

Modules - Règles de packaging et validation DoliStore

Sommaire

Introduction

Règles

- Nommage du package

- Règle dans le code

 - 1 - Structure des fichiers

 - 2 - Inclusion des fichiers main ou master

 - 3 - Inclusion

 - 4 - Lien

 - 5 - Pas d'écriture dans l'arborescence standard

 - 6 - Modifications de fichier core Dolibarr

 - 7 - Règles sur les fonctionnalités implémentés

 - 8 - Qualité du code

- Publications sur DoliStore

 - MetaPackages

 - Langue

 - Support

 - Règles d'apparition sur la page d'accueil



Introduction

Voici les règles à suivre pour réaliser un module de qualité sur Dolibarr. Certaines de ces règles sont contrôlées lors du déploiement d'un module via l'assistant de déploiement en ligne d'un module externe. Elles sont aussi vérifiées pour valider un module sur <https://www.dolistore.com>.
Note: Les fichiers d'autre nature que des modules (documentation, outils externes, autres), même si mis à disposition sur DoliStore, ne sont pas concernés.



Nommage du package

Tous les modules pour Dolibarr doivent être nommé **module_mymodulename-VERSION.zip** (où VERSION peut être x or x.y or x.y.z). Si le module est pour un autre logiciel, le nom du zip doit être **moduleothersoftware_mymodulename-VERSION.zip** (par exemple moduleprestashop_mymodulename-1.0.zip).

Les packages du type application android (.apk), document PDF (.pdf) sont libres d'utiliser le nom de leur choix.

Dans tous les cas, le nom des modules ne doit pas contenir ou ressembler à des mots vulgaires ou offensants, ni à des noms propres ou de nom de marque déposées (sauf si vous êtes le titulaire de cette marque bien sur, ou si le module offre un service d'intégration en avec le produit protégé par la marque, par exemple *module_visualiation_paypal-1.0.zip* qui s'interface aux API Paypal sera accepté mais le nom *module_mieux_que_paypal-1.0.zip* qui s'interface avec un service concurrent non).

Le reste du document ne s'applique que pour les modules Dolibarr uniquement.

Règle dans le code

1 - Structure des fichiers

Le premier répertoire dans le fichier zip **module_mymodulename-VERSION.zip** doit s'appeler **mymodulename**.

Tous les fichiers du module seront ensuite dans ce répertoire, comme fichier ou sous-répertoire, en suivant l'arborescence exemple proposée dans **htdocs/modulebuilder/template** :

- **mymodule/*** contains php pages (note that you can also add any other subdir of your choice). Note: If your module is a metapackage (a module that will embed other modules in same zip, you must put here a file metapackage.conf)
- **mymodule/build/** can contains any file you develop for compiling or building package
- **mymodule/core/modules/** contains module descriptor file **modMyModule.class.php**
- **mymodule/core/triggers** contains triggers provided by module
- **mymodule/admin/** contains pages to setup module
- **mymodule/class/** contains PHP class files provided by module
- **mymodule/css** contains CSS files provided by module
- **mymodule/js** contains javascript files provided by module to add new functions
- **mymodule/docs** to provide doc and licence files
- **mymodule/img** contains images files provided by module

- **mymodule/langs/xx_XX** contains language files for language xx_XX (try to put at least en_US)
- **mymodule/lib** contains libraries provided and used by module
- **mymodule/scripts** to provide command line tools or scripts. Note: Command lines script must start with line `#!/usr/bin/env php`
- **mymodule/sql** contains SQL file provided by module to add new tables or indexes
- **mymodule/theme/mytheme** if module provide its own theme/skin

2 - Inclusion des fichiers main ou master

Un module externe de Dolibarr doit pouvoir se mettre dans un répertoire **htdocs/custom** (cas standard), tout comme dans **htdocs/** (cas alternatif), et il doit fonctionner idéalement dans les deux cas mais obligatoirement dans le premier (htdocs/custom) car c'est la que sont mis les modules déployés par la fonction "Déployer module externe" de Dolibarr.

Pour cette raison, lorsque vous essayez de charger l'environnement Dolibarr en incluant le fichier **main.inc.php** ou **master.inc.php**, vous devez vous assurez que l'inclusion, qui fonctionne par défaut quand le module est dans htdocs/custom, marchera aussi dans le cas du module déployé à la racine dans htdocs/:

Nous recommandons d'utiliser la méthode suggérée dans les exemples, dans les fichiers htdocs/modulebuilder/*.php et présentée ici pour charger le **main.inc.php**, mais vous pouvez remplacer le **main.inc.php** par **master.inc.php** pour les scripts de ligne de commande. Cette portion de code est connue pour fonctionner dans toutes les situations (avec Apache, Nginx, IIS, en utilisant un hôte virtuel de sous-répertoires, après une redirection proxy ou non, si la racine Web est auto-détectée ou forcée par le fichier conf, et toute combinaison de cela).

```
// Load Dolibarr environment
$res = 0;
// Try main.inc.php into web root known defined into CONTEXT_DOCUMENT_ROOT (not always defined)
if (!$res && !empty($_SERVER["CONTEXT_DOCUMENT_ROOT"])) $res = @include
$_SERVER["CONTEXT_DOCUMENT_ROOT"]."/main.inc.php";
// Try main.inc.php into web root detected using web root calculated from SCRIPT_FILENAME
$tmp = empty($_SERVER['SCRIPT_FILENAME']) ? '' : $_SERVER['SCRIPT_FILENAME']; $tmp2 = realpath(__FILE__);
$i = strlen($tmp) - 1; $j = strlen($tmp2) - 1;
while ($i > 0 && $j > 0 && isset($tmp[$i]) && isset($tmp2[$j]) && $tmp[$i] == $tmp2[$j]) { $i--; $j--; }
if (!$res && $i > 0 && file_exists(substr($tmp, 0, ($i + 1))."/main.inc.php")) $res = @include
substr($tmp, 0, ($i + 1))."/main.inc.php";
if (!$res && $i > 0 && file_exists(dirname(substr($tmp, 0, ($i + 1)))."/main.inc.php")) $res = @include
dirname(substr($tmp, 0, ($i + 1)))."/main.inc.php";
// Try main.inc.php using relative path
if (!$res && file_exists("../main.inc.php")) $res = @include "../main.inc.php";
if (!$res && file_exists("../../main.inc.php")) $res = @include "../../main.inc.php";
if (!$res && file_exists("../../../main.inc.php")) $res = @include "../../../main.inc.php";
if (!$res) die("Include of main fails");
```

3 - Inclusion

- Toutes les inclusions de fichiers du core doivent être réalisées par

```
include_once/require_once/include/require DOL_DOCUMENT_ROOT."/pathtocorefile";
```

- Toutes les inclusions de fichiers spécifiques au module par les pages du modules doivent

être faites avec

```
include_once/require_once/include/require './monmoduledir/...';
```

- Toutes les inclusions de fichiers spécifiques à un autre module externe doivent être faites par

```
dol_include_once('/extmoduledir/...');
```

4 - Lien

Tous les liens générés pour le HTML vers les pages spécifiques (href, src,...) du module devraient utiliser getNomURL de la classe d'un objet ou à minima utiliser dol_buildpath

5 - Pas d'écriture dans l'arborescence standard

Le module ne doit pas écrire dans l'arborescence des fichiers "**programmes**" de Dolibarr mais uniquement dans des fichiers situés dans le répertoire "**documents**". Y compris pour des fichiers temporaires. Il ne faut pas oublier que sur une installation correcte sécurisée de Dolibarr, l'ensemble de l'arborescence des programmes (à l'exception du répertoire custom) est positionné en lecture seule.

6 - Modifications de fichier core Dolibarr

Compte tenu du point précédent, si des modifications de fichiers du coeur de Dolibarr sont nécessaires au module, elles doivent être soumises au projet Dolibarr via github. Elles seront acceptées :

- Si elles sont poussées dans la branche dev du GitHub de Dolibarr (cas d'enrichissement de fonctionnalité Dolibarr)
- Si il s'agit d'ajout de hook, trigger ou fonction, ou ajout de paramètres optionnels à des fonctions existantes. L'ajout de hook permet de permettre à un module externe d'intégrer son propre code n'importe où dans le code de Dolibarr.

7 - Règles sur les fonctionnalités implémentés

Le module peut solliciter des sites externes, pour envoyer ou collecter des informations uniquement dans les cas suivant:

- Besoin en rapport avec les fonctionnalités même du module. Un module de synchro ou de notification peut transférer des messages dans le cadre de la synchronisation.
- Besoin d'ordres statistiques pour l'auteur du module. Dans ce cas, le module doit offrir une option permettant à l'utilisateur de désactiver cette fonction.
- Les serveurs avec lesquels le module communiquent ne doivent pas utiliser un nom de domaine en violation avec les règles d'utilisation de la marque dolibarr dans les noms de domaines (Voir Règles_d'utilisation_du_nom_ou_de_la_marque_"Dolibarr"_(ou_"DoliStore"))

8 - Qualité du code

Le code doit respecter les mêmes règles de langage et normes que l'application Dolibarr. Voici ces règles: Langages_et_normes

Publications sur DoliStore

MetaPackages

Si votre module est un "metapackage", donc un module qui comprend plusieurs modules, vous devez inclure un fichier appelé **metapackage.conf** dans le répertoire principal avec le nom de la liste de tous les autres modules fournis par votre package. Par exemple, l'image de votre module s'appelle "mymetapack". Lors de l'installation de ce module, il installe également les modules "abc" et "def". Vous devez donc inclure dans le zip de votre package un fichier `*mymetapack/metapackage.conf*` avec le contenu suivant:

```
# Ce fichier décrit tous les modules inclus dans le zip module_mymetapack-x.y.zip
mymetapack
abd
def
```

Le zip aura donc dans son arborescence, à sa racine le répertoire mymetapack qui lui ne contiendra que le fichier metapackage.conf, et au même niveau que ce répertoire mymetapack, les n répertoires de chaque module.

⚠ Le déploiement d'un métapackage à partir de l'assistant de déploiement de Dolibarr n'est possible qu'avec Dolibarr v11+

Langue

La description du produit est obligatoire en Anglais sur DoliStore. Les autres langues sont recommandées pour augmenter l'adoption de votre module par plus d'utilisateurs.

Support

Si votre module est payant, il doit obligatoirement figurer dans la description du produit un mail ou un canal pour le support aux utilisateurs (comme un lien vers un site web de contact)

Règles d'apparition sur la page d'accueil

Certains modules peuvent apparaître dès la page d'accueil. Les conditions pour cela sont définies sur la page suivantes <https://www.dolistore.com/fr/content/3-conditions-generales-de-ventes> dans la section 8.