzano questi browser, le immagini verranno visualizzate in formato PNG o JPG.



## Compatibilità con Prestashop 1.6

Poiché Prestashop 1.6 non esegue l'azione `actionOnImageResizeAfter`, è necessario copiare il seguente frammento di codice nella `ImageManager` classe `resize` metodo, prima dell'istruzione `return $write_file`:

```
Hook :: exec ('actionOnImageResizeAfter', array ('dst_file' => $dst_file, 'file_type' => $file_type));
```



```
if (!ImageManager::checkImageMemoryLimit($src_file)) {
    return !($error = self::ERROR_MEMORY_LIMIT);
}

$tgt_width = $dst_width;
$tgt_height = $dst_height;

$dest_image = imagecreatetruecolor($dst_width, $dst_height);

// If image is a PNG and the output is PNG, fill with transparency. Else fill with white background.
if ($file_type == 'png' && $type == IMAGETYPE_PNG) {
    imagealphablending($dest_image, blendmode: false);
    imagesavealpha($dest_image, saveflag: true);
    $transparent = imagecolorallocatealpha($dest_image, red: 255, green: 255, blue: 255, alpha: 127);
    imagefilledrectangle($dest_image, x1: 0, y1: 0, $dst_width, $dst_height, $transparent);
} else {
    $white = imagecolorallocate($dest_image, red: 255, green: 255, blue: 255);
    imagefilledrectangle($dest_image, x1: 0, y1: 0, $dst_width, $dst_height, $white);
}

$src_image = ImageManager::create($type, $src_file);
if ($rotate) {
    $src_image = imagerotate($src_image, $rotate, bgd_color: 0);
}

if ($dst_width >= $src_width && $dst_height >= $src_height) {
    imagecopyresized($dest_image, $src_image, (int)((($dst_width - $next_width) / 2), (int)((($dst_height - $next_height) / 2), src_x: 0, src_y: 0));
} else {
    ImageManager::imagecopyresampled(&$dest_image, $dest_image, $src_image, (int)((($dst_width - $next_width) / 2), (int)((($dst_height - $next_height) / 2), dst_x: 0, dst_y: 0));
}
$write_file = ImageManager::write($file_type, $dest_image, $dst_file);

@imagedestroy($src_image);
Hook::exec('hook_name: 'actionOnImageResizeAfter', array('dst_file' => $dst_file, 'file_type' => $file_type));
return $write_file;
}
```