



Octobre
2019

Documentation Book Of Dreams

Réseau social interne Sonatel centré sur **l'innovation collaborative**

Encadré par :

El Hadji Birahim GUEYE

Papa Saliou MBODJ

Réalisé par :

Mouhamadou T. MANGASSOUBA

Sakhite MBOUP

Amadou Mbagnick SARR

PLAN

I- Architecture de base et système de fichiers

II- Personnalisation du thème : Sonatel

III- Personnalisation des modules

- Gallery (Catalogue Innovation)
- Calendar (Agenda)
- Intaller (ILAB et Mur d'idées)
- Mostactiveusers (Top 3 idées)
- Autres paramétrages

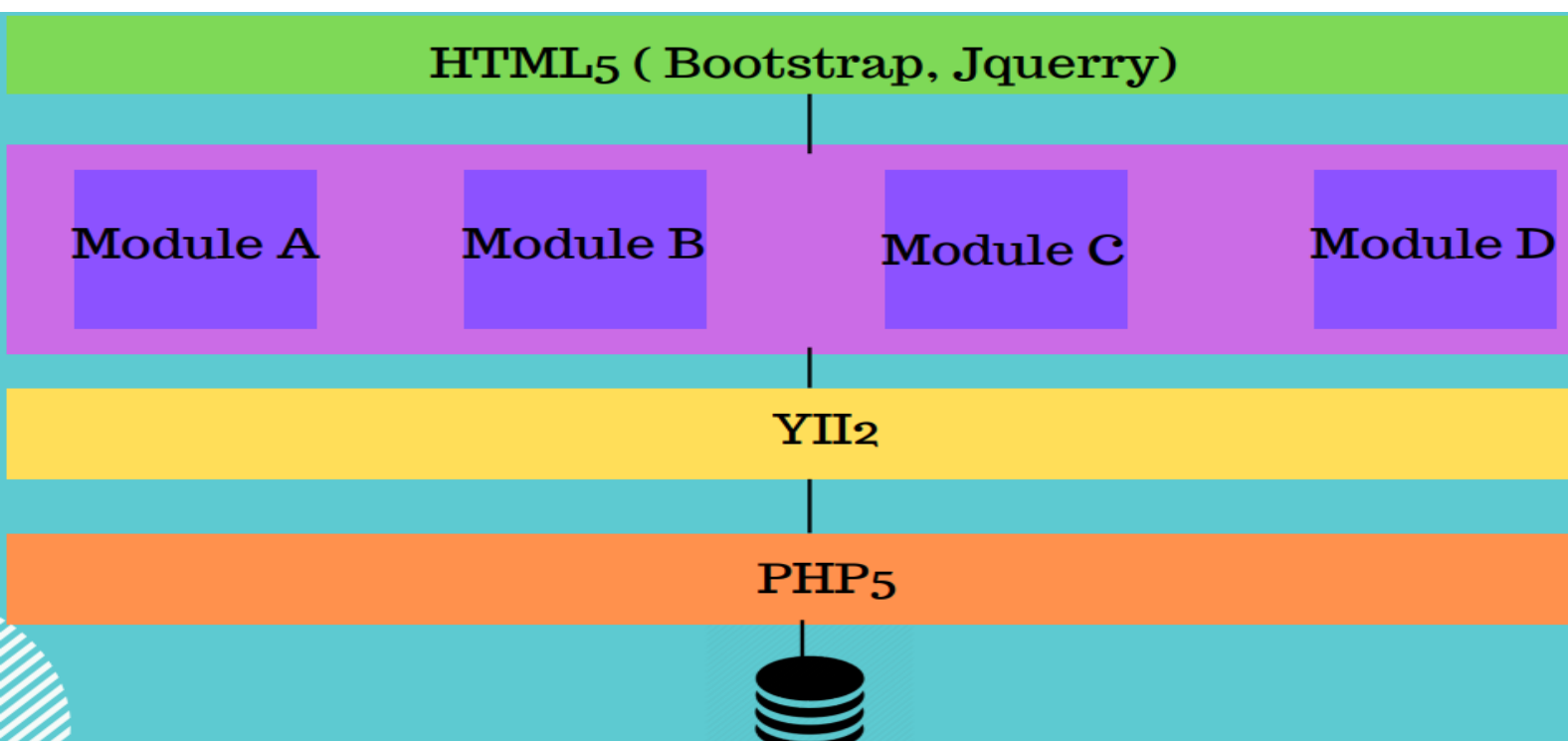
I- Architecture de base et système de fichiers

HumHub fournit une plate-forme modulaire puissante basée sur le framework Yii2. La nature modulaire de la plate-forme HumHub nous permet d'ajouter de nouvelles fonctionnalités ou de modifier les fonctionnalités principales existantes au moyen de modules personnalisés.

Outre PHP, les langages utilisés sur la plate-forme sont JavaScript, HTML, SQL et CSS / Less. HumHub est basé sur le modèle Model-View-Controller (MVC) et utilise des technologies frontales telles que jQuery, Bootstrap et Less.

HumHub étant basé sur le framework PHP Yii 2.0, nous devons nous assurer également de connaître les concepts de base de ce framework.

➔ Architecture de base de la plateforme



→ Système de fichiers

BookOfDreams/

- assets/** - contient les fichiers publiés par les utilisateurs
- protected/** - contient les fichiers sources, modules, configuration, etc.
 - protected/config** - configuration de base et celle utilisateur
 - protected/humhub** - répertoire principal humhub
 - protected/modules** - répertoire des modules par défaut et utilisateurs
 - protected/runtime** - fichiers d'exécution en cache, index de recherche, etc.
 - protected/vendor** - bibliothèques tierces chargées par le compositeur
- static/** - fichiers statiques (javascript, less, etc).
- themes/** - contient des thèmes autonomes (non regroupés dans un module)
- uploads/** - images de profil des fichiers téléchargés, etc.

Le noyau HumHub consiste en un ensemble de composants, modules, widgets, assistants et vues de base. Par les modules intégrés de base sont les suivant :

- **activity**: activités récentes du réseau social
- **admin**: responsable des problèmes liés à la configuration de l'administrateur
- **comment**: module d'extension de commentaire
- **content**: module de base pour tous les types de contenu (Post, Wiki, ...)
- **dashboard**: fonctionnalité liée au tableau de bord
- **directory**: répertoire utilisateur / espace / groupe
- **file**: module de fichier de base pour accéder au système de fichiers
- **friendship**: module d'amitié utilisateur
- **installer**: module d'installation HumHub
- **like**: mentions j'aime
- **live**: utilisé pour les mises à jour en direct frontend
- **notification**: Notifications utilisateur
- **post**: fonctionnalité simple liée à l'utilisateur post
- **queue**: pilotes de file d'attente et interfaces
- **search**: module de gestion des barres de recherche
- **space**: fonctionnalité liée aux espaces
- **stream**: flux de contenu et murs
- **topic**: les sujets sont utilisés pour classer et filtrer le contenu
- **tour**: guide utilisateur HumHub
- **ui**: composants de base de l'interface utilisateur comme widgets et logique de thème
- **user**: module utilisateur de base

II- Personnalisation du thème : Sonatel

Les thèmes par défaut HumHub peuvent être utilisés directement pour créer des réseaux sociaux propres et professionnels. Toutefois, nous pouvons définir notre propre style, qui correspond à notre esthétique personnelle ou à celle d'un projet d'entreprise existant.

Dans le système de fichier, tous les thèmes ont leurs fichiers situés dans la racine **/themes** ou dans dossier du module en question. Ainsi, chaque thème a son propre sous-dossier, dans le dossier des thèmes principaux. De plus, chaque thème est composé de fichiers index, de fichiers images, d'un ou de plusieurs fichiers LESS / CSS et parfois même de fichiers JavaScript (.js).

Il a donc deux approches différentes pour changer l'apparence :

- Modifier les feuilles de style par défaut (basées sur CSS et LESS) pour ajuster les couleurs, les polices ou les styles des éléments utilisés
- Remplacer les modèles par défaut utilisés dans HumHub pour modifier des éléments, ajuster des positions, supprimer des éléments existants ou ajouter de nouveaux éléments.

Ainsi, dans le cadre de la customisation du front end de notre plateforme nous avons créé un dossier **Sonatel** dans **BookOfDreams/themes** (que nous avons activé à travers l'interface [Administration -> Paramètres -> Apparence](#)) avec les sous-dossiers suivants :

thèmes/

Sonatel/

css/ - Fichiers css de thèmes

theme.css

less/ - Fichiers .less utilisés pour construire theme.css

build.less - Utilisé pour la compilation du less

variables.less - variables de thème comme couleurs de texte et d'arrière-plan

mixins.less - Utilisé pour définir nos propres mixins

theme.less - Contient vos propres définitions de thème

js/ - Fichiers javascript supplémentaires

img/ - Images (facultatif)

views/ - Vues redéfinies des autres modules de la plateforme

user/ - Id du module

auth/

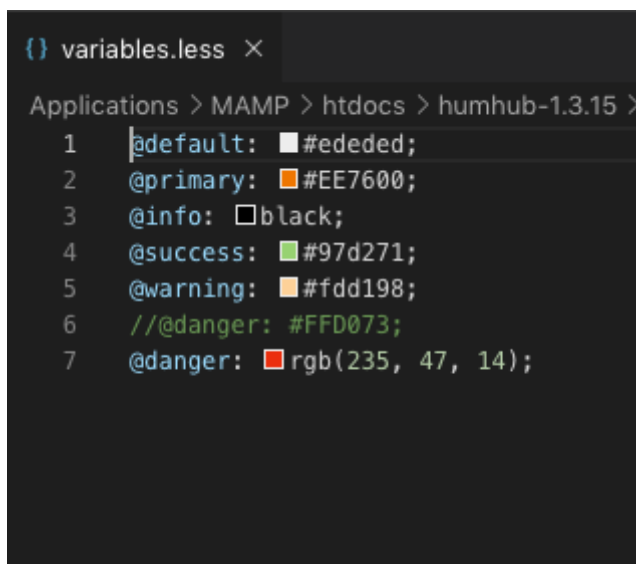
login.php - Modifications apportés au login page

password-recovery/

Par la suite, nous avons eu à :

- ➔ Adapter les couleurs et les font à celui de la Sonatel
 - Modifier le fichier **login.php** dans le répertoire Sonatel
 - Changer la mise en forme dans **static/less/login.less**
- ➔ Customiser le login page
 - Shell : *npm install less -g*
 - Ajouter les variables modifiées dans **Sonatel/less/variable.less**
 - Compiler les fichier less avec le shell : *lessc build.less ../css/themes.css*

- **variables.less**



```
{ } variables.less ×
Applications > MAMP >htdocs > humhub-1.3.15 >
1 @default: #ededed;
2 @primary: #EE7600;
3 @info: black;
4 @success: #97d271;
5 @warning: #fdd198;
6 // @danger: #FFD073;
7 @danger: rgb(235, 47, 14);
```

III- Personnalisation des modules

• Gallery (Catalogue Innovation)

ILab dispose de plusieurs services dans son catalogue et souhaite le publier à destination des collaborateurs de la Sonatel en self-Service.

L'idée est de pouvoir choisir le service du catalogue que le collaborateur souhaite mettre en œuvre en donnant une description claire de besoin et des objectifs attendus.

Ainsi, pour disposer de ce catalogue d'innovation, nous avons téléchargé le module **gallery** au niveau de la partie admin du site et nous avons opéré quelques modifications là-dessus afin de répondre aux besoins du cahier des charges :

- Changer le nom du module de **Gallery** à **Catalogue Innov'** au niveau du fichier `/protected/modules/gallery/messages/fr/base.php`
- Pour éviter que le contenu de chaque image ajoutée dans la galerie ne s'affiche sur le fil de l'actualité, nous avons ajouté une variable **protected \$streamChannel** dans le fichier `/protected/modules/gallery/models/Media.php` et nous l'avons initialisé à **null**.
- Changer le nombre de caractères maximal de la description de 1000 à 20000 (varchar) dans la fonction rules contenue dans le fichier `/protected/modules/gallery/models/BaseGallery.php` et appliquer ce changement dans les tables `gallery_gallery` et `gallery_media` (pour la migration)
- Modifier le header d'affichage du catalogue au niveau de la classe « panel-heading clearfix » dans `/protected/modules/gallery/views/custom-gallery/gallery-view`
- Rendre le module visible uniquement pour l'espace ILAB en ajoutant dans les fichiers `/protected/humhub/modules/space/views/create/modules.php` et `/protected/humhub/modules/space/modules/manage/views/module/index.php`, le bout de code qui suit :

```
<?php if ($moduleId == 'gallery' && !Yii::$app->user->isAdmin())
{
    continue;
}
?>
```

- Enfin désactiver le module dans tous les espaces sauf celui de l'ILAB.

• Calendar (Agenda)

Afin de rendre possible une demande de bootcamp au niveau du département ILAB et de pouvoir gérer les événements au sein de la plateforme, nous avons d'abord installé le module **Calendar** à partir de la partie admin, et nous avons opéré aux modifications suivantes :

- Changer le nom du module de **Calendar** à **Agenda** au niveau du fichier `/protected/modules/calendar/messages/fr/base.php`
- Pour éviter aussi de ne pas afficher tous les événements de chaque espace sur le fil d'actualités, nous avons mis à **false** la variable **\$upcomingEventsSnippetShow**

présente dans le fichier

/protected/modules/calendar/models/SnippetModuleSettings.php.

Nous avons aussi ajouté une variable **protected \$streamChannel** dans le fichier **/protected/modules/calendar/models/CalendarEntry.php** et nous l'avons initialisé à **null**.

- Activer ce module dans tous les profils et espaces à partir de l'interface admin.

• Intaller (ILAB et Mur d'idées)

Chaque nouvel utilisateur de la plateforme doit être par défaut dans les espaces Mur d'idées et ILAB. Pour activer cette fonctionnalité, nous avons :

- Modifié la section **// Create Welcome Space** afin de la personnaliser pour l'espace Mur d'idées (nom, description, etc) dans le fichier **/protected/humhub/modules/installer/controllers/ConfigController.php**
- Fait une copie de cette précédente section pour créer l'espace ILAB et changer le champ **auto_join** à 1 au niveau de base de données.

```
// Create Welcome Space
$space = new Space();
$space->name = Yii::t("InstallerModule.controllers_ConfigController", "Mur d'idées");
$space->description = Yii::t("InstallerModule.controllers_ConfigController", "Your first sample space to discover the platform.");
$space->join_policy = Space::JOIN_POLICY_FREE;
$space->visibility = Space::VISIBILITY_ALL;
$space->created_by = $adminUser->id;
$space->auto_add_new_members = 1;
$space->color = '#6fdb8';
$space->save();
```

• Mostactiveusers (Top 3 idées)

Afin de pouvoir afficher sur le menu de gauche les utilisateurs les plus actifs, nous avons installé le module **mostactiveusers**. Nous avons aussi modifié ce module pour avoir y afficher aussi le top 3 des posts qui ont le plus de « like » sur l'espace Mur d'idées. Pour réaliser cette dernière, nous avons :

- Ajouté deux nouveaux champs à la table post : **nb_like** et **space**

Dans le fichier **/protected/humhub/migrations/m131023_165755.php** :

```
$this->createTable('settings', [
    'id' => 'pk',
    .
    .
    .
    'nb_like' => 'int(100) DEFAULT 0',
    'space' => 'varchar(100) DEFAULT NULL'
])
```

id	message_2trash	message	url	created_at	created_by	updated_at	updated_by	nb_like	space [▲] ₁
1	NULL	Yay ! Je viens d'installer HumHub ;Cool;	NULL	2019-09-27 09:21:20	1	2019-09-27 09:21:20	1	0	NULL
2	NULL	Mon amie qui joue à la guitare	NULL	2019-09-27 09:37:08	1	2019-10-08 12:16:30	1	0	NULL
3	NULL	hg	NULL	2019-09-27 09:38:02	1	2019-09-27 09:38:02	1	0	NULL

- Renseigné sur un post créé dans l'espace de bienvenue (mur d'idées) au niveau de la méthode **actionPost()** dans le fichier **/protected/humhub/modules/post/controllers/PostController.php**

```
public function actionPost()
{
    // Check createPost Permission
    if (!$this->contentContainer->getPermissionManager()->can(new CreatePost())) {
        return [];
    }

    $post = new Post($this->contentContainer);
    $post->message = Yii::$app->request->post('message');
    $post->space = substr($_SERVER['REQUEST_URI'], 17, 19);

    return WallCreateContentForm::create($post, $this->contentContainer);
}
```

- Incrémenté l'attribut nb_like de chaque post après un chaque nouveau « like » sur ce post au niveau de la fonction **actionLike()** dans le fichier **/protected/humhub/modules/like/controllers/LikeController.php**

```

public function actionLike()
{
    $this->forcePostRequest();

    $like = Like::findOne(['object_model' => $this->contentModel, 'object_id' => $this->contentId,
    $post = Post::findOne(['id' => $this->contentId]);
    if ($like === null) {

        // Create Like Object
        $like = new Like([
            'object_model' => $this->contentModel,
            'object_id' => $this->contentId
        ]);
        $post->nb_like++;
        $like->save();
        $post->save();
    }

    return $this->actionShowLikes();
}

```

- Créé une méthode **catch()** dans le fichier **/protected/modules/mostactiveusers/models/ActiveUsers.php** qui va nous permettre de récupérer tous les posts contenus dans le mur d'idées.

```

public static function catch()
{
    $posts = Post::find()->all();
    // var_dump($posts);
    usort($posts, function ($a, $b){
        if($a->nb_like == $b->nb_like)
        {
            return 0 ;
        }
        return ($a->nb_like < $b->nb_like) ? -1 : 1;
    });
}

```

- Créé le fichier **Topbar.php** dans **protected/modules/mostactiveusers/widgets** pour récupérer les informations de la méthode **catch()** et les trier suivant le nombre de « like » et renvoyer le résultat au niveau de la vue suivante.
- Créé le fichier **topbar.php** dans **protected/modules/mostactiveusers/widgets/views** pour pouvoir afficher sur le menu de droite les 3 premiers objets du résultat précédent et y ajouter un peu de style. Ce même fichier est inséré dans une balise « div » du fichier **sidebar.php** contenu dans le même dossier.

```

<div class="panel-body" style = "border: 2px solid orange; border-radius: 10px" >
  <?php
  // run through the array of users
  ?>

  <a href="<?php echo Url::to(['/content/perma', 'id' => $posts[$i]->content-

    <p style="color:black;">
      <?php echo $posts[$i]->message; echo "<hr />";
      ?>
      <b> <?php
        echo $posts[$i]->nb_like;
        if ($posts[$i]->nb_like>1) {echo " votes";}
        else {echo " vote";}
      ?>
      </b>
    </p>
  </a>

  <?php
  }

  ?>
</div>

```

• Autres paramétrages

- Installer le module Chat Wee pour le messagerie instantannée
- Désactiver la possibilité de registration dans le site
- Gestion des cron et de queue :

Les tâches CronJobs sont utilisées pour exécuter des tâches planifiées, telles que l'envoi d'e-mails de synthèse ou l'optimisation d'index de recherche. Elles doivent être configurées dans votre environnement de serveur.

```
> /usr/bin/php /path/to/humhub/protected/yii cron/run
```

Les travaux asynchrones sont utilisés pour exécuter des tâches potentiellement longues en arrière-plan.

```
> /usr/bin/php /path/to/humhub/protected/yii queue/run
```

- Gestion des urls :

Renommer le fichier **.htaccess.dist** résidant dans votre dossier racine Book of Dreams en **.htaccess** et modifier la configuration locale située dans le fichier protected / config / common.php comme suit:

```
<?php

return [
    'components' => [
        'urlManager' => [
            'showScriptName' => false,
            'enablePrettyUrl' => true,
        ],
    ]
];
```