Comment développer et intégrer un module à PhpMyLab?

La structure du fichier

Afin de conserver une homogénéité et une cohérence entre chaque module, une structure commune est utilisée pour chacun des modules développés. Voici les différentes parties obligatoires que l'on distingue dans chaque module :

- Le **commentaire** de départ qui précise qui a crée le fichier, quand et pourquoi,
- Le **plan** des différentes parties du fichier pour une meilleure lisibilité du code,
- La gestion de la **déconnexion** (à conserver),
- Les différentes **fonctions** utilisées dans le module (à compléter),
- L'**initialisation générale** (à conserver),
- L'**initialisation de la session** et des variables (à conserver),
- La gestion des variables de **recherche** (à compléter),
- La gestion des variables relatives au **module** (à compléter),
- Le **choix du module** pour les redirection (à conserver),
- l'**HTML** (à compléter).

```
<?php

/**
* Fichier source de départ à la création d'un nouveau module.
*
* La structure indiquée est à respecter pour garder une homogénéité avec les autres modules.
*
* Date de création : 17 Aout 2012
* Date de dernière modification : 17 Aout 2012
* @author Cedric Gagnevin <http://www.cedric_gagnevin.franceserv.fr>
*/

/*************************************************************************************
******
****************************** PLAN
**********************************************
*************************************************************************************
******/

// | -A- Gestion de la déconnexion
// | -B- Fonctions
// | -C- Initialisation generale (configuration et php)
// | -D- Initialisation Session et variables
// | -E- Gestion des variables Recherche
// | -F- Gestion des variables du module
// | -G- Choix du module
// | -H- HTML
-- Code à mettre à chaque début de fichier --
```

Les parties indiquées comme "à conserver" ne sont pas à modifier, il faut les inclure telles quelles. Par contre les autres parties sont modifiables selon vos besoins, vous pouvez créer d'autres parties dans le fichier si vous le jugez nécessaire pour mieux structurer votre programme.

Voici la structure de base que votre module doit avoir : télécharger la source.

Explication des principales parties

-A- Gestion de la déconnexion

Cette partie gère la déconnexion de l'utilisateur au logiciel. S'il s'est connecté par CAS, l'utilisateur se verra déconnecté du service en cours. Dans tous les cas, la session sera détruite et il sera redirigé vers la page d'authentification du logiciel.

-B- Fonctions

C'est ici que toutes les fonctions nécessaires au fontionnement du module sont développées (exemples : envoi d'emails, initialisation du formulaire, ajout d'une demande...). Ce type de séparation du code permet d'avoir toutes les fonctions au même endroit et de pouvoir les réutiliser facilement par la suite.

-C- Initialisation générale

Les fichiers de configuration sont inclut dans cette partie et le jeu de caractères utilisé y est défini.

-D- Initialisation session

La session est initialisée grâce au **sid** (numéro de session) passé en paramètre.

-E- Gestion des variables de recherche

A compléter si votre module utilise une fonctionnalité de recherche.

-F- Gestion des variables du module

C'est là que toutes les variables du module sont gérer (contrôles, passage dans des varaibles de session...).

-G- Choix du module

Redirige l'utilisateur vers le module voulu en faisant transiter le **sid**.

-H- HTML

Développement de l'architecture HTML du module dans cette partie. Le fichier "en_tete.php" est à inclure systématiquement et le sid doit être mis dans un input caché. C'est ici que vous développerez les différentes interfaces de votre module.

Les variables de session

Lorsque l'utilisateur s'authentifie au logiciel, une session s'itilisalise et des variables se remplissent dans **$_SESSION['connection']**.
Voici les variables les plus utiles:

- **$_SESSION['connection']['utilisateur']** contient le login de l'utilisateur,
- **$_SESSION['connection']['nom']** contient le nom de l'utilisateur,
- **$_SESSION['connection']['prenom']** contient le prénom de l'utilisateur,
- **$_SESSION['connection']['groupe']** contient son groupe (équipe/service),
- **$_SESSION['connection']['status']** contient son statut (niveau d'accréditation),
- **$_SESSION['connection']['admin']** contient un booléen pour savoir s'il administre le logiciel.

Le style du module

Le fichier "style.css" situé à la racine regroupe le style des différents modules. C'est donc dans cette partie que se rajoutera le code CSS concernant votre module. Prenez soin de faire une partie et de commenter votre code.

Le style doit être compatible avec :

- IE 8+
- Firefox 11+
- Opéra 5+
- Safari 5+
- Chrome 18+

NB : Il est fortement recommandé d'utiliser les identifiants des éléments pour leur appliquer un style afin d'éviter toutes incompatibilités avec le style précédemment établi.

```
#id_de_mon_element
{
  //Code CSS appliqué à l'élément
}
```

L'intégration dans le logiciel existant

L'intégration de votre module se fait dans le fichier "en_tete.php". Dans un premier temps, il s'agit d'afficher le libellé de votre module si l'utilisateur se dirige vers ce module en question.

- **L 116** Rajouter :

```
elseif($pageCourante == 'nom_de_votre_module.php')
{
  $libelle_module = 'Ce que fait votre module | ';
  echo $libelle_module;
}
```

Ensuite il faut afficher le nom de votre module en haut à gauche :

- **L 116** Rajouter :

```
elseif($pageCourante == 'nom_de_votre_module.php')
{
  echo '<span id="classe_du_titre_de_votre_module">Nom_de_votre_module';
}
```

NB : Il est important que le nom du fichier contenant le code de votre module soit le même que le nom de votre module.

La configuration du module

Il est nécessaire que votre module apparaisse dans la procédure d'installation automatique du logiciel.

Le code minimal à rajouter est le suivant :

```
/* L 75 - etape3.php - dans la partie : Stockage du choix des modules */
if(isset($_POST[ 'nom_de_votre_module' ]))
    $_SESSION[ 'etape3' ][ 'modules' ][ 'nom_de_votre_module' ] = 1;
else $_SESSION[ 'etape3' ][ 'modules' ][ 'nom_de_votre_module' ] = 0;
```

```
/* L 267 - etape3.php - dans l'HTML */
<td>
<input type="checkbox" name="nom_de_votre_module" id="nom_de_votre_module"
<?php
  if(isset($_SESSION[ 'etape3' ][ 'modules' ][ 'nom_de_votre_module' ]))
      echo '';
  else echo 'checked';
?>
/>
<label for="nom_de_votre_module">Nom_de_votre_module</label>
</td>
```

```
/* L 160 - finConfig.php - dans l'écriture du fichier config.php, le choix des modules */
if($_SESSION[ 'etape3' ][ 'modules' ][ 'nom_de_votre_module' ] == 1)
  $module .= '"NOM_DE_VOTRE_MODULE",';
```

NB : Si votre module nécessite des variables de configuration à paramètrer, vous devrez stocker le choix de l'utilisateur dans l'étape 3 qui concerne votre module et gérer l'écriture de ces variables dans "config.php" au cours de la fin de la configuration.

Quelques conseils

- Ne pas oublier de tester le mode (production/test) en cas d'envoi d'emails

  ```
  if($mode_test) $mail_du_destinataire = $mel_test;
  else $mail_du_destinataire = $login_destinataire.'@'.$domaine;
  ```

- Placer le fichier contenant le code de votre module à la racine ("phpmylab_db/")
- Utiliser des chemins relatifs pour les liens
- Remplacer les accents par leur code HTML (é -> é)
- Bien commenter le code pour une relecture et une lisibilité facile
- Prendre en compte le fait que les adresses emails des utilisateurs sont formées à partir de leur login et du domaine défini lors de l'installation (Il est néanmoins possible de modifier cela en se servant de la colonne MEL dans la table T_UTILISATEUR).