



Drupal, traduire facilement un module

Le 16 novembre 2007 à 09:56.

Autant le système de traduction de Drupal est efficace, autant cela peut vite devenir l'enfer lorsqu'il s'agit de créer et de maintenir un fichier de traduction spécifique à son module. Utiliser pour cet tâche l'export de Drupal est déjà casse-bibine (car on va y trouver toutes les traductions, celles de vos modules et le reste...) mais surtout pas exhaustif car les textes sont découverts par Drupal "à la volée", au moment de leur affichage.

La solution passe donc par un outil d'extraction prévu pour cet usage mais malheureusement pas pas assez connu et documenté.

Les fichier PO et POT

Comme chacun sait (évidement 😊), lorsque l'on construit un module Drupal, il est de bonne pratique d'utiliser la fonction `t("...")` qui opérera la traduction, si disponible, de la chaîne passée en paramètre. Lors de l'affichage, la chaîne va ainsi passer dans moulinette à Drupal qui, en fonction de la langue courante (ex. français), de la disponibilité d'un dictionnaire pour cette langue, et de la présence d'une traduction dans ce dictionnaire, va retourner par retourner un `t("This is a test")`, la valeur "Ceci est un test".

Tout le problème va donc maintenant résider en le fait de pouvoir fabriquer un tel dictionnaire contenant la liste exhaustive des textes de notre module et de celui-ci seulement. Ce travail est réalisé par [POT eXtractor](#) [1].

POTX est un outil qui peut fonctionner en mode WEB (peut conseillé pour l'instant, pas encore sec), ou plus simplement en ligne de commande. Il va scanner tous les fichiers contenant du texte dans un dossier module et fabriquer un fichier .POT.

Rapidement, les fichier PO et POT sont un standard de traduction d'application du monde Unix (ou plus particulièrement GNU). Un Portable Objects (PO) est un simple fichier texte qui contient une liste de traductions littérales de phrases complètes, d'une langue vers une autre langue, généralement de l'anglais vers une langue cible.

Le Portable Objects Template (POT), quant à lui, est un fichier PO qui dont les données de la langue cible ne sont pas traduit. Une sorte de fichier vierge.

La syntaxe d'un fichier .po (ou pot) donne ceci :

```
#: Commentaire fichier:ligne
msgid "English text"
msgstr "Text français"
```

L'outil POTX, ne générant que des fichier .pot, va donc nous donner un fichier texte contenant une série de choses du genre :

```
#: mon_module.module:45
msgid "This is a text"
msgstr ""
```

Là il ne reste plus alors au traducteur qu'à copier ce fichier en un `fr.po` et de remplir les trous.

Mis en oeuvre

POTX est téléchargeable ici [POTX](#) [1]. Il se décompresse où vous voulez (soit dans `~/mes_outils/potx`, soit dans votre dossier modules si vous voulez tester le mode WEB).

Une fois décompressé, vous y trouverez un script `potx-cli`. Ce dernier a besoin de PHP en mode ligne de commande pour fonctionner (`urpmi php-cli`).

Ensuite, pour plus de simplicité vous pouvez le modifier et ajouter en première ligne un `#!/usr/bin/php` et lui appliquer un `chmod +x potx-cli.php`.

Ceci fait, il suffit d'aller dans le dossier de votre module à traduire et de lancer la commande `~/mes_outils/potx/potx-cli.php` pour que l'outil vous génère un fichier `general.pot` ... Et c'est fini.

POTX va extraire de votre code, tous les textes contenus dans des `t(...)`, mais aussi ceux contenus dans les fonctions comme `format_plural(...)`. En revanche il ne prendra pas les chaînes non encadrées par `t()`, ni celle contenant de variables (pratique déconseillée). Par exemple pour le fragment php suivant :

```
print [2] t("Test for t() function");  
print [2] format_plural(2, "Test for format_plural()", "Tests for format_plural()");  
print [2] t("Test #!number", array [3]("number"=>12));  
print [2] t("Test that should not work because of this : $pas_bien");
```

```
print [2] "Test of simple string";  
print [2] foo("Test of simple string");
```

Le fichier `general.pot` contiendra donc :

```
gaston$ cat general.pot | grep "Test"  
msgid "Test for t() function"  
msgid "Test #!number"  
msgid "Test for format_plural()"  
msgid_plural "Tests for format_plural()"  
gaston$
```

Conclusion

Voilà donc un outil simple et efficace qui vous fera j'espère gagner pas mal de temps. La seule chose qui lui manque est la fabrication de fichier `.po` pour une langue donnée. En effet, lorsque certaines traductions sont déjà connues de Drupal, il serait agréable que l'outil fabrique un fichier `.po` contenant les traductions déjà réalisées... Pour une prochaine version ou a faire à la mimine...

*<http://artisan.karma-lab.net/node/1261>
(C) artisan numerique - CC BY-SA*

Liens:

[1] <http://drupal.org/project/potx>

[2] <http://www.php.net/print>

[3] <http://www.php.net/array>