

TP de programmation web : forum

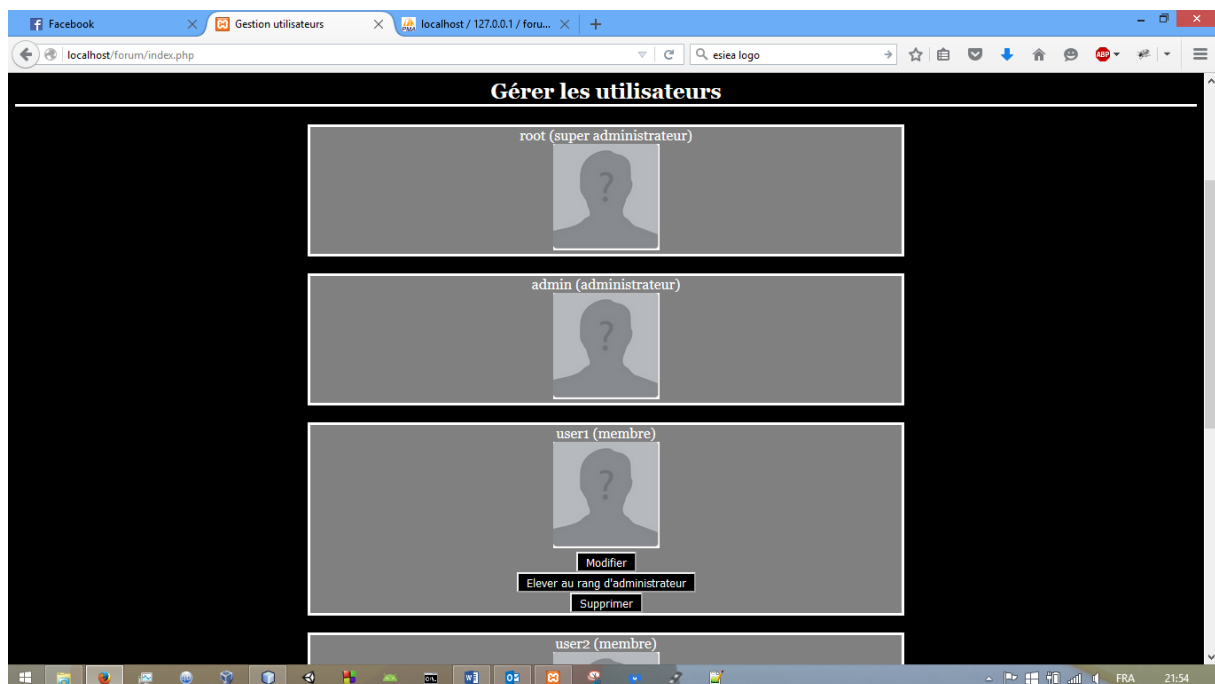


Table des matières

Organisation	3
Architecture.....	5
Fonctionnalités.....	8

Organisation

Il existe trois types de membres : membres, administrateurs, et super administrateur, mais le forum gère aussi le cas d'une personne non connectée

Les sujets disposent des droits suivants :

Pour les non connectés	Pour les connectés
Aucun droit	Lecture seule
Lecture seule	Lecture et écriture

Voici les droits des utilisateurs sur chaque fonctionnalité :

	Non connecté	Membre	Administrateur	Super Administrateur
Accéder à un sujet	Seulement si le sujet est en lecture pour les non connectés	oui	oui	oui
Poster sur un sujet	non	Seulement si le sujet est en lecture et écriture pour les connectés	oui	oui
Créer un sujet	non	oui	oui	oui
Supprimer un sujet	non	non	Seulement si l'auteur du sujet n'est pas administrateur ou super administrateur	oui
Passer un membre en administrateur	non	non	oui	oui
Passer un administrateur en membre	non	non	non	oui
Supprimer un utilisateur	non	non	Seulement si l'utilisateur n'est pas administrateur ou super administrateur	

En résumé :

- Si un sujet n'est pas accessible à l'utilisateur courant, il n'est pas affiché
- Le super administrateur est unique et est le seul compte pouvant destituer un administrateur
- Un administrateur peut élever les droits d'un membre mais ne peut pas le destituer (puisqu'il entre au même niveau hiérarchique)
- Lorsqu'une option n'est pas accessible à l'utilisateur courant, elle n'est pas affichée
- Pour créer un administrateur, on ne peut qu'élever les privilèges d'un membre. Ainsi, il est nécessaire qu'un autre administrateur élève ses droits. Cette propriété de ne pas pouvoir élever soi-même ses privilèges permet de renforcer la sécurité du forum

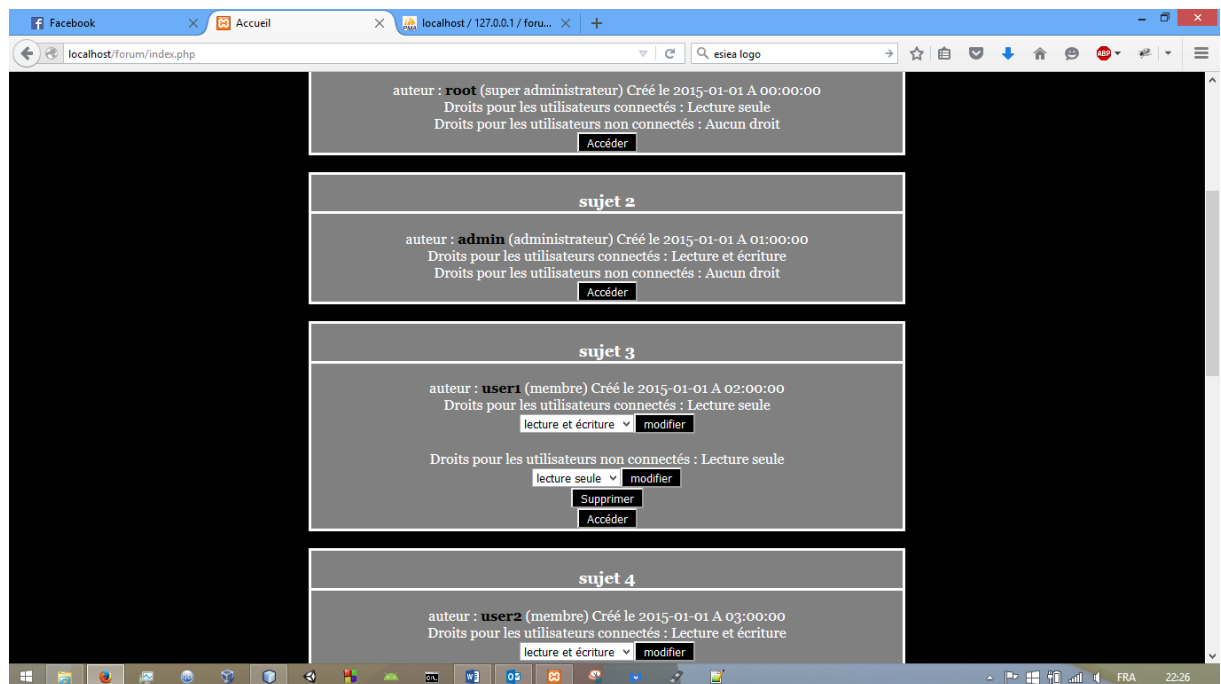
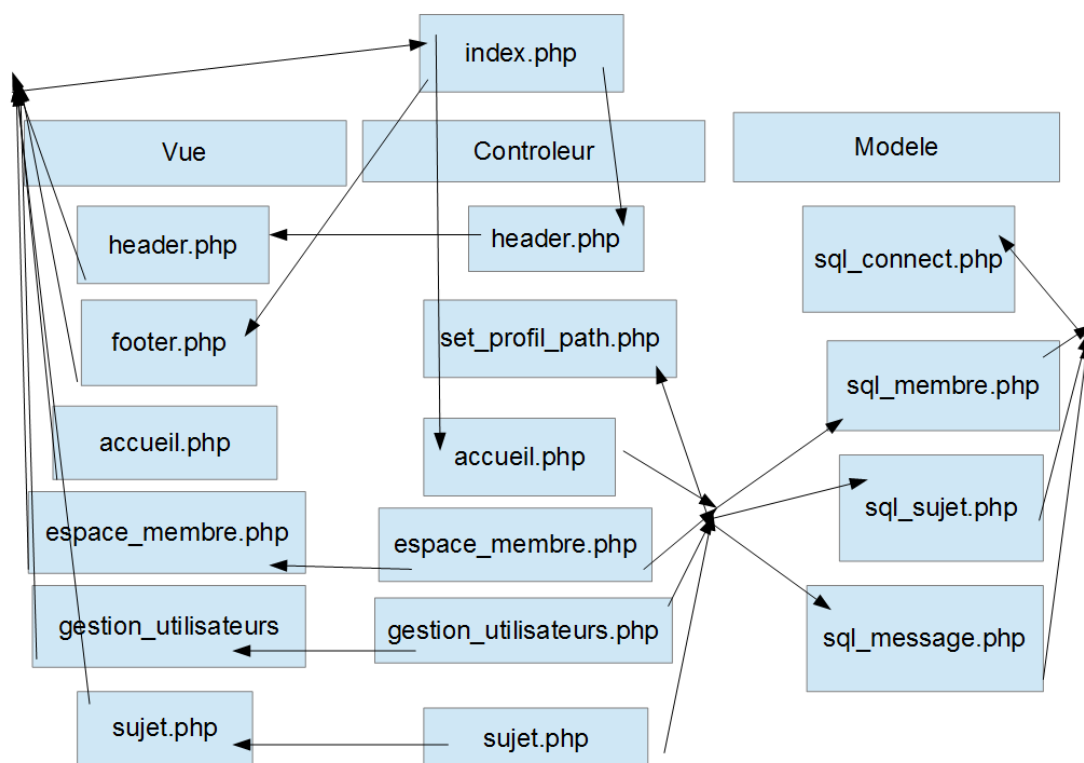


Figure 1 : Certaines options sont affichées ou masquées selon les droits

Architecture

Le forum ne contient qu'une seule page, index.php, mais celle-ci change de contenu en faisant appel à des fonctions du contrôleur organisées en cinq catégories :

- Accueil : gère l'inscription et la connexion, affiche la liste des sujets et permet d'y accéder, de modifier leurs droits ou de les supprimer quand l'utilisateur connecté en a les droits
- Espace membre : affiche les informations relatives au l'utilisateur connecté, et permet de les modifier et d'uploader une image de profil
- Gestion utilisateurs : disponible uniquement pour les administrateurs et le super administrateur, elle permet d'afficher la liste des utilisateurs, et de modifier leur statut, leurs informations et de les supprimer quand ils en ont le droit
- Sujet : affiche un sujet, et par conséquent tous les messages postés sur ce sujet. Quand l'utilisateur connecté en a le droit, il peut supprimer certains messages
- Header : visible à tout moment sur la page, il permet de s'inscrire ou de se connecter lorsque l'on n'est pas connecté, et de se déconnecter, d'accéder à l'espace membre et à la gestion des utilisateurs en fonction des droits de l'utilisateur connecté



Toutes les informations préalables à un traitement sont envoyées par formulaire des vues vers la page index.php. Celle-ci, à l'aide d'un champ « hidden » nommé « form_type », identifie l'opération effectuée et appelle le fichier de fonctions concernées du contrôleur. Puis elle appelle les fonctions nécessaires en utilisant les variables envoyées dans \$_POST depuis la vue.

Les fonctions du contrôleur peuvent faire appel aux fonctions des vues, mais également aux fonctions du modèle, qui gèrent l'accès à la base de données et la récupération des données qui y sont contenues.

La base de données « forum_db » est constituée des trois tables suivantes :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
☐ messages	★ Afficher Structure Rechercher Insérer Vider Supprimer	5	InnoDB	latin1_bin	48 Kio	-
☐ topics	★ Afficher Structure Rechercher Insérer Vider Supprimer	5	InnoDB	latin1_bin	32 Kio	-
☐ users	★ Afficher Structure Rechercher Insérer Vider Supprimer	5	InnoDB	latin1_bin	16 Kio	-
3 tables	Somme	15	InnoDB	latin1_swedish_ci	96 Kio	0 o

Table messages :

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐ 1	id	int(10)		UNSIGNED	Non	Aucune	AUTO_INCREMENT
☐ 2	topic	int(10)		UNSIGNED	Non	Aucune	
☐ 3	auteur	int(10)		UNSIGNED	Oui	NULL	
☐ 4	date_creation	date			Oui	NULL	
☐ 5	heure_creation	time			Oui	NULL	
☐ 6	contenu	varchar(500) latin1_bin			Oui	NULL	

Topic contient l'id correspondant de la table topics, et auteur contient l'id correspondant de la table users

Table topics :

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐ 1	id	int(10)		UNSIGNED	Non	Aucune	AUTO_INCREMENT
☐ 2	titre	varchar(40) latin1_bin			Non	Aucune	
☐ 3	auteur	int(10)		UNSIGNED	Non	Aucune	
☐ 4	date_creation	date			Oui	NULL	
☐ 5	heure_creation	time			Oui	NULL	
☐ 6	droits_nc	int(10)		UNSIGNED	Non	Aucune	
☐ 7	droits_c	int(10)		UNSIGNED	Non	Aucune	

Droits_nc renseigne les droits pour les personnes non connectées :

- 0 : lecture seule
- 1 : aucun droit

Droits_c renseigne les droits pour les personnes connectées :

- 0 : lecture et écriture
- 1 : lecture seule

Table users :

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
☐ 1	id	int(10)		UNSIGNED	Non	Aucune	AUTO_INCREMENT
☐ 2	prenom	varchar(30) latin1_bin			Oui	NULL	
☐ 3	nom	varchar(30) latin1_bin			Oui	NULL	
☐ 4	age	int(11)			Oui	NULL	
☐ 5	email	varchar(40) latin1_bin			Non	Aucune	
☐ 6	email_secours	varchar(40) latin1_bin			Oui	NULL	
☐ 7	login	varchar(30) latin1_bin			Non	Aucune	
☐ 8	mdp	varchar(40) latin1_bin			Non	Aucune	
☐ 9	admin	tinyint(1)			Non	Aucune	

Admin renseigne le statut du compte :

- 0 : membre
- 1 : administrateur
- 2 : super administrateur

Particularité : `set_profil_path.php` est une fonction de type contrôleur utilisée par les autres pages du contrôleur afin de déterminer le chemin vers la photo de profil à afficher

Trois variables de session sont utilisées pour naviguer sur le site : `login`, `admin`, et `sujet_id`, qui permet à la fonction de listage des sujets de savoir quels messages charger.

Fonctionnalités

Accueil

- Créer sujet : vérifie qu'aucun sujet n'existe déjà avec le titre qu'on essaye de renseigner, et fait appel au modèle pour l'insérer en base
- Gestion connexion : vérifie les champs entrés lors de la connexion, si le login existe bien en base, et si le mot de passe correspond au login
- Gestion inscription : vérifie chacun des champs entrés lors de l'inscription, et insère le nouvel utilisateur en base
- Lister sujets : en fonction des variables de session, fait appel à la vue pour afficher les sujets avec les options correspondantes (modifier les droits, supprimer)
- Modifier droits sujets : la vérification des droits sur le sujet ayant été faite au préalable par Lister sujets, fait simplement appel au modèle pour modifier en base les droits du sujet
- Set nouveau sujet : si l'utilisateur est connecté, fait appel à la vue pour afficher le formulaire pour créer un sujet
- Supprimer sujet : Pour les mêmes raisons que Modifier droits sujets, appelle directement le modèle pour supprimer le sujet en base

Espace membre

- Modifier utilisateur : gère le formulaire de modification des informations personnelles, et fait appel au modèle pour répercuter les modifications en base. Si une image de profil a été uploadée, traite la validité du fichier, et crée un répertoire dans vue/utilisateurs ayant pour nom le login de l'utilisateur, dans lequel va être stockée la photo de profil
- Set espace membre : récupère en faisant appel au modèle les informations relatives à l'utilisateur, puis fait appel à la vue pour les afficher dans l'espace membre

Gestion utilisateurs

- Lister utilisateurs : fait appel au modèle pour récupérer les utilisateurs, puis à la vue pour les afficher, en entrant en paramètre le chemin vers l'image de profil récupérée grâce à set_profil_path.php
- Supprimer utilisateur : vérifie grâce au modèle que plus aucun message n'est associé à l'utilisateur que l'on essaie de supprimer afin de garder une cohérence dans la base (il ne peut exister de message sans auteur), puis appelle le modèle pour supprimer l'utilisateur. Supprime également l'image de profil et le dossier ayant pour nom le login de l'utilisateur dans vue/utilisateurs

Header

- Deconnexion : détruit les variables de session, ce qui déconnecte l'utilisateur
- Set header : en fonction des variables de session, fait appel à la vue avec options pour déterminer les boutons à afficher (connexion, inscription, gestion_utilisateurs si administrateur...)

Sujet

- Créer message : vérifie le champ de contenu du message (il ne doit pas être vide), et appelle le modèle pour l'insérer en base
- Lister messages : récupère tous les messages associés à un `topic_id` passé en paramètre, et en fonction des variables de session, appelle la vue pour afficher les messages avec les options correspondantes (option supprimer). Fait appel à `set_profil_path.php` pour indiquer à la vue où chercher l'image de profil
- Set nouveau message : en fonction des variables de session, décide d'appeler ou non la vue pour afficher le formulaire pour poster un message sur le sujet
- Supprimer message : puisque Set nouveau message a déjà contrôlé le droit de supprimer le message, va simplement appeler le modèle pour supprimer le message en base

Conclusion

L'architecture MVC permet de gagner en efficacité et de fractionner le travail de développement. Le fait de partir de zéro (sans framework) permet de maîtriser entièrement son site sur le plan structurel et architectural, mais nécessite davantage de réflexion en amont et de rigueur. En me renseignant sur le fonctionnement et de développement d'un site professionnel en MVC, je constate une nette différence de complexité, mais il n'empêche que je pense que cela reste une bonne expérience, et que je pourrais gagner en structuration et en rigueur en continuant sur la lancée. La décision de faire ou non de l'orienté objet fut difficile, mais j'ai finalement estimé que vu le niveau de complexité relativement faible de mon site n'en nécessitait pas. A l'avenir, sur un projet plus ambitieux et plus professionnel, je pense que le choix de l'orienté objet couplé pourquoi pas à un outil UML sera judicieux. Le framework reste également une solution intéressante dans ce cas.