

Projet Scripts-library

Compte tenu de la proximité de la fin de l'année, le projet PHP de cette année se transforme en une évaluation orale/pratique à préparer :

Principe d'évaluation :

Lors de la dernière séance de TP, chaque étudiant passe un oral pratique au cours duquel il devra :

- répondre aux questions posées (techniques ou théoriques)
- réaliser des manipulations (à effectuer sur le projet Scripts-library)

Ces questions et réalisations sont demandées par le chargé de TP, et sont inspirées des TPs réalisés, et/ou des spécifications demandées dans le projet.

Support de l'évaluation :

Le projet Script-library servira de base à l'interrogation. Les manipulations demandées seront issues des spécifications demandées. Vous avez donc intérêt à travailler sur le projet Script-Library pour préparer votre oral.

Durée :

10 minutes d'interrogation par étudiant

Exemples de manipulations pouvant être demandées :

Implémenter des fonctionnalités :

- Afficher la liste des Utilisateurs à l'adresse **users/all**
- Afficher les scripts par catégorie à l'adresse **scripts/by/category**
- ...

Créer des éléments techniques précis :

- Créer un contrôleur
- Créer une action (et la vue associée)
- Créer/générer un model
- Charger une ou des instances de model depuis la base, les afficher dans une vue
- Ajouter et configurer la gestion des droits sur un contrôleur
- Personnaliser la gestion des droits
- Créer un CRUD controller
- Personnaliser un CRUD controller
- ...

En tant qu'administrateur système, vous utilisez régulièrement des scripts permettant d'automatiser des tâches, qu'ils soient créés par vos soins, ou trouvés sur Internet. Vous êtes également amenés à créer/modifier des fichiers de configuration.

Après quelques années d'expérience, vous en avez créé quelques-uns... que vous ne voudriez pas perdre, compte tenu du travail effectué.

Vos collègues du service sont dans le même cas que vous.

Vous décidez à plusieurs de créer une application web qui permettra de recenser et de sauvegarder les scripts créés par les différents intervenants.

Résumé

Outils à utiliser	<ul style="list-style-type: none">• Ubiquity• Semantic-UI• phpMv-UI
Principales fonctionnalités	<ul style="list-style-type: none">• Gestion des scripts (Ajout, modification, suppression),• recherche,• gestion des droits
Livraison	<ul style="list-style-type: none">• Pas de livraison

Ressources

- [Ubiquity guide](#)
- [Ubiquity api](#)
- [Twig documentation](#)
- [Base de données \(à importer sur votre serveur local mysql via phpmyadmin\)](#)

Règles de gestion

Les utilisateurs peuvent créer un compte, et se connecter à l'application.

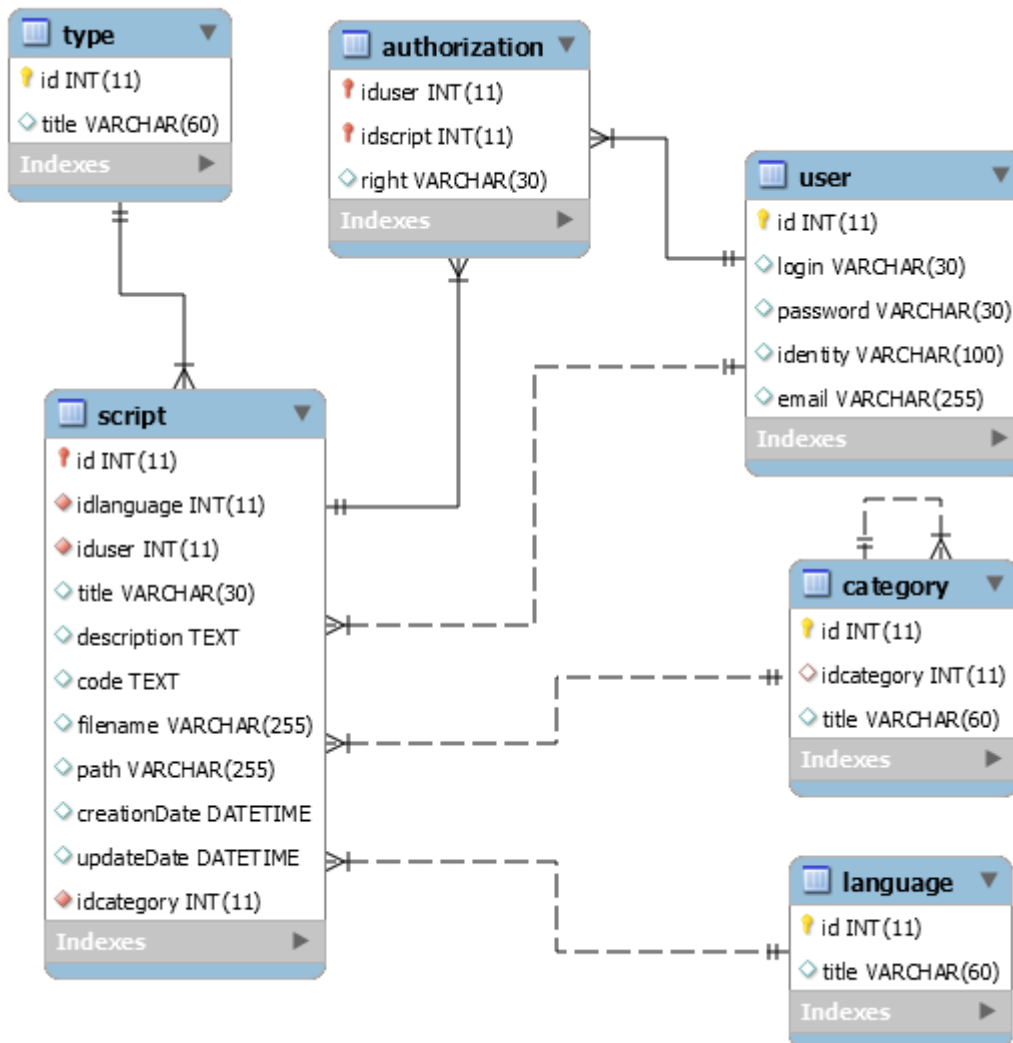
Un script :

- est créé par un utilisateur
- Appartient à une catégorie (config files, bash scripts...)
- est écrit dans un langage (php, bash, python...)
- est d'un certain type (Apache config file, DNS file, CRON...)

Une catégorie peut avoir une catégorie parent : les catégories sont donc structurées de manière arborescente. Les scripts sont classés par catégorie.

Comme avec un système d'exploitation, le propriétaire d'un script peut en modifier les droits :

Schéma de la base



Fonctionnalités à mettre en place

Sur lesquelles vous pourrez potentiellement être interrogé :

- Connexion à l'application
- Affichage des scripts de l'utilisateur connecté, éventuellement classés par catégorie
- Affichage des scripts partagés avec l'utilisateur (sur lesquels il a des droits), éventuellement classés par catégorie
- Parcours par catégories
- Recherche de script
- Affichage des détails d'un script
- Ajout/modification d'un script (uniquement ceux qui appartiennent à l'utilisateur)
- Attribution de droits sur un script
- Gestion des accès (prise en compte des droits)

From:
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:
<http://slamwiki2.kobject.net/php-rt/projets/projet-2018?rev=1527761003>

Last update: **2019/08/31 14:26**



