Sitemap CLI

Plugin for Joomla! 4, 5 & 6

This manual documents **version 1.x** of the Joomla! extension.

https://www.aimy-extensions.com/joomla/sitemap-cli.html

1 Introduction

The Joomla! console plugin Aimy Sitemap CLI allows to use certain features of Aimy Sitemap on the command line. This is especially useful when combined with a command scheduler like *cron* or Microsoft's *task scheduler*.

2 Supported Joomla! Versions

Aimy Sitemap CLI supports Joomla! 4.0 and up, 5.0 and up as well as 6.0 and up.

3 Requirements

Aimy Sitemap CLI requires at least PHP 8.0 and Aimy Sitemap v39.0.

Some commands (i.e. `crawl-website`) require the PRO version of Aimy Sitemap to be installed.

More specific requirements are documented along with the commands.

4 Installation

The installation of the extension follows the common Joomla! procedures.

In case you are not familiar with these procedures, proceed as follows:

1. Download the extension's ZIP archive
2. Log into your Joomla! backend as "Super User"
3. From the menu, choose "*System*" → "*Extensions*" (below "*Install*")
4. Click on the "*Or browse for file*" button and select the ZIP archive

The extension's archive will be uploaded and installed afterwards.

For further information, please have a look at the Joomla! documentation Installing an Extension.

NOTE: All plugins are disabled by Joomla! when installed for the first time. To enable Aimy Sitemap CLI, proceed with a click on the "Enable Aimy Sitemap CLI now!" button, set the plugin *State* to "Enabled" and save your changes.

5 Configuration

At this time, there are no configuration options to be set for the plugin. All tuneables are either set in Aimy Sitemap's options or passed as options directly on the command line.

6 Available Commands

All commands are made available using the Joomla! CLI and share the "`aimysitemap`" prefix, i.e.:

```
php cli/joomla.php aimysitemap:crawl-website
```

6.1 `crawl-website` Command

The `crawl-website` command allows to crawl a website just like Aimy Sitemap's crawler does when started in the Joomla! backend of your website.

This command requires the PRO version of Aimy Sitemap to be installed and can be used as a replacement for the obsolete `crawl-website.php` script that is included in Aimy Sitemap PRO.

`crawl-website` supports the following options:

`--write-sitemap`

Write the XML sitemap to disk after the crawl. Settings made for Aimy Sitemap apply.

`--notify-indexnow`

Notify `indexnow.org` if changes are found during the crawl. Settings made for Aimy IndexNow apply.

When this option is given, Aimy IndexNow PRO is required.

`--debug`

Enables Aimy Sitemap's debugging mode. The website's index will not be updated, but a protocol of the crawl will be created for debugging purposes.

A complete invocation may look like this (with the current working directory being your website's root directory):

```
php cli/joomla.php aimysitemap:crawl-website --write-sitemap
```

7 Scheduling Commands for Periodic Execution

In order to schedule an automatic command execution, a **system service** is required. On a Unix or Unix-like system (i.e. Linux, FreeBSD, Mac OS X, Solaris) **cron** is usually used for such tasks. On a Microsoft Windows server **Task Scheduler** is available. Most hosters provide an interface to schedule jobs for periodical execution as well.

Whatever way you choose to start the periodic crawl, be sure to always select the **Command Line Interface** (CLI) version of the **PHP interpreter** to run the script. For security reasons, the script will refuse to run if the Common Gateway Interface (CGI) version is used as an interpreter.

If in doubt, ask your webhoster, webmaster or administrator for the path of the command line interface PHP interpreter.

7.1 Setting Up a Cron-Job (Unix)

If you have a Unix or Unix-like server with shell access, log into your server using the user account that should be running the command periodically. This may be the same user your Joomla! website is running as.

First, determine the path of your PHP **CLI** interpreter. If you do not already know where the binary is located, run the following command to find out:

```
which php
```

Then, add a new cron-job running the following command:

```
crontab -e
```

Your default editor will open up, allowing you to enter a new cron-job. Each cron-job consists of the following six required fields: *minute*, *hour*, *day of month*, *month*, *day of week*, *command*. The special value of "*" can be used as placeholder for any valid value in one of the first five fields, i.e. any hour or any day of the week.

Set up your cron-job to be run at your desired time and be sure to use the full path to your PHP binary followed by the full script path as a *command*.

To suppress all non-error output during execution, you can pass either `-q` or `--quiet` as an argument.

Save your changes and exit your editor afterwards.

Example:

To run your periodic crawls each night at 23:55 you would enter:

```
55 23 * * * $php /path-to/cli/joomla.php aimysitemap:crawl-website
```

NOTE: Replace `$php` with the full path of your PHP interpreter (i.e. `/usr/bin/php`) and `/path-to/cli/joomla.php` with the full path to `cli/joomla.php` within your website's root directory.

7.2 Setting Up a New Task (Windows)

On a Microsoft Windows server, **Task Scheduler** can be used to schedule automatic command executions periodically. To start the utility, click "Start" → "Accessories" → "System Tools" → "Task Scheduler".

Create a new "Basic Task" and enter your desired configuration.

When asked what *Action* should be performed, choose "*Start a program*". Choose your PHP interpreter as a program and enter the full path to the `joomla.php` script within the `cli` directory beneath your website's root directory in the "*Add arguments*" field below.

To suppress all non-error output, add `--quiet` as an argument (after the script's path) in the "*Add arguments*" field.

Verify your settings in the next mask and save the task by clicking "*Finish*".

7.3 Using Your Hosters Interface

Most hosters provide an interface to the periodic command execution system services in their management frontends. First, verify that your hoster provides such an interface for your website and have a look at your hoster's documentation.

When asked for it, choose the highest version of PHP available supported by your Joomla! version (i.e. 8.4) as an interpreter and use the full path of the `joomla.php` script along with the desired parameters as periodic task.

By default, PHP processes started from the command line do not have a time limit set. However, if your hoster has overwritten this default setting (`max_execution_time`), be sure to either restore the default (0) or at least set the highest limit available for the directive (>60) to decrease the probability that the script will get aborted by the system.

8 Copyright & Trademark Notice

The Joomla!® name and logo are trademarks of Open Source Matters, Inc. in the United States and other countries.

Mentioned hard- and software as well as companies may be trademarks of their respective owners. Use of a term in this manual should not be regarded as affecting the validity of any trademark or service mark. A missing annotation of the trademark may not lead to the assumption that no trademark is claimed and may thus be used freely.