

Multilingual messages with Polylang

PHP, Polylang

Web Skills : 4

When creating a site, the registration, login or validation of an action often requires the sending of an e-mail message to the visitor, subscriber or user. If the website is multilingual, it is essential to send correspondence messages in the language of the website. Polylang offers on-the-fly translation functions that avoid having to write messages or their HTML code in all the languages of the site.

Published Wednesday October 21st 2020, 10:23
Modified Monday August 26th 2024, 10:04



By Olivier Paudex

Introduction

Polylang comes with a library of PHP functions that allows, among other things, to write a message in several languages. This avoids the boring task of writing the HTML code of the messages in all the languages offered by the site. Using these features of the Polylang plugin allows to shorten the time spent coding and assigning the translation task to a native speaker who is not, in principle, a person who knows programming.

English as main language

Most, if not all, developers will write PHP code for a WordPress site in English. As such, it is also applicable to the writing of the various emails sent to the users of the site.

In the language settings of WordPress, there are several options concerning the language.

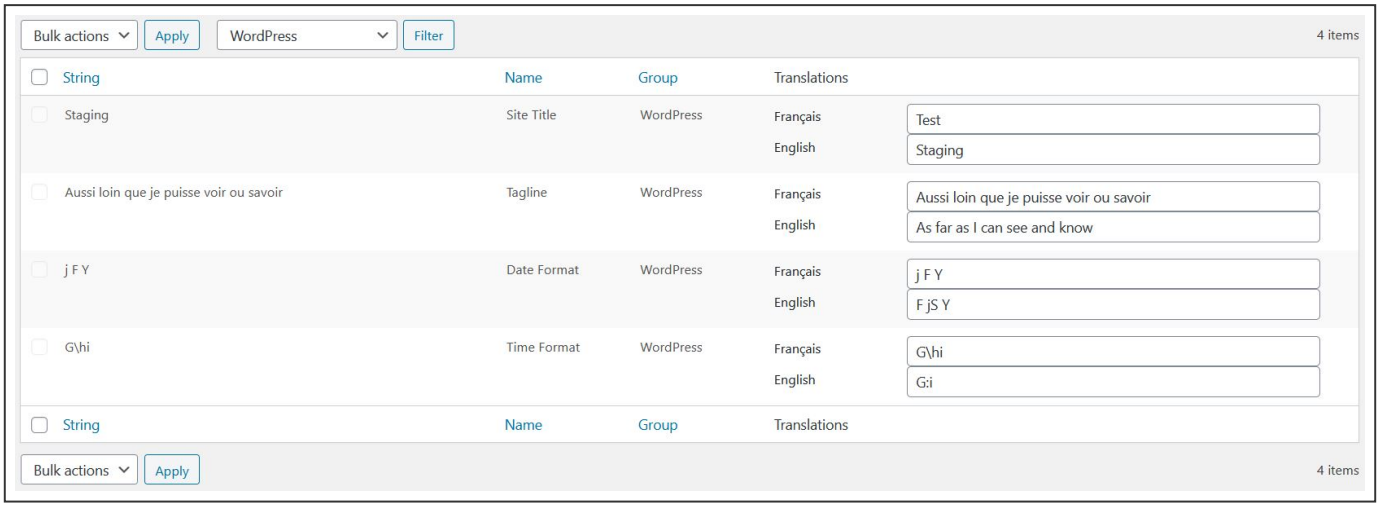
- **The main language of the site (Settings – General)** This setting allows you to change the main settings according to the country.
- **The main language of the user (Users)** This setting allows you to choose the display language of the WordPress administration console for a user. This means that it is quite possible to create multiple users, each displaying the WordPress console in their own language.
- **The theme's display language. (Languages – Languages)** This Polylang setting allows you to choose the default display language for posts and other messages. At first glance, it should always be **"English"**, but it will also display in English all plugins that offer translation, to follow the above recommendation.

Full name	Locale	Code	★	Order	Flag	Posts
English	en_US	en	★	2		543
Français	fr_FR	fr		1		252
Full name	Locale	Code	★	Order	Flag	Posts

Language settings in Polylang

Strings translations

Polylang offers a tab called **"String Translation"**, in which all the expressions of the site will be displayed in the main language. Then each string has a field per language added to the website in which it is possible to translate the word or expression into the other language. The expressions are sorted into different groups. The main group is **"WordPress"**, in which all expressions created by WordPress are listed. It is quite possible to create a formatting expression, like here for the date and time format. Other groups are created by third party plugins, such as **"CPT UI"**. It is also possible to create your own group, as explained in the next chapter.



Strings translations of the WordPress group

Create an expression

As said above, expressions should always be written in English, then translated. To create an expression, you need to create a new PHP file and then add it to the child theme. Expressions will be executed when the site is loaded via the **"after_setup_theme"** action, which will run just after the main theme is loaded.

The Polylang function used to create expressions is **"pll_register_string"**. This function asks as parameters, the name of the expression, the expression and the name of the group. The fourth parameter is a boolean type to pass the expression in multiline. Expressions should always be short and concise, so this one should always be negative.

Here is an example to create an expression whose name is **"WP Mails"** and the group is **"Custom strings"**.

```
<?php
  /**
  * Plugin Name: Polylang
  * Description: Customized text strings translation
  * Author: Olivier Paudex
  * Author Web Site: https://www.fuyens.ch
  */
  // Register Polylang strings translations
  function register_polylang_strings_translations() {
    if (function_exists('pll_register_string')) {
      // Mails
      pll_register_string('WP Mails', 'Your post is awaiting validation', 'Custom strings', fa
    }
  }
  add_action('after_setup_theme', 'register_polylang_strings_translations');?>
```

Once the above code is added, the expression will be found under the **"strings translations"** tab of Polylang. All that remains is to translate it into French.



The expression and its translation

Use expressions in an e-mail message

Once the expressions have been created in Polylang, it is possible to use them to send an e-mail message. To do this, you need to use a new Polylang feature called **"pll_translate_string"**. This feature allows to translate an expression previously stored in Polylang. It requires two parameters, the expression and the desired language.

Thus, to take the example above, the code below will assign to the variable **"$message"**, the expression **"Your post is awaiting validation"**, translated into French.

```
$message = pll_translate_string('Your post is awaiting validation', 'fr_FR')
```

Send a message in the author's language

To finish this post, here is the complete code to send to a contributor, an email, in his language, that his post is awaiting validation.

- This function uses the **"transition_post_status"** action which is executed each time a post changes status. Here, the post changes to **"pending"** mode, when it is saved by the writer.
- The code will retrieve some information about the writer, including his ID, email address, first name, last name and the language he speaks.
- The message is then written and translated using the **"pll_translate_string"** function and sent to the contributor in HTML format.

Since strings should be as simple as possible, it is advisable not to add punctuation marks and other spaces in them, but rather to use multiple instead. Thus, it is preferable to separate long expressions into several strings. A concrete example would be two strings for the expression **"turn right"**, **"at the next intersection"**. It will then be possible to reuse the second string in the expression **"turn left"**, **"at the next intersection"**.

```
<?php
  /**
  * Plugin Name: OP_Mails
  * Description: Send mails
  * Author: Olivier Paudex
  * Author Web Site: https://www.fuyens.ch
  */
  // Send email to contributor when post is created or updated
  function new_post_mail($new_status, $old_status, $post) {

    // Bail early if not a custom post type form
    $custom_post_type = array('travels', 'tracks', 'it');
    if (in_array(get_post_type($post), $custom_post_type)) {
      // Send email when post is created or updated
      if ($old_status != 'pending' && $new_status == 'pending') {
        // Get author data
        $author_id = $post->post_author;
        $email = get_the_author_meta('user_email',$author_id);
        $firstname = get_the_author_meta('user_firstname',$author_id);
        $lastname = get_the_author_meta('user_lastname',$author_id);
        $language = get_user_locale($author_id);
        // Add a subject to the mail
        $subject = pll_translate_string('New post on', $language) . ' ' . get_bloginfo('name');
        // Get date and time of the current post
        $date_time = get_the_date('F jS Y', $post) . ", " . get_the_time('H:i', $post);
        // Create the message
        $message = pll_translate_string('Hello', $language) . ' ' . $firstname . ' ' . $lastnam
        $message .= pll_translate_string('Your post is awaiting validation', $language) . '.'
        $message .= pll_translate_string('Publication title', $language) . ' : <strong>' . get_
        $message .= pll_translate_string('Publication date', $language) . ' : <strong>' . $date
        $message .= pll_translate_string('We will send you a new email to inform you of its pub
        $message .= pll_translate_string('If this is not the case, we will contact you to expla
        $message .= pll_translate_string('Thanks for your contribution', $language) . '.' . '<b
        $message .= pll_translate_string('The administrator of', $language) . ' ' . get_bloginf

        // Add Text/HTML filter
        add_filter ('wp_mail_content_type', function($content_type) {return 'text/html';});
        // Send mail
        wp_mail($email, $subject, $message);
        // Reset content-type filter to avoid conflicts
        remove_filter('wp_mail_content_type', 'set_html_content_type');
      }
    }
  }  add_action('transition_post_status', 'new_post_mail', 10, 3);
```

This action allows you to write the email message in English only once, and thus also the function code. Polylang will automatically translate the message according to the user's language. If the message had to be modified, it would be enough to adapt only the English message and its translation. There is no risk of having to modify the code and create a programming error.