

TP Projet Informatique : Persistance en Php et prise en main SVN / Redmine

Objectif : faire un site pour gérer des articles par catégorie. Chaque catégorie a un nom, chaque catégorie est constitué d'articles (titre, auteur, texte), un article est dans une seule catégorie. Il y aura 3 pages :

- une page pour créer : créer une catégorie, ajouter un article à une catégorie existante
- une page pour afficher tous les articles de toutes les catégories, avec des tris possibles
- une page pour faire rechercher (d'articles, de catégories)

Vous aurez à vous répartir le travail ensuivant des directives, à manipuler le Redmine (version, demandes) et à utiliser le SVN (pour partager vos avancées, avec les bons commentaires de « commit », en gérant des conflits).

Note : les problèmes d'accès concurrents, de sécurité et de login sont laissés de côté.

La persistance des groupes se fera : soit dans un (ou des) fichiers (texte ou csv) rangés dans un ou plusieurs dossiers, soit dans un (ou des) fichiers xml (texte à balise) rangés dans un ou plusieurs dossiers, soit dans une base de données (mysql) soit par un choix parmi tous ces moyens dans la version finale (une page de configuration).

Pour toutes les formes de persistance, vous devez définir la mise en œuvre (organisation des fichiers, des tables, etc.). Ceci sera à indiquer dans le wiki du projet (c.f. plus étape 1).

Vous allez travailler par itération.

L'itération 1 est : faire une page pour afficher toutes les articles de toutes les catégories, faire la lecture des données selon les 3 façons (juste ce qu'il faut).

L'itération 2 est : d'ajouter une page pour « créer et affecter » un article dans une catégorie (sans qu'il y ait de doublon) et faire l'écriture des données selon les 3 façons (juste ce qu'il faut).

Vous auriez pu avoir à définir la ou les autres itérations pour atteindre les objectifs en incorporant au fur et à mesure (et dans un ordre et par étapes à définir) les fonctionnalités suivantes : tri(s), recherche(s), ajout catégorie, feuille de style, page de configuration, compléter les articles (date), gérer les auteurs (recherche, liste par auteur), etc. Mais il ne s'agit pas d'un projet ☺, donc pour ce TP, nous nous contenterons des deux premières itérations.

Etape 1 : découpage du travail (pour itération 1) & description

Le travail se découpe en deux parties :

- la partie « front end », qui génère les pages pour manipuler les catégories / les articles. Le « front end » est utilisé par les internautes.

- la partie « back end », qui gère l'enregistrement des données et leur accès. Le « back end » est masqué aux internautes.
- Dans le cadre de ce tp, le « front end » utilise le « back end » : le « front end » demande des informations au « back end ». L'utilisation est uniquement dans ce sens.

Pour faire la liaison entre les deux, nous allons définir une interface logicielle (ou API du « back end », c.f. vos cours de POO) pour définir un « back end ». Ainsi toutes les implémentations du « back end » auront à implémenter cette interface et cela cachera la façon dont le service est rendu (csv, xml, bd).

Voici une interface de départ (pour l'itération 1) que vous aurez à faire évoluer :

Fichier : Persistence.inc
<pre> interface Persistence { /** * permet de charger toutes les catégories et tous les articles * @return boolean : vrai en cas de succès, faux sinon */ public function charger(); /** * @return array retourne un tableau de toutes les catégories lues */ public function getCategories(); /** * @return array retourne un tableau de tous les articles lus de toutes les catégories lues */ public function getArticles(); } </pre>

Notez que vous pouvez faire évoluer l'interface en classe abstraite si vous le souhaitez.

A vous de définir les classes Catégorie (dans le fichier Catégorie.inc) et Article (dans Article.inc) en choisissant leurs attributs et leurs méthodes (comme par exemple une méthode pour obtenir un code html les représentant).

Pourquoi des fichiers .inc
<p>L'objectif de ne pas utiliser l'extension .php est de faire la différence entre les pages du site (.php) et les fichiers qui structurent la programmation du site (.inc). Par ailleurs, avec une bonne configuration du serveur web, il est possible de rendre inaccessible directement certains fichiers :</p> <pre> <Files *.inc> Order allow,deny Deny from all </Files> </pre> <p>Ce principe s'applique aussi aux fichiers xml, csv, txt, etc.</p>

Dans les premiers temps du tps, vous devez viser la première itération. Aussi vous devez planifier celle-ci. **Il s'agit de l'objectif de l'étape 1.**

Connectez-vous à Redmine (<https://reunion.unice.fr>) avec votre login/mot de passe usuel (ent, machine du 3^{ième}, etc.). Si les pages sont en anglais, cliquez sur My Account (en haut à droite) et

choisissez Français. Allez sur votre projet (bouton Projets en haut à gauche ou raccourci sur la flèche après « Projet de développement L3 Informatique 2012-13 »).

Dans Redmine, sur votre projet, allez dans l'onglet Configuration, et créez une version « priseEnMain ». (Indiquez un nom pour la page wiki)

En suivant les indications ci-dessous, créez les demandes dans l'onglet « nouvelles demandes » en vous assurant bien que vous créez des « Fonctionnalités » pour la version cible « priseEnMain ». Pour l'itération une, la date d'échéance est aujourd'hui. Affectez toujours une personne (choix restreint à une seule). Si une demande doit être partagée entre plusieurs membres de l'équipe, créez des « demandes filles » en utilisant le champ « Tâche parente ».

Les différentes demandes (avec la version « priseEnMain ») et affecter les 8 tâches « individuelles » :

- Créer un projet eclipse (pour tous)
- Créer une des « classes ou interface » Persistance / Catégorie / Article, « commit »
 - o « Update » et compléter avec une des « classes ou interface » manquantes
 - o « Update » et compléter avec une des « classes ou interface » manquantes (l'autre)
- Travailler sur le wiki (pour décrire ce tp, vos approches et solutions, et le principe de répartition des demandes) et Définir des données initiales (Catégorie / Article)
 - o Lors de la création de la base de données, choisissez bien innoDB en moteur (pour pouvoir gérer les relations entre les tables)
 - o Pour le(s) fichier(s) XML, incluez la DTD, ainsi vous pourrez vérifier la validité du document XML avec php (méthode **DOMDocument::validate**)
- Faire la page d'affichage
- Faire lecture fichier csv
- Faire lecture fichier xml
- Faire lecture bd

Etape 2 : débiter le travail : eclipse & première itération

Votre eclipse se situe dans le dossier suivant : /usr/local/eclipse/eclipse-php-3.0.2

Lancez cet eclipse-php et **placez votre workspace dans votre dossier ~/www** (directement ou dans un sous-dossier). Ainsi vos projets seront accessibles via une url du type :

`http://www-mips.unice.fr/~<login>/<workspace>/<nom de projet>/<chemin vers le fichier php>`

Pour créer un projet, allez dans le menu File > New > Project... (ne choisissez pas php project).

Dans la boîte de dialogue « Select a wizard », choisissez « SVN » / « Checkout projects from SVN ».

Créez une nouvelle adresse (« create a new repository location »). Tapez :

`https://reunion.unice.fr/svn/<votre-projet>`

Entrez votre login, puis choisissez « select a wizard » pour sélectionner un projet php. Entrez alors un nom de projet et cliquez sur « finish ». Répondez oui à l'importation depuis le svn (pour l'instant il n'y a rien).

Pour exclure du SVN des fichiers liés à eclipse et à votre compte, allez dans le menu « Windows > / « show view » / « navigator ». La fenêtre s'ouvre généralement en bas d'eclipse. Dans cette nouvelle fenêtre, cliquez droit sur les fichiers avec un « ? » (ils peuvent être .buildpath et / ou .project, et / ou le dossier .settings). Dans le menu contextuel qui vient d'apparaître, allez dans team > add to svn :ignore... Choisissez alors « by name ». **Ceci est à vérifier à chaque création de projet.**

Le membre du groupe qui doit créer l'interface Persistance doit faire les actions suivantes :

- Il doit créer un nouveau fichier php (New Php File) en changeant l'extension (.inc).
- Il le complète (et le sauvegarde localement) c.f. la note ci-dessous sur l'édition du php.
- Il est apparu avec un « ? ». Il doit le partager avec les autres membres du groupe via le SVN.
 - soit clic droit sur le fichier « Persistance.inc » > team > add to Version Control puis commit (clic droit sur le projet > team > Commit). Ajouter un commentaire pour clore la demande associée (fixes #nb)
 - soit commit direct : et sélectionner le fichier « Persistance.inc »

Une fois ce premier « Commit » effectué, les autres membres du groupe peuvent faire un « Update », via un clic droit sur le projet, menu « team » > « Update to HEAD » pour avoir la dernière version.

Edition du php

Placez votre workspace dans votre dossier ~/www (sur votre site web).

Placez des commentaires « intelligents » dans php : devant une variable ou une méthode ou une fonction, tapez : « /**<entrée> ». Vous pouvez alors rentrer les « types » avec l'annotation @var ou @param ou @return (à ajouter pour chaque méthode qui a une valeur de retour). Ces commentaires sont utiles pour la complétion (pas autant qu'en Java) et ce seront les commentaires pour la génération de documentation – API avec phpDocumentor (vous verrez cela lors du dernier cours).

Pour éviter de mettre tous les « include » liés aux classes (surtout en cas d'utilisation de SESSION), vous pouvez ajouter la fonction __autoload. Si vos classes sont dans des fichiers avec le même nom et que tous vos fichiers sont aux mêmes endroits :

```
function __autoload($classname)
{
    // il faut indiquer le chemin depuis le script qui utilise les classes
    include_once "$classname.inc";
}
```

Pour exécuter vos codes php / pages web, vous pouvez consulter vos pages via un navigateur à l'URL `http://www-mips.unice.fr/~<votre login>/<votre workspace>/[chemin de la page dans le projet]`. Vous pouvez aussi utiliser le navigateur interne à eclipse : cliquez droit sur un fichier à essayer, choisissez « Run as » / « Php Web Application ». Dans la boîte de dialogue « Launch », modifiez l'URL pour qu'elle ressemble à `http://www-mips.unice.fr/~<votre login>/<votre workspace>/[reste du texte = chemin de la page dans le projet]`.

Mettez à jour les demandes (en commentaire des « commit » : commencez avec closes #num pour marquer les demandes « résolues », il restera à les fermer manuellement). Faites évoluer le wiki en

corrigeant les intentions initiales ou en complétant avec des informations d'implémentation et d'utilisation résultantes.

Avant chaque « Commit », pensez à faire un « Update » pour ne pas propager des conflits dans le svn, mais de les traiter localement.

En cas de conflit, vous devez résoudre le conflit en éditant le fichier incriminé (manuellement ou en utilisant les outils qui vous seront présentés plus tard en cours). Une fois le conflit résolu, il faut le « marquer » pour SVN : « menu team » / « mark resolved » ou la même action en vue « team synchro ».

Vous rencontrez des conflits lorsque vous utiliserez une persistance plutôt qu'une autre. Pour résoudre ce conflit, propre à chaque membre (celui qui travaille sur la version BD, celui qui travaille sur la version XML, etc.), il faut que le choix se fasse en dehors du code. Il y a plusieurs façons de faire, mais le plus rapide reste le paramètre placé dans l'url : « ?persistance=<nom de la classe> ». Pour récupérer ce paramètre, et l'utiliser, voici un code « minimal » (car il est possible de faire des tests sur le type pour ne pas utiliser n'importe quoi) :

```
/**
 * indique le type de persistance utilisée, avec une valeur par défaut
 * @var string
 */
$typepersistance = "PersistanceFichier";
if (isset($_GET["persistance"]))
{
    $typepersistance = $_GET["persistance"];
}

/**
 * persistance utilisée
 * @var Persistance
 */
$persistance = new $typepersistance();
```

Un autre cas de conflit possible est la configuration de l'accès à la base de données entre un serveur distant (le vôtre euterpe.unice.fr) et un serveur local (si certain développe sur un ordinateur portable... ou chez eux). Dans ce cas, le 1^{er} paramètre de mysql_connect variera d'une configuration à une autre. Une façon de la résoudre / ou de l'éviter :

```
$server = "euterpe.unice.fr";
if ($_SERVER["SERVER_NAME"]=="localhost") $server = "localhost";
// vous pouvez ajouter un test avec le port du server
$connexion = mysql_connect($server, $user, $mdp);
```

Etape 3 : itération 2

Effectuez un travail équivalent aux étapes 1 et 2 mais appliquées cette fois à l'itération 2 : en quelques mots, il s'agit de définir les demandes, de compléter le wiki, de coder, et d'utiliser le SVN.

Etape 4 : finir pour le 5 novembre

Vous devez finir les deux itérations et avoir un Redmine correctement utilisé pour le lundi 5 novembre à 23h59. Uniquement le code « commité » avant cette heure-là sera pris en compte, de même que les demandes et l'édition du wiki. Ce qui suit devra être fait.

A faire par un seul d'entre vous

Pour préparer votre SVN à votre projet, assurez-vous que tous les membres sont à jour. L'un de vous passe dans eclipse en perspective « SVN Repository Exploring ». Sélectionnez toutes les sources, cliquez droit et Choisissez « Branch/Tag ». Dans la boîte de dialogue, complétez l'url pour qu'il ressemble à : <https://reunion.unice.fr/svn/<votre projet>/tag/priseEnMain>

Cochez l'option « Create any intermediate folders that are missing ». Cliquez sur « Next », choisissez « Head revision in the repository ». Cliquez encore une dernière fois sur « Next » et ajoutez un commentaire (par exemple « création d'une version stable de la prise en main »). Et cliquez sur « Finish ». La version a été mise de côté dans « tag », on peut « nettoyer » le SVN.

Toujours dans la perspective « SVN Repository Exploring », sélectionnez tous les dossiers et fichiers sauf « tag ». Cliquez droit et choisissez « Delete... ». Entrez un commentaire (par exemple : « on repart de zero pour le projet ») et validez.

Toujours dans la perspective « SVN Repository Exploring », créez un nouveau dossier appelé « trunc ». Il contiendra la version courante de votre projet. Pour la réalisation de votre projet à proprement parlé, chacun d'entre vous créera un nouveau projet eclipse. Vous choisirez comme base de dépôt (repository) : <https://reunion.unice.fr/svn/<votre-projet>/trunc> Il vous sera possible de « tagguer » des versions intermédiaires « stables » (ou exécutable) en créant de nouvelles « Branch/Tag ».