

Инструменты отладки

В «ДокуВики» есть инструменты отладки. Некоторые необходимо настроить.

Включить отладку

Если опция конфигурации [allowdebug](#) включена, вы можете:

- Добавляем `?do=debug` или `&do=debug` к URL-адресу, чтобы «ДокуВики» выводила огромный список строк отладки.
- (Вплоть до версии *Hogfather 2020*: посмотрите выходные данные `dbglog()` в `data/cache/debug.log`)
- Получать дополнительные сообщения об ошибках, например, от некоторых серверов аутентификации.
- Добавление `?httpdebug` или `&httpdebug` к URL-адресу для включения вывода отладки в HTTPClient DokuWiki
- Вернуть несколько сообщений, если таблицы стилей отсутствуют
- Получайте дополнительные комментарии в выводе средства визуализации `xhtml`.
- Получите несколько резервных попыток в скомпилированном файле JavaScript с помощью `js.php`
- Получите отладку в некоторых других местах, а также в плагинах или шаблонах, см. поиск кода для использования [разрешенного дебага](#).

Включение ведения журнала

С момента выпуска Игоря 2022 года ::

«ДокуВики» имеет функцию ведения журналов. Какая информация не регистрируется, можно установить с помощью [dontlog](#) настройки конфигурации.

- По умолчанию регистрируются только сообщения об ошибках и устаревшие сообщения, а журнал отладки отключен.
- Вы также можете включить журнал отладки для получения дополнительных результатов отладки.
- Через интерфейс администратора вы можете использовать [LogViewer](#). Если доступ к LogViewer невозможен, вы можете найти файлы журналов в вашей файловой системе, например, в формате `[wiki_folder]/data/log/error/<date>.log`.

Получить информацию о странице и пользователе

Добавление `?do=check` или `&do=check` к URL запускает [проверку режима действия](#), которая предоставляет некоторую информацию о разрешениях пользователей и настройке «ДокуВики».

Отладка TaskRunner

См. [информацию об отладке](#) на странице запуска задач.

Функции отладки

- **[dbg\(\\$msg\)](#)**

Вставьте блок кода на вики-страницу, заполненный `$msg` в виде обычного текста.

Добавление дополнительного аргумента `dbg($msg, $hidden=true)` вернет `$msg` в виде комментария html.

- **[dbglog\(\\$msg\)](#)**

Записывает `$msg` в журнал отладки, отображаемый в [LogViewer](#), который сохраняется `[wiki_folder]/data/log/debug/<date>.log` когда `debug` этот флажок снят в [dontlog](#) настройке.

(Вплоть до версии *Hogfather 2020*: он записывался в журнал в каталоге кеша `data/cache/debug.log` когда опция [allowdebug](#) включена.) Имейте в виду, что журнал не очищается автоматически.

Используйте `dbglog($msg, 'Header above log message')` для включения заголовка над `$msg`.

- **[dbg_deprecated\(\\$alternative = ''\)](#)**

Функция для обозначения устаревших функций. Каждый вызов записывается в устаревший журнал, отображаемый в [LogViewer](#), если `deprecated` это флажок не установлен в [dontlog](#) настройке. В сообщении журнала упоминается: имя устаревшей функции, использующая ее функция, ее файл и строка, а также время `$alternative` предоставления.

- **[dbg_backtrace\(\)](#)**

Напечатайте перевернутую, красиво напечатанную обратную трассировку.

- **[debug_guard\(\\$data\)](#)**

Когда вы пишете собственную функцию отладки, используйте эту функцию для удаления всех данных из массива, ключ которого указывает на конфиденциальные данные.

Плагин «Страж»

Плагин [Sentry Plugin](#) позволяет регистрировать обширную информацию в [Sentry](#) сервисе. Это особенно полезно при запуске бета-тестирования в реальной среде.

Смотрите также

- Конфигурацию опции [allowdebug](#)
- Конфигурацию опции [dontlog](#) и [LogViewer](#) для включения журналов
- [Кэширование](#)

- У меня [пустая страница](#) или [ошибка 500](#) (внутренняя ошибка сервера)

From:

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

Permanent link:

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

Last update: **2023/08/31 19:09**

