

Projet Programmation Orienté Objet

Description générale du projet

Objectif : réaliser une application destinée à héberger des fichiers multimédia.

Depuis l'avènement du HTML5 utiliser des fichiers multimédia est devenu très simple coté client. C'est pourquoi nous allons mettre en place une application destinée à héberger ce type de contenu.

Ainsi nous voulons développer un système tel que celui-ci permette :

- de visualiser des vidéos,
- de visualiser des photos,
- d'écouter des sons.

Contraintes d'utilisation :

- pour fonctionner il faudra disposer d'un navigateur HTML5,
- les vidéos devront être au format WebM,
- les images devront être au format gif, Jpeg, png ou SVG,
- les sons devront être au format OGG.

Utilisation de l'application :

- Il existe deux types d'utilisations :
 - l'utilisation non authentifié permettant de visualiser n'importe quel contenu présent. La recherche de celui-ci se fait par un formulaire de saisie contenant les champs suivants :
 - autor pour rechercher un élément déposé par un utilisateur particulier, et alimenté à partir du contenu de la base de données,
 - type pour rechercher un élément de type particulier : vidéo, son ou image
 - descr pour rechercher un élément dont la description contient un mot saisi ici (limité à 50 caractères).

Dans le cas ou la saisie présente plusieurs éléments un système de liens affichant une description sommaire permettra de choisir l'un des éléments.

Suite à la saisie d'un élément unique celui-ci sera visualisé en utilisant la puissance du HTML5, c'est à dire par l'un des éléments video, audio ou img.

- L'utilisation authentifiée permettra en plus de la consultation précédemment décrite, de déposer un contenu. Dans ce cas :
 - un premier formulaire d'authentification demandant identifiant et mot de passe sera rempli,
 - en cas de validité l'identification sera maintenue en session,
 - alors un formulaire de dépôt de contenu sera accessible, celui-ci présentant les champs suivants :
 - fic de type file permettant d'envoyer un fichier (upload HTTP),
 - descr de type text permettant d'indiquer la description du fichier.

Structuration de la base de données :

- les éléments multimédia seront archivés sur le système de fichiers, dans une répertoire application/multimedia segmenté en 3 sous répertoires :
 - videos pour les vidéos,
 - audio pour les sons,
 - images pour les images.
- Ils seront indexés à partir d'une base de données SQL dans laquelle seront présentes les tables suivantes :
 - la table datas destinée aux informations sur les contenus multimédia,
 - la table users destinée aux informations sur les utilisateurs.

Description de la table datas.

Dans cette table seront renseignés les champs suivants :

- chemin_relatif indiquant le nom relativement à la racine ou se situe le média en fonction de son type. Ainsi un fichier de type image ayant pour valeur chemin_relatif=ma_photo.jpg sera localisé dans application/multimedia/image/ma_photo.jpg
- mime_type contenant le type mime du fichier :
 - image/jpeg pour une image de type jpeg,
 - image/png pour une image de type png,
 - video/webm pour une vidéo de type webm...
- pour plus d'infos suivre [ce lien](#)
- description contenant la description textuelle du contenu,
- auteur_id contenant l'identifiant de la personne ayant déposé le contenu,
- date contenant la date de dépôt de ce contenu,

Description de la table users.

Dans cette table seront renseignés les champs suivants :

- id indiquant l'identifiant de l'utilisateur,
- nom indiquant le nom de l'utilisateur,
- passwd indiquant le mot de passe de l'utilisateur haché par du md5 «avec grain de sel»,

Structuration en objet :

Afin de permettre une réutilisation du code et un découpage du travail le code sera structuré en classes de la façon suivante :

- les sessions seront gérées par un singleton Session assurant ce rôle comme vu dans les exercices précédents,
- l'authentification sera également gérée par un singleton nommé Authentification exploitant la classe Session et permettant :
 - de vérifier si l'utilisateur est authentifié par la methode isAuth() retournant true en cas de succès,
 - de vérifier si un identifiant et un mot de passe sont bien valide par la methode checkUser(user, pass) retournant true en cas de succès et validant l'authentification en session,
 - de permettre la déconnexion par la méthode disconnect() retournant true en cas de succès, c'est à dire si précédemment connecté.
- les formulaires seront produits par des classes dédiées permettant :
 - de valider les valeurs saisies par des contraintes internes à partir des tableaux \$_GET et/ou \$_POST par la méthode check() et retournant true en cas de succès,
 - de produire le code HTML à partir des tableaux \$_GET et/ou \$_POST par la methode __toString() tout en présentant les valeurs erronées en cas de saisie incorrecte.
- la présentation des données sera réalisée par des classes dédiées permettant :
 - de visualiser les champs extraits de la base de données,
 - en fonction du contenu le HTML produit s'adaptera au type de média.

La visualisation de la page devra s'adapter au système d'affichage en n'utilisant que les CSS et les directives de type «media query» ; nous sommes en développement d'applications destinées aux dispositifs mobiles !

Afin de se forcer à plus de rigueur dans la génération des pages, celles ci seront servies au navigateur en mode XML ce qui se fait par l directive header("Content-type: application/xhtml+xml") placée évidemment avant tout envoi de contenu.

Constitutions des équipes

le tableau ci dessous donne la liste des équipes travaillant sur le projet.

| prénoms des participants au projet | | | | |
|------------------------------------|----------------------------|------------|----------|-----------|
| équipes | chef de projet et membre 1 | membre 2 | membre 3 | membre 4 |
| équipe 1 | Sandra | Laurent | Yasmine | Florent |
| équipe 2 | Tian | Cyril | Gérard | Bastien |
| équipe 3 | Guilhem | Benoit | Fabio | Christine |
| équipe 4 | Thierry | Christophe | Adeline | Jonathan |

La gestion et le partage des sources se fera par git.

stop