

Présentation

L'application Docu-Mission permet pour une entreprise de créer un gestionnaire de documentation, qui permettra aux salariés de gérer/créer puis consulter/rechercher dans une documentation partagée.

Source

...

Contraintes techniques

- PHP/Mysql
- Doctrine pour le mappage relationnel/Objet
- CodeIgniter pour la mise en place MVC
- BootStrap pour le fonctionnement côté client

Documents

Script de la base de données

[docu.sql](#)

Diagramme de cas d'utilisation



Listages thèmes / documents par domaine

Modèle

Travail sur 3 classes métiers:

1. Domaine
 2. Thème
 3. Document
- Un thème appartient à un domaine
 - Un thème peut contenir zéro, un, ou plusieurs documents
 - Un document appartient à un thème

Vue

Pour le listage des thèmes / documents par domaine, 1 seule vue est utilisée:

- v_listages (Vue d'ensemble)



DocuMission

Vous etes ici: Accueil->Domaine

Connexion

Listages Themes/Documents par domaine

Domaines:

- [Développement](#)
- [Analyse](#)
- [Réseau](#)
- [Cinéma](#)
- [Sculpture / Architecture](#)
- [Dessin / Arabesque](#)
- [Peinture](#)
- [Musique](#)
- [Danse](#)
- [Littérature / Poésie](#)

Retour

Copyright © | Tous droits reserves | 2014

- Exemple (Cinéma)

Listages Themes/Documents par domaine

Themes:

- [Comédie](#)
- [Tragédie](#)

Retour

- Exemple (Comédie)

Listages Themes/Documents par domaine

Documents:

Les films comiques

Retour

Contrôleur

Le listage des thèmes et des documents par domaine est effectué dans "Listages.php". Le contrôleur possède plusieurs méthodes pour optimiser la mise en place des fonctionnalités. (Voir ci-dessous)

- **index** : Méthode appelée par défaut, appelle les vues supplémentaires tel que le header, footer...
- **refresh** : Méthode appelée par l'index et utilisée pour l'affichage initial et le rafraîchissement après une modification (Paramètre de la BDD/fonction JS...)
- **selectDomaine** : Méthode qui permet d'afficher les thèmes correspondant au domaine qui lui est associé.
- **selectTheme** : Méthode qui permet d'afficher les documents correspondant au thème qui lui est associé. (Si pas de documents dans le thème, affichage vide).

jsUtils

- **jsUtils** : Les fonctions jsUtils sont appelées pour rendre le contenu dynamique.

```
public function __construct()
{
    parent::__construct();
    $this->load->library('jsUtils');
}
```

1. **\$library_src** doit être ajouté à la vue pour charger la librairie jQuery
2. **\$script_foot** script compilé pour effectuer les actions.

```
<?php echo $library_src;?>
<?php echo $script_foot;?>
```

Liste des méthodes utilisées :

1. **getAndBindTo** : Effectue une action suite à un événement (les paramètres de l'id Html sont envoyés automatiquement)
2. **compile** : Ajoute la portion de code écrite vers la vue
3. **doSomethingOn** : Peut servir à ajouter ou supprimer des éléments dans un contenu existant

- Les méthodes jsUtils sont appelées de la façon suivante : `$this->jsUtils->`

```
$this->jsutils->getAndBindTo(".listDomaine", "click", "/dokuMission/Listages/selectDomaine", ".space");
echo $this->jsutils->compile();
$this->jsutils->doSomethingOn(".space", "append", "<a href=\"#\"\" class=\"doc\" id=\"\$do\">\".$dom->getLibelle().\"</a><br>\"");
```

Requête doctrine

Les méthodes doctrine sont appelées de la manière suivante : `$this->doctrine->em->`

1. **createQuery** : Création de la requête
2. **getResult** : Récupération multiple

- Exemple 1:

```
$query = $this->doctrine->em->createQuery("SELECT doc FROM Document doc JOIN doc.theme t WHERE t.id=".$param);  
$theme = $query->getResult();
```

- Exemple 2:

```
$query = $this->doctrine->em->createQuery("SELECT doc FROM Document doc JOIN doc.theme t WHERE t.id=".$param);  
$theme = $query->getResult();
```

From:

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

Permanent link:

<http://slamwiki2.kobject.net/etudiants/2014/croullier/docuwiki?rev=1418889549>

Last update: **2019/08/31 14:32**

