

Wednesday, 23 March 2005

Article sur patForms dans le International PHP Magazine

Hier j'ai reçu la confirmation que mon article sur patForms (en) va paraître dans le prochain numéro (03.05) du International PHP Magazine (en). Pour ceux qui ne connaissent pas encore patForms, c'est une librairie qui permet de générer des formulaires. Ce qui la démarque des autres, c'est avant tout sa simplicité et le fait qu'on a le contrôle total de la mise en page et du visuel du formulaire. patForms est un projet de l'équipe de PHP Application Tools (en), développé par Stéphane (en), gERD (en|de) et moi-même.

Mon article est scindé en deux parties, et la première partie qui va sortir bientôt présente l'architecture et les possibilités de base façon tutoriel avec beaucoup d'exemples. Dans la deuxième partie, j'insisterai sur la gestion des évènements, les observateurs, filtres et autres. Pour découvrir patForms c'est l'idéal, surtout que la seule source d'informations à l'heure actuelle est la collection d'exemples (en), qui est très fournie et bien documentée.

Et pourquoi encore une librairie de gestion de formulaires, me direz-vous - il est vrai qu'il y en a beaucoup, mais j'en avais marre de me battre avec des solutions jamais optimales, et je trouve que même des outils très plébiscités comme PEAR::QuickForm sont trop hermétiques et pas assez flexibles. Pourquoi? Parce-que je veux que ce soit intuitif. Si il faut que je tape 60+ de lignes de code super-cryptique juste pour afficher trois champs de formulaire, non merci. Et c'est là la vraie force de patForms: c'est super simple à utiliser - voilà tout ce qu'il faut pour créer un formulaire qui se valide tout seul:

```
// inclure les classes nécessaires
require_once 'pat/patForms.php';
require_once 'pat/patForms/Parser.php';
require_once 'pat/patErrorManager.php';

// définir les options essentielles
patForms_Parser::setNamespace( "patForms" );
patForms_Parser::setCacheDir( "cache" );

// créer l'objet formulaire à partir d'un template
$form =& patForms_Parser::createFormFromTemplate(
    'SimpleRenderer',
    'templates/form.fhtml',
    'templates/form.html'
);

// dire au formulaire de se valider tout seul
$form->setAutoValidate( 'save' );

// afficher le formulaire
echo $form->renderForm();

// le formulaire a-t-il été envoyé?
if( $form->isSubmitted() ) {
    // des erreurs de validation?
    $errors = $form->getValidationErrors();
    if( $errors ) {
        // afficher les erreurs
    } else {
        // faire qq-chose avec les données
        $values = $form->getValues();
    }
}
```

L'intéressant ici, c'est surtout le template de formulaire. Le parseur de patForms permet de générer un objet patForms à partir d'un fichier html ou encore un template patTemplate. Dans ce cas-ci, c'est un fichier html qui pourrait avoir le contenu suivant:

Faut être logué...

Pseudo:

Passe:

On peut donc écrire ses formulaires comme on le faisait avec un formulaire traditionnel, et ainsi avoir le contrôle complet du visuel. Bien sûr patForms offre aussi des méthodes simples pour créer des formulaires de manière dynamique, le parseur n'est qu'un addon.

Allez hop, j'en retourne à mes occupations mondaines - plus précisément la préparation des crêpes au sarrazin

Posté par Argh dans PHP à 17:36