



Translating WordPress

I've been working on a new WordPress theme for a few weeks. Now I'm at the point where I can start worrying about the last finishing touches before I can start dogfooding it.

One of the things left in the to-do list is internationalizing the theme. Making sure that it's translated to Spanish and that it is ready for people who want to translate it to other languages to do so.

In this post we'll concentrate on theme i18n (internationalization) for a theme. Gutenberg and blocks will be discussed in a separate post.

Translating Themes

For the translations to work you have to make sure that `textdomain` is set on your `styles.css` main stylesheet. You also have to change the way that you write the strings using PHP.

Textdomain

When building the CSS stylesheet for your theme, make sure that you add a Text Domain field to the header of the stylesheet. Wordpress will use it as the text domain to tie all translations to.

A minimal example of the theme header looks like this:

```
/*!  
Theme Name: rivendellweb  
Version: 1.0.0  
Text Domain: rivendellweb  
*/
```

Changing how you write strings

In order to translate our themes, we need to change the way we write our strings.

Using `echo` on its own, like this:

```
<?php
if ($req) echo ('(required)');
```

Will always produce the string required, regardless of the language we are using in WordPress. Instead use one of the localization functions built into Wordpress.

- [__\(\)](#)
- [_e\(\)](#)
- [_x\(\)](#)
- [_ex\(\)](#)
- [_n\(\)](#)

Using `_e()` the string to localize now looks like this:

```
<?php
if ($req) ( _e( 'Required', textdomain ) );
```

The first parameter is the string to localize, the second is the text domain that we want to use the localization with since we may have different localization domains for themes, plugins and WordPress itself.

As always, test the localization functions to see which one works best for your project and what results they produce.

Getting the translations ready

You're not required to use Poedit or to pay for the Pro version. I've found out this is the easiest way to do it but your mileage may vary.

WordPress uses [GNU Gettext](#) to manage the translations.

I downloaded [Poedit](#) and chose to upgrade to the Pro version because it allows you to automatically create the translation files for a WordPress theme or plugin without having to do the work manually.

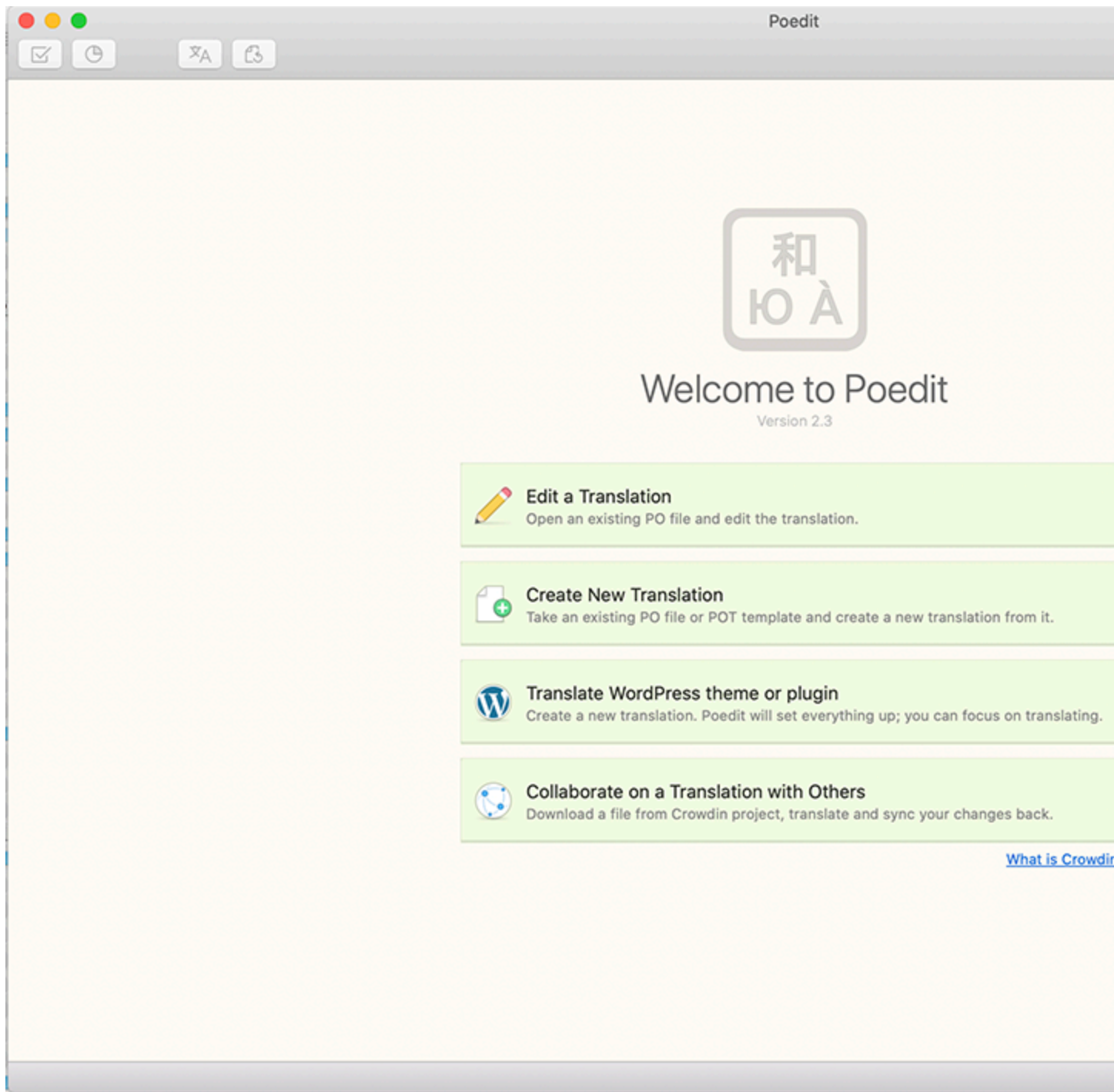


Figure 1: Opening screen for Poedit application

In this case we'll select Translate WordPress theme or plugin and move on to the next step.

When we choose to work with a WordPress theme or plugin we're prompted for the following information

- If we want to work with a remote server or locally
 - If we want to work locally then enter the location of the theme or plugin we want to work with

- If we want to create a new translation or a POT template to use as the basis for further translations

For this post, we'll create a POT template for our theme.

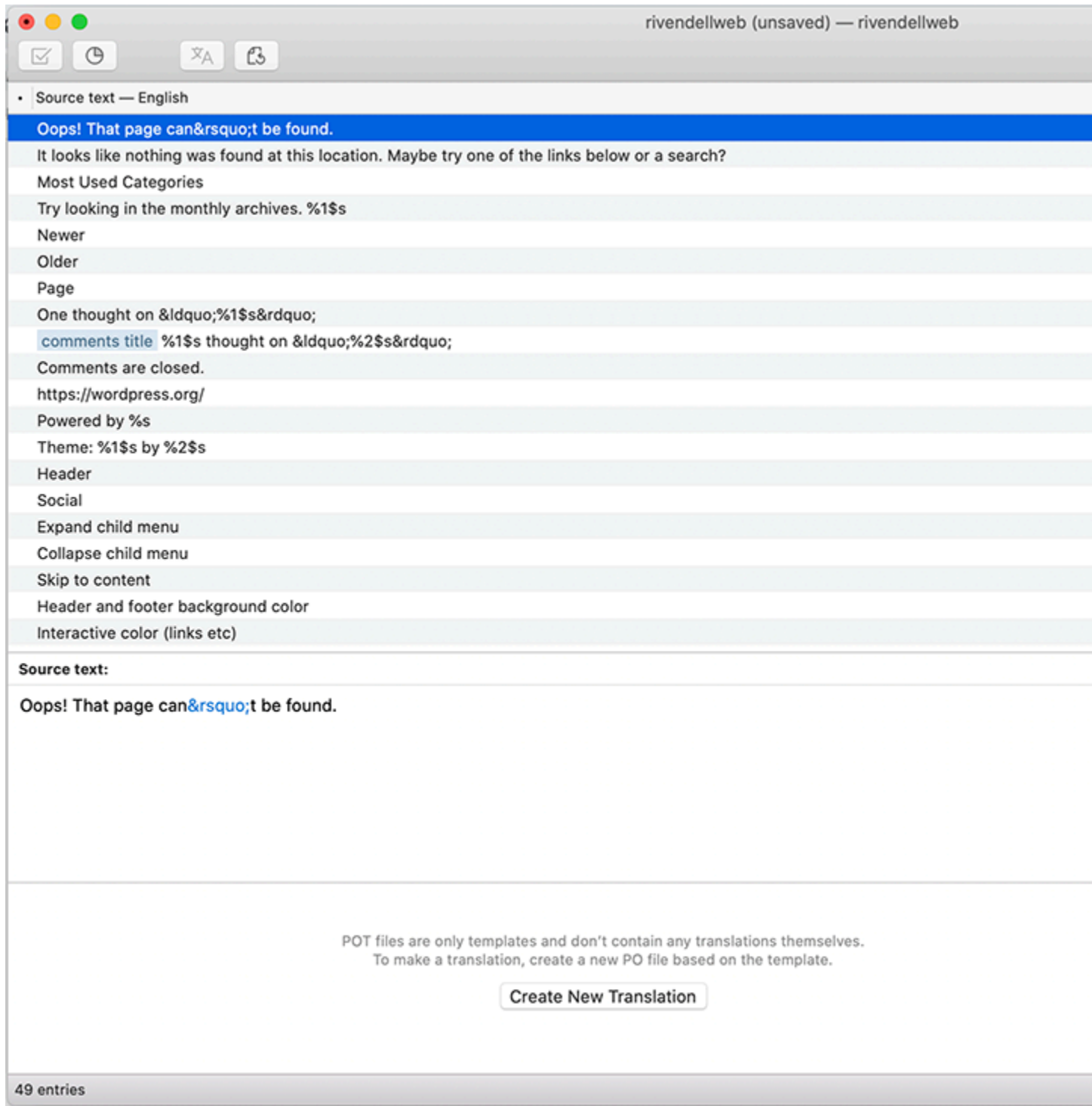


Figure 2: POT Screen in Poedit

When we click on create POT, we see all the translated strings available on the theme. At the bottom, we have a create new translation button.

You will be prompted to save the POT file and when you do, you will be asked what language are you translating for.

✓

🕒

🌐

🔍

Source text — English

Oops! That page can't be found.

It looks like nothing was found at this location. Maybe try one of the links below

Most Used Categories

Try looking in the monthly archives. %1\$s

Newer

Older

Page

One thought on " %1\$s "

[comments title](#) %1\$s thought on " %2\$s "

Comments are closed.

<https://wordpress.org/>

Powered by %s

Theme: %1\$s by %2\$s

Header

Social

Expand child menu

Collapse child menu

Skip to content

Header and footer background color

Interactive color (links etc)

Source text:

Oops! That page can't be found.

Language of the source text

Spanish

POT files are only templates and don't contain any translations.
To make a translation, create a new PO file based on this POT file.

Create New Translation

Figure 3:
Poedit
Language
Selection

We can then enter translations for each of the strings we chose to translate in our code and the translation that we want for it.

The Pro version allows you to pull translation from online sources. But, even without the online sources, you can translate the content yourself or you can handle the POT file to a commercial translator to work with.

☑

🕒

🔍A

📄

• Source text — English

Trans

Most Used Categories

Try looking in the monthly archives. %1\$s

Newer

Older

Page

One thought on “%1\$s”;

comments title

 %1\$s thought on “%2\$s”;

Comments are closed.

<https://wordpress.org/>

Powered by %s

Theme: %1\$s by %2\$s

Header

Social

Expand child menu

Collapse child menu

Skip to content

Header and footer background color

Interactive color (links etc)

Theme Options

Change how much of a post is displayed on index and archive pages.

Source text:

Oops! That page can’t be found.

Translation:

Figure 4:
Poedit
Translation
Screen

You can use the POT to translate to multiple languages. Place the translations on your theme's languages directory.

Links and Resources

- [Translating the theme you created](#)
- [Poedit](#)
- [The Text Domain in WordPress Internationalization](#)
- <https://pascalbirchler.com/wordpress-internationalization-workflows/>
- [Improving Our Translation Workflow](#)