

Pourquoi le projet openMairie ?

23/03/2006

1- objectifs d'openMairie: créer un framework commun aux applications métiers

- compatible toutes bases de données
- customizable
- compatible tout navigateur (IE et Mozilla)
- compatible php4 php5
- orienté php objet (objet générique et objet métier)
- qui permet le triple contrôle: client (javascript), serveur (php), base de données
- qui permet une ergonomie réactive et dynamique: onglet (multi sous formulaire) = recherche dynamique

2- Construire un framework en utilisant des composants existants :

- dbpear : connecteur de bases de données (incomplet comme copix)
- fpdf: pour la génération de pdf
- menu (sylvain machefer)
- upload
- ajax
- construire un composant « formulaire » et édition » : openMairie

3- Qu apporte le composant openMairie ?

1. une compatibilité complète entre diverses bases en stockant les requêtes dans des fichiers individualisés suivant chaque base de données

- assurer une compatibilité totale entre toutes les bases (dbpear est limité comme copix au requete mise a jour et select si pas d utilisation de fonction spécifique (concatenation, extraction ...)
 - Les requêtes servent pour
 - l'affichage de données en table : tab.class.php
 - l'edition de données: tableau, etiquette, lettre type en pdf
 - l'affichage en formulaire dbform.class.php

2. implémenter des objets métiers définis dans l'analyse UML

- La construction de l'objet métier se fait au travers d'une requete stockée dans un repertoire suivant la base de données utilisée (respect de la compatibilité)
 - l'ordre des champs dans le formulaire ou le tableau est suivant l'ordre de la requête
- l'implémentation en php se fait strictement en fonction de l'analyse UML: les propriétés et les méthodes de chaque objet métier sont traités dans un script : nomOBJET.class.php qui hérite de la classe générique dbform.class.php (interface db / formulaire d'openmairie) qui construit le formulaire avec la classe générique formulaire.class.php (openmairie)

- utiliser par défaut les éléments de la base pour les formulaires afin de réaliser une maquette (quickform: redefinir chaque contrôle)
 - par défaut input text en modification et création, un hidden en suppression
 - par défaut longueur et taille du contrôle = longueur du champ
 - par défaut nom du contrôle = nom du champ dans la base

- les paramètres par défaut de la base sont modifiables en surchargeant les méthodes de dbform dans l'objet métier
 - setLibelle : permet de changer le libellé du champ
 - setMax : permet de fixer la longueur de saisie
 - setTaille: permet de fixer la taille du contrôle
 - setType: permet de changer le type du contrôle:
 - text: contrôle par défaut
 - textdisabled
 - password
 - hidden: contrôle caché
 - static : contrôle affiché en texte, valeur non conservé
 - hiddenstatic: contrôle affiché en texte et valeur récupéré
 - textarea: champ texte
 - pagehtml; champ texte, les \n sont transformés en

 - textareamulti: recupere les valeurs d une zone
 - date: zone date + calendrier.php
 - hiddenstaticdate
 - comboG combo gauche + combo.php
 - comboD combo droite + combo.php
 - localisation: localisation.php
 - select: contrôle select
 - selectdisabled
 - setSelect : passage de paramètre pour combo, select, textareamulti ...
 - setVal: valeur par défaut (création)
 - setOnChange: traitement javascript si mise à jour de la zone
 - setGroupe: groupe des controles sur une même ligne
 - setRegroupe : regroupe des controles dans un fieldset dans un sous tableau

- l'objet métier peut surcharger les autres méthodes de dbform
 - setvalF: passage des valeurs du formulaire dans la base
 - verifier, verifierAjout: méthodes de vérification de données (obligatoire, champ date, numérique ...)
 - clessecondaire: protection des cles secondaires
 - trigger: triggermodifier, triggerajouter, triggersupprimer
 - setId: numérotation automatique d'après une séquence
 - formulaire, ajouter, modifier, supprimer, retour ...

3. Customizer l interface

- l utilisation de css permet une customization façon « plone » (css et img en dyn/custom) plus facile à mettre en oeuvre que les templates (copix, quickform)

4. Dynamiser l'interface avec des techniques AJAX

- recherche dynamique
- présentation en onglet des sous formulaires

Annexe 1 : Interface php et objet métier

affichage en tableau

scr/tab.php?obj=bureau

la requete d affichage est decrite en sql/.../bureau.inc

edition d'un tableau pdf

pdf/pdf.php?obj=bureau

la requete d edition est decrite en sql/..../bureau.pdf.inc

edition etiquette en pdf

pdf/pdfetiquette.php?obj=bureau

la requete d edition est decrite en sql/..../bureau.pdfetiquette.inc

edition de lettre type en pdf

pdf/pdflettretype.php?obj=bureau

la requete d'edition est decrite en sql/..../bureau.pdflettretype.inc

affichage formulaire bureau

scr/form.php?obj=bureau

l'objet bureau est decrit en obj/bureau.class.php

la requete de construction de l objet en sql/.../bureau.form.php

Annexe 2: Description des repertoires du framework

dyn: éléments dynamiques

- menu.inc
- feuilles css : cg.css, menu.css et /custom
- variables applicatives : var.inc
- connexion.php base.php

pdf: interface pdf

- pdf.php : tableau
- pdfetiquette.php : etiquette
- pdflettretype.php

scr : interface ecran

- tab.php: affichage en table
- form.php : affichage d un formulaire
- soustab.php affichage d un sous tab
- soustabdyn.php: affichage d un sous tab en onglet
- sousform.php: affichage d un sous formulaire

spg: interface générique

- login.php
- loginchangepwd.php : changement du mot de passe
- localisation.php
- upload.php + upload.class.php : upload
- combo.php : combo
- calendrier.php
- doc.php

doc

fichier html de la doc en ligne

obj

objets metiers

sql/mysql ou sql/pgsql

requetes SQL et paramétrage: tab pdf ...

trt

traitement

data mysql pgsql

initialisation des bases

rec

recuperation de données

trs

fichiers uploadés

img

image de base de l application

uml

analyse et schémas UML

Annexe 3: Description du composant openMairie

- tab.class.php : affichage en table (en html)
 - méthode afficher, afficherpdf, affichersousformulaire, afficheronglet
- dbform.class.php : méthode abstraite
 - initialise un formulaire
 - modifie les propriétés d un objet stockées en base de données (méthodes ajouter, modifier supprimer)
- formulaire.class.php
 - affiche les controles d un formulaire
 - db_fpdf.php: édition d un tableau en pdf
 - fpdf_etiquette.php: édition d etiquette