

Silktide Umbraco plugin requirements

Introduction:

Silktide is a company that helps customers improve their websites by analyzing the content of their website in a real browser to give recommendations on how they can improve their SEO, check for mistakes in content and ensure the page is accessible. Silktide tests these sites every 5 days.

Silktide makes use of plugins for popular CMS's to allow our customers to inform us when a page has changed so we check it in under a minute instead of every 5 days. They also use this plugin so when they have a problem, they can click within our tool on the page and be taken directly to the edit screen within their CMS (if they're logged in) to fix the issue we flag.

Silktide have been requested by one of our customers to build an Umbraco plugin.

Requirements:

We do not require multi language support and there is a 1-1 relationship between API key and website.

Settings area

The plugin will need a settings screen within the Umbraco settings area to allow the user to provide an API key (32 chars) from our platform. This is used later on when the API call is made. This should also link to an area of our support site (<https://support.silktide.com/guides/cms-install-umbraco-guide/>) so we can provide a guide for them.

On publish of a new or updated page

When a page is updated or added, an API call should be made to our platform that informs us of the URLs of the pages that have been changed.

Method: POST

Endpoint: <https://api.silktide.com/cms/update>

Body: `apiKey=[as provided]&urls%5B0%5D=https%3A%2F%2Fexample.com%2Fmyfirstpage`

The URLs are just an array of urls.

You do not need to do anything specific if the hook fails or succeeds. The user does not need to be notified at all.

On a page

On each page of the site, we need a meta tag injecting.

E.g. `<meta name="silktime-cms" content=".....">`

This meta tag will hold a JSON encoded block of text that contains the "editorUrl" (i.e the admin URL for editing of the page). This allows a user of ours to click the "Edit" button in our own platform, and get taken straight to the page editor inside Umbraco to fix issues we report to them. If they are not logged in, they will just need to login first. We don't do any form of auto login or anything special.

Ideally the content of this header should be hashed to be non-obvious to viewers of the CMS code. I'll post Richard's prior explanation in regards to the SiteCore integration here, as it's the same way we ideally wish to hash this

"This is then encrypted using OpenSSL using the AES-256-CBC encryption method to the API key (line 114). We then take the "ciphertext_raw" and create a sha256 hash of it using the API Key. We then take the ciphertext and the hmac (and the ciphertext's randomly generated initialization vector) and base64 encode it (line 121) before adding the header `<meta name="silktime-cms" content=".....">` to the outputted html.

If the OpenSSL encryption method isn't possible in Umbraco , or it is not possible to generate an "admin url" on the page, then this stage can be skipped: however, if there is another way this output can be achieved, please let me know and I'll discuss it with the team here.

"

The line numbers he refers to are an example in our Wordpress CMS plugin which I have attached in our email. You can probably copy the code of how this was done in that. If this is NOT possible, just let us know and we can decide on another hashing method.

Installation instructions

Installation instructions will be required so our development team can easily provide support for our customers when they're trying to install this plugin.

In addition to this, if a brief guide could be provided as to how we can make modifications to this plugin, that would be great. Just mostly an overview of what the files you have created are doing. We are normally Mac based PHP developers, so working in the microsoft eco-system is not where we are comfortable.

To conclude

The plugin is very simple and just needs an interface to store an API key, a hook to be sent to our system when a page is published, and finally a meta-tag applying to each web page in the CMS so we can get the editor URL for that page.

Ideally we need this building prior to the 26th of September. If this is not possible, we can figure out other arrangements.