Обслуживание

Вот несколько советов по автоматизации некоторых повседневных операций по обслуживанию, необходимых или рекомендуемых для DokuWiki.

См. также плагины: cleanup и clearhistory

Поддерживайте черный список в актуальном состоянии

См. раздел «Черный список» о том, как настроить cronjob для поддержания черного списка антиспама в актуальном состоянии.

Скрипт автоматической очистки

Рекомендуется настроить процесс очистки для загруженных DokuWiki. Следующий скрипт оболочки Bash (Unix shell) служит примером. Он удаляет старые ревизии с чердака, служит примером. Он удаляет старые ревизии из cache[1].

cleanup.sh

```
#!/bin/bash

cleanup()
{
    local data_path="$1"          # полный путь к каталогу данных вики
    local retention_days="$2"     # количество дней, по истечении которых
старые файлы должны быть удалены

    # удалить файлы старше ${retention_days} дней из attic и media_attic
(старые ревизии)
    find "${data_path}"/{media_,}attic/ -type f -not -name _dummy -
mtime +"${retention_days}" -delete

    # удалить устаревшие файлы блокировки (файлы, которым 1-2 дня)
    find "${data_path}"/locks/ -name '*.lock' -type f -mtime +1 -delete

    # удалить пустые каталоги
    find
"${data_path}"/{attic,cache,index,locks,media,media_attic,media_meta,me
ta,pages,tmp}/ \
        -mindepth 1 -type d -empty -delete

    # удалить файлы старше ${retention_days} дней из кэша
    if test -n "$(find "${data_path}"/cache/?/ -maxdepth 1 -print -quit
&> /dev/null)"
```

```
    then
        find "${data_path}"/cache/?/ -type f -not -name _dummy -mtime
+"${retention_days}" -delete
    fi
}


# очистка установок DokuWiki (путь к datadir, количество дней)
# несколько примеров:

cleanup /home/user1/htdocs/doku/data    256
cleanup /home/user2/htdocs/mywiki/data  180
cleanup /var/www/superwiki/data         180
```

Для автоматического запуска настройте cronjob. Следующий пример вызывает скрипт каждый день через 7 минут после полуночи. Для запуска от имени пользователя без прав root удалите `root`.

```
7 0 * * *   root  /full/path/to/cleanup.sh
```

Обязательно настройте все правильно — вы ведь не хотите удалить не то, не правда ли?

Windows -- warmzip

Скрипт для очистки старых файлов в системах Windows — waRmZip, доступен here on SourceForge. Напишите пакетный файл для его вызова и запланируйте его запуск на каждый день. И как говорит человек: «Обязательно настройте все правильно»' 

Я воспользовался приведенным выше предложением waRmZip и написал этот пакетный файл — возможно, он поможет.

Мой любимый способ запуска заданий cron в Windows — PyCron.

dw-cleanup.bat

```
@echo off
set waRmZip="c:\Program Files\waRmZip\waRmZip.wsf"
set wikiHome="c:\path\to\htdocs\wiki\data"

rem Переместить файлы attic старше 30 дней в архивное местоположение
%waRmZip% %wikiHome%\attic /ma:30 /md:%wikiHome%_archive\attic /r /q

rem Параметр: удалить файлы attic старше 30 дней
rem %waRmZip% %wikiHome%\attic /da:30 /dc /r /q

rem Удалить пустые каталоги attic; waRmZip требует флаг /da при использовании
rem /df, поэтому добавьте фильтр для *.zzz, чтобы /da не удалял никакие
файлы
```

```
%waRmZip% %wikiHome%\attic /r /da:31 /df /fo:*.zzz /q

rem Удалить устаревшие файлы блокировки
%waRmZip% %wikiHome%\locks /da:1 /fo:*.lock /r /q

rem Удалить пустые каталоги
%waRmZip% %wikiHome%\pages /da:365 /df /fo:*.zzz /r /q
```

Windows -- batch script

Это еще один скрипт командной оболочки Windows для поддержки вашей базы dokuwiki в среде Windows. Скрипт использует бесплатную и открытую утилиту find, которую можно получить черезhttp://gnuwin32.sourceforge.net/

Все пути считываются из файла конфигурации DokuWiki. Файлы, которые нужно удалить, можно показать перед удалением, чтобы предотвратить случайное удаление файлов.

maintain_dokuwiki.cmd

```
@echo off
setlocal

REM Этот скрипт выполняет некоторые основные функции обслуживания DokuWiki

REM Авторские права (C) 2012 Питер Мосманс

REM Эта программа является свободным программным обеспечением: вы можете
распространять ее и/или изменять
REM в соответствии с условиями GNU General Public License, опубликованной
REM Free Software Foundation, либо версии 3 Лицензии, либо
REM (по вашему выбору) любой более поздней версии.

REM Эта программа распространяется в надежде, что она будет полезна,
REM но БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ; даже без подразумеваемой гарантии
REM ТОВАРНОЙ ПРИГОДНОСТИ или ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ ЦЕЛИ.
Подробнее см. в REM GNU General Public License.

REM Вы должны были получить копию GNU General Public License
REM вместе с этой программой. Если нет, см. <http://www.gnu.org/licenses/>.

REM Пожалуйста, свяжитесь со службой поддержки по адресу go-forward.net для
вопросов и/или отзывов



REM Последнее изменение: 02-05-2012 (Питер Мосманс)
set NAME=maintain_dokuwiki
set VERSION=0.13
```

```
REM путь к файлу конфигурации dokuwiki, заключенный в двойные кавычки
set DOKUWIKICONFIG="\full\filename\of\your\dokuwiki\conf\local.php"
REM сохранить все файлы, которые моложе DAYSTOKEEP дней
set DAYSTOKEEP=31
REM установите значение true, если вы хотите отобразить результаты и сделать
паузу перед удалением любых файлов
set SHOWRESULTSFIRST=true
set FIND=c:\tools\find.exe
set TEMPFILE=%TMP%\%NAME%.tmp

REM посмотрите, все ли инструменты присутствуют
for %%i in (%FIND%) do (
    if not exist %%i (
        echo sorry, could not find %%i - exiting
        echo you can obtain the free GNU tools from
gnuwin32.sourceforge.net
        exit /b
    )
)

REM проверяем, можно ли прочитать файл конфигурации dokuwiki
if not exist %DOKUWIKICONFIG% (
    echo sorry, could not find DokuWiki config at %DOKUWIKICONFIG% -
exiting
    exit /b
)

REM извлекаем правильные пути из файла конфигурации
for /f "usebackq delims=' tokens=2,4" %%i in (%DOKUWIKICONFIG%) do (
    if /i "%%i"=="datadir" set DOCUMENTROOT=%%j
    if /i "%%i"=="olddir" set ATTICDIR=%%j
    if /i "%%i"=="cachedir" set CACHEDIR=%%j
    if /i "%%i"=="lockdir" set LOCKDIR=%%j
)
if "%DOCUMENTROOT%" == "" (
    echo sorry, could not find datadir variable in %DOKUWIKICONFIG%,
exiting...
    exit /b
)

REM использовать значения по умолчанию, если пути не указаны
if /i "%ATTICDIR%" == "" set ATTICDIR=%DOCUMENTROOT%/attic
if /i "%LOCKDIR%" == "" set LOCKDIR=%DOCUMENTROOT%/lock
if /i "%CACHEDIR%" == "" set CACHEDIR=%DOCUMENTROOT%/cache

REM удалить файлы старше DAYSTOKEEP дней из attic
%FIND% "%ATTICDIR%" -type f -mtime +%DAYSTOKEEP% -print > %TEMPFILE%
REM снять блокировки старше одного дня
%FIND% "%LOCKDIR%" -name "*.lock" -type f -mtime +1 -print >>
%TEMPFILE%
REM удалить файлы кэша старше DAYSTOKEEP
```

```
%FIND% "%CACHEDIR%" -type f -mtime +%DAYSTOKEEP% -print >> %TEMPFILE%

REM показать результаты, если таковые имеются
for /f "usebackq" %%i in (`%FIND% "%TMP%" -size +1 -name %NAME%.tmp`)
do (
    if /i "%SHOWRESULTSFIRST%"=="TRUE" (
        echo files to be deleted:
        type %TEMPFILE%
        pause
    )
    for /f "delims=#" %%i in (%TEMPFILE%) do del "%%i"
)

REM очистить
del /f /q %TEMPFILE%

endlocal
```

Поддержание чистоты на детской площадке

Чтобы поддерживать чистоту playground и других страниц вики , используйте задание cron, например, каждые 30 минут, которое восстанавливает Playground и другие страницы до их первоначального содержимого.

Пример: восстанавливать игровую площадку каждые 30 минут::

```
0,30 * * * * cp -f /path/to/savedwiki/data/pages/playground/playground.txt
/path/to/dokuwiki/data/pages/playground/
```

Пример: восстановление всех страниц в пространстве имен «wiki» каждые 30 минут:

```
0,30 * * * * cp -rf /path/to/savedwiki/data/pages/wiki/
/path/to/dokuwiki/data/pages/wiki/
```

Проблемы с плагином CAPTCHA

Использование плагина CAPTCHA и обслуживаниярекомендуемого метода для поддержания чистоты игровой площадки может привести к невозможности редактирования игровой площадки.

Если это произошло, проблему можно легко решить, удалив соответствующие файлы Playground в папке meta с помощью следующего задания cron.

Пример: удаляет метафайлы Playground каждые 30 минут:

```
0,30 * * * * rm -f /path/to/dokuwiki/data/meta/playground/playground.*
```

Когда cronjob недоступен

Если ваш хостинг не позволяет использовать cronjobs, рассмотрите возможность использования плагин cronojob.

Обсуждение

Не могли бы вы предоставить PHP-версии этих скриптов для использования с плагином cronojob?

> Что касается приведенного выше скрипта очистки, который использует время изменения файла (mtime), не будет ли безопаснее использовать временную метку в имени файла, чтобы определить, следует ли удалять файл на чердаке или нет?

С одной стороны, я бы сказал, что это можно сделать, но, конечно, это сложнее в настройке. Для многих установок будет нормально использовать mtime. С другой стороны, некоторые могут захотеть убедиться, что они очищают старые файлы, несмотря ни на что (например, файлы, оставшиеся после сбоя или критической ошибки PHP).

Может ли кто-нибудь добавить соответствующую строку для обслуживания кэша в скрипт Windows waRmZip?

Справляется ли cleanup Plugin со всеми вышеперечисленными задачами? Будет ли он рекомендован вместо запуска этих скриптов?

Это пример PHP-скрипта для очистки старых файлов кэша. Полезно, когда .sh недоступен для запуска.

cleanup.php

```
<?php
/*
 * mrlemonade ~
 */
function getFilesFromDir($dir) {
  $files = array();
  if ($handle = opendir($dir)) {
    while (false !== ($file = readdir($handle))) {
        if ($file != "." && $file != "..") {
            if(is_dir($dir.'/'.$file)) {
```

```
                $dir2 = $dir.'/'.$file;
                $files[] = getFilesFromDir($dir2);
            }
            else {
              $files[] = $dir.'/'.$file;
            }
        }
    }
    closedir($handle);
  }
  return array_flat($files);
}
function array_flat($array) {
  foreach($array as $a) {
    if(is_array($a)) {
      $tmp = array_merge($tmp, array_flat($a));
    }
    else {
      $tmp[] = $a;
    }
  }
  return $tmp;
}

// Определяем папку для очистки
$captchaFolder = 'data/cache';
// Здесь можно определить, через сколько
// дней файлы должны быть удалены
$expire_time = 5;
// Найти все файлы указанного типа
foreach (getFilesFromDir($captchaFolder) as $Filename) {
        // Считать время создания файла
        $FileCreationTime = filectime($Filename);
        // Вычислить возраст файла в секундах
        $FileAge = time() - $FileCreationTime;
        // Файл старше указанного промежутка времени?
        if ($FileAge > ($expire_time*60*60*24 )) {
            // Теперь что-нибудь сделаем со старыми файлами...
            print "The file $Filename is older than $expire_time days
\n";
            // Например, удаление файлов:
            // unlink($Filename); ;
        }
}
echo 'ran';
?>
```

используйте это на свой страх и риск. — S.C. Yoo *2012/02/10 12:49*

Привет, я хотел бы добавить, что это хорошая идея — очищать потерянные метаданные, как вы думаете? Я делаю следующее (в скрипте R):

рекурсивно вывести список всех файлов в каталоге страниц добавьте в этот список столбец «pagename», который снова будет содержать имя файла, но без базового каталога. в имени страницы замените '/' (или '\') на ':' и удалите расширение файла сделать то же самое для мета-каталога + исключить некоторые дополнительные файлы удалить все записи из мета-списка, из которых имя страницы находится в списке страниц удалить все файлы, оставшиеся в мета-списке Конечно, можно было бы добавить ограничение по времени, чтобы метаданные не использовались немедленно.

Clemo *2016/09/23 sometime*

[1)]

Обсуждение обслуживания кэша можно также найти на форуме.

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

Permanent link:
https://wwoss.ru/doku.php?id=wiki:tips:maintenance

Last update: **2025/02/16 21:29**