

Конфигурация PHP для DokuWiki

DokuWiki будет работать «из коробки» практически при любой установке PHP, а [установщик](#) проверит доступность минимально необходимого набора функций PHP.

На этой странице даны советы по настройке параметров PHP для влияния на функциональность, производительность, безопасность и переносимость DokuWiki.

Более подробную информацию можно найти в онлайн- [PHP manual](#).

PHP-расширения

Ниже вы можете найти, какие расширения должны быть включены для вашей настройки PHP. В некоторых системах эти расширения могут быть установлены отдельно, другие системы могут объединить их в один установочный пакет и их нужно будет включить только в `php.ini`

необходимый

Эти расширения обычно включены по умолчанию.

- json- JSON декодирование, кодирование
- pcntl- Обработка регулярных выражений
- session- Обработка сеанса

рекомендуется

- bz2- обработка сжатия для файлов Attic, установка плагинов
- gd- изменение размера изображения
- intl- обработка интернационализации, например, для сортировки с учетом локали
- mbstring- более быстрая обработка UTF-8
- openssl- SSL/TLS-соединения, используемые для загрузки плагинов
- zlib- обработка сжатия для файлов Attic, установка плагинов

особые случаи использования

Следующие расширения часто используются плагинами и могут оказаться полезными.

- pdo_sqlite - используется плагином [sqlite](#)
- xml

php.ini

Основным средством настройки PHP является файл конфигурации (`php.ini`).

Для версий PHP с серверным модулем этот файл загружается только один раз при запуске веб-сервера. Для версий CGI и CLI это происходит при каждом вызове.

Этот файл содержит список директив, которые управляют тем, как функционирует PHP. Вы можете посмотреть страницу [директив php.ini](#) в Интернете для подробного описания этих директив.

Хотя в большинстве случаев DokuWiki будет работать «из коробки» с типичными настройками PHP дистрибутива, ряд параметров конфигурации имеет для DokuWiki особое значение.

С другой стороны, всегда предупреждайте, что то, что может быть хорошим или даже рекомендуемым значением для DokuWiki, может в некоторых случаях нарушить работу других приложений PHP, которые вы также размещаете. Это особенно верно при включении директив, которые повышают безопасность PHP, при наличии других приложений PHP, которые полагаются на небезопасные функции PHP, такие как register_globals и т. д.

short_open_tag

Разрешить <? тег. В противном случае распознаются только теги <?php и .<script>

DokuWiki будет работать с short_open_tag отключенной функцией.

- Предлагается: short_open_tag = Off

Однако следует отметить, что некоторые [templates](#) or [plugins](#) могут полагаться на установку этой функции в значение On.

open_basedir

Это ограничивает доступ вашего процесса PHP к настроенным путям. Это подразумевает дополнительную функцию безопасности, но на нее не следует полагаться. Это отключит кэширование readdir и может значительно замедлить обработку PHP. Подробности см. в разделе [производительность](#).

- Предлагается: open_basedir =

output_buffering

Буферизация вывода позволяет отправлять строки заголовков (включая файлы cookie) даже после отправки основного содержимого. DokuWiki будет работать с любой настройкой.

- Предлагается: output_buffering = Off

output_handler

Redirect all the output of all scripts to a function. Setting output_handler automatically turns on output_buffering.

Перенаправить весь вывод всех скриптов в функцию. Настройка `output_handler` автоматически включается `output_buffering`.

Этот параметр можно использовать для автоматического сжатия всего контента перед отправкой его в браузер клиента. Это может создать проблемы с изображениями и загрузками. Вместо этого включите функцию DokuWiki [gzip_output](#).

- Предлагается: `output_handler` =

zlib.output_compression

Обеспечивает прозрачное сжатие выходных данных с помощью библиотеки zlib. В целом настройка `zlib.output_compression = On` работает довольно хорошо с DokuWiki.

Однако DokuWiki поддерживает сжатие выходных данных с помощью опции [gzip_output](#).

- Предлагается: `zlib.output_compression = Off`

implicit_flush

Сообщает PHP, что нужно указать выходному слою автоматически очищать себя после каждого выходного блока. Включение этой опции имеет серьезные последствия для производительности и обычно рекомендуется только для целей отладки. DokuWiki заботится об очистке буфера при необходимости.

- Предлагается: `implicit_flush = Off`

allow_call_time_pass_reference

Предупреждать ли, когда аргументы передаются по ссылке во время вызова функции, так как этот метод устарел. Аргументы, которые должны передаваться по ссылке, следует указывать в объявлении функции, а не во время вызова функции.

- Предлагается: `allow_call_time_pass_reference = Off`

max_execution_time

Максимальное время выполнения каждого скрипта в секундах.

- Предлагается: `max_execution_time = 30`

max_input_time

Максимальное количество времени, которое каждый скрипт может потратить на анализ данных запроса.

- Предлагается: `max_input_time = 60`

max_input_vars

При определенных условиях (например, при работе с большими таблицами с помощью плагина editable) быстро достигается значение по умолчанию 1000.

- Предлагается: `max_input_vars = 10000`

memory_limit

Максимальный объем памяти, который может потреблять скрипт.

- Предлагается: `memory_limit = 128M`

error_reporting

О каких ошибках сообщать.

- Предлагается: `error_reporting = E_ALL & ~E_NOTICE`

display_errors

Распечатайте ошибки (как часть вывода). Для рабочих веб-сайтов настоятельно рекомендуется отключить эту функцию и вместо этого использовать регистрацию ошибок.

- Предлагается: `display_errors = Off`

display_startup_errors

Выводит ошибки, возникающие во время последовательности запуска PHP (`display_errors` не имеет никакого контроля над ними). Настоятельно рекомендуется отключать `display_startup_errors`, за исключением случаев отладки.

- Предлагается: `display_startup_errors = Off`

log_errors

Записывать ошибки в файл журнала. Также настроить `error_log` соответствующим образом.

- Предлагается: `log_errors = On`

variables_order

Порядок, в котором PHP регистрирует переменные GET, POST, Cookie, Environment и встроенные переменные (G, P, C, E и S соответственно).

- Предлагается: variables_order = «EGPCS»

register_argc_argv

Следует ли объявлять переменные argv & argc (которые будут содержать информацию GET).

- Предлагается: register_argc_argv = Off

file_uploads

Разрешать ли загрузку файлов по протоколу HTTP.

- Предлагается: file_uploads = On

upload_max_filesize

Максимально допустимый размер загружаемых файлов. Он должен соответствовать тому, что вы ожидаете от максимального размера загружаемых медиафайлов.

session.use_cookies

Использовать ли файлы cookie.

- Предлагается: session.use_cookies = 1

session.cache_limiter

Определяет, какие заголовки HTTP управления кэшем отправляются клиенту. Эти заголовки определяют правила, по которым содержимое страницы может кэшироваться клиентом и промежуточными прокси-серверами. Допустимые значения: none, nocache (по умолчанию) private, private_no_expire или public. Установка ограничителя кэша на nocache запрещает любое кэширование клиента/прокси. Значение public разрешает кэширование прокси-серверами и клиентом, тогда как private запрещает кэширование прокси-серверами и разрешает клиенту кэшировать содержимое. В приватном режиме заголовок Expire, отправляемый клиенту, может вызвать путаницу в некоторых браузерах, включая Mozilla. Вы можете избежать этой проблемы, используя private_no_expire режим. Заголовок expire никогда не отправляется клиенту в этом режиме.

В случае проблем с кэшированием (т. е. вы редактируете страницу, но изменения не отображаются при просмотре этой страницы) вы можете попробовать установить: session.cache_limiter = nocache

extension

Динамические расширения. Рекомендации см. [PHP extensions](#) в разделе PHP-расширения выше.

From:

<https://wwoss.ru/> - **worldwide open-source software**



Permanent link:

<https://wwoss.ru/doku.php?id=wiki:install:php>

Last update: **2025/02/22 18:29**