2025/10/30 17:05 1/5 DokuMission

## **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**

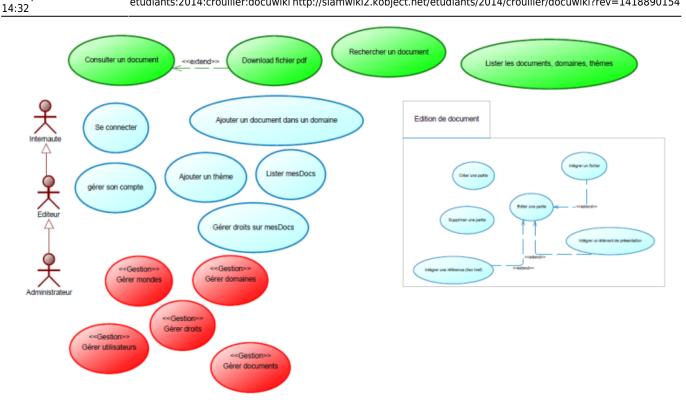
- PHP/Mysql
- Doctrine pour le mappage relationnel/Objet
- Codelgniter pour la mise en place MVC
- BootStrap pour le fonctionnement côté client

## **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:

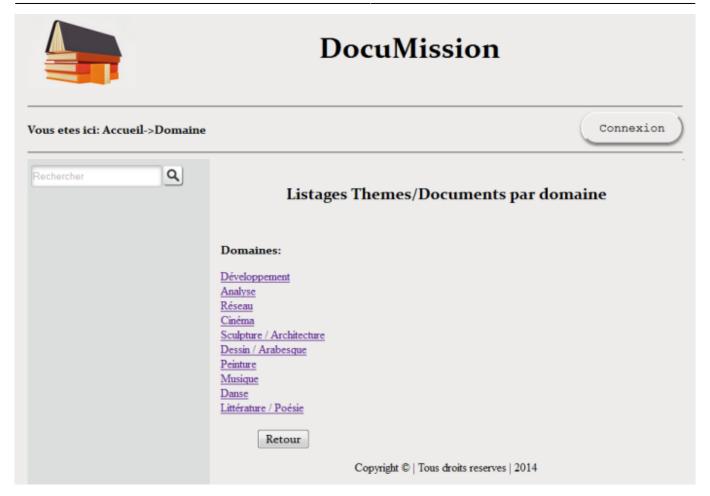
- 1. Domaine
- 2. Thème
- 3. Document
- Un thème appartient à un domaine
- Un thème peut contenir zéro, un, ou plusieurs documents
- Un document appartient à un thème

#### Vue

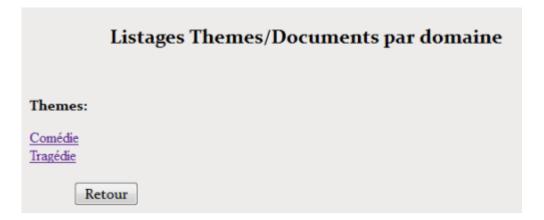
Pour le listage des thèmes / documents par domaine, 1 seule vue est utilisée:

• v\_listages (Vue d'ensemble)

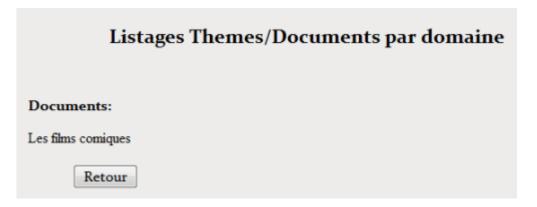
2025/10/30 17:05 3/5 DokuMission



• Exemple (Cinéma)



• Exemple (Comédie)



#### 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 défault, appel les vues supplémentaire (header, left, footer...)
- refresh : Méthode appelée par l'index et utilisé pour l'affichage initial et le rafraîchissement après une modification (Paramêtre 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 la vue pour charger à 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 méthode 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 méthode doctrine sont appelées de la manière suivante : \$this→doctrine→em→

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

2025/10/30 17:05 5/5 DokuMission

```
$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=1418890154



