

Инструкция по настройке WP Super Cache






Дата создания: 19 декабря 2025

Окружение: DEV (<https://dev.maslovka.org/>)

Версия плагина: WP Super Cache 3.0.3

Краткое резюме

Статус до настройки:

-  Плагин установлен и активен
-  Кеширование выключено (`$cache_enabled = false`)
-  Компрессия выключена
-  Preload не настроен
-  Размер кеша: 20KB (пустой)

Цель: Включить кеширование, настроить preload, garbage collection и mobile cache

Пошаговая настройка


Доступ к настройкам

URL: <https://dev.maslovka.org/wp-admin/options-general.php?page=wpsupercache>

Меню WordPress: Настройки → WP Super Cache

ШАГ 1: Включение кеширования (вкладка "Простой")

Блок "Кеширование":

-  Выбери: "**Включить кеширование**"
- Нажми: "**Обновить статус**"




Результат: Кеширование будет активировано

ШАГ 2: Основные настройки (вкладка "Дополнительно")

Блок "Метод доставки кеша":

Выбери один из вариантов:

ВАРИАНТ А: Простой режим (РЕКОМЕНДУЕТСЯ для DEV)

- ☐ **Простой** (работает через PHP)
-  Проще в настройке
-  Не требует изменений .htaccess
-  Безопаснее при деплое

ВАРИАНТ В: Режим эксперта (для PROD)

- ☐ **Эксперт** (использует mod_rewrite)
- ☐ Требуется настройки .htaccess
- ☐ Нужно нажать "Обновить правила Mod_Rewrite" в самом низу страницы
- ⚡ Максимальная производительность

Блок "Разное" - Ограничения кеша:

Включить (поставить галочки):

- ☒ **Отключить кеширование для авторизованных пользователей** (Рекомендовано)
- ☒ **Не кешировать страницы с параметрами GET** (?x=y в конце URL)
- ☒ **Сжимать файлы кэша чтобы ускорить работу** (Рекомендовано)
- ☒ **Авто перестройка кэша** (Рекомендовано)
- ☒ **Кеширование браузера 304** (Рекомендовано)

НЕ включать:

- ☒ **Считать известных пользователей анонимными** (не рекомендуется)
- ☒ **Включить кеширование для всех посетителей** (оставить выключенным)
- ☒ **Отключение кеширования для посетителей, имеющих в браузере установленные куки** (оставить выключенным)

Блок "Расширенные":

Включить:

- ☒ **Поддержка мобильных устройств**
- ☒ **Дополнительная сверка кэша** (Рекомендовано)
- ☒ **Обновлять страницу при добавлении нового комментария к ней**

НЕ включать:

- ☒ **Включить динамическое кеширование** (не нужно)
- ☒ **Создать список страниц в кэше** (не обязательно)
- ☒ **Жесткая блокировка файлов** (замедляет работу)

Блок "Просроченные страницы & Очистка мусора":

Таймаут кеширования: 1800 секунд (30 минут)

Как долго файлы кеша должны оставаться актуальными

Планировщик:

- ☐ Выбери: **Таймер**
- **Таймер:** 600 секунд (10 минут между проверками)

Электронные адреса для уведомлений:

- ☒ **Оставь выключенным** (или включи если нужны уведомления)

✓ Нажми внизу: "Обновить статус"

✓ ШАГ 3: Настройка предварительного кеширования (вкладка "Preload")

Параметры предзагрузки:

Обновлять общий кэш каждые: 600 минут (10 часов)

Как часто обновлять весь кэш

Кэшировать: 0 записи

0 = кэшировать ВСЕ записи (рекомендуется)

Галочки:

- ✓ **Режим предзагрузки** (сбор мусора отключен. Рекомендуется.)
- ✓ **Предзагрузка тегов, категорий и других таксономий**

Уведомления:

- ✗ Отправлять мне сообщения с уведомлениями об обновлении кэша (опционально)

✓ Нажми: "Обновить настройки"

Затем нажми: "Предварительно кэшировать сейчас"

🕒 **Процесс займет несколько минут** - на странице будет показан прогресс создания кэша для каждой страницы.

✓ ШАГ 4: Параметры отслеживания (вкладка "Дополнительно", блок внизу)

Блок "Параметры отслеживания":

Список параметров (должен быть уже заполнен):

```
fbclid
ref
gclid
fb_source
mc_cid
mc_eid
utm_source
utm_medium
utm_campaign
utm_term
utm_content
utm_expid
mtm_source
mtm_medium
```

```
mtm_campaign  
mtm_keyword  
mtm_content  
mtm_cid  
mtm_group  
mtm_placement  
ysclid  
srsltid  
yclid
```


-  **Включить** - поставь галочку

Назначение: Позволяет кешировать страницы, игнорируя параметры из Facebook, Google Analytics и других систем отслеживания.

ШАГ 5: CDN (вкладка "CDN")

Для DEV-окружения: Можно **пропустить** - оставь всё выключенным.

Для PROD (опционально):


-  Enable CDN Support
 - **Off-site URL:** укажи CDN домен (например, через Cloudflare)
 - **Include directories:** `wp-content,wp-includes`
-

ШАГ 6: Проверка и тестирование

6.1. Очистка кеша (вкладка "Содержимое"):

1. **Удалить кеш** - нажми для полной очистки
2. **Удалить просроченные файлы** - для очистки только устаревших

6.2. Проверка статуса (вкладка "Простой"):

 Должно быть зеленое: **"Кэширование включено"**

 Внизу будет статистика: количество закэшированных страниц

6.3. Тестирование в браузере:

1. Открой в режиме инкогнито: `https://dev.maslovka.org/`
2. Обнови страницу 2-3 раза (Cmd+R)
3. Второй раз должен загружаться заметно быстрее
4. Открой DevTools (F12) → Network
5. Посмотри на главный HTML документ
6. В Headers должно быть: `"WP-Super-Cache: Served supercache file from PHP"`

6.4. Проверка через SSH:

```
# Размер кеша
ssh a1182962@141.8.192.186 "du -sh
/home/a1182962/domains/dev.maslovka.org/public_html/wp-content/cache/"

# Количество закешированных страниц
ssh a1182962@141.8.192.186 "find
/home/a1182962/domains/dev.maslovka.org/public_html/wp-
content/cache/supercache/ -name 'index*.html' -o -name 'index*.html.gz' |
wc -l"

# Список закешированных страниц
ssh a1182962@141.8.192.186 "find
/home/a1182962/domains/dev.maslovka.org/public_html/wp-
content/cache/supercache/ -type f -name 'index*' | head -20"

# Очистка кеша через WP-CLI
ssh a1182962@141.8.192.186 "/home/a1182962/wp-cli.phar --
path=/home/a1182962/domains/dev.maslovka.org/public_html cache flush"
```

Ожидаемые результаты

После правильной настройки:

Производительность:

- ⚡ Время загрузки сократится на **30-50%**
- ⚡ TTFB (Time to First Byte) уменьшится в **2-3 раза**
- ⚡ Второй визит будет загружаться в **5-10 раз быстрее**

Нагрузка на сервер:

- 📉 Нагрузка на PHP снизится на **80%**
- 📉 Нагрузка на MySQL снизится на **80%**
- 📉 CPU usage уменьшится на **60-70%**

Размер кеша:

- 📁 Для небольшого сайта (~50 страниц): **5-20 MB**
- 📁 Для среднего сайта (~200 страниц): **50-100 MB**
- 📁 Для большого сайта (~1000 страниц): **200-500 MB**

Дополнительные настройки (опционально)

Не кешировать определенные типы страниц (вкладка "Дополнительно", блок "Допустимые имена & Запрещенные адреса"):

По умолчанию **НЕ кешируются** (можно оставить так):

- ☒ Записи (is_single) - отключено
- ☒ Страницы (is_page) - отключено
- ☒ Основная страница (is_front_page) - отключено

Рекомендуется **ОТКЛЮЧИТЬ кеширование** (поставить галочки):

- ☒ Страницы поиска (is_search)
- ☒ Фиды (is_feed)

Исключить URL из кеша:

Если нужно не кешировать определенные страницы, добавь их в блок "Настройки отклонения URL":

```
/wp-admin/  
/cart/  
/checkout/  
/my-account/
```

⚠ Важные моменты

После каждого деплоя:

```
# Очистить кеш через WP-CLI  
ssh a1182962@141.8.192.186 "/home/a1182962/wp-cli.phar --  
path=/home/a1182962/domains/dev.maslovka.org/public_html cache flush"  
  
# Или через SSH и плагин  
ssh a1182962@141.8.192.186 "rm -rf  
/home/a1182962/domains/dev.maslovka.org/public_html/wp-  
content/cache/supercache/*"
```

Добавить в deployment скрипты:

В файл [www/deployment-scripts/deploy-dev.sh](#) уже должна быть команда:

```
$WP_CLI_REMOTE cache flush
```

Особенности работы кеша:

1. **Залогиненные пользователи** не получают кешированные страницы (это правильно)
2. **Первый визит** на страницу всегда медленнее (кеш создается), второй и последующие - быстрые
3. **Preload** работает в фоне через wp-cron и обновляется каждые 10 часов

4. **Garbage collection** проверяет устаревшие файлы каждые 10 минут
5. **Мобильные устройства** получают отдельную версию кеша

Мониторинг:

```
# Проверить статус preload (в админке WP)
# Настройки → WP Super Cache → вкладка Preload
# Внизу будет "Предзагрузка в процессе..." или "Последнее завершение: ..."
```



```
# Проверить размер кеша
ssh a1182962@141.8.192.186 "du -sh
/home/a1182962/domains/dev.maslovka.org/public_html/wp-content/cache/"
```



```
# Проверить последние созданные файлы кеша
ssh a1182962@141.8.192.186 "find
/home/a1182962/domains/dev.maslovka.org/public_html/wp-
content/cache/supercache/ -type f -name 'index*' -mmin -60 | wc -l"
```

Troubleshooting (Решение проблем)

Проблема: Кеш не работает

Проверить:

1. Зайди в админку → Настройки → WP Super Cache → вкладка "Простой"
2. Должно быть зеленое "Кэширование включено"
3. Если красное - посмотри сообщение об ошибке

Решение:

```
# Проверить права на папку
ssh a1182962@141.8.192.186 "ls -la
/home/a1182962/domains/dev.maslovka.org/public_html/wp-content/cache/"
```



```
# Должно быть: drwxr-xr-x a1182962 a1182962
```



```
# Если нет – исправить
ssh a1182962@141.8.192.186 "chmod 755
/home/a1182962/domains/dev.maslovka.org/public_html/wp-content/cache/"
```

Проблема: Не создаются файлы кеша

Проверить:

```
# Количество файлов
ssh a1182962@141.8.192.186 "find
```

```
/home/a1182962/domains/dev.maslovka.org/public_html/wp-content/cache/supercache/ -type f | wc -l"
```

```
# Если 0 – проблема с созданием
```

Решение:

1. Очисти весь кеш (вкладка "Содержимое" → "Удалить кеш")
2. Проверь что не залогинен (открой инкогнито)
3. Открой главную страницу
4. Проверь снова количество файлов - должен появиться хотя бы 1

Проблема: "Права на запись должны быть обновлены" (для режима Эксперт)

Решение А (простое):

- Переключись на "Простой" режим вместо "Эксперт"

Решение В (если нужен Эксперт):

- Прокрути в самый низ страницы "Дополнительно"
- Нажми кнопку **"Обновить правила Mod_Rewrite"**
- Плагин сам добавит нужные правила в .htaccess

Проблема: Preload не работает

Проверить:

```
# Проверить wp-cron
ssh a1182962@141.8.192.186 "/home/a1182962/wp-cli.phar --
path=/home/a1182962/domains/dev.maslovka.org/public_html cron event list |
grep preload"
```

Решение:

1. Вкладка "Preload"
2. Нажми "Предварительно кэшировать сейчас" вручную
3. Подожди пока завершится процесс

Проблема: Сайт показывает устаревший контент

Решение:

```
# Очистить весь кеш
ssh a1182962@141.8.192.186 "/home/a1182962/wp-cli.phar --
path=/home/a1182962/domains/dev.maslovka.org/public_html cache flush"

# Или через админку
# Настройки → WP Super Cache → Содержимое → Удалить кеш
```

Метрики для отслеживания

До оптимизации:

- 🕒 TTFB (Time to First Byte): **~100-500ms**
- 🕒 Полная загрузка: **~1-3 секунды**
- 🇮🇹 Размер кеша: **0-20KB** (пустой)

После оптимизации (ожидаемые значения):

- 🕒 TTFB: **~20-50ms** (первый визит) / **~5-15ms** (из кеша)
- 🕒 Полная загрузка: **~300-800ms**
- 🇮🇹 Размер кеша: **50-200MB** (в зависимости от количества страниц)

Инструменты для замера:

1. **Chrome DevTools** (F12) → Network tab
2. **GTmetrix**: <https://gtmetrix.com/>
3. **Google PageSpeed Insights**: <https://pagespeed.web.dev/>
4. **WebPageTest**: <https://www.webpagetest.org/>

Итоговая конфигурация (checklist)

После всех настроек у тебя должно быть:

Основные настройки:

- ✅ Кеширование включено
- ✅ Метод: Простой (или Эксперт с настроенным .htaccess)
- ✅ Отключено для авторизованных пользователей
- ✅ Не кешируются страницы с GET параметрами
- ✅ Сжатие gzip включено
- ✅ Авто перестройка кеша
- ✅ 304 браузерное кеширование

Расширенные:

- ✅ Поддержка мобильных устройств
- ✅ Дополнительная сверка кеша
- ✅ Обновление при новых комментариях

Garbage Collection:

- ✅ Таймаут: 1800 секунд (30 минут)
- ✅ Планировщик: Таймер, 600 секунд (10 минут)

Preload:

- ☒ Обновление каждые 600 минут (10 часов)
- ☒ Кешировать все записи (0)
- ☒ Режим предзагрузки включен
- ☒ Предзагрузка таксономий включена

Параметры отслеживания:

- ☒ Игнорирование utm, fbclid и других параметров

Что дальше (следующие оптимизации)

После настройки WP Super Cache:

1. **Настроить CDN** (Cloudflare или собственный)
2. **Оптимизировать изображения** (WebP, lazy loading)
3. **Добавить Redis Object Cache** (для кеширования запросов к БД)
4. **Настроить OPcache** (кеширование PHP)
5. **Минифицировать CSS/JS** (через плагины или build процесс)

См. файл [PERFORMANCE_OPTIMIZATION.md](#) для полного плана оптимизации.

Полезные ссылки

- [WP Super Cache документация](#)
- [FAQ](#)
- [GitHub репозиторий](#)
- [Форум поддержки](#)

Дата последнего обновления: 19 декабря 2025

Автор инструкции: GitHub Copilot

Статус: ☒ Готово к использованию