SEO and URL rewriting

PHP, Wordpress

Web Skills : 5

SEO, everyone talks about it in the web community. SEO is an English acronym that literally means 'search engine optimization'. This chapter is not going to talk about SEO though, but a part about the search filter. Rewriting a URL that contains parameters into something much more readable, is a guarantee of finishing at the top in search engine indexing.

Published Monday January 13th 2020, 18:16
Modified Monday August 26th 2024, 10:06

By Olivier Paudex

Introduction

When it comes to positioning in search engines, we immediately think of SEO, for **"Search engine optimization"**. Behind this barbaric name is hidden one of the main aspects of referencing by search engines, including Google. In order for a search engine to rank you in a certain category, it needs a list of keywords that it retrieves in different ways, including a simple reading of the URL.

If the URL of the page has some words like **"travels"**, the name of the custom type in this example, the SEO is very clear. At the same time, if the URL has search parameters like **"?s=bern&travel_country=switzerland"**, then there, it is not going to understand the word **"travel_country"**. The first rule of SEO is to correct these URLs to rewrite them in a more readable way.

Little reminder

For those of you who have been following me since the beginning, we have created a search form on the archive page, with Elementor. This one builds on the fly, a URL that will filter the posts according to the requested criteria.

The code below will give an URL that looks like this :

*https://www.fuyens.ch/travels/?search=bern&travel_country=switzerland&travel_month=january&travel_year=2010&travel_*

The URL is made up of the domain name, the type of publications and all the criteria that begin with a question mark, separated by **"&"**. This is not very readable and, above all, does not mean much to a search engine.

```
// Set up redirect url with query vars
$redirect_url = home_url('travels' . '/');
$search_text ? $redirect_url .= rawurldecode('?search=' . $search_text) : $redirect_url .= rawu
$travel_country ? $redirect_url .= rawurldecode('&travel_country=' . $travel_country) : $redire
$travel_month ? $redirect_url .= rawurldecode('&travel_month=' . $travel_month) : $redirect_ur
$travel_year ? $redirect_url .= rawurldecode('&travel_year=' . $travel_year) : $redirect_url .=
$travel_taxonomy ? $redirect_url .= rawurldecode('&travel_taxonomy=' . $travel_taxonomy) : $re
$travel_order ? $redirect_url .= rawurldecode('&travel_order=' . $travel_order) : $redirect_ur
```

URL correction

The URL, when corrected, should look more like this :

https://www.fuyens.ch/travels/search/bern/country/switzerland/month/january/year/2010/type/cultural/order/desc/

Most developers, designers, computer scientists will create a site in English, because English is the language of the web in general. In a second step, the site and the posts are translated into other languages.

In French, we can also translate English words, like **"country"**. Fuyens.ch takes this into account and uses the **"Polylang"** plugin to translate URLs from English to French. But this is another chapter.

Here is the code to correct the URL :

```
// Set up redirect url with query vars
$redirect_url = home_url('travels') . '/' . 'search');
$search_text ? $redirect_url .= rawurldecode('/' . $search_text . '/') : $redirect_url .= rawu
$travel_country ? $redirect_url .= rawurldecode('country' . '/' . $travel_country . '/') : $re
$travel_month ? $redirect_url .= rawurldecode('month' . '/' . $travel_month . '/') : $redirect_
$travel_year ? $redirect_url .= rawurldecode('year' . '/' . $travel_year . '/') : $redirect_ur
$travel_taxonomy ? $redirect_url .= rawurldecode('type' . '/' . $travel_taxonomy . '/') : $redi
$travel_order ? $redirect_url .= rawurldecode('order' . '/' . $travel_order . '/') : $redirect_
```

URL rewriting

Now, if the syntax of the URL looks like this, WordPress won't understand it and will open a 404 error page, page not found.

In order for WordPress to understand your request, it will have to be corrected back to what was initially written. And yes, the URL has been corrected once so that it is more pleasant to read for human beings, as well as for SEO engines that also like words rather than syntax, and then must return to its primary form so that WordPress can redirect the visitor cleanly to the right page.

For this, WordPress has provided a function whose name, **"add_rewrite_rule"**, speaks for itself.

In the code below, some important points :

- The rewrite rule is executed at the initial loading of the site, on the**"init"** action.
- The rewrite rule must always start with a reset function called **"flush_rewrite_rules"**.
- Apart from **line 5**, only **lines 15 and 16** are essential. Everything else is just comments. In fact, these two lines are one line, separated by a comma. **You should not try to break the line**, otherwise it may not work.

Regex

To rewrite a URL, the use of **Regex** is very appropriate. **Regex** allows to replace on the fly a search pattern as in a **"search – replace"** tool.

To check the syntax of a Regex expression, you can use an online tool like **https://regex101.com/**.

Some explanations

There are two strings in a **Regex** expression, the first one **(line 15)**, is the search pattern, the second one **(line 16)**, is the string that will replace it. Here are some explanations.

- Important, the transformed URL must always start with **"index.php"**, which is the default page.
- Then, we write in plain text what the URL must correspond to, namely: **"?post_type=travel&search="**.
- Next, we will use a dynamic variable by writing: **"$matches[1]"**, for the first variable. This variable will replace the group between brackets **"([^/]*)"**, which literally means all characters until the next slash **"/"**. The **"/?"** before and after the group means an optional slash. Indeed, the visitor may not have entered a search word for the **"search"** variable.
- Then, we start again with the parameters **"country, month, year, type and order"**. All these parameters are uncaptured groups, unlike **"search"**, represented by a **"?:"**, placed before.
- We end the Regex expression with the page number whose syntax is **"(?:page/([0-9]+))?"**. This means to add the word **"page"** and a number from 0 to 9 at the end of the URL. It also works if you are on the 10th page.

```
// Rewrite rules for the travels search bar
function travels_rewrite_rules() {
  // Flush the rules
  flush_rewrite_rules();
  // Regex
  // ([^/]*)                 = All characters until next /
  // /?                      = Optional /
  // (?:page/([0-9]+))/?$    = Literally 'page/[page_number]' with optional / and positioned a
  //                           Page is a non captured-group (?:)
  // Rule with query search string as first match
  // country as second match, travel month as third match, travel year as fourth match, taxonom
  add_rewrite_rule ('travels/search/?([^/]*)/?(?:country)/?([^\/]*)/?(?:month)/?([^\/]*)/?(?:ye
  'index.php?post_type=travels&search=$matches[1]&travel_country=$matches[2]&travel_month=$mat
}add_action ('init', 'travels_rewrite_rules');
```

That covers the rewriting of URLs. It requires a little understanding of the **Regex** expression language, but once you've mastered it, the method of searching and replacing all sorts of strings will be a major asset.

The next chapter continues the **page map** that my blog was talking about in the post of the same name. Rather than talking about creating a single post, to display its content, the topic will expand on **creating and customizing a comment area**.