extension "extensible_sitemap"

Christian Zenker

extension "extensible_sitemap"

Christian Zenker
Copyright © 2010 Christian Zenker

Abstract

extensible_sitemap is an implementation for sitemaps for search engines as specified by http://www.sitemaps.org/protocol.php. It is extensible for you to include pages generated by your extensions very easily.

Table of Contents

List of Tables

List of Examples

Chapter 1. Introduction

What does it do?

This extension is yet another sitemap generator for search engines.

The main thing that makes this extension special from all the other implementations is that it is extensible. With the other extensions you have no possibility to add pages generated by your extension. (except by *really nasty* XCLASSing, but what if there are two or more extensions trying to XCLASS?) So this extension has an interface where you can add your extensions to the sitemap generator.

It is also possible to generate multiple sitemaps. So if you have a gallery extension with multiple galleries you could stuff all the gallery pages in the default sitemap and create an additional image sitemap for google to make the images available in the Google Image Search.

Feature List

This is a selection of features of this extension

- link sitemaps from extensions that generate pages dynamically
- extend sitemap with namespaces (for example for news, images, videos, mobile content, etc.)
- Use of eID, so the load on your server is minimized
- configuration via TypoScript
- native tt_news support

Sources

Home in forge	http://forge.typo3.org/projects/extension-extensible_sitemap
Distribution in TER	http://typo3.org/extensions/repository/view/extensible_sitemap/current/
Bugtracker	http://forge.typo3.org/projects/extension-extensible_sitemap/issues
Official Git Repository	http://github.com/czenker/extensible_sitemap

Chapter 2. Administration

QuickStart

If you are just keen on getting the extension running and don't care much on *why* it (-hopefully-) works, this is for you.

1. **Download and install the extension extensible_sitemap**

   ### Note

   If you are new to TYPO3 and don't know how to install an extension, the Web-Empowered Church has a nice tutorial on how to install a TYPO3 extension [https://webempoweredchurch.org/clientcenter/knowledgebase/213/How-Do-I-Install-an-Extension.html].

2. **Include the template `Auto Configuration (extensible_sitemap)` to the rootpage of your website**

   ### Note

   In TYPO3 4.4 this page is marked by a globe in the pagetree.

3. **View the sitemap**

   Well, actually this is all it takes. Have a look at `http://example.org/index.php?eID=extensible_sitemap` to see the sitemap for your website. If you have tt_news installed you can view a news sitemap at `http://example.org/index.php?eID=extensible_sitemap&sitemap=news`.

   ### Note

   Don't panik if you have tt_news configured, but just get an empty `<urlset>`-Tag. Only news from the last 7 days are displayed by default. This is because Google™ only indexes news if they are not older than 2 days. So displaying older records seems useless. You can configure this value though.

Configuration

All configuration of the extension is done via TypoScript inside `plugin.tx_extensiblesitemap`.

Table 2.1. Configuration for `plugin.tx_extensiblesitemap`

Property	Data type	Description
`default`	`sitemap-configuration`	This is the configuration for the default sitemap that will be displayed if you call `http://example.org/index.php?eID=extensible_sitemap`.

Property	Data type	Description
[sitemap]	sitemap-configuration	Additional sitemaps that might be called by visiting http://example.org/index.php?eID=extensible_sitemap&sitemap=**[sitemap]**

Table 2.2. Configuration for `sitemap-configuration`

Property	Data type	Description
[generator]	classname/ +config	You can define a list of multiple generators here that might return records for your sitemap. Each of them can be defined by a configuration array. The configuration options depend on the generator - so have a look at the corresponding documentation.

Example 2.1. Example configuration

```
plugin.tx_extensiblesitemap {
 # this is the first sitemap
 default {
  # this is a generator that generates a sitemap for all TYPO3 pages
  # it has no further configuration
  page = Tx_ExtensibleSitemap_Generator_Page_Recursive

  # additionally to the TYPO3 pages tt_news generates pages by its single
  # view there is a different class taking care of submitting the
  # corresponding data to the sitemap creator
  tt_news_article = Tx_ExtensibleSitemap_Generator_TtNews_SimpleArticle
  # this is some additional configuration for the class
  tt_news_article {
   pid_list = {$plugin.tt_news.pid_list}
   singlePid = {$plugin.tt_news.singlePid}
  }
 }

 # this is a second sitemap - completely independent from the first one
 news {
  # it uses yet another generator and has some configuration
  tt_news_article = Tx_ExtensibleSitemap_Generator_TtNews_NewsArticle
  tt_news_article {
   publicationName = {$company.name}
   maxAge = 365
   pid_list = {$plugin.tt_news.pid_list}
   singlePid = {$plugin.tt_news.singlePid}
  }
 }
}
```

Generators

Tx_ExtensibleSitemap_Generator_Page_Recursive

A generator for "extensible_sitemap" that indexes pages of the TYPO3 page-tree recursively.

Table 2.3. parameters for Tx_ExtensibleSitemap_Generator_Page_Recursive

Property	Data type	Description	Default
pidList	comma-seperated integer	A comma-seperated list of pageIds that serve as the root for the indexer. These pages and all below will be submitted to the sitemap	current

Example 2.2. Example using Tx_ExtensibleSitemap_Generator_Page_Recursive

```
plugin.tx_extensiblesitemap {
 default {
  # this is a very simple way to use the sitemap
  # if you call the eid-parameter on any pageId in the frontend it will
  # output a sitemap of this page and all its siblings
  page = Tx_ExtensibleSitemap_Generator_Page_Recursive
 }

 fixed-pid {
  # this will output a sitemap of the pages 42 and 1337 and all its
  # siblings no matter from which page you call it
  page = Tx_ExtensibleSitemap_Generator_Page_Recursive
  pidList = 42,1337
 }
}
```

Tx_ExtensibleSitemap_Generator_TtNews_SimpleArticle

a generator for "extensible_sitemap" that indexes tt_news articles. This is a basic version that does NOT extend the XML-Scheme by additional tags - so these are just the bare webpages created and do *not* generate a sitemap for google news .

Warning

Please notice that this is for tt_news from version 3.0.0 up - the parameters have changed with this version.

Table 2.4. parameters for `Tx_ExtensibleSitemap_Generator_TtNews_SimpleArticle`

Property	Data type	Description	Default
`singlePid`	`integer`	the page id of the single view	
`pid_list`	`comma-seperated integer`	the pids the records reside in (set to **`0`** to take all records)	`0`
`recursive`	`integer`	how many levels under the above given `pid_list` should be looked for records too	`0`
`defaultPriority`	`priority`	the default priority assigned to each item.	

Example 2.3. Example using `Tx_ExtensibleSitemap_Generator_TtNews_SimpleArticle`

```
plugin.tx_extensiblesitemap {
 default {
  # takes records from pages 42 and 1337 and links them
  # to the page 42
  news = Tx_ExtensibleSitemap_Generator_TtNews_SimpleArticle
  news {
   singlePid = 42
   pid_list = 42,1337
   # this has to be set always
   publicationName = TYPO3 News
  }

  news2 = Tx_ExtensibleSitemap_Generator_TtNews_SimpleArticle
  news2 {
   # you could copy the configuration from the tt_news-
   # plugin configuration
   pid_list < plugin.tt_news.pid_list
   recursive < plugin.tt_news.recursive
   singlePid < plugin.tt_news.singlePid

   publicationName = TYPO3 News
  }

  news3 = Tx_ExtensibleSitemap_Generator_TtNews_SimpleArticle
  news3 {
   # ... or even better if you take the constants
   pid_list = {$plugin.tt_news.pid_list}
   recursive = {$plugin.tt_news.recursive}
   singlePid = {$plugin.tt_news.singlePid}

   publicationName = TYPO3 News

   # if this is not set it is tried to auto-determine
   # from your pages settings
   publicationLanguage = en

   # your news article is only available for registered
   # users. Registration is free.
   access = Registration

   # your news articles are opinion-based blog articles
   genres = Blog,Opinion

   # only display news from the last 3 days
   maxAge = 3

   # such a high value means, the news are rather important
   # for your page
   defaultPriority = 0.8
  }
 }
}
```

Tx_ExtensibleSitemap_Generator_TtNews_NewsArticle

A generator for "extensible_sitemap" that indexes tt_news articles for a news sitemap.

Warning

Please notice that this is for tt_news from version 3.0.0 up - the parameters have changed with this version.

Table 2.5. parameters for Tx_ExtensibleSitemap_Generator_TtNews_NewsArticle

Property	Data type	Description	Default
singlePid	integer	the page id of the single view	
pid_list	comma-seperated integer	the pids the records reside in (set to **0** to take all records)	0
recursive	integer	how many levels under the above given pid_list should be looked for records too	0
defaultPriority	priority	the default priority assigned to each item.	
publicationName	string	the name of the publication.	
publicationLanguage	string	the 2- or 3-signed ISO 639 Language Code of the language this news is in. Leave blank for autodetection.	
access	access-string	if access to the article is not public, set one of Subscription, Registration. Also see glossary entry.	
genres	comma-seperated genre-string	might be a comma-seperated list of PressRelease, Satire, Blog, OpEd, Opinion, UserGenerated. Also see glossary entry.	
maxAge	integer	the maximum age in days of the news in order to be displayed. Google™ states it won't add news if they were	7

Property	Data type	Description	Default
		published more than 2 days ago [http://www.google.com/support/news_pub/bin/answer.py?answer=74496].	

Chapter 3. Development

Basics

You need to create a class implementing the `Tx_ExtensibleSitemap_Generator_Interface` in order to link it to the sitemap generator. There are four methods that need to be implemented:

`init($conf, $parent, $cObj = null)`

This method is called before fetching the first item. So here you might develop a database connection, fetch your results or initialize variables.

There are three parameters given:

`$conf` An array with configuration for the generator. It is a php-transformation of the configuration options set via TypoScript.

`$parent` A link back to the calling class. Most notable is the

```
throwError($code,
           $message = null);
```

method. If something goes wrong use this to throw a HTTP Status Code and a message.

`$cObj` A reference to a cObject you might use to generate links.

`finish()`

This method is called after fetching the items. Close your database connection and unset all variables you don't need anymore. There might be other Generators that need their space.

If there is nothing you need to do - just give this method an empty body.

`getNext()`

This implements a very simple `Iterator`-Interface. It should return an array for each page to add to the sitemap or `NULL` if there is nothing more to submit. See below on how this array is supposed to be formed.

`getRequiredNamespaces()`

The controller calls this method to ask if your generator makes use of additional namespaces to the standard sitemap tags. If you don't just give this method an empty body. If so see the next section for more details.

The `getNext()` method will be called repeatedly by the controller untill it returns `NULL`. It should return an array with values in every other case.

`uid`

If you return *real* pages from the `pages` table, you can return just its uid and the URL will be generated for you.

`_OVERRIDE_HREF`

In all other cases you should set this field to the path from the pages root. The domain and protocol will be appended by the controller.

> **Note**
>
> You should use the `cObj` given to the `init()`-method to generate the link. Else it might not work with `realURL`.

`SYS_LASTCHANGED` — The time of the last change of the contents of this page as a unix-timestamp.

`tx_extensiblesitemap_frequency` — The expected frequency of changes to the contents of this page.

> **Note**
>
> You might want to use the constants in `Tx_ExtensibleSitemap_Utility_Config`.

See glossary for valid values.

`tx_extensiblesitemap_priority` — The priority of this page compared to the remains of the website. You most likely would make this configurable via the config.

See glossary for valid values.

`TX_EXTENSIBLESITEMAP_ADDITIONAL_FIELDS` — This is were you could put custom extensions to the sitemaps protocol. It takes *any* string you submit and appends it. So you are responsible for the validity of the output. The controller will just pass this content through to the output.

All the above parameters are optional, as long if either `uid` or `_OVERRIDE_HREF` are present.

You can use your class by setting the name of the generator in TypoScript to your classname.

Namespaces

You can extend the sitemap protocol through namespaces. To avoid slowing of the sitemap generation for large sites this is simply done by returning a string in the field `TX_EXTENSIBLESITEMAP_ADDITIONAL_FIELDS` of the return array. This string will be added to the <url>-tag.

> **Warning**
>
> As your returned string won't be treated any further, you should make use of `htmlspecialchars()` when necessary.

Additionally you have to let the Controller know you make use of namespaces. The Controller calls the `getRequiredNamespaces()` method before writing the opening tag. You can return `NULL` if you don't make use of namespaces. If you do you should return an array where each line is a namespace you require.

You can either use an associate entry where the `key` is the name of the namespace and the `value` the uri for that namespace.
Or you can use a simple array where the `value` is the name of the namespace. If the namespace is known the uri will be added automatically.

See `Tx_ExtensibleSitemap_Controller_Eid::$namespaces` for a list of known namespaces.

Note

`getRequiredNamespaces()` will be called prior to `init()`.

Glossary

access-string

The access-string takes one of the following values:

`Subscription` (visible) — an article which prompts users to pay to view content.

`Registration` (visible) — an article which prompts users to sign up for an unpaid account to view content.

If you don't set the access-string or set it to an empty value this means that the full article is accessible to all users for at least thirty days.

This definition was taken from the official Google™ documentation on the news namespace [http://www.google.com/support/webmasters/bin/answer.py?answer=93992].

> **Note**
>
> Developers will find class constants for these in `Tx_ExtensibleSitemap_Utility_Config`.

changefreq

How frequently the page is likely to change. This value provides general information to search engines and may not correlate exactly to how often they crawl the page. Valid values are: `always`, `hourly`, `daily`, `weekly`, `monthly`, `yearly`, `never`.

> **Note**
>
> The value `always` should be used to describe documents that change each time they are accessed. The value `never` should be used to describe archived URLs.

> **Note**
>
> Please note that the value of this tag is considered a hint and not a command. Even though search engine crawlers may consider this information when making decisions, they may crawl pages marked `hourly` less frequently than that, and they may crawl pages marked `yearly` more frequently than that. Crawlers may periodically crawl pages marked `never` so that they can handle unexpected changes to those pages.

This definition was taken from sitemaps.org [http://www.sitemaps.org/protocol.php] and is licensed under the Creative Commons Attribution-ShareAlike License (version 2.5) [http://www.sitemaps.org/terms.php].

> **Note**
>
> Developers will find class constants for these in `Tx_ExtensibleSitemap_Utility_Config`.

genre-string

Valid values for Genres are:

`PressRelease` (visible) — an official press release.

`Satire` (visible)	an article which ridicules its subject for didactic purposes.
`Blog` (visible)	any article published on a blog, or in a blog format.
`OpEd`	an opinion-based article which comes specifically from the Op-Ed section of your site.
`Opinion`	any other opinion-based article not appearing on an Op-Ed page, i.e., reviews, interviews, etc.
`UserGenerated`	newsworthy user-generated content which has already gone through a formal editorial review process on your site.

This definition was taken from the official Google™ documentation on the news namespace [http://www.google.com/support/webmasters/bin/answer.py?answer=93992].

> **Note**
>
> Developers will find class constants for these in `Tx_ExtensibleSitemap_Utility_Config`.

priority

The priority of this URL relative to other URLs on your site. Valid values range from `0.0` to `1.0`. This value does not affect how your pages are compared to pages on other sites—it only lets the search engines know which pages you deem most important for the crawlers.

The default priority of a page is `0.5`.

Please note that the priority you assign to a page is not likely to influence the position of your URLs in a search engine's result pages. Search engines may use this information when selecting between URLs on the same site, so you can use this tag to increase the likelihood that your most important pages are present in a search index.

Also, please note that assigning a high priority to all of the URLs on your site is not likely to help you. Since the priority is relative, it is only used to select between URLs on your site.

This definition was taken from sitemaps.org [http://www.sitemaps.org/protocol.php] and is licensed under the Creative Commons Attribution-ShareAlike License (version 2.5) [http://www.sitemaps.org/terms.php].