

Personnalisation du menu administrateur

Traduction libre de l'article de [Randy Carey](#) sur le [Joomla!® Community Magazine](#) du mois d'Avril 2012 : [Customizing the Admin Men](#)

Donnez à vos clients un menu backend taillé pour eux. Voici comment :

Il y a différentes raisons selon lesquelles nous souhaitons proposer à nos clients un menu administration personnalisé. Peut être la plus importante : Nous souhaitons fournir des éléments intuitivement nommés reliant directement les écrans d'édition couramment utilisés. Mais, nous souhaitons également supprimer certains de ces éléments qui encombrant la barre d'administration. Et enfin, nous souhaitons supprimer des éléments de menu qui permettraient à un utilisateur non initié de faire des dégâts importants sur le site.

Comme la plupart des développeurs le savent, le menu administration n'est pas configurable comme le sont les autres menus du site (à une exception, le template Admin Praise qui fournit son propre menu, qui est configurable). Heureusement, le texte, les liens, et la hiérarchie du menu administration sont stockés dans un module. En surchargeant sa vue, nous pouvons ajouter des éléments de menu ainsi qu'une logique ACL. Cette approche nécessite du code, mais les additions et changements seront simples pour un développeur php compétent.

Dans Joomla 2.5, la vue de menu est composée de trois fichiers. *default.php* est un fichier simple qui invoque *default_enabled.php* quand la barre de menu est activée, et inversement, il invoque *default_disabled.php* quand le menu est désactivé. L'état désactivé s'applique, par exemple, lorsqu'un article est édité. Nous sommes avant tout préoccupé avec *default_enabled.php*, mais une solution complète devrait imiter nos changements dans *default_disabled.php*.

Suivez l'approche standard pour la surcharge d'une vue. Dans ce cas, copiez les deux fichiers mentionnés (*default_enabled.php* et *default_disabled.php*) situés dans le répertoire */administrator/modules/mod_menu/tmpl* et placez-les dans le répertoire html du template admin. En supposant que nous utilisons le template Bluestork, les deux fichiers vont ici :

```
/administrator/bluestork/html/mod_menu/default_enabled.php
```

```
/administrator/bluestork/html/mod_menu/default_disabled.php
```

Lorsque vous modifiez ces nouveaux fichiers, les changements apparaîtront dans le menu backend. Si par hasard vous faites une erreur en codant et que vous ne pouvez pas la corriger, il suffit de renommer le fichier qui pose problème avec un autre nom. Si vous changez le nom d'un fichier dans le répertoire de surcharge du template, Joomla! reviendra au fichier original et vous serez donc toujours capable de récupérer le template d'origine.

Ajouter des éléments de menu

Regardez le code dans *default_enabled.php* et observez comment le menu et ses éléments sont construits. Voici un fragment de code pour mieux comprendre:

```
1 $menu->addChild(  
2     new JMenuNode(JText::_('JSITE'), '#'), true  
3 );  
4 $menu->addChild(  
5     new JMenuNode(JText::_('MOD_MENU_CONTROL_PANEL'), 'index.php', 'class:cpanel')  
6 );  
7 ...  
8 $menu->getParent();
```

Ceux qui comprennent le php seront capables de comprendre le modèle. Le constructeur pour *JmenuNode* prend en argument le texte et le lien de cet élément. Il peut prendre des arguments additionnels, mais ils sont optionnels.

Un détail à souligner est la façon dont les niveaux de menu sont déclarés. La valeur 'true' en ligne 2 déclare que ce nouvel élément devra être le parent des éléments suivants. Un élément de menu n'est pas un parent si sa valeur booléenne est fausse ou absente (comme à la ligne 5). Le code en ligne 8 indique au menu de remonter d'un niveau dans la hiérarchie des menus et réassigne ce grand-parent comme le parent des éléments qui suivent.

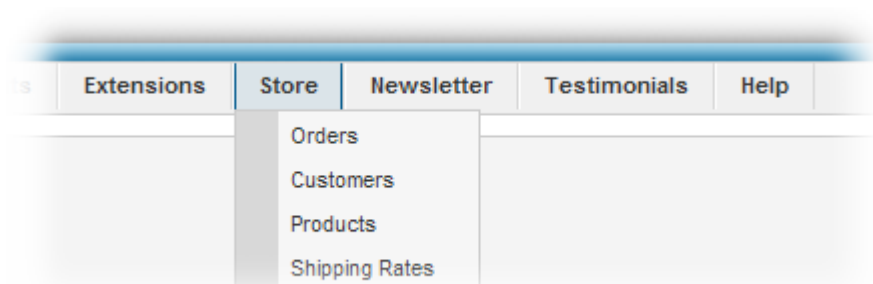
Une bonne place pour ajouter nos nouveaux éléments de menu est juste avant l'élément de menu Help (actuellement vers la ligne 343). Copiez un morceau de code que vous trouverez ailleurs, collez-le ici, et changez avec le texte et les liens que vous souhaitez. Actualisez l'écran d'administration. Ensuite, regardez vos ajouts et testez les nouveaux liens. Cela ne devrait pas être long pour ajouter quelques éléments de menu.

Pour trouver le lien que vous souhaitez utiliser, naviguez sur cet écran, copiez et utilisez son adresse URL en commençant par "index.php" (regardez l'exemple ci-dessous). Si le label de l'élément de menu doit être multilingue, vous aurez besoin de créer des entrées dans vos fichiers de langue et de les utiliser avec la fonction *Jtext::_()*. Je ne l'ai pas fait dans mon exemple afin de simplifier la visibilité du code.

Voici un extrait des éléments de menu ajoutés dans mon exemple

```
1 // STORE  
2 $menu->addChild(  
3     new JMenuNode('Store', 'index.php?option=com_redshop'), true // has children  
4 );  
5 $menu->addChild(  
6     new JMenuNode('Orders', 'index.php?option=com_redshop&view=order')  
7 );  
8 $menu->addChild(  
9     new JMenuNode('Customers', 'index.php?option=com_redshop&view=user')  
10 );  
11 // additional submenu items  
12  
13 $menu->getParent(); // finished adding items at this level, go up one level
```

Voici une vue des nouveaux éléments insérés dans le menu admin



Personnel vs administrateur

Cela devrait être tout ce que vous avez à faire pour ajouter des éléments de menu. Mais sans doute qu'un menu plus pratique est celui qui ne montre au personnel que les liens auxquels il a besoin d'avoir accès. Par exemple, si l'utilisateur a simplement besoin de l'élément de menu que nous venons juste d'ajouter, alors nous allons améliorer la convivialité en supprimant les autres éléments de menu tels que "Menu" et "Extensions". Nous pouvons faire cela.

Dans cet exemple, nous implémentons une règle: le personnel ne voit que les éléments de menu "Site", "Contenu", ainsi que celui que nous venons de créer. Inversement, Le Super Utilisateur a besoin d'accéder au menu standard d'administration et probablement pas aux liens réservés au personnel. Voyons comment nous pouvons implémenter cela.

Vers le début du fichier enabled – après que \$user ait été défini et avant qu'aucun des éléments de menu ne soit créé – ajoutez ce code:

```
$groups = $user->get('groups');  
$isSuper = in_array(8,$groups);  
$isStaff = !$isSuper;
```

L'essentiel est que nous segmentions les utilisateurs backend en groupes significatifs et que nous définissions comment les distinguer. Ici, nous définissons la distinction entre le Super Utilisateur et le Personnel. Utilisez la logique appropriée pour votre projet ainsi que pour votre client. Dans cet exemple, nous supposons que si quelqu'un n'est pas Super Admin, il fait partie du personnel.

Supposons qu'au moins un membre du personnel doit avoir accès au menu admin par défaut, mais sans être un super utilisateur. Nous créons simplement un nouveau groupe d'utilisateur (appelé StaffAdmin, qui dans ce cas est assigné à l'ID "9") et créons une autre variable dans ce fichier :

```
$isStaffAdmin = in_array(9,$groups);
```

Maintenant, le client peut assigner les personnes de son personnel au groupe StaffAdmin, et une telle désignation fait passer cette variable à vrai.

Qui voit quels éléments de menu

Étudier le code pour voir comment l'ACL est appliqué pour déterminer quel élément de menu est affiché pour chaque utilisateur donné. Nous allons suivre cette approche.

Pour afficher tous nos nouveaux éléments de menu uniquement au personnel, insérez le bloc de ces éléments de menu au sein de l'instruction conditionnelle

```
if($isStaff){ /* menu items */ }
```

Pour supprimer certains des éléments de menu au personnel ajouter la logique appropriée à la condition que ces éléments encapsulent. Par exemple, voici le code qui restreint l'accès aux éléments de menu User (ce que j'ai ajouté en jaune)

```
if ($user->authorise('core.manage', 'com_users') && !$isStaff)
```

Mais si vous souhaitez que cette option soit affichée pour StaffAdmin (en plus de Super Admin), alors ajoutez la logique appropriée

```
if ($user->authorise('core.manage', 'com_users') && ($isSuper || $isStaffAdmin))
```

Faites de petits changements que vous pouvez contrôler. Ensuite testez. Puis recommencez jusqu'à ce que le menu d'admin ressemble à ce que vous souhaitez.

Et à propos du fichier disabled?

Nous devrions suivre l'exemple donné par le code dans le fichier original default_disabled.php. Seuls les éléments de menu du premier niveau sont listés et chacun d'entre eux est réglé sur l'état désactivé. Le menu désactivé ne fait rien si ce n'est ressembler à une version désactivée du menu actif.

Nous pouvons être tenté de laisser ce fichier inchangé. Après tout, chacun de ces éléments est désactivé. Mais si le menu désactivé affiche un ensemble différent d'éléments de menu, alors nous risquons de mettre en confusion nos clients ainsi que leur personnel. La meilleure des solutions est d'éditer le menu désactivé afin qu'il affiche les mêmes éléments de niveau supérieur par groupe d'utilisateurs. Souvenez-vous, les sous menus ne seront pas affichés, vous devez donc inclure seulement le premier niveau d'éléments de menu. Si cela représente trop de travail pour vous, une alternative plus simple (mais moins "pure") consiste à n'afficher aucun élément de menu dans la version désactivée. N'afficher aucun élément est toujours mieux que d'afficher un ensemble incohérent d'éléments de menu.

Une meilleure expérience de Joomla!

En surchargeant le menu d'administration, nous pouvons ajouter des liens directs vers les parties fréquemment utilisées de l'administration, utiliser des termes que le client comprend afin de masquer les détails techniques au personnel. Cela nous permet d'offrir à nos clients ainsi qu'à leur personnel une expérience plus simple et plus intuitive avec Joomla!.

Auteur Simon Grange



Cet article de [Simon Grange](#) est mis à disposition selon les termes de la [licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France](#).