



Le plus grand magazine sur PHP au monde

phpsolutions

Nouvelles technologies et solutions pour les développeurs PHP

PHP N° 3/2009 (33) ISSN 1731-4593 Prix 7,5 EUR CD offert France Metro : 7,50 EUR DOM : 8,80 EUR MAR : 80 MAD TOM/S : 990 XPF

ALLEZ PLUS LOIN AVEC AJAX

CAKE PHP

DÉVELOPPEZ RAPIDEMENT
VOS APPLICATIONS

CONTENT MANAGEMENT SYSTEM

CRÉEZ ET GÉREZ VOTRE SITE
AVEC JOOMLA!

YAHOO! SEARCH MONKEY

AMÉLIORER LA PRÉSENTATION
DE RECHERCHE

PRATIQUE

EXPLORER LA GALAXIE DRUPAL

CRÉEZ DES SITES WEB
DE QUALITÉ PROFESSIONNELLE

PROJETS

QUERCUS

INTÉGRER DU PHP DANS
UNE APPLICATION WEB JAVA



SUR LE CD

UNIQUEMENT CHEZ NOUS
COURS VIDÉO SUR AJAX

HUMAN PICTCHA

VERSION COMPLÈTE
DE L'APPLICATION
D'UNE VALEUR DE 20 \$
ILLIMITÉE DANS LE TEMPS

14389 - 33 - F: 7,50 € - RD



Misez sur vos compétences

10% de réduction

pour les lecteurs de
ce magazine

Administration JBoss 5

3 jours - 1545 € HT

Prochaines sessions

- Lyon 15 septembre 2009
- Paris 9 juin 2009, 29 septembre 2009
- Genève 25 mai 2009, 21 septembre 2009

Programme

- Introduction à JavaEE (rappels)
- Bases de l'administration JBoss 5
- Gestion des configurations
- Monitoring et diagnostic du serveur
- Tuning de Java et de JBoss 5
- Sécurité du serveur et des applications

Intégration Java / PHP

2 jours - 1030 € HT

Prochaines sessions

- Lyon 25 juin 2009
- Paris 19 mai 2009, 3 septembre 2009
- Genève 28 mai 2009

Programme

- Objectif de l'intégration
- WebService : SOAP et REST
- Pont Java-PHP
- Intégration dans Tomcat avec Quercus
- Synthèse et comparatif

Spring Framework

5 jours - 2150 € HT

Prochaines sessions

- Lyon 15 juin 2009, 7 septembre 2009
- Paris 22 juin 2009
- Bordeaux 21 septembre 2009

Programme

- Principes fondamentaux
- Manipuler les beans Spring
- Spring JDBC et Spring / Hibernate
- Gestion des transactions
- Intégration Web : Struts, JSF et Spring Web Flow
- Techniques avancées : AOP et remoting
- Spring Security
- Web Services

Analyse et conception avec UML

4 jours - 2060 € HT

Prochaines sessions

- Lyon 2 juin 2009, 6 octobre 2009
- Paris 8 septembre 2009
- Genève 15 juin 2009, 19 octobre 2009

Programme

- Approche Objet
- Introduction à UML
- Modélisation du comportement
- Modélisation de la structure
- Modélisation des interactions
- Finalisation du système
- Compléments de conception

PHP continue à évoluer en 2009

L'arrivée imminente de PHP 5.3.0 et de ses nouveautés, l'apparition de versions spécialement étudiées à *Microsoft Windows* et l'adaptation aux nouvelles technologies du Web 2.0 sont trois orientations importantes de PHP, qui sont des arguments de poids pour l'adoption de la technologie dans les entreprises.

Si de gros efforts sont déployés pour que PHP reste à la pointe, le chemin n'est pas terminé, car PHP n'est toujours pas une solution d'avant plan pour des projets stratégiques qui touchent au cœur des entreprises. Malgré l'apparition de frameworks sérieux, l'image de PHP dans le monde professionnel en général n'est pas toujours à la hauteur des efforts fournis.

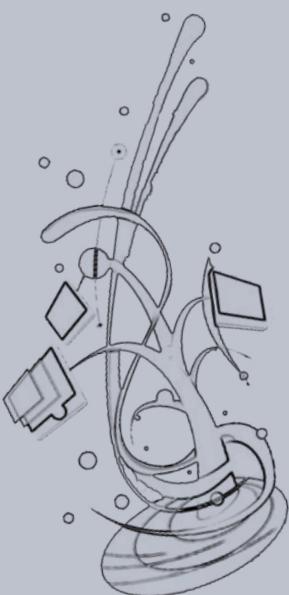
Un autre point ambigu concerne les méthodes d'adoption de PHP pour de gros projets. Plus que jamais, un projet PHP doit être réalisé avec organisation et rigueur. Une bonne connaissance du langage et de ses composants toujours plus nombreux, une architecture organisée et cohérente, une bonne culture générale en informatique et en génie logiciels sont des pré-requis nécessaires.

Aujourd'hui, on peut affirmer que les succès de PHP concernent presque exclusivement des projets liés au web. Les entreprises qui parviennent à s'enrichir en utilisant PHP proposent des gros portails, des intranets ou des solutions de gestion avec interface web. L'interface est importante car elle est le premier élément que l'on voit et souvent le premier argument de vente. PHP, en proposant une intégration parfaite à Flex et autres solutions AJAX, est une solution idéale pour le développement d'interfaces dynamiques.

Le revers de la médaille concerne les systèmes d'information. Des solutions compilées en C# ou en Java sont toujours préférées à PHP pour tout ce qui concerne les manipulations de données métier. Ces technologies, bien qu'interfaçables, ne sont dans la pratique que très rarement mélangées avec PHP. PHP 5.3 offre de nouvelles possibilités objet tel que les espaces de noms, des corrections importantes et une SPL qui évolue. Mais le développement, intuitif et agile, de projets web tels qu'on les connaît ne s'avère pas adapté au développement d'un système d'information complet, plus théorique et plus prédictif.

La nouvelle marche à franchir pour PHP concerne donc l'adoption totale de la POO avec PHP, aidée de frameworks solides, de la SPL, d'outils de maintenance tels que PHAR, et surtout d'une évolution des méthodes de développement copiée sur les grands projets J2EE et .NET.

Guillaume Ponçon, expert PHP, auteur de *Best practices PHP 5* et *Zend Framework, bien développer en PHP* aux éditions *Eyrolles*. <http://www.openstates.com>.



TABLES DES MATIÈRES

VARIA

6 Actualités

Actualités du monde du développement.

8 Description du CD

Présentation du contenu du CD joint au magazine.

OUTILS

10 Yahoo! SearchMonkey

Christophe Villeneuve

En 2008, un nouvel outil réalisé par Yahoo a été mis à la disposition de tous et surtout pour les développeurs et développeuses du monde entier. Cet outil s'appelle SearchMonkey.



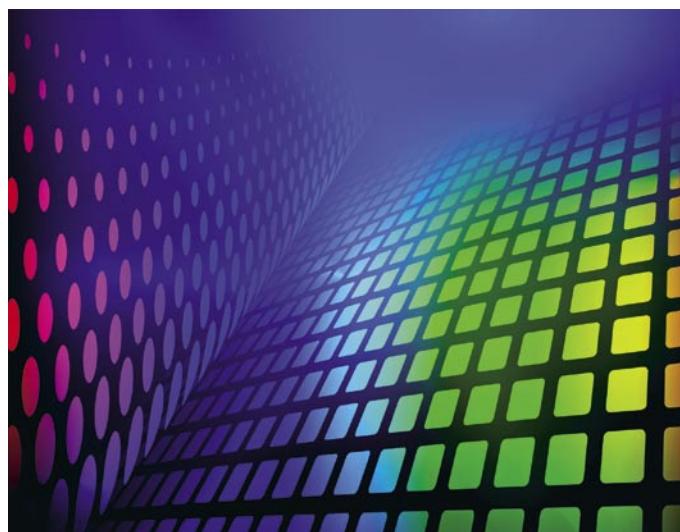
Comme vous allez le voir, il est très souple en ce qui concerne les données : une fois que vous créez une source de données, elle est réutilisable à travers la présentation des demandes. Enfin, Yahoo a adopté une approche très intéressante de la gestion de la plate-forme : elle vous permet d'exécuter votre code PHP sur les serveurs Yahoo.

PROJETS

16 PHP avec Quercus

Alexis Hassler

Contrairement à la plupart des techniques d'interopérabilité entre Java et PHP, Quercus nous propose de marier les deux langages au sein d'un *runtime* commun. Vous verrez comment intégrer du PHP dans une application Web Java, ou comment utiliser des composants Java dans une application PHP.



DOSSIER

20 Messagerie instantanée web avec AJAX et PHP

Nicolas Faugout

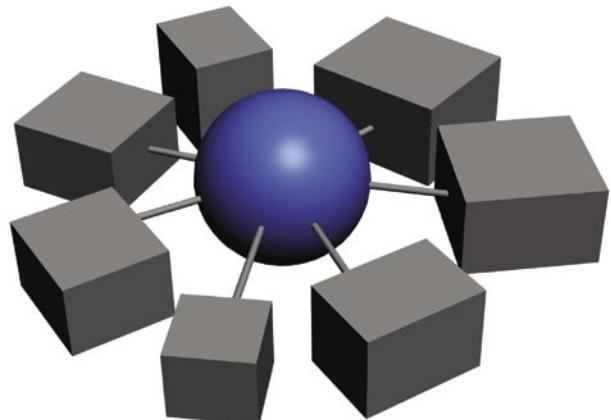
AJAX n'est pas une technologie en elle-même, mais un terme qui évoque l'utilisation conjointe d'un ensemble de technologies libres couramment utilisées sur le Web. La synchronisation des données entre clients est un enjeux essentiel des applications AJAX. Dans ce dossier vous allez étudier, via un cas pratique, la méthode Push Server. C'est celle utilisée par la plupart des messageries web.

PRATIQUE

26 Création de template Joomla! 1.5

Panya Sayavongsa

La création de site Internet avec le CMS Joomla! rencontre un vif succès en partie grâce à sa capacité d'évolution et d'adaptation. On compte à ce jour plus de 5000 extensions de tout type : composants, modules, *plugins*, langues et templates. Nous allons nous intéresser à ces derniers dans cet article.



36 Barre d'avancement de téléchargement en PHP

Lucas Corbeaux

L'envoi de fichier sur un serveur par l'intermédiaire de PHP a toujours été une tâche pénible et hasardeuse pour l'utilisateur. Depuis l'arrivée de la version 5.2 de PHP, il est désormais possible, à l'aide de quelques extensions, de mettre en place une barre d'avancement de téléchargement. Vous allez voir comment dans cet article.

42 Créer votre blog avec Drupal

Brice Favre

Le choix d'un CMS se fait avant tout en fonction des besoins des sites que vous comptez mettre en place avec lui. Drupal permet de déployer rapidement des sites web dynamiques. C'est un CMS de plus en plus utilisé à un niveau professionnel. Complet, il bénéficie d'un support communautaire important et de modules À travers cet article vous découvrirez comment l'utiliser au mieux.

E-COMMERCE

48 Hébergement E-Commerce

François-Xavier Bois

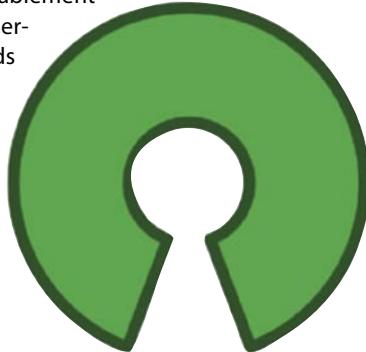
Souvent considéré comme la dernière roue du carrosse, l'hébergement est pourtant un aspect essentiel de la réussite d'un projet e-commerce. Cet article vise à présenter les différents points sur lesquels votre attention doit se porter lors du choix de votre futur hébergeur.

50 Quand l'Open Source révolutionne la distribution

Cyril Fouillot, Samuel Paccoud

À l'heure où le e-commerce est en plein essor, les sites marchands doivent être à la pointe de l'innovation pour conquérir les marchés et conserver leurs marges. La rencontre entre CMS et solution ERP matures constitue indéniablement

l'une des clés majeures pour permettre aux cyber-marchands de développer de nouveaux avantages concurrentiels.



54 Bien choisir sa boutique en ligne

Sébastien Joncheray

Vendre en ligne c'est urgent, simple et très rentable si vous faites les bons choix. Voici un panorama des types de solutions existantes, et des critères à prendre en compte pour choisir et passer à l'action. Suivez le guide !

FICHE TECHNIQUE

60 CakePHP le framework pour développer rapidement vos applications

Pierre-Emmanuel Fringant, Pierre Martin

CakePHP a été conçu afin de faciliter au maximum la vie du développeur qui l'utilise. Cependant il faut au préalable comprendre la philosophie du framework et la suivre au départ de manière presque aveugle. Dans cet article vous verrez les points clés de CakePHP permettant de se lancer dans son apprentissage.

68 Introduction au partitionnement avec MySQL 5.1

Olivier Dasini

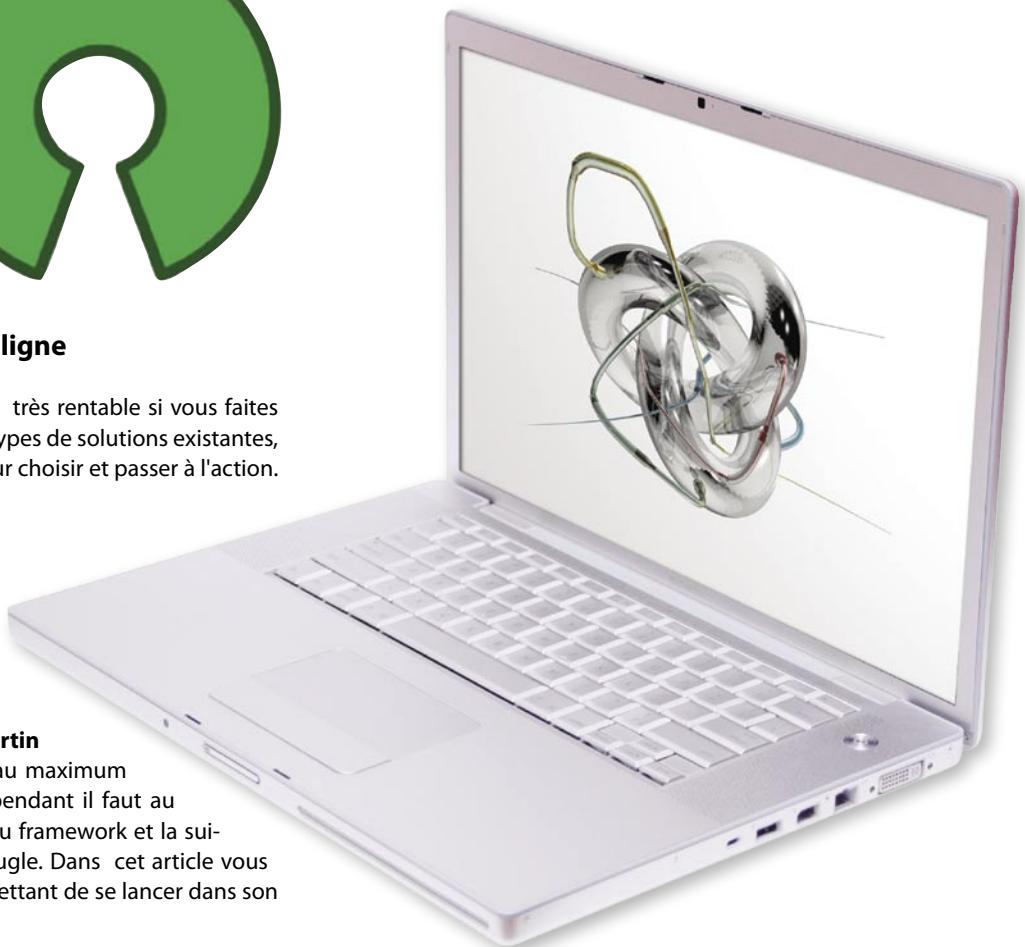
Après de longs mois d'attente, MySQL 5.1 est (enfin) sortie en *Generally Available* (GA). L'une des fonctionnalités phare de cette nouvelle mouture est la possibilité de partitionner ses tables. Découvrez comment manipuler le partitionnement avec MySQL 5.1.

POUR LES DÉBUTANTS

74 Les nouveautés de HTML5

Cécile Odero, Magali Contensin

HTML5 est la 5ème version majeure de HTML. C'est le successeur de HTML 4.01 et de XHTML 1.0. Ce standard est encore en cours de définition mais certaines fonctionnalités sont déjà disponibles dans les navigateurs. Il intègre *Web Forms* 2.0, des éléments multimédias et permet une meilleure description du contenu.



PHP 5.2.9

La version 5.2.9 vient de sortir avec une cinquantaine de correctifs dont plusieurs touchant la sécurité. Il est indispensable de passer à cette version par rapport aux anciennes versions. Par ailleurs, les extensions PECL ne sont plus distribuées automatiquement avec la version Windows mais vous pouvez toujours utiliser les extensions distribuées avec les anciennes versions de PHP 5.2.x
<http://www.php.net/archive/2009.php>

Forum PHP 2009

Le prochain *Forum PHP* organisé par AFUP se déroulera les 12 et 13 novembre 2009 prochains à la cité des sciences à Paris. Mais pour patienter, un *Barcamp PHP* serait prévu en juin 2009.
<http://www.afup.org/>

10 erreurs classiques

Trouver des sources ou des scripts sur internet, c'est assez facile, surtout avec des sites internet qui proposent de regrouper l'ensemble de ces scripts. Cependant la qualité peut poser quelques soucis si on les utilise sans aucune modification. Le site *Nettuts.com* montre comment repérer les erreurs classiques et surtout comment les éviter.
<http://nettuts.com/articles/are-you-making-these-10-php-mistakes/>

Activité autour de Joomla!

Depuis le 1er janvier 2009, Joomla! 1.5 est l'objet de toutes les attentions car c'est le projet qui évolue le plus mais aussi car c'est la fin du développement de Joomla! 1.0.x. De nouvelles extensions ont été converties pour devenir compatibles, un convertisseur automatique pour passer vos sites de la version 1.0 à la version 1.5, une nouvelle documentation *Joomla pour les nuls 1.5...* Et pour regrouper l'ensemble des utilisateurs de Joomla!, une association : AFUJ (Association Francophone des utilisateurs de Joomla!).
<http://www.joomla.fr/>

PHP Live DOCX

PHP Live DOCX est une nouvelle API proposant l'exportation des formats DOC et DOCx (formats utilisés dans les traitements de textes) ainsi que les formats standards comme RTF et PDF. De nombreux exemples et documentations pour l'insérer dans vos applications très facilement.
<http://www.phpliv/docx.org/>

PHPquery 0.9.5.343

PHPquery est basé sur *Jquery*. Il est basé sur DOM et réalisé en PHP 5 avec une gestion de commande en ligne intégrée. Son intégration est très facilement réalisable dans vos applications.
<http://code.google.com/p/phpquery/>

LDAPau 1.1

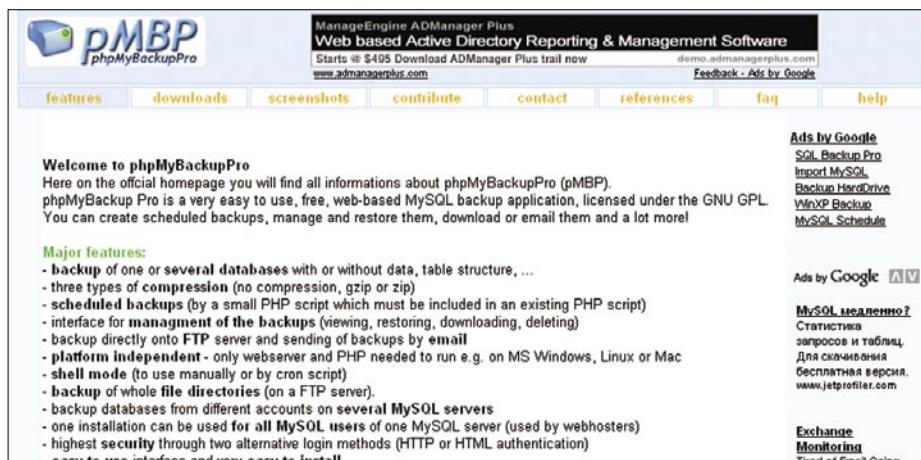
LDAP est le protocole d'annuaire sur TCP/IP et *LDAPau* un projet IHM réalisé en PHP vous permettant d'administrer les utilisateurs et les groupes d'un serveur LDAP.
<http://www.sourceforge.net/projects/ldapau/>

phpMyBackupPro

PhpMyBackupPro est une application internet permettant de sauvegarder l'ensemble de vos bases de données MySQL avec PHP. Une interface va vous permettre de configurer et programmer suivant vos envies et surtout vos besoins. Les fonctions disponibles sont accessibles par une interface en français et proposent un certain nombre de fonctionnalités.

Les fonctions disponibles sont :

- La compression au format GZIP,
- La sauvegarde des répertoires de fichiers ZIP,



The screenshot shows the homepage of phpMyBackupPro. At the top, there's a banner for 'ManageEngine ADManager Plus' with a 'Starts at \$495' offer. Below the banner, the main navigation menu includes 'features', 'downloads', 'screenshots', 'contribute', 'contact', 'references', 'faq', and 'help'. The 'features' section is expanded, showing a list of major features: backup of one or several databases with or without data, table structure, three types of compression (no compression, gzip or zip), scheduled backups (a small PHP script), interface for management of the backups (viewing, restoring, downloading, deleting), backup directly onto FTP server and sending of backups by email, platform independent (only webserver and PHP needed), shell mode (use manually or by cron script), backup of whole file directories (on a FTP server), backup databases from different accounts on several MySQL servers, one installation can be used for all MySQL users of one MySQL server, highest security through two alternative login methods (HTTP or HTML authentication), and an easy-to-use interface and very easy to install. To the right of the features list, there are several ads: 'Ads by Google' with links to SQL Backup Pro, Import MySQL, Backup HardDrive, WinXP Backup, and MySQL Schedule; 'MySQL 2008' with a link to MySQL 2008; and 'Exchange Monitoring' with a link to Exchange Monitoring.

RAYNETTE : nouvelles offres E-Commerce originales, clé en main

La société *Raynette* commercialise une nouvelle version de sa propre solution de boutique en ligne, fruit de 3 ans de développement et de 10 ans d'expérience. L'offre consiste en une boutique en ligne hébergée avec assistance, mises à jour très régulières, sans aucun frais d'installation ni engagement.

L'originalité de l'offre consiste en une formule de base très abordable au vu de la qualité technique (49 EUR HT/mois), à laquelle vous pouvez ajouter toutes sortes de modules optionnels selon vos besoins (envois de SMS, stocks, vente d'articles à télécharger, factures client, export vers les comparateurs de prix, etc). Ainsi vous obtenez une boutique en ligne qui ne comprend que les fonctionnalités dont vous avez besoin.

Deux avantages :

- cela optimise le coût de l'abonnement mensuel,
- cela évite que vous soyez confronté à toutes sortes d'options qui vous sont inutiles et qui vous compliqueraient la tâche.

- La gestion de sauvegarde,
- L'enregistrement sur un serveur FTP ou vers un autre base de données MySQL...,
- Le téléchargement par courriel,
- Les sauvegardes classiques ou programmées.

L'interface se veut intuitive pour vous faciliter son utilisation avec la possibilité d'utiliser de nombreuses langues différentes. L'installation est automatique et une aide en ligne est disponible : <http://www.phpmybackuppro.net/>.

Des modules optionnels peuvent être ajoutés ou supprimés chaque mois sur simple demande.

Des packs sont également proposés, ils contiennent une sélection de module selon le profil de boutique souhaité.

La société *Raynette* existe depuis 10 ans, et a environ 1600 clients.

Site web : www.raynette.fr



The logo for Raynette features a stylized green frog with large white eyes and a wide smile. Below the frog, the word 'raynette' is written in a lowercase, bold, sans-serif font. Underneath 'raynette', the words 'Applications serveur' are written in a smaller, bold, sans-serif font, all contained within a dark red rectangular box.

Serveur Web d'applications PHP pour les applications critiques

Zend Server est un serveur d'applications Web stratégiques qui permet de faire fonctionner et de gérer les applications PHP qui nécessitent un haut niveau de fiabilité, de performance et de sécurité.

- PHP professionnel – Un stack PHP à jour, testé et supporté qui assure une haute fiabilité, améliore la sécurité et améliore la productivité de vos équipes ;
- Pour un déploiement en confiance – Un environnement complet et consistant utilisé en développement, en test et en production élimine de nombreux problèmes rencontrés durant la phase de déploiement ;
- Une réponse rapide aux problèmes – Le monitoring des applications couplé à des

outils de diagnostic avancés permettent de détecter rapidement les problèmes et leurs causes ;

- Performance d'applications – Les outils d'optimisation et d'accélération embarqués permettent d'assurer de hautes performances et une faible utilisation des ressources.

Donc à découvrir sur le site :

<http://www zend.com/fr/products/server/>



Magix cjQuery

Magix cjQuery, une bibliothèque PHP pour intégrer jQuery réalisée en PHP 5 et Open Source. Magix cjQuery peut intégrer l'ensemble des types de conceptions pouvant exister. Par ailleurs il est compatible avec de nombreux frameworks PHP. En plus, il va vous permettre de contrôler vos données échangées dans les champs de formulaire.

Cette application va vous permettre de contrôler et de manipuler jQuery dans votre code PHP. Son utilisation a été conçue pour faciliter la vie des développeurs et par conséquent un *plug-in* est disponible pour manipuler jQuery. Il a été conçu pour optimiser des performances de vos sites web afin de garantir un affichage



rapide dans votre site internet. La configuration se traduit par l'appel d'une seule ligne de commande. Et l'utilisation a été optimisée et utilise directement le cache des visiteurs sans que vous interveniez directement. Une documentation est disponible pour vous aider à intégrer et utiliser Magix cjQuery. Le site officiel montre aussi que des tests de performances ont été réalisés pour vous montrer l'intérêt de vous en servir.

<http://magix-cjquery.com/>

A screenshot of the Magix cjQuery website. The header features a navigation bar with links for Accueil, Installation, Téléchargement, Changelog, Blog, and Contact. Below the header are two small boxes: 'Programista PHP Praca' and 'Document X'. The main content area has a heading 'Magix cjQuery, une bibliothèque php pour intégrer jQuery'. It includes a paragraph of text, a list of bullet points, and a section titled 'Manipuler des plugins jQuery'. On the right side, there are two forms: 'Rechercher' and 'Lettre d'informations', and a 'Sondage' section with a radio button poll.

Projet Bespin

Le projet *Bespin* est un projet signé par Mozilla. Il s'agit d'un éditeur de code collaboratif pour la réalisation de sites internet. La version actuelle permet de couvrir l'ensemble des fonctionnalités du JavaScript. En plus il permet d'ouvrir le même fichier en simultané par plusieurs personnes.
<http://labs.mozilla.com/projects/bespin/>

Une Reste API

Le site *Gen X Design* propose une API appelé REST API. Elle peut être utilisée comme une couche d'interface sur le haut d'une application PHP. De nombreuses possibilités sont offertes comme l'utilisation de XML et de JSON.
<http://www.gen-x-design.com/archives/create-a-rest-api-with-php/>

Livre blanc international

Inspiré du livre blanc PHP de l'AFUP, la communauté PHP Irlande a traduit cet ouvrage en version anglaise. Pour le découvrir, il est disponible sur le CD joint au numéro ou à partir du lien suivant <http://www.php.ie>.

PHP Edit 3.2

La nouvelle version de *PHP Edit* est actuellement disponible. Cette version majeure apporte de nombreuses évolutions comme le support du framework symfony ainsi que des améliorations dans la façon de travailler. En plus l'utilisation de YAML va vous permettre d'édition facilement vos fichiers de configuration avec la syntaxe, le smart tiret, niveau de colorisation...
<http://fr.phpedit.com/>

Yslow 2.0 aperçu

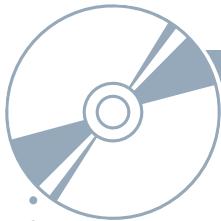
Yslow est une API vous permettant de mesurer les performances d'un site internet. La version 2 a été présentée voici quelques semaines, et ne va pas tarder à être disponible. Cette nouvelle version apporte un certain nombre de corrections mais aussi de nombreuses évolutions dont le blog *performance.survol* parle.
<http://performance.survol.fr/2009/01/apercu-de-yslow-20/>

PHP TV

PHP TV, met à disposition un grand nombre de conférences au format vidéo concernant la couverture du Forum PHP 2008 organisé par AFUP. Grâce à ce dispositif, et si vous n'avez pas pu vous rendre à cet événement, vous pourrez quand même voir les présentations.
<http://www.phptv.fr/>

Alternative à Google Analytic

Google Analytic est une API (application) pour obtenir des statistiques de vos sites internet et pages visitées. Le blog *ilogan* montre qu'il n'y a pas que *Google Analytic* pour obtenir des statistiques de vos sites internet et pages visitées et qu'il existe d'autres outils permettant d'analyser vos pages, réalisés en PHP et MySQL par exemple : *Piwik*, *Firestats*, *Bbclone*, *Woopra*.
<http://www.ilogan.info/4-alternatives-a-google-analytics/>



Cours vidéo : messagerie web instantanée PHP/AJAX

Nous vous présentons le cours vidéo réalisé spécialement pour PHP Solutions par Nicolas Faugout, de la société *Lucca*. Pour mettre en place ce tutoriel, vous devez posséder ou installer un serveur web (Apache par exemple) et si possible, être visible de l'extérieur (sur Internet).

Dans la plupart des applications AJAX, les échanges de données client/serveur sont toujours à l'initiative du client. Nous allons voir comment on peut simuler un échange de données à l'initiative du serveur, autrement dit un *Push Serveur*. L'exemple parfait pour mettre en valeur un *Push Serveur* est la messagerie web. Chaque client doit recevoir les nouveaux messages en provenance des autres clients. C'est donc bien le serveur qui doit envoyer aux clients les nouveaux messages.

Organisation des fichiers

Nous avons un seul fichier client HTML et 2 fichiers serveurs. Le premier pour récupérer les informations en provenance du serveur, le fichier réalisant le *Push Serveur*. Le deuxième, plus classique, pour se connecter, insérer un nouveau message, se déconnecter. En plus, nous avons une librairie pour gérer les variables d'application, quelques images et des fichiers de données.

Architecture de l'application côté client

Côté client, nous avons un fichier HTML classique avec inclusion de SCRIPT JavaScript, définitions CSS en ligne et squelette HTML de la page. Nous utilisons principalement des DIV imbriquées pour assurer la disposition des différents éléments de l'application. Nous avons également besoin de quelques éléments de formulaire pour que l'utilisateur puisse saisir ses entrées.

Mise en place du Push Serveur

Le fichier servant au Push serveur permet de retarder la réponse au client et donc de la déclencher à l'initiative du serveur. Le délai de réponse au client est donc variable et intervient dès qu'un nouveau message est enregistré. Côté client, au chargement initial de la page comme à chaque fin d'écoute, une fonction JavaScript lance une requête AJAX vers le fichier *Push Serveur*.

Pour un premier essai, nous simulons l'écriture dans le fichier de données d'un nouveau message pour voir comment le serveur et le client réagissent. Le serveur interroge le fichier de données régulièrement et répond au client dès que le fichier est non vide. Le client affiche alors dans la zone des messages ce qui a été écrit dans le fichier de données.

Récupération des Input client par un fichier serveur

Ce fichier gère les données entrantes du client vers le serveur. Il s'agit des connexions/déconnexions et des nouveaux messages. Sa mission principale est de transférer ces données à tous les clients, via le *Push Serveur*. Il va donc écrire les données dans le fichier de données. La boucle est ainsi bouclée. Nous simulons donc l'envoi de données au serveur, la propagation des données du fichier entrant au fichier sortant et la réponse de ce dernier au client.

Implémentation des fonctionnalités de la messagerie

Un champ permet de saisir son *login* et de se connecter en appuyant sur *Entrée*. Le client écoute le serveur qui propage alors la nouvelle connexion à tous les clients. Notre client reçoit donc également l'information de sa propre connexion. Au chargement de la

page comme à toute nouvelle connexion, nous devons récupérer la liste des utilisateurs connectés. Cette liste est stockée côté serveur grâce à la librairie de gestion des variables d'application.

Pour simuler plusieurs utilisateurs, nous ouvrons une *Session Firefox* et une *Session Internet Explorer*. L'un et l'autre voient la connexion de chacun. L'information de connexion passe d'un client à l'autre par l'intermédiaire du serveur. Si le premier utilisateur poste un message, le serveur le stocke dans le fichier de données et le *Push Serveur* le propage à tous les clients, dont le deuxième utilisateur.

Enfin, quand un utilisateur ferme son navigateur ou change de page, il doit être déconnecté de la messagerie. Sur ces événements, on envoie une requête AJAX de déconnexion au serveur. Le serveur propage alors l'information aux clients encore en ligne.

ZZEE Human Pictcha (20 \$)

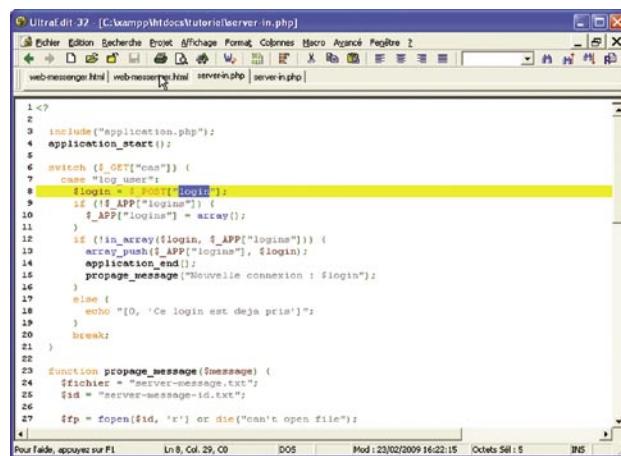
Human Pictcha est une image de protection (un *captcha*) dans un formulaire, qui peut être intégrée dans vos formulaires, et qui va filtrer les spams de forum, livre d'or, formulaire, inscription, attaques de mots de passe, etc. Il s'agit d'une bibliothèque et d'un ensemble d'outils et d'exemples pour PHP qui peuvent être également utilisés avec Perl. Vous trouverez la configuration requise ainsi que la description détaillée dans le fichier d'aide disponibles à l'adresse suivante : <http://zzee.com/human-pictcha/help.html>.

Ce logiciel est assez facile à installer et à intégrer dans vos formulaires PHP ou Perl. Human Pictcha est livré avec le code source libre, ce qui est un gros avantage. Voici les plus importantes caractéristiques de Human Pictcha :

- Sécurisé et difficile à vaincre par OCR. Vous pouvez modifier de manière significative le niveau de protection par l'activation ou la désactivation des fonctions de sécurité et l'évolution des différents paramètres.
- Vous pouvez intégrer Human Pictcha dans vos formulaire PHP ou Perl.
- Compatible avec le serveur web de configuration typique (PHP/MySQL).
- Le code de l'image peut être composé de chiffres et de lettres ou de chiffres seulement, ce qui permet de s'adapter aux marchés internationaux.

Nous vous proposons la version complète de cette application d'une valeur de 20 \$. Pour plus d'informations sur le produit nous vous invitons à visiter le site : <http://www.zzee.com>.

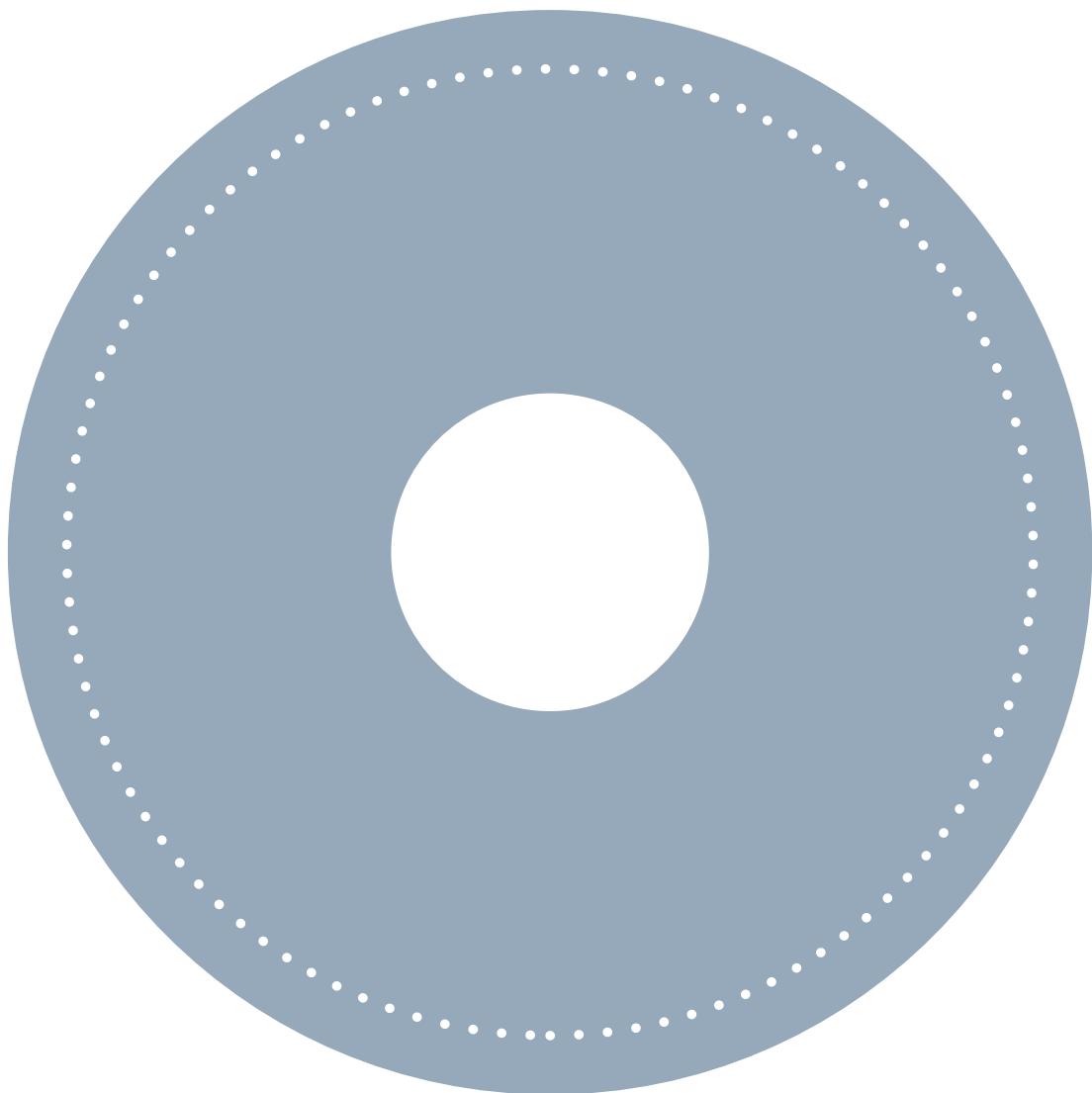
Bon apprentissage !



```

1 <?
2
3 include("application.php");
4 application_start();
5
6 switch ($_GET["can"]){
7     case "log_user":
8         $login = $_POST["login"];
9         if (!$_APP["logins"]){
10             $_APP["logins"] = array();
11         }
12         if (!in_array($login, $_APP["logins"])){
13             array_push($_APP["logins"], $login);
14             application_end();
15             propage_message("Nouvelle connexion : $login");
16         }
17         else{
18             echo "[0, 'Ce login est déjà pris']";
19         }
20         break;
21     }
22
23 function propage_message($message){
24     $fichier = "server-message.txt";
25     $fid = "server-message-id.txt";
26
27     $fp = fopen($fid, 'r') or die("can't open file");
28
29     $id = $fp
30
31     $fp = fopen($fid, 'w') or die("can't open file");
32     $fp
33
34     $fp
35
36     $fp
37
38     $fp
39
40     $fp
41
42     $fp
43
44     $fp
45
46     $fp
47
48     $fp
49
50     $fp
51
52     $fp
53
54     $fp
55
56     $fp
57
58     $fp
59
60     $fp
61
62     $fp
63
64     $fp
65
66     $fp
67
68     $fp
69
70     $fp
71
72     $fp
73
74     $fp
75
76     $fp
77
78     $fp
79
80     $fp
81
82     $fp
83
84     $fp
85
86     $fp
87
88     $fp
89
90     $fp
91
92     $fp
93
94     $fp
95
96     $fp
97
98     $fp
99
100    $fp
101
102    $fp
103
104    $fp
105
106    $fp
107
108    $fp
109
110    $fp
111
112    $fp
113
114    $fp
115
116    $fp
117
118    $fp
119
120    $fp
121
122    $fp
123
124    $fp
125
126    $fp
127
128    $fp
129
130    $fp
131
132    $fp
133
134    $fp
135
136    $fp
137
138    $fp
139
140    $fp
141
142    $fp
143
144    $fp
145
146    $fp
147
148    $fp
149
150    $fp
151
152    $fp
153
154    $fp
155
156    $fp
157
158    $fp
159
160    $fp
161
162    $fp
163
164    $fp
165
166    $fp
167
168    $fp
169
170    $fp
171
172    $fp
173
174    $fp
175
176    $fp
177
178    $fp
179
180    $fp
181
182    $fp
183
184    $fp
185
186    $fp
187
188    $fp
189
190    $fp
191
192    $fp
193
194    $fp
195
196    $fp
197
198    $fp
199
200    $fp
201
202    $fp
203
204    $fp
205
206    $fp
207
208    $fp
209
210    $fp
211
212    $fp
213
214    $fp
215
216    $fp
217
218    $fp
219
220    $fp
221
222    $fp
223
224    $fp
225
226    $fp
227
228    $fp
229
230    $fp
231
232    $fp
233
234    $fp
235
236    $fp
237
238    $fp
239
240    $fp
241
242    $fp
243
244    $fp
245
246    $fp
247
248    $fp
249
250    $fp
251
252    $fp
253
254    $fp
255
256    $fp
257
258    $fp
259
260    $fp
261
262    $fp
263
264    $fp
265
266    $fp
267
268    $fp
269
270    $fp
271
272    $fp
273
274    $fp
275
276    $fp
277
278    $fp
279
280    $fp
281
282    $fp
283
284    $fp
285
286    $fp
287
288    $fp
289
290    $fp
291
292    $fp
293
294    $fp
295
296    $fp
297
298    $fp
299
299    $fp
300
300    $fp
301
301    $fp
302
302    $fp
303
303    $fp
304
304    $fp
305
305    $fp
306
306    $fp
307
307    $fp
308
308    $fp
309
309    $fp
310
310    $fp
311
311    $fp
312
312    $fp
313
313    $fp
314
314    $fp
315
315    $fp
316
316    $fp
317
317    $fp
318
318    $fp
319
319    $fp
320
320    $fp
321
321    $fp
322
322    $fp
323
323    $fp
324
324    $fp
325
325    $fp
326
326    $fp
327
327    $fp
328
328    $fp
329
329    $fp
330
330    $fp
331
331    $fp
332
332    $fp
333
333    $fp
334
334    $fp
335
335    $fp
336
336    $fp
337
337    $fp
338
338    $fp
339
339    $fp
340
340    $fp
341
341    $fp
342
342    $fp
343
343    $fp
344
344    $fp
345
345    $fp
346
346    $fp
347
347    $fp
348
348    $fp
349
349    $fp
350
350    $fp
351
351    $fp
352
352    $fp
353
353    $fp
354
354    $fp
355
355    $fp
356
356    $fp
357
357    $fp
358
358    $fp
359
359    $fp
360
360    $fp
361
361    $fp
362
362    $fp
363
363    $fp
364
364    $fp
365
365    $fp
366
366    $fp
367
367    $fp
368
368    $fp
369
369    $fp
370
370    $fp
371
371    $fp
372
372    $fp
373
373    $fp
374
374    $fp
375
375    $fp
376
376    $fp
377
377    $fp
378
378    $fp
379
379    $fp
380
380    $fp
381
381    $fp
382
382    $fp
383
383    $fp
384
384    $fp
385
385    $fp
386
386    $fp
387
387    $fp
388
388    $fp
389
389    $fp
390
390    $fp
391
391    $fp
392
392    $fp
393
393    $fp
394
394    $fp
395
395    $fp
396
396    $fp
397
397    $fp
398
398    $fp
399
399    $fp
400
400    $fp
401
401    $fp
402
402    $fp
403
403    $fp
404
404    $fp
405
405    $fp
406
406    $fp
407
407    $fp
408
408    $fp
409
409    $fp
410
410    $fp
411
411    $fp
412
412    $fp
413
413    $fp
414
414    $fp
415
415    $fp
416
416    $fp
417
417    $fp
418
418    $fp
419
419    $fp
420
420    $fp
421
421    $fp
422
422    $fp
423
423    $fp
424
424    $fp
425
425    $fp
426
426    $fp
427
427    $fp
428
428    $fp
429
429    $fp
430
430    $fp
431
431    $fp
432
432    $fp
433
433    $fp
434
434    $fp
435
435    $fp
436
436    $fp
437
437    $fp
438
438    $fp
439
439    $fp
440
440    $fp
441
441    $fp
442
442    $fp
443
443    $fp
444
444    $fp
445
445    $fp
446
446    $fp
447
447    $fp
448
448    $fp
449
449    $fp
450
450    $fp
451
451    $fp
452
452    $fp
453
453    $fp
454
454    $fp
455
455    $fp
456
456    $fp
457
457    $fp
458
458    $fp
459
459    $fp
460
460    $fp
461
461    $fp
462
462    $fp
463
463    $fp
464
464    $fp
465
465    $fp
466
466    $fp
467
467    $fp
468
468    $fp
469
469    $fp
470
470    $fp
471
471    $fp
472
472    $fp
473
473    $fp
474
474    $fp
475
475    $fp
476
476    $fp
477
477    $fp
478
478    $fp
479
479    $fp
480
480    $fp
481
481    $fp
482
482    $fp
483
483    $fp
484
484    $fp
485
485    $fp
486
486    $fp
487
487    $fp
488
488    $fp
489
489    $fp
490
490    $fp
491
491    $fp
492
492    $fp
493
493    $fp
494
494    $fp
495
495    $fp
496
496    $fp
497
497    $fp
498
498    $fp
499
499    $fp
500
500    $fp
501
501    $fp
502
502    $fp
503
503    $fp
504
504    $fp
505
505    $fp
506
506    $fp
507
507    $fp
508
508    $fp
509
509    $fp
510
510    $fp
511
511    $fp
512
512    $fp
513
513    $fp
514
514    $fp
515
515    $fp
516
516    $fp
517
517    $fp
518
518    $fp
519
519    $fp
520
520    $fp
521
521    $fp
522
522    $fp
523
523    $fp
524
524    $fp
525
525    $fp
526
526    $fp
527
527    $fp
528
528    $fp
529
529    $fp
530
530    $fp
531
531    $fp
532
532    $fp
533
533    $fp
534
534    $fp
535
535    $fp
536
536    $fp
537
537    $fp
538
538    $fp
539
539    $fp
540
540    $fp
541
541    $fp
542
542    $fp
543
543    $fp
544
544    $fp
545
545    $fp
546
546    $fp
547
547    $fp
548
548    $fp
549
549    $fp
550
550    $fp
551
551    $fp
552
552    $fp
553
553    $fp
554
554    $fp
555
555    $fp
556
556    $fp
557
557    $fp
558
558    $fp
559
559    $fp
560
560    $fp
561
561    $fp
562
562    $fp
563
563    $fp
564
564    $fp
565
565    $fp
566
566    $fp
567
567    $fp
568
568    $fp
569
569    $fp
570
570    $fp
571
571    $fp
572
572    $fp
573
573    $fp
574
574    $fp
575
575    $fp
576
576    $fp
577
577    $fp
578
578    $fp
579
579    $fp
580
580    $fp
581
581    $fp
582
582    $fp
583
583    $fp
584
584    $fp
585
585    $fp
586
586    $fp
587
587    $fp
588
588    $fp
589
589    $fp
590
590    $fp
591
591    $fp
592
592    $fp
593
593    $fp
594
594    $fp
595
595    $fp
596
596    $fp
597
597    $fp
598
598    $fp
599
599    $fp
600
600    $fp
601
601    $fp
602
602    $fp
603
603    $fp
604
604    $fp
605
605    $fp
606
606    $fp
607
607    $fp
608
608    $fp
609
609    $fp
610
610    $fp
611
611    $fp
612
612    $fp
613
613    $fp
614
614    $fp
615
615    $fp
616
616    $fp
617
617    $fp
618
618    $fp
619
619    $fp
620
620    $fp
621
621    $fp
622
622    $fp
623
623    $fp
624
624    $fp
625
625    $fp
626
626    $fp
627
627    $fp
628
628    $fp
629
629    $fp
630
630    $fp
631
631    $fp
632
632    $fp
633
633    $fp
634
634    $fp
635
635    $fp
636
636    $fp
637
637    $fp
638
638    $fp
639
639    $fp
640
640    $fp
641
641    $fp
642
642    $fp
643
643    $fp
644
644    $fp
645
645    $fp
646
646    $fp
647
647    $fp
648
648    $fp
649
649    $fp
650
650    $fp
651
651    $fp
652
652    $fp
653
653    $fp
654
654    $fp
655
655    $fp
656
656    $fp
657
657    $fp
658
658    $fp
659
659    $fp
660
660    $fp
661
661    $fp
662
662    $fp
663
663    $fp
664
664    $fp
665
665    $fp
666
666    $fp
667
667    $fp
668
668    $fp
669
669    $fp
670
670    $fp
671
671    $fp
672
672    $fp
673
673    $fp
674
674    $fp
675
675    $fp
676
676    $fp
677
677    $fp
678
678    $fp
679
679    $fp
680
680    $fp
681
681    $fp
682
682    $fp
683
683    $fp
684
684    $fp
685
685    $fp
686
686    $fp
687
687    $fp
688
688    $fp
689
689    $fp
690
690    $fp
691
691    $fp
692
692    $fp
693
693    $fp
694
694    $fp
695
695    $fp
696
696    $fp
697
697    $fp
698
698    $fp
699
699    $fp
700
700    $fp
701
701    $fp
702
702    $fp
703
703    $fp
704
704    $fp
705
705    $fp
706
706    $fp
707
707    $fp
708
708    $fp
709
709    $fp
710
710    $fp
711
711    $fp
712
712    $fp
713
713    $fp
714
714    $fp
715
715    $fp
716
716    $fp
717
717    $fp
718
718    $fp
719
719    $fp
720
720    $fp
721
721    $fp
722
722    $fp
723
723    $fp
724
724    $fp
725
725    $fp
726
726    $fp
727
727    $fp
728
728    $fp
729
729    $fp
730
730    $fp
731
731    $fp
732
732    $fp
733
733    $fp
734
734    $fp
735
735    $fp
736
736    $fp
737
737    $fp
738
738    $fp
739
739    $fp
740
740    $fp
741
741    $fp
742
742    $fp
743
743    $fp
744
744    $fp
745
745    $fp
746
746    $fp
747
747    $fp
748
748    $fp
749
749    $fp
750
750    $fp
751
751    $fp
752
752    $fp
753
753    $fp
754
754    $fp
755
755    $fp
756
756    $fp
757
757    $fp
758
758    $fp
759
759    $fp
760
760    $fp
761
761    $fp
762
762    $fp
763
763    $fp
764
764    $fp
765
765    $fp
766
766    $fp
767
767    $fp
768
768    $fp
769
769    $fp
770
770    $fp
771
771    $fp
772
772    $fp
773
773    $fp
774
774    $fp
775
775    $fp
776
776    $fp
777
777    $fp
778
778    $fp
779
779    $fp
780
780    $fp
781
781    $fp
782
782    $fp
783
783    $fp
784
784    $fp
785
785    $fp
786
786    $fp
787
787    $fp
788
788    $fp
789
789    $fp
790
790    $fp
791
791    $fp
792
792    $fp
793
793    $fp
794
794    $fp
795
795    $fp
796
796    $fp
797
797    $fp
798
798    $fp
799
799    $fp
800
800    $fp
801
801    $fp
802
802    $fp
803
803    $fp
804
804    $fp
805
805    $fp
806
806    $fp
807
807    $fp
808
808    $fp
809
809    $fp
810
810    $fp
811
811    $fp
812
812    $fp
813
813    $fp
814
814    $fp
815
815    $fp
816
816    $fp
817
817    $fp
818
818    $fp
819
819    $fp
820
820    $fp
821
821    $fp
822
822    $fp
823
823    $fp
824
824    $fp
825
825    $fp
826
826    $fp
827
827    $fp
828
828    $fp
829
829    $fp
830
830    $fp
831
831    $fp
832
832    $fp
833
833    $fp
834
834    $fp
835
835    $fp
836
836    $fp
837
837    $fp
838
838    $fp
839
839    $fp
840
840    $fp
841
841    $fp
842
842    $fp
843
843    $fp
844
844    $fp
845
845    $fp
846
846    $fp
847
847    $fp
848
848    $fp
849
849    $fp
850
850    $fp
851
851    $fp
852
852    $fp
853
853    $fp
854
854    $fp
855
855    $fp
856
856    $fp
857
857    $fp
858
858    $fp
859
859    $fp
860
860    $fp
861
861    $fp
862
862    $fp
863
863    $fp
864
864    $fp
865
865    $fp
866
866    $fp
867
867    $fp
868
868    $fp
869
869    $fp
870
870    $fp
871
871    $fp
872
872    $fp
873
873    $fp
874
874    $fp
875
875    $fp
876
876    $fp
877
877    $fp
878
878    $fp
879
879    $fp
880
880    $fp
881
881    $fp
882
882    $fp
883
883    $fp
884
884    $fp
885
885    $fp
886
886    $fp
887
887    $fp
888
888    $fp
889
889    $fp
890
890    $fp
891
891    $fp
892
892    $fp
893
893    $fp
894
894    $fp
895
895    $fp
896
896    $fp
897
897    $fp
898
898    $fp
899
899    $fp
900
900    $fp
901
901    $fp
902
902    $fp
903
903    $fp
904
904    $fp
905
905    $fp
906
906    $fp
907
907    $fp
908
908    $fp
909
909    $fp
910
910    $fp
911
911    $fp
912
912    $fp
913
913    $fp
914
914    $fp
915
915    $fp
916
916    $fp
917
917    $fp
918
918    $fp
919
919    $fp
920
920    $fp
921
921    $fp
922
922    $fp
923
923    $fp
924
924    $fp
925
925    $fp
926
926    $fp
927
927    $fp
928
928    $fp
929
929    $fp
930
930    $fp
931
931    $fp
932
932    $fp
933
933    $fp
934
934    $fp
935
935    $fp
936
936    $fp
937
937    $fp
938
938    $fp
939
939    $fp
940
940    $fp
941
941    $fp
942
942    $fp
943
943    $fp
944
944    $fp
945
945    $fp
946
946    $fp
947
947    $fp
948
948    $fp
949
949    $fp
950
950    $fp
951
951    $fp
952
952    $fp
953
953    $fp
954
9
```

**S'il vous est impossible de lire un CD,
alors qu'il n'a pas de défaut apparent,
essayez de le lire dans un autre lecteur.**



**Pour tout problème concernant les CDs,
écrivez-nous à l'adresse :
cd@phpsolmag.org**

Yahoo! SearchMonkey

Les résultats de recherche sont proposés par les moteurs de recherche. SearchMonkey inverse la tendance en permettant la personnalisation de ses résultats. Ainsi vous pourrez enrichir la fiche de vos sites ou blogs préférés sur la page de résultats du moteur de recherche Yahoo.

Cet article explique comment :

- Exploiter les contenus du web sémantique.
- Améliorer la présentation de recherche.

Ce qu'il faut savoir :

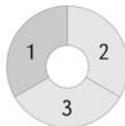
- Connaître la syntaxe PHP.
- Avoir la notion de DOM.

les choix de nombreuses entreprises, même aux côtés de Java où d'autres langages.

Configuration

Pour utiliser SearchMonkey, quelques configurations sont nécessaires à exploiter.

Niveau de difficulté



En 2008, un nouvel outil réalisé par Yahoo a été mis à la disposition de tous et surtout pour les développeurs et dévelopeuses du monde entier. Cet outil s'appelle SearchMonkey, représenté par une tête de singe.

Ces caractéristiques premières sont :

- La rapidité des recherches,
- Plus attrayant,
- Plus pertinent,
- Une meilleure dynamique pour le trafic sur internet.

L'utilisation de cet outil est possible avec le langage PHP, c'est pourquoi il était indispensable que votre magazine préféré consacre quelques colonnes sur ce merveilleux outil.

Pourquoi SearchMonkey ?

Avec la multitude de microformats qui apparaissent, la complexité d'obtenir des informations précises et pointues dans les manipulations de la fonction recherche, il devenait utile de réfléchir à quelque chose pour passer à l'étape suivante.

Actuellement, les résultats obtenus par les différents moteurs de recherche, fournissent des informations utiles dans les résultats de recherche. Bien sûr, si vous saisissez quelques

mots, il est souvent rare de tomber sur le reflet d'information que vous auriez aimé obtenir dans une page. Et là Yahoo peut avoir trouvé la solution idéale, autoriser l'utilisateur à prendre la décision pour obtenir le bon résultat, et là, vous allez vous trouver devant une plate-forme web et non plus juste devant un simple moteur de recherche.

Cette plate-forme sera créée par vous-même ou encore par d'autres personnes qui pourront partager leurs plate-formes, et vous pourrez créer aussi des mini applications de présentation en PHP. Rasmus Lerdorf, pour ne pas le citer, a réalisé un exemple avec SearchMonkey pour montrer la différence avec 2 résultats obtenus. La Figure 2 affiche le résultat obtenu avant d'utiliser SearchMonkey, et la Figure 3 montre le résultat après.

Comme vous pouvez le voir, une nette différence apparaît au niveau du résultat. Yahoo profite de la technologie du web sémantique pour produire un système de recherche le plus avancé actuellement et surtout disponible.

Si on se réfère à la définition du site Wikipédia concernant le web sémantique, la définition donnée est : *Le Web sémantique désigne un ensemble de technologies visant à rendre le contenu des ressources du World Wide Web accessible et utilisable par les programmes et agents logiciels, grâce à un système de métadonnées formelles, utilisant notamment la famille de langages développés par le W3C.*

Compte tenu de l'élan général vers le Web, l'utilisation dans SearchMonkey du langage PHP peut très bien devenir un élément clé dans

Ouverture d'un compte

Pour SearchMonkey, si vous ne possédez pas déjà un compte Yahoo, l'ouverture de ce dernier sera nécessaire, vous permettant ainsi d'utiliser les mêmes paramètres d'identification. Pour l'ouverture d'un compte SearchMonkey, vous devez vous rendre à l'adresse suivante : <http://developer.yahoo.com/searchmonkey>.

PHP

Lors de la mise en route de l'outil, votre serveur devra posséder la version 5.2 ou suivante de PHP. Ces deux points sont obligatoires pour pouvoir continuer la lecture de l'article. Et un point supplémentaire sera étudié un peu plus loin pour les personnes qui souhaitent en faire plus avec SearchMonkey.



Figure 1. Logo SearchMonkey

Approche

En plus de pouvoir afficher des résultats de recherche vers des flux XML ou intégrer dans les pages elles-mêmes avec les microformats (eRDF ou RDFa), SearchMonkey possède deux composants principaux :

- Présentation des applications sous la forme de morceau de code PHP, permettant de générer les résultats de recherches personnalisés.
- Les données sont tirées des données structurées venant du Web.

Concernant le deuxième point, de nombreux services par défaut sont proposés pour obtenir une approche personnalisée. Avec bien sûr, la possibilité de concevoir le votre.

Cas pratique

Après avoir expliqué les différentes possibilités qui vous sont offertes par SearchMonkey, passons maintenant à l'étape de mise en route en s'appuyant sur un cas pratique. Lorsque vous vous connectez sur votre compte, plusieurs étapes sont nécessaires pour réaliser l'application. Les étapes étudiées sont :

- Informations basic,
- Les URLs des pages,
- Service de données,
- Apparence de la page,
- Demande de confirmation.

Mais avant de commencer, vous pouvez choisir :

- *Présentation d'une application,*
- *Custom Data Services,*
- *Data Feeds.*

Pour chacune des propositions, vous pouvez créer, utiliser ou importer un de ces choix. Cependant, l'exemple va porter sur la présentation d'une application comme ceci, nous utiliserons le simple d'application et il ne vous restera plus qu'à créer votre application après la lecture de cet article.

Étape 1 : informations basic

L'écran de présentation d'informations basic représenté par la Figure 4, montre ce que vous devez remplir. Vous visualiserez chaque partie grâce à un chiffre qui sera utilisé comme repère.

- Saisir le nom : Ce champ est très important, car c'est le nom qui apparaîtra dans la galerie de recherche.
- Choisir le type *infobar* : Nous avons choisi de créer une application, et donc vous devez sélectionner *Enhanced résultats*. L'autre option est utilisée pour *Custom Data Services*.

Listing 1. Présentation de la fonction

```
<?php
    public static function getOutput() {
        $ret = array();
        define("SMDEFAULT", "");

        /* If you leave these blank, the default title and summary will be shown */
        $ret['title'] = SMDEFAULT;
        $ret['summary'] = SMDEFAULT;
        /* Now you fill in the rest. Use Data::get and Data::xpath to get data */
        // Image
        $ret['image']['src'] = SMDEFAULT;
        $ret['image']['alt'] = SMDEFAULT;
        $ret['image']['title'] = SMDEFAULT;
        $ret['image']['allowResize'] = true;
        // Deep links - up to 4
        $ret['links'][0]['text'] = SMDEFAULT;
        $ret['links'][0]['href'] = SMDEFAULT;
        $ret['links'][1]['text'] = SMDEFAULT;
        $ret['links'][1]['href'] = SMDEFAULT;
        $ret['links'][2]['text'] = SMDEFAULT;
        $ret['links'][2]['href'] = SMDEFAULT;
        // Key Value pairs - up to 4
        $ret['dict'][0]['key'] = SMDEFAULT;
        $ret['dict'][0]['value'] = SMDEFAULT;
        $ret['dict'][1]['key'] = SMDEFAULT;
        $ret['dict'][1]['value'] = SMDEFAULT;
        $ret['dict'][2]['key'] = SMDEFAULT;
        $ret['dict'][2]['value'] = SMDEFAULT;
        $ret['dict'][3]['key'] = SMDEFAULT;
        $ret['dict'][3]['value'] = SMDEFAULT;

        You can put a subset of HTML in here
        See the docs for more details */
        $ret['infobar']['summary'] = SMDEFAULT;
        $ret['infobar']['blob'] = SMDEFAULT;
        return $ret;
    }
?>
```

Listing 2. Utilisation par paire

```
<?php
    $ret['dict'][0]['key'] = "Category";
    $ret['dict'][0]['value'] = Data::get('yahoo:index/rel:Listing/dc:title')." (" . Data:
        :get('yahoo:index/rel:Listing/dc:type')." );
    $ret['dict'][1]['key'] = "Language";
    $ret['dict'][1]['value'] = Data::get('yahoo:index/dc:language');
    $ret['dict'][2]['key'] = "MIME Type";
    $ret['dict'][2]['value'] = Data::get('yahoo:index/dc:format');
?>
```

Higuma Japanese Restaurant - Redwood City, CA 94063

Higuma Japanese Restaurant Redwood City, CA - Ruth B. says "Some of the ... Higuma is what I would bill as a Japanese style neighborhood restaurant sushi ...
www.yelp.com/biz/higuma-japanese-restaurant-redwood-city - 152k - [Cached](#)

Figure 2. Résultat avant

Figure 3. Résultat après

- Ajouter un descriptif : Le descriptif saisi, sera visible dans la galerie de recherche comme le champ nom. Vous devez aussi enregistrer la catégorie de rangement.
- Sélectionner un icône : L'icône permet de repérer plus facilement le type d'application. Le fichier utilisé et conseillé se nomme *favicon.ico*.
- Condition d'utilisation : Il est nécessaire de cocher la case pour accepter son utilisation et surtout pouvoir continuer la procédure.
- Clic sur le bouton *NEXT* : Pour finir cliquez sur le bouton *Next* pour terminer la saisie.

Étape 2 : Les URLs des pages

Cet écran permet de faciliter les recherches pour l'affichage des résultats en utilisant un filtre :

- Type URL : Ce champ permet de trouver facilement les informations, si vous possédez un blog ou un site... Il est important de spécifier le nom de domaine par une étoile devant ou après le nom de domaine. Exemple **.NomDomaine.com/**
- Les images : Ce champ n'est à compléter que si vos images sont stockées sur un site web distant comme *flick'r*.
- Les liens externes : Si certains de vos liens sont gérés sur un autre site, vous pouvez le signaler en les rajoutant.
- Ajout des tests URL : Lorsque vous avez rempli le type URL, il est nécessaire d'effectuer un test pour être certain de ne pas avoir effectuer d'erreurs. Un bouton *Autofind URL's* permet d'obtenir 10 URLs et de vérifier si les paramètres sont corrects. Il sera possible par rapport aux 10 URLs proposées de choisir

Figure 4. Informations basic

la présentation de son choix (comme le montre la Figure 6).

- Clic sur le bouton *NEXT* : Pour finir cliquez sur le bouton *Next* pour terminer la saisie.

Étape 3 : Service de données

Cet écran permet de compléter les liens que vous avez spécifiés précédemment, c'est-à-dire, que les informations que vous avez retenues, peuvent être trouvées aussi dans les flux RSS provenant d'autres sites comme Yahoo.

- Sélection des flux : Il suffit de sélectionner le ou les sites que vous désirez mémoriser par rapport à votre sélection précédente. Et de cliquer sur le lien *Add More Data Services* pour les ajouter dans la présentation.
- Clic sur le bouton *NEXT* : Pour finir cliquez sur le bouton *Next* pour terminer la saisie.

Étape 4 : Apparence de la page

Vous arrivez maintenant dans la partie visuelle. Cette partie va vous permettre de préparer l'affichage des résultats de recherche de SearchMonkey. Vous pourrez la modifier comme vous le désirez puisqu'il existe une série de fonctions pré-définies sur la partie droite de l'image (repère 2 de la Figure 8).

- Zone modifiable : Cette zone vous permet de personnaliser les résultats de recherche de SearchMonkey en réalisant votre propre *template*.
- Fonctions : Ce sont des fonctions disponibles, permettant l'évolution de la présentation et de modifier le *template* comme vous le désirez.
- Clic sur le bouton *NEXT* : Pour finir cliquez sur le bouton *Next* pour terminer la saisie.

L'utilisation et la zone modifiable sera étudiée un peu plus loin dans la partie *Réalisation du template*.

Étape 5 : Demande de confirmation

Vous voici arrivé à la dernière étape de la création de ce *template* représentée par la Figure 9.

- Zone de confirmation : La présentation qui sera affichée dans la galerie est montrée pour que vous puissiez la contrôler. Cette présentation peut être représentée sous le nom de la Figure 6 par exemple.
- Validation : Dans la Figure 9, deux propositions vous sont offertes : Le bouton *TRY* vous permettant de recommencer l'application SearchMonkey. Le bouton *Make*, par l'intermédiaire de la galerie vous permettant de la partager aux autres personnes.

Figure 5. Informations basic

Étape 6 : Validation

Une étape supplémentaire existe pour être certain d'apparaître dans la galerie et que tout le monde puisse en bénéficier, vous recevrez un email des modérateurs de cette galerie SearchMonkey qui validera ou pas votre présentation avec son contenu.

Bien sûr la présentation et son contenu ne sont pas assez pertinents, vous avez la possibilité de modifier votre *template* pour le présenter un autre jour.

Réalisation du template

L'étape 4 *Apparence de la page* vous permet de réaliser et de personnaliser la présentation d'affichage des résultats, qui peuvent apparaître comme sur la Figure 6.

Les positions

Le code fourni est un langage PHP, et ces quelques lignes de programmations dans le Listing 1, montrent la réalisation d'un tableau associatif. Avant tout, vous devez préparer le *template* pour les champs suivants :

- `$Ret ['title']` : affiche le titre et sera affiché en haut.
- `$Ret ['summary']` : affiche la description de la recherche.

- `$Ret ['image']` : est une extension de propriété SearchMonkey avec la possibilité de définir la position de son choix comme à côté du résultat de recherche.
- `$Ret ['links']` : grâce à 4 lignes positionnables à votre gré, vous pouvez placer des liens de texte.
- `$Ret ['dict']` : c'est un tableau de paires valeur-clé qui permet d'afficher des informations structurées.
- `$Ret ['infobar']` : est la barre d'information de SearchMonkey pour les applications.

Quand vous avez modifié le champ `SMDEFAULT` par la fonction pré définie, vous pouvez cliquer sur le bouton *Save & refresh*, pour voir le résultat changé.

Pour rendre l'application encore plus facile, Yahoo utilise la technique du *Plug and Play*,



Figure 6. Modèle de présentation

ainsi, vous n'aurez plus qu'à placer la fonction que vous souhaitez et la positionner à l'endroit que vous désirez. Comme ceci, la ligne retenue, le mot `SMDEFAULT` disparaîtra pour laisser la place à cette nouvelle fonction.

Particularité

La fonction `$Ret ['dict']` sera utilisée par paire, et plus exactement une ligne clef et une ligne valeur.

Comme le montre le Listing 2, une catégorie, un langage et un format seront présentés lors de l'affichage de recherche. Enfin pour visualiser le résultat, il ne vous reste plus qu'à cliquer le bouton *save & refresh*.

Custom Data Service

Cette partie, va vous montrer comment vous pouvez utiliser des extensions externes pour insérer la configuration de *SearchMonkey Custom Data Service*. Le format qu'utilise SearchMonkey s'appelle *DataRSS*, incorporé dans un flux *Atom*. Le format *DataRSS* peut générer automatiquement des données sémantiques qui sont intégrées ou formatées en *Microformats*.

Pour extraire les *DataRSS*, il est nécessaire d'utiliser la transformation avec *XSLT*, qui vous permet d'extraire structurées sur les sites existants.

Tout ceci concerne les microformats avec une superposition des données structurées sur les sites existants. L'article va vous parler de 2 extensions sans rentrer dans les détails, car le principe reste toujours identique à ce qui a été vu dans les lignes précédentes.

Par effectuer l'opération, il existe plusieurs extensions et nous choisirons les extensions :

- *Xpather*,
- *MashMake*.

Xpather

Xpather est une extension qui génère des *XPaths* pendant la navigation ou bien, il peut être utilisé comme un inspecteur de documents HTML / XML / * ML pour en extraire un contenu. Le *XPather* est une simple extension Firefox qui s'intègre à la fois dans le navigateur et sa *DOM Inspector*. Avec *Xpather*, il sera possible d'utiliser la technique de *Custom Data Services* pour SearchMonkey. Mais avant tout, pour utiliser l'extension *Xpather*, il est préférable de posséder Firefox 3.

L'utilisation de cette extension, impose d'installer :

Figure 7 consists of two screenshots. Part 1 is a screenshot of the Yahoo! Application Platform interface, specifically the 'Data Services' step of a presentation application. It shows a list of available data services, including 'Feed: yahoo:index', 'Feed: com.yahoo.page.uf.adr', 'Feed: com.yahoo.page.uf.hcalendar', 'Feed: com.yahoo.page.uf.hcard', 'Feed: com.yahoo.page.uf.hresume', and 'Feed: sm031-LinkedIn'. Each service has a checkbox and a 'Contains' link. A red number '1' is placed to the right of the interface. Part 2 is a screenshot of a LinkedIn profile for 'Amit Kumar - President, IIT Delhi Alumni Association, North America ...'. The profile includes a photo, a summary, and a list of connections. A red number '2' is placed to the right of the LinkedIn profile.

Figure 7. Service de données

- L'extension Xpather (<https://addons.mozilla.org/fr/firefox/addon/1192>),
- Plugin Dom Inspector (<https://addons.mozilla.org/en-US/firefox/addon/6622>).

L'extension Xpather va vous permettre de découvrir les XPath de n'importe quel élément, de voir immédiatement les noeuds xpath dans un document. Pour utiliser cette extension, vous pouvez vous rendre sur le site officiel de celui-ci pour consulter la documentation (<http://xpath.alephzarro.com>).

Lorsque vous avez trouvé les noeuds sur la liste, vous pouvez manipuler les noeuds que vous aurez sélectionnés. Avec l'expression Xpath, vous pouvez modifier dans la zone de texte et cliquer sur *Eval* pour voir les résultats. Lors de l'insertion dans *MonkeySearch*, il ne vous restera plus qu'à copier le code pour obtenir le résultat désiré.

MashMaker

MashMaker est une barre d'outils pour actuellement Firefox et Internet Explorer. Elle a pour but d'enrichir les informations de la page que vous affichez avec différentes sources combinées. Cette application va être utile pour faciliter les différentes recherches et utilisations

Tout d'abord, vous devez passer sur le site internet de Intel qui a réalisé l'application sur le lien suivant : <http://mashmaker.intel.com/web/download.html>. Vous recevrez sur votre messagerie vos paramètres d'identification (login et mot de passe) en choisissant le type de votre navigateur. Actuellement, deux navigateurs sont proposés :

- Firefox,
- Internet Explorer.

Pour effectuer la dernière étape, vous devez installer manuellement dans le navigateur Firefox (Outils -> options -> MashMaker) en enregistrant vos identifiants que vous avez reçus par email. Son utilisation est différente de xpather, même si vous pourrez trouver les noeuds xpath, les manipulations seront un peu plus limitées.

Lors des repérages de ces noeuds, vous pourrez effectuer les mêmes opérations que précédemment. Pour terminer dans la partie *Custom Data Services*, il est très difficile de vous conseiller d'utiliser une extension plutôt qu'une autre, mais les deux peuvent être complémentaires.

Galerie

Une galerie regroupe l'ensemble des SearchMonkey que les développeurs ont réalisé permettant ainsi à chacun de s'en servir. Elle est accessible par l'URL suivant : <http://gallery.search.yahoo.com/>. Il faudra vous connecter sur votre compte pour en bénéficier de l'ensemble des possibilités. Dans cette galerie se trouve les liens validés par Yahoo mais il est possible de vérifier

Figure 8 consists of three screenshots of the Application Dashboard. Part 1 is a screenshot of the 'Presentation PHP' step, showing a code editor with PHP code for mapping data services. A red number '1' is placed to the right of the code editor. Part 2 is a screenshot of the 'Data Inputs' list, showing a list of available data inputs such as 'yahoo:index', 'com.yahoo.ex...ction.social', and 'sm031-LinkedIn'. A red number '2' is placed to the right of the list. Part 3 is a screenshot of the 'Appearance' step of the application dashboard, showing a preview pane and a code editor for modifying appearance with PHP. A red number '3' is placed to the right of the dashboard.

Figure 8. Appearance

The screenshot shows the Yahoo! SearchMonkey Application Platform. At the top, there are links for 'Yahoo!', 'My Yahoo!', 'Mail', and 'More'. The main header says 'YAHOO! APPLICATION PLATFORM' and 'SearchMonkey'. The top right shows 'Welcome, hellosd1 | Sign out | Help'. Below the header, it says 'Make Y! Your Home Page' and 'Search' with a 'Web Search' button. The main content area is titled 'Application Dashboard / Liked in exemple'. Below this, there are tabs: 'Basic Info', 'URLs', 'Data Services', 'Appearance', and 'Confirmation'. The 'Confirmation' tab is selected, indicated by a yellow bar. The content within the 'Confirmation' tab shows a preview of the application's output, which is a simple text box containing 'Now you have a SearchMonkey application.' A large red number '1' is overlaid on the top right of this preview area. Below the preview, there is a note: 'Now you have a SearchMonkey application. Check it out above. We are using the first URL in your list of [Test URLs](#) to generate the preview.' Underneath that, it says 'Here's what to do next:'. It includes a note: 'Try this Presentation Application in Yahoo! Search. (this link only works for you). Make the latest version of your application sharable so that others may use it.' and a small note: 'Don't forget to kick back and have a banana.' A small banana icon is shown next to the note. At the bottom of the confirmation tab, there are 'Back' and 'Home' buttons. A large red number '2' is overlaid on the bottom right of the 'Confirmation' tab area.

Figure 9. Demande de confirmation

votre mise en forme avec la sélection *show experimental enhancements* sur d'autres moteurs de recherche.

Bien sûr, comme tout le monde, si vous désirez apparaître dans cette galerie, cela est possible, mais souvent il vous faudra plusieurs tentatives pour apparaître dans cette galerie car la validation effectuée est manuelle comme ceci lorsque c'est validé vous êtes sûr d'avoir bien respecté les normes.

Futur

Comme dans de nombreuses applications, Yahoo! SearchMonkey va progresser pour vous faciliter son utilisation. À l'heure où cet article est écrit une nouvelle version verra le jour. Cette nouvelle version de Yahoo! SearchMonkey proposera :

- Une amélioration au niveau du partage avec des modules par Email,

- Une évolution de la galerie avec l'affichage des pages de résultats de recherche sous la forme de boutons à ajouter sur votre site internet,
- Le code source verra aussi une certaine amélioration avec un partage et une réutilisation du code d'un module existant pour en créer un autre.

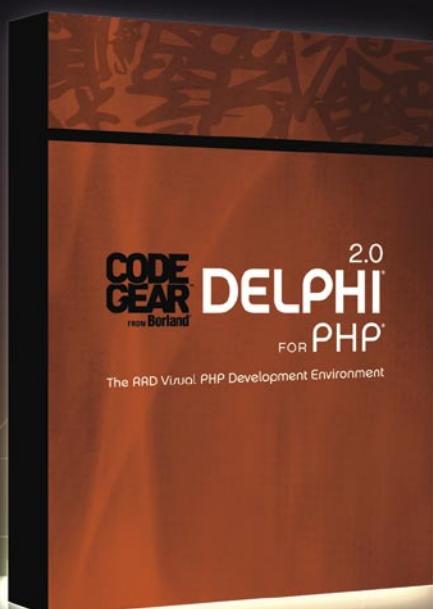
Conclusion

SearchMonkey, comme vous avez pu le voir, est très souple en ce qui concerne les données : une fois que vous créez une source de données, elle est réutilisable à travers la présentation des demandes. Enfin, Yahoo a adopté une approche très intéressante de la gestion de la plate-forme : elle vous permet d'exécuter votre code PHP sur les serveurs Yahoo. Naturellement, il y a toute sorte de restriction de sécurité en place, mais de nombreuses fonctions sont disponibles sur la documentation officielle de SearchMonkey : <http://developer.yahoo.com/searchmonkey/smguide/>.

CHRISTOPHE VILLENEUVE

Auteur du livre PHP & MySQL-MySQLi-PDO, Construisez votre application, livre français aux Éditions ENI. Travaille depuis de nombreuses années dans le développement d'outils de gestion en tant que prestataire ; il est aussi contributeur de nombreux sites touchant PHP dont Nexen, PHP Team, PHPTV... Contactez l'auteur : <http://www.hello-design.fr>.

DELPHI FOR PHP 2.0



l'environnement de
développement RAD pour PHP

WWW.CODEGEARFRANCE.COM

codegear@open-way.com
01 49 97 51 28

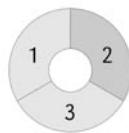
PHP avec Quercus

Contrairement à la plupart des techniques d'interopérabilité entre Java et PHP, Quercus nous propose de marier les deux langages au sein d'un runtime commun.

Cet article explique :

- Comment développer des pages PHP avec Quercus.
- Comment déployer des applications mixtes Java / PHP dans Tomcat.

Niveau de difficulté



Dans le numéro 6/2008 (30), j'avais exposé mon point de vue sur la cohabitation entre les langages Java et PHP dans une architecture commune. Nous avions vu comment tirer le meilleur de chaque langage, en développant sur une architecture en couches, où PHP serait utilisé pour la couche de présentation, c'est-à-dire les pages Web, alors que Java serait utilisé pour les couches métier et d'accès aux données, afin d'exploiter les capacités transactionnelles et de gestion des connexions du langage (Figure 1).

Nous avions cité quelques techniques d'interopérabilité : *WebServices* avec SOAP et REST, pont Java-PHP, ainsi que la *Java scripting API*. Nous avions ensuite détaillé le fonctionnement du pont Java-PHP qui permet de faire communiquer une application PHP, exécutée dans un processus Apache traditionnel, avec des composants Java, exécutés dans leur machine virtuelle.

Nous avions aussi cité les solutions *in-process*, sans pour autant détailler leur fonctionnement. C'est sur ce type de solution que nous allons nous concentrer, avec Quercus.

Présentation de Quercus

Quercus est un logiciel Open Source permettant de faire fonctionner une application PHP dans une machine virtuelle Java.

Ce qu'il faut savoir :

- Développer en PHP.
- Comprendre les bases de fonctionnement de Java.

Société Cauchoo

Quercus a été développé par la société *Cauchoo*, connue des habitués de Java pour leur serveur d'applications *Resin* et auteur des protocoles de communication *Burlap* et *Hessian*, basés sur HTTP, avec du contenu XML ou binaire. Tous ces produits sont distribués sous licence Open Source, avec des versions dites professionnelles, sous licence commerciale.

Resin est un serveur d'applications Java, destiné au déploiement d'applications Web. De ce fait, il est en concurrence directe avec *Apache Tomcat*, mais aussi avec tous les serveurs d'applications *JavaEE*, comme *IBM Websphere*, *Oracle Weblogic* ou *RedHat JBoss*. *Resin* est le premier serveur d'applications Java capable de supporter le déploiement d'applications PHP, en mode *in-process*, grâce à l'intégration de *Cauchoo*. *Glassfish*, le serveur d'applications *JavaEE Open Source* de *Sun*, lui emboîte le pas puisqu'il embarque aussi Quercus pour la version 3.

IBM a lancé un projet concurrent, nommé *Project Zero*, probablement pour intégrer le

support de PHP dans une future version de *Websphere*.

Mode de fonctionnement

Dans toutes les solutions concurrentes, les pages PHP et les classes Java sont exécutées dans des processus séparés et communiquent via les couches du réseau. À l'opposé, avec Quercus, le PHP est exécuté au sein du même processus que Java, ce qui économise toutes les communications inter-processus (Figure 2).

Dans la version Open Source, Quercus interprète le code PHP, alors que la version professionnelle est capable de le compiler en *byte-code Java*, ce qui améliore encore les performances. Quercus est donc une implémentation du moteur PHP, indépendante de celle fournie par *php.net*.

Pour valider le fonctionnement, *Cauchoo* a testé un certain nombre d'applications PHP classiques, comme *Typo3*, *MediaWiki*, *Drupal* ou *WordPress*. Les résultats semblent probants, puisque sur certains tests, Quercus est quatre fois plus performant que le moteur PHP traditionnel.

Avantages

Outre l'interopérabilité des langages, les avantages de Quercus sont à chercher du côté des performances, ainsi que de certains services fournis par Java, comme la gestion des

Terminologie

- *Java EE = Java Enterprise Edition* constitue l'offre standard de Java pour les serveurs, Web ou transactionnel, implémentée par différents serveurs d'applications (IBM Websphere, Oracle Weblogic, RedHat JBoss,...).
- *Java SE = Java Standard Edition* est le tronc commun aux applications Java, elle contient les bibliothèques standard et la machine virtuelle de Java.
- *JDK = Java Development Kit* constitue le runtime pour l'exécution de *Resin* ou de *Tomcat*.
- *EJB = Enterprise Java Bean* est un type de composants de *JavaEE*.
- *JNDI = Java Naming and Directory Interface* est utilisé dans les serveurs d'applications *JavaEE* pour accéder au registre des composants.

connexions aux bases de données par pools, grâce aux *datasources*.

Premiers pas avec Quercus et Resin

Dans un premier temps, nous allons utiliser la version de Quercus embarquée dans *Resin*, car c'est la forme la plus simple à installer.

Présentation de Resin

Resin est le serveur d'applications *JavaEE* de *Caucho*. Il est fourni par la société *Caucho* en deux versions : la version standard est distribuée en Open Source, sous licence *GPL v2*, et la version professionnelle, qui apporte quelques fonctionnalités supplémentaires, est vendue sous licence commerciale.

Un serveur d'applications *JavaEE* permet de déployer des applications Web développées en Java, ainsi que des composants EJB. Il leurs fournit divers services, comme un registre de composant JNDI, des pools de connexions aux bases de données, appelées *DataSource*, de la messagerie inter-applications, une gestion des transactions, ... (Figure 3)

Installation de Resin

Nous allons donc installer *Resin*, en version standard, qu'il faut au préalable télécharger sur le site de *Caucho*. L'installation de *Resin* est très simple, puisqu'il suffit de décompresser l'archive téléchargée dans le répertoire de notre choix. Sous Windows, pour faciliter l'écriture de scripts, on préférera un répertoire sans espace, ce qui exclut *c:\Program Files*. Pour ma part, j'ai installé *Resin* dans un répertoire *DevPhp*.

Avant de pouvoir démarrer *Resin*, il faut que Java soit installé, en version 5 ou ultérieure.

Démarrage de Resin

Parmi les façons de démarrer *Resin*, ma préférée est la technique universelle façon Java, car elle fonctionne sur tous les systèmes d'exploitation et elle permet de confectionner facilement nos propres scripts (Listing 1). Pour les autres modes de démarrage, en particulier comme service Windows, je renverrai le lecteur à la documentation de *Resin*.

Pour tester si le démarrage s'est bien passé, on appelle la page par défaut depuis un navigateur Web, à l'adresse locale *http://localhost:8080/*. Cette page par défaut est très simple, puisqu'elle n'est constituée que d'un titre et de deux liens hypertextes, vers la documentation intégrée et vers la console d'administration.

Application PHP

Après l'installation de *Resin*, on crée une application en ajoutant un répertoire dans son répertoire *webapps*. Les pages peuvent être ajoutées directement dans ce répertoire. Par exemple, une page *world.php* dans l'application *hello*, pourra être appelée localement depuis le navigateur à l'adresse locale *http://localhost:8080/hello/world.php*.

On a donc installé un moteur PHP, près à accueillir des applications Web, sans *Apache Web Server*, ni le moteur *PHP traditionnel*.

Utilisation de Quercus

Pouvoir déployer des pages PHP dans un serveur d'applications Java ne présente, en tant que tel, que peu d'intérêt. Cet exercice devient vraiment productif si le code PHP et le code Java peuvent interagir facilement, et si l'application PHP peut exploiter les ressources du serveur d'applications.

Configuration de PHP

La configuration de PHP se fait dans un fichier *php.ini* ; ce fichier, optionnel, doit être déclaré dans le fichier *WEB-INF/resin-web.xml* de l'application (Listing 2).

Connexion aux bases de données

En Java, les connexions aux bases de données relationnelles se font via les drivers Java et les datasources. Toute connexion se fait en utilisant une API standard qui s'appelle JDBC, et le code Java est strictement identique pour les différents types de bases de données (Oracle, MySQL, Microsoft SQL Server...). Pour les applications Web, les connexions sont gérées au sein de pools et partagées entre les différents utilisateurs, via des composants appelés *DataSource*. Ces composants sont paramétrés dans le serveur d'applications et mis à disposition des applications déployées via le registre JNDI (Figure 4).

Pour accéder à une base de données MySQL, le développeur PHP utilise la fonction *mysql_connect*, en lui passant les paramètres de connexion (*host*, *username*, *password* et *dbname*). La même fonction peut être utilisée pour établir une connexion via une *datasource* ; dans ce cas, seul le nom de la *datasource* est passé en paramètre. Pour que cela fonctionne, il faut que la *datasource* ait été créée dans le serveur d'applications (Listing 3).

Par configuration, dans le fichier *WEB-INF/resin-web.xml*, il est possible d'ignorer les paramètres et d'imposer une base de données. Ceci peut être fort utile dans le cadre du portage d'une application *PHP traditionnelle* vers Quercus (Listing 4).

La classe PDO (*Portable Data Object*) permet de fournir des connexions à des bases de données, via des *datasources*, sans se préoccuper de leur type. Pour utiliser une connexion gérée par une *datasource*, il faut instancier un objet de type PDO en lui passant le nom complet de la *datasource* (Listing 5).

Appel de classe Java, classes standard

Il est possible d'instancier une classe Java standard, il faut utiliser la classe Java : `$a = new Java ("java.util.Date", 123);`. Cette façon de procéder ressemble beaucoup à celle que nous avions découverte pour le PHP / Java Bridge.

Il existe une autre façon de faire, plus simple, et certainement plus lisible car elle utilise directement les classes Java. La seule différence de programmation, par rapport à l'utilisation des classes PHP est l'existence de packages, qui

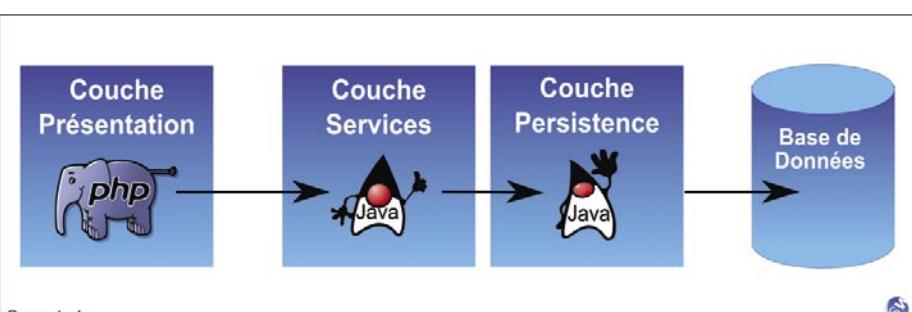


Figure 1. Architecture applicative avec PHP et Java

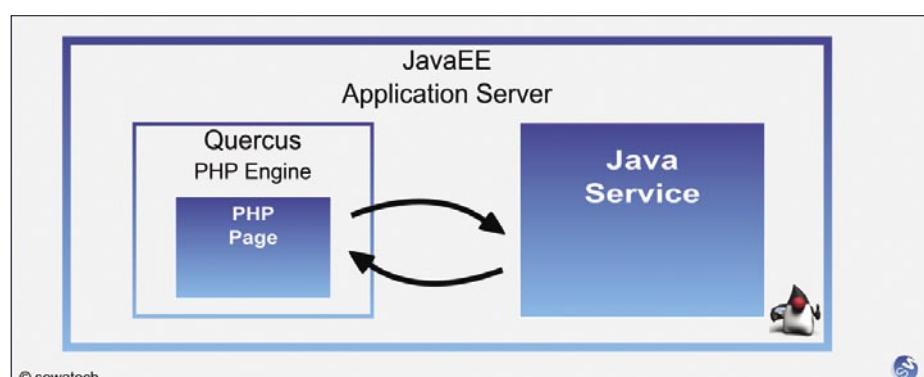


Figure 2. Interaction in-process entre Java et PHP

Listing 1. Démarrage depuis le répertoire d'installation de Resin

```
java -jar lib/resin.jar
```

Listing 2 Configuration WEB-INF/resin-web.xml

```
<web-app xmlns="http://caucho.com/ns/resin">
  <servlet-mapping url-pattern="*.php"
    servlet-class="com.caucho.quercus.
      servlet.QuercusServlet">
    <init>
      <ini-file>WEB-INF/php.ini</ini-file>
    </init>
  </servlet-mapping>
</web-app>
```

Listing 3. Connexion à une base de données MySQL

```
// Connection traditionnelle, en PHP
mysql_connect($host, $username, $password, $dbname);
// Connection via une datasource
mysql_connect("java:comp/env/jdbc/sewa-ds");
```

Listing 4. Configuration WEB-INF/web.xml

```
<web-app xmlns="http://caucho.com/ns/resin">
  <database jndi-name="jdbc/mysql">
    <driver type="org.gjt.mm.mysql.Driver">
      <url>jdbc:mysql://localhost:3306/test</url>
      <user></user>
      <password></password>
    </driver>
  </database>
  <servlet-mapping url-pattern="*.php" servlet-class=
    "com.caucho.quercus.servlet.QuercusServlet">
    <init>
      <textbf{\ensuremath{<}\database\ensuremath{>}}>
        java:comp/env/jdbc/myDatabaseName\ensuremath{<}/
      </init>
    </servlet-mapping>
</web-app>
```

Listing 5. Connexion portable à une base de données

```
$connection = new PDO("java:comp/env/jdbc/my-database");
```

Listing 6. Appel de classes Java

```
<?php
  // Avec la classe Java
  $unJour = new Java("java.util.Date", 123);
  // Avec les imports
  import java.util.Date;
  $leMemeJour = new Date(123);
?>
```

constituent des espaces de nommage. Avant de pouvoir utiliser une classe, il faut donc importer son nom, avec le mot clé `import` (Listing 6).

Les classes sont accessibles via le système de *classloaders* de Resin. Elles doivent être dans une des

- *WEB-INF/classes/* de l'application,
- *WEB-INF/lib/*.jar* de l'application,
- *lib/*.jar* de Resin.

Par ailleurs, il est possible de mélanger des classes PHP et des classes Java dans le répertoire *WEB-INF/classes/*.

Méthodes

L'appel de méthode peut se faire de façon classique, via les variables locales. L'accès aux propriétés peut être fait de façon simplifiée, sans utiliser le `get` ou le `set` explicitement.

Les méthodes static sont accessibles via la classe `java_class : $class = java_class("java.lang.System")`; puis `$in = $class->in;` ou directement sur la classe, si un import a été fait au préalable : `$calendar = Calendar::getInstance();`

Lorsque des méthodes sont surchargées, c'est-à-dire qu'elles ont le même nom et des arguments différents, le nombre d'arguments est prédominant. Par contre, si plusieurs méthodes ne sont différenciées que par le type des arguments, l'aspect non typé peut poser problème.

Fonctions

Les méthodes des classes qui héritent de `AbstractQuercusModule` sont exposées sous forme de fonctions PHP.

Composants Spring

Comme pour le pont PHP/Java, l'interopérabilité devient intéressante lorsqu'on peut appeler des composants évolués, tels que des EJB ou des beans Spring.

Pour accéder à un composant de Spring, il faut commencer par charger sa configuration en instanciant son contexte, puis lui demander une référence au *bean*. Il faut donc que les librairies nécessaires à Spring soient présentes dans le répertoire *WEB-INF/lib* de l'application et que le fichier de configuration soit dans le répertoire *WEB-INF/classes*. L'appel se fait comme si nous étions dans du code Java, avec les imports, par l'instanciation de la classe de contexte `Spring`, `ClassPathXmlApplicationContext` ou `context`, et l'appel de la méthode `getBean` sur l'objet de contexte (Listing 7).

Installation de Quercus dans Tomcat

Apache Tomcat est un serveur d'applications Web plus répandu que *Resin*. Il est donc intéressant de tester l'utilisation de Quercus avec *Tomcat*.

Listing 7. Appel de composants Spring

```

<?php
    import org.springframework.context.support.ClassPathXmlApplicationContext;
    // Instanciation du contexte applicatif de Spring
    $ctx = new ClassPathXmlApplicationContext("application-context.xml");
    // Demande d'une référence au bean courseService inscrit dans Spring
    $service = $ctx->getBean("courseService");
    // Utilisation du bean
    $course=$service->findById(1);
    echo ($course->toString());
?>

```

Listing 8. Configuration WEB-INF/web.xml

```

<servlet>
    <servlet-name>Quercus Servlet</servlet-name>
    <servlet-class>com.caucho.quercus.servlet.QuercusServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Quercus Servlet</servlet-name>
    <url-pattern>*.php</url-pattern>
</servlet-mapping>

```

Installation

Pour démarrer, il faut un environnement *Tomcat* qui fonctionne, avec *Tomcat* 5 ou 6 et un *JDK* 5 ou 6. Il faut aussi télécharger *Quercus Standalone* sur le site de *Caucho*.

Pour installer *Quercus*, il faut décompresser le fichier war téléchargé (*quercus-3.2.1.war*) sous forme de répertoire (*quercus*, par exem-

ple), puis copier ce répertoire dans le répertoire *webapps* de *Tomcat*. Les pages PHP pourront être ajoutées dans ce répertoire ; elles seront appelées par l'URL <http://localhost:8080/quercus/nom-de-la-page.php>.

Cette procédure peut s'appliquer à n'importe quel serveur d'applications qui supporte le déploiement en mode répertoire (*JBoss*,

Sur Internet

- <http://www.caucho.com/> – Société Caucho, éditeur de Resin et de Quercus,
- <http://quercus.caucho.com/> – Accès direct à Quercus,
- <http://www.projectzero.org/> – Project Zero.

Glassfish,...). Pour les serveurs qui imposent le mode archive, il faut décompresser l'archive de *Quercus*, ajouter les fichiers *php*, puis reconstituer l'archive.

Ajout de PHP dans une application

Pour ajouter le support de PHP à une application existante ou en cours de développement, il faut surtout ajouter la servlet *Quercus Servlet*.

Pour cela, il faut prendre les fichiers *quercus.jar* et *resin-util.jar* dans le répertoire *WEB-INF/lib* de *web.xml* du répertoire *WEB-INF*. Enfin, dans le même fichier, il faut déclarer la prise en charge des URL avec l'extension *php* par cette servlet (Listing 8).

Ajout de PHP dans toutes les applications

Dans *Resin*, toutes les applications supportent PHP. Pour avoir le même comportement dans *Tomcat*, il faut placer les fichiers *quercus.jar* et *resin-util.jar* dans le répertoire *lib* de *Tomcat*. Ensuite, il faut déclarer la servlet de *Quercus* qu'une seule fois, au niveau global, dans le fichier *web.xml* du répertoire *conf* de *Tomcat*.

Tomcat étant utilisé au sein d'autres serveurs d'applications, comme *JBoss*, cette procédure d'installation peut aussi s'appliquer à ceux-ci. Attention, dans le cas d'une déclaration globale, il ne faut plus déclarer localement la servlet car cela créerait une conflit.

Conclusion

Nous avons découvert comment intégrer du PHP dans une application Web Java, ou comment utiliser des composants Java dans une application PHP. La technique que nous avons abordée est certainement une des plus prometteuses pour exploiter au mieux les capacités des deux langages. Il reste à surveiller l'évolution de cette solution, en particulier son futur support de PHP 6, ainsi que l'implémentation de nouveaux modules optionnels de PHP5.

ALEXIS HASSLER

Alexis Hassler (alexis.hassler@sewatech.fr) est consultant et formateur en technologies Objet, spécialisé en architectures orientées services, en Java et en conception objet avec UML. En 2005, il a fondé SEWATECH (<http://www.sewatech.fr>), société de conseil et de formation.

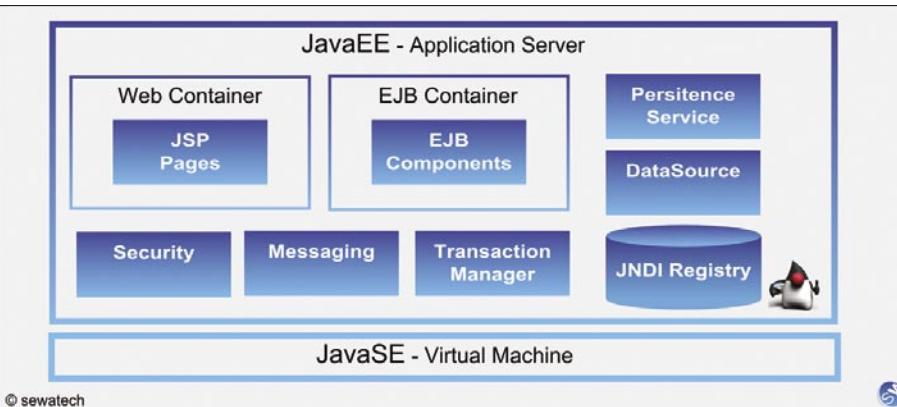


Figure 3. Serveur d'applications Java EE

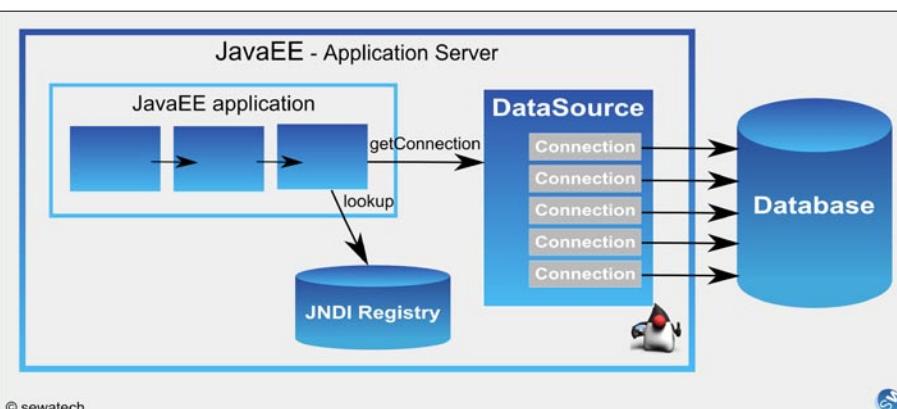


Figure 4. DataSource

Messagerie instantanée web avec AJAX et PHP

La synchronisation des données entre clients est un enjeux essentiel des applications AJAX. Nous allons étudier, via un cas pratique, la méthode Push Server. C'est celle utilisée par la plupart des messageries web.

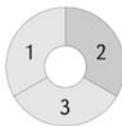
Cet article explique :

- Comment implémenter une messagerie instantanée web.
- Comment initier un transfert de données depuis le serveur web.

Ce qu'il faut savoir :

- Les bases de JavaScript.
- Les bases de PHP.
- Les grands principes du protocole HTTP.

Niveau de difficulté



La recette d'une application web AJAX réussie est simple. Côté client, il faut un bon squelette HTML à base de `DIV`, `SPAN`, `IMG`, `A` et `TABLE` ; une feuille de style CSS ayant recours aux classes multiples, pour optimiser son poids ; du code JavaScript orienté objet s'appuyant sur un framework AJAX comme *Prototype* ou *jQuery*.

Côté serveur, je préconise un fichier unique servant d'interface entre les requêtes client AJAX entrantes et le code métier serveur à proprement dit. C'est la seule chose qui change côté serveur entre une application AJAX et une application web classique !

Dans notre messagerie instantanée, nous allons rajouter un ingrédient : la possibilité d'envoyer des données du serveur vers le client à l'initiative du serveur ! En temps normal, pour que le serveur envoie des données au client, il faut que le client en fasse la demande via une requête AJAX. À ce moment là, le serveur peut envoyer ses données en réponse.

Construction du squelette de l'application

Notre application sera constituée, côté client, d'une seule page HTML, nommée *web-messenger.html*. Pour commencer, nous choisissons le DOCTYPE *Transitional*, il permet une meilleure compatibilité entre Internet Ex-

plorer et Firefox au niveau des CSS sans être trop strict !

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML xmlns="http://www.w3.org/1999/xhtml">
```

Viennent ensuite les balises contenues dans `HEAD` présentées dans le Listing 1. Une inclusion de la librairie *jQuery*, la balise `SCRIPT` qui contiendra le code client de notre application et enfin la balise `STYLE` pour tout ce qui est CSS.

Le Listing 2 présente le corps de notre application, qui est composée : d'une zone à fond bleue affichant les connexions/déconnexions ainsi que les messages envoyés par tous les utilisateurs ; d'une zone de saisie d'un nouveau message ; à droite, de la zone de *login* et dessous, de la liste des utilisateurs connectés ; enfin, de deux zones pour les messages de confirmation et d'erreur provenant du serveur.

Presque toutes les `DIV` ont des `ID` car nous allons y faire référence soit depuis le code JavaScript, soit depuis les déclarations CSS. Cette application est mono-conversation, c'est-à-dire qu'on ne peut démarrer plusieurs conversations avec des utilisateurs différents. Dans le cas d'une messagerie multi-conversations, on pourra donc avoir plusieurs `DIV` *conversation* et il faudra donc passer par l'attribut `CLASS` plutôt que par `ID` pour définir le style.

Les `DIV` conversation et message ont chacune trois `DIV` enfant. La première et la dernière per-

mettent de gérer les arrondis, celle du milieu recevra les données à afficher. Cette pratique est courante sur le web. Le fait d'utiliser trois `DIV` permet d'avoir une zone fixe en largeur mais variable en hauteur. Si on voulait que la zone puisse être variable en largeur, il faudrait utiliser un damier de neuf `DIV`, sur le même principe. Le Listing 3 présente tous les styles CSS permettant de présenter notre application.

Principe de fonctionnement de la messagerie

Maintenant que l'interface cliente est montée, voyons comment nous allons l'interfacer avec le serveur. Il faut bien garder en mémoire qu'on peut avoir plusieurs clients simultanés mais un seul serveur.

Les interactions possibles sont les suivantes :

- quand on saisit un pseudo et qu'on valide, le client doit envoyer une demande de connexion au serveur,
- au chargement initial comme à chaque nouvelle connexion, le serveur doit envoyer aux clients la liste des utilisateurs connectés,
- quand un client saisit un nouveau message, il doit l'envoyer au serveur qui répercute sur tous les clients,
- enfin, quand un client se déconnecte en fermant la page web, le client doit notifier



Figure 1. Notre messagerie web en cours d'utilisation

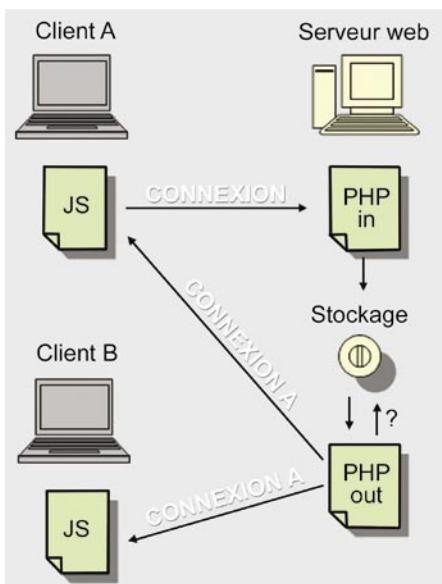


Figure 2. Architecture client/serveur de la messagerie

le serveur qui répercute l'information sur tous les autres clients.

Côté serveur se dégagent deux rôles distincts : le premier consiste à réceptionner les informations en provenance de chaque client (connexion, déconnexion, nouveau message) ; le deuxième, à l'inverse, consiste à informer tous les clients de l'activité d'un client (connexion, déconnexion, nouveau message). Nous allons donc créer deux fichiers serveur : *server-in.php* et *server-out.php*.

Ainsi, quand un client se connecte, il fait une requête AJAX vers *server-in.php* qui stocke l'information dans un espace d'échange (fichier ou base de données). À ce moment-là, *server-out.php* doit être capable de lire l'information dans cet espace d'échange pour la communiquer à tous les clients, y compris le client apellant.

En HTTP, il est impossible qu'un serveur envoie des informations à un client qui n'en a pas fait la demande. Aussi la notification à tous les clients de la part de *server-out.php* semble impossible.

Une solution simple consiste pour chaque client, à lancer une requête AJAX à intervalles réguliers, vers *server-out.php*, qui, s'il a de nouvelles informations à communiquer, renverra ces informations, sinon un message indiquant qu'aucune nouvelle connexion ou message n'est arrivé depuis la dernière demande. Si on considère que les intervalles de temps entre deux requêtes sont très faibles, alors on a une impression de simultanéité. Le problème est que plus on abaisse la durée de l'intervalle, plus les clients font de requêtes vers le serveur. Et dans 99% des cas, pour rien, puisqu'il n'y a pas de nouvelle connexion ou de nouveaux messages toutes les millisecondes !

Notre solution, qui s'approche très près d'un véritable *push* serveur, consiste pour les clients

à lancer une seule requête AJAX à laquelle le serveur ne répond pas immédiatement. En effet, le code de *server-out.php* qui est appelé tourne en boucle jusqu'à l'arrivée du prochain message ou connexion dans l'espace de stockage.

Interrogation de l'espace de stockage

L'espace de stockage permet aux deux scripts PHP de communiquer. Quand une information arrive du client (connexion, nouveau message), *server-in.php* la stocke pendant 1 seconde dans le fichier *server-message.txt*. De ce fait, *server-out.php*, qui interroge *server-message.txt* toutes les 200 millisecondes, a le temps de récupérer l'information avant qu'elle ne soit détruite par *server-in.php* une fois la seconde écoulée.

Afin que le client ne reçoive jamais deux fois la même information, chaque information possède un identifiant qui est donné par

server-in.php et stocké dans le fichier *server-message-id.txt*.

On a donc, côté serveur, un *ping* régulier qui était réalisé auparavant par le client. Si au bout de 50 secondes, soit 250 interrogations, *server-out.php* ne trouve toujours aucune information à diffuser, alors le script se termine en notifiant le client que le délai est expiré avec l'identifiant de message 0. Le Listing 4 correspond à ce script.

Réception et enregistrement d'un message client

Lorsque le client se connecte, se déconnecte, demande la liste des utilisateurs connectés ou envoie un nouveau message, il fait une requête AJAX vers le fichier *server-in.php* présenté dans le Listing 5. Nous avons vu que les messages à diffuser étaient stockés dans un fichier TXT. La liste des utilisateurs connectés est, quant à elle, mémorisée en variable d'application. Nous

Listing 1. Contenu de la balise HEAD

```
<HEAD>
  <SCRIPT TYPE="text/javascript" SRC="jquery-1.2.6.js">
  </SCRIPT>
  <SCRIPT TYPE="text/javascript">
  </SCRIPT>
  <STYLE>
  </STYLE>
</HEAD>
```

Listing 2. Contenu du BODY de notre application

```
<BODY>
  <DIV id="main">
    <DIV id="login">
      Choisissez votre pseudo<br />
      <INPUT TYPE="text" VALUE="" ID="pseudo" />
    </DIV>
    <DIV id="contacts">
      Utilisateurs connectés
      <DIV id="contacts_main"></DIV>
    </DIV>
    <DIV id="conversation">
      <DIV id="conversation_top"></DIV>
      <DIV id="conversation_main"></DIV>
      <DIV id="conversation_bottom"></DIV>
    </DIV>
    <DIV id="message">
      <DIV id="message_top"></DIV>
      <DIV id="message_main">
        <INPUT TYPE="text" id="message_text" />
        <INPUT TYPE="button" VALUE="SEND" id="message_button" />
      </DIV>
      <DIV id="message_bottom"></DIV>
    </DIV>
    <DIV ID="confirm"></DIV>
    <DIV ID="erreur"></DIV>
  </DIV>
</BODY>
```

Listing 3. Style CSS de notre application

```

#main {
    width: 720px;
    /* Permet d'avoir des éléments enfants placés en absolute */
    position: relative;
}
#main > div {
    margin: 10px;
}
/* Le même style peut être affecté à plusieurs éléments */
#conversation, #message {
    width: 327px;
}
#conversation_top, #message_top {
    height: 5px;
    /* L'image de fond des arrondis supérieurs fixe la largeur
       de la zone */
    background: url(conversation_top.gif) 0 0 no-repeat #FFF;
}
#conversation_bottom, #message_bottom {
    height: 5px;
    background: url(conversation_bottom.gif) 0 0 no-repeat #FFF;
}
#conversation_main {
    /* Les bordures sont de la même couleur que les bords des
       arrondis */
    border-left: 1px solid #39638C;
    border-right: 1px solid #39638C;
    /* La hauteur de la zone peut être modifiée en JS par la suite */
    height: 300px;
    /* Le fond est également de la même couleur que les arrondis */
    background-color: #DEE7EF;
    padding: 0px 6px;
    /* Permet d'afficher un scroll vertical si besoin */
    overflow: auto;
}
#message_main {
    border-left: 1px solid #39638C;
    border-right: 1px solid #39638C;
    height: 26px;
    background-color: #DEE7EF;
    padding: 0px 6px;
}
#message_text {
    height: 18px;
    width: 248px;
    /* Permet d'afficher le champ et le bouton sur la même ligne */
    float: left;
}
#message_button {
    width: 54px;
    height: 26px;
    margin-left: 5px;
}
#login {
    position: absolute;
    /* Par défaut, top est à 0px également */
    right: 0px;
    text-align: right;
}
#contacts {
    width: 150px;
    position: absolute;
    right: 0px;
    top: 60px;
    text-align: right;
}
#contacts_main {
    border: 1px solid #7F9DB9;
    height: 220px;
    overflow: auto;
    text-align: left;
    padding: 4px;
}
#erreur {
    background-color: #F66;
    color: #099;
    height: 20px;
    padding: 4px;
    /* La zone ne s'affichera qu'en cas de besoin */
    display: none;
}
/* Zone de message positif de la part du serveur */
#confirm {
    background-color: #6F6;
    color: #909;
    height: 20px;
    padding: 4px;
    display: none;
}

```

utilisons le script disponible sur le site <http://www.leosingleton.com> pour simuler en PHP les variables d'application. En effet, ces variables doivent être partagées par toutes les sessions client. In fine, elles sont aussi stockées dans un fichier, *application-variables.txt*.

Par défaut, le fichier d'échange *server-message.txt* est vide. Lorsque le fichier *server-in.php* reçoit un message, il l'écrit dans le fichier d'échange et patiente 1 seconde. Pendant cette seconde, tous les *server-out.php* en cours d'exécution ont le temps d'interroger le fichier d'échange, de constater qu'il contient un message et de renvoyer ce message au client.

Une amélioration de ce cas pratique consisterait à trouver un moyen pour les scripts PHP en cours d'exécution de communiquer direc-

tement entre eux, sans devoir passer par un fichier intermédiaire.

Gestion des événements côté client

Il ne nous reste plus qu'à connecter les pièces, c'est-à-dire les éléments d'action HTML avec les cas serveurs correspondant. Pour se faire, nous utilisons la librairie *jQuery* qui permet de simplifier l'accès aux DOM et la gestion des requêtes AJAX. Pour commencer, nous allons gérer dans notre SCRIPT quelques variables globales, présentées dans le Listing 6.

Pour éviter tout problème de cache client ou serveur, je rajoute toujours dans mes applications un paramètre aléatoire à la fin des URL serveur appelées.

```

function urlend() {
    var r = Math.random()*99999999;
    return '',
    '&r=' + Math.floor(r);
}

```

La fonction du Listing 7 permet de récupérer la liste des utilisateurs connectés. Nous l'appellerons lors du premier chargement de l'application ainsi qu'après toute connexion ou déconnexion.

Dans la fonction de connexion des utilisateurs, décrite dans le Listing 8, il faut retenir l'utilisation de *keyCode* qui permet de détecter que l'utilisateur appuie sur la touche *Entrée*.

Cette fonction est appelée quand l'utilisateur appuie sur *Entrée*. Dans un événement

Listing 4. Code du fichier server-out.php

```

<?
//Nom du fichier de stockage
$fichier = "server-message.txt";
//Début des interrogations
$start = time();
//Boucle infinie
while (true) {
    //Au bout de 50 secondes,
    //on arrête
    if (time() > $start+50) {
        //On informe le client
        //de l'arrêt
        echo "[0, 'TROP LONG']";
        break;
    } else {

        $fp = fopen($fichier, "r");
        //On lit son contenu
        $data = fgets($fp, 512);
        fclose($fp);
        //S'il contient une
        //information
        if ( strlen($data) > 0 ) {
            //On renvoie cette
            //information au client
            echo $data;
            break;
        } else {
            //Sinon, on attend 200
            //millisecondes
            sleep(0.2);
        }
    }
}
?>

```

DOM, *this* fait référence à l'élément HTML. On reçoit en paramètre l'événement souris, qu'on transmet grâce au tableau *arguments* qui contient tous les paramètres envoyés à notre code. Il faut donc modifier le champ de saisie du *login* en conséquence.

```

<INPUT TYPE="text" VALUE="" ID="pseudo"
    ONKEYPRESS="log_user(this.value,
    arguments[0]);" />

```

Je n'ai volontairement pas créé de contrôle visuel qui permette à l'utilisateur de se déconnecter. Il faut donc détecter automatiquement la fermeture de la fenêtre. L'événement *window.onbeforeunload* va nous permettre d'exécuter du code JavaScript lorsque la page est déchargée, ce qui se produit, entre autres, quand on ferme la fenêtre. Pour gérer la déconnexion, il faut donc implémenter deux fonctions, décrites dans le Listing 9.

Lorsque le client réceptionne des données en provenance du serveur, c'est sous forme de chaîne de caractères. Si les données contiennent une déclaration HTML, on peut l'incorporer directement dans la page en l'affectant en *innerHTML* d'un élément déjà présent. Mais si les données sont des commandes JavaScript ou des objets JavaScript, alors il faut utiliser

Listing 5. Code du fichier server-in.php

```

<?
//Librairie qui gère les variables d'application
include("application.php");
application_start();
//Récupération du contexte
switch ($_GET["cas"]) {
    //Le client se loggue
    case "log_user":
        $login = $_POST["login"];
        //Si c'est le premier, on crée le tableau
        if (!$_APP["logins"]) {
            $_APP["logins"] = array();
        }
        //S'il n'est pas déjà loggué, on le rajoute
        if (!in_array($login, $_APP["logins"])) {
            array_push($_APP["logins"], $login);
            application_end();
            //On informe tous les clients
            propage_message("Nouvelle connexion : $login");
        } else {
            //Message d'erreur pour le client
            echo "[0, 'Ce login est déjà pris']";
        }
        break;
    //On retourne la liste des utilisateurs
    case "list_users":
        $html = "";
        if ($_APP["logins"]) {
            $html = implode($_APP["logins"], "<br />");
        }
        echo $html;
        break;
    //Nouveau message
    $pseudo = $_GET["pseudo"];
    $message = $_POST["message"];
    //On propage le message à tous les clients
    propage_message("Nouveau message de <b>$pseudo</b> : $message");
    break;
    //Un client se déconnecte
    case "logout_user":
        $pseudo = $_GET["pseudo"];
        //On supprime son login du tableau
        $_APP["logins"] = array_diff($_APP["logins"], array($pseudo));
        application_end();
        //On informe tous les clients
        propage_message("Déconnexion : $pseudo");
        break;
    }
}

function propage_message($message) {
    $fichier = "server-message.txt";
    $id = "server-message-id.txt";
    //On incrémente le numéro de message
    $fp = fopen($id, 'r') or die("can't open file");
    $count = fread($fp, filesize($id));
    $count++;
    fclose($fp);
    $fp = fopen($id, 'w') or die("can't open file");
    fwrite($fp, "$count");
    fclose($fp);
    //On écrit le message avec son numéro
    $fp = fopen($fichier, 'w') or die("can't open file");
    fwrite($fp, "[ $count, '$message' ]");
    fclose($fp);
    //On laisse le temps aux clients de consulter le message
    sleep(1);
    $fp = fopen($fichier, 'w') or die("can't open file");
    fclose($fp);
}

application_end();
?>

```

Listing 6. Variables globales côté client

```
//Numéro de message, pour ne pas
//afficher 2 fois le même message
var GLO_message_count = 0;
//Timers pour gérer le fadeIn/
//fadeOut des messages
var GLO_timer_erreur = null;
var GLO_timer_confirm = null;
//On mémorise le pseudo de
//l'utilisateur connecté
var GLO_pseudo = null;
```

Listing 7. Fonction qui récupère la liste des utilisateurs

```
function load_liste_utilisateurs() {
    //$.get permet de lancer une
    //requête AJAX vers l'URL spécifiée

    $.get('server-in.php?cas=list_
    users'+urlend(), function(reponse) {
        //$.get permet de récupérer
        //un élément DOM
        //et html() permet de lui
        //affecter un nouveau contenu
        $('#contacts_
        main').html(reponse);
    });
}
```

Listing 8. Fonction qui loggue un utilisateur

```
function log_user(pseudo, evt) {
    //Récupération de l'évènement
    //souris IE et Firefox
    evt = evt || event;
    //Si c'est la touche entrée
    //qui est pressée
    if ( evt.keyCode == 13 ) {
        //Si le pseudo saisi n'est pas vide
        if ( pseudo.length > 0 ) {
            //On loggue l'utilisateur
            //en envoyant le pseudo choisi
            $.post('server-in.php?cas=
            log_user'+urlend(), {
                login: pseudo
            });
            function(reponse) {
                //Le serveur renvoie
                //un message d'erreur ou vide
                if ( reponse.length > 0 ) {
                    erreur(getServerMessage
                    (reponse));
                } else {
                    ok("Vous êtes connecté
                    sous le pseudo "+pseudo);
                    //On vide le champ
                    //de saisie du pseudo
                    $('#pseudo').val('');
                    //On mémorise le pseudo
                    //dans la variable globale
                    GLO_pseudo = pseudo;
                }
            });
        } else {
            erreur("Votre pseudo est
            vide !!");
        }
    }
}
```

Listing 9. Fonctions de gestion de la déconnexion

```
function logout_user() {
    //Si l'utilisateur était loggué
    if ( GLO_pseudo != null ) {
        //On indique sa déconnexion au serveur
        $.post('server-in.php?cas=logout_user&pseudo='+GLO_pseudo+urlend());
    }
}
window.onbeforeunload = function() {
    logout_user();
}
```

Listing 10. Décapsulation des données retournées par le serveur

```
function getServerMessage(response) {
    //La réponse du serveur est un tableau
    //Avec le numéro de message et le texte
    var tabMessage = eval(response);
    //On ne s'intéresse qu'au texte du message
    return tabMessage[1];
}
```

Listing 11. Fonction de gestion des messages d'erreur du serveur

```
function erreur(message) {
    //On supprime tout message d'erreur prévu pour s'afficher
    clearTimeout(GLO_timer_erreur);
    //On affecte le message à l'élément correspondant
    $('#erreur').html(message);
    //On affiche lentement le message
    $('#erreur').fadeIn('slow');
    //On prévoit sa disparition rapide dans 5 secondes
    GLO_timer_erreur = setTimeout(function() {
        $('#erreur').fadeOut('fast');
    }, 5000);
}
```

Listing 12. Fonction d'écoute du serveur

```
function start_ecoute() {
    //Lancement de la requête AJAX vers server-out.php
    $.get('server-out.php?q=1'+urlend(), function(strTabMessage) {
        //Une fois la requête terminée, on décapsule le message
        tabMessage = eval(strTabMessage);
        //S'il a le numéro 0, c'est un message d'erreur
        if (tabMessage[0] == 0) {
            //On l'ignore et on relance une requête
            start_ecoute();
        } else {
            //Si le message est nouveau
            if ( GLO_message_count != tabMessage[0] ) {
                //On mémorise son numéro
                GLO_message_count = tabMessage[0];
                //On rajoute le message à la conversation
                $('#conversation_main').html($('#conversation_main').html()+tabMessage[1]+
                "<br />");
                //Si des actions spécifiques doivent être menées, on transfert
                //le message à la fonction
                traiteMessage(tabMessage[1]);
                //On relance l'écoute
                start_ecoute();
            } else {
                //Si le message n'est pas nouveau, on l'ignore et on relance
                start_ecoute();
            }
        }
    });
}
```

Listing 13. Traitement du message serveur selon le contexte

```

function traiteMessage(message) {
    //Si c'est un message de connexion ou déconnexion
    if ( message.indexOf("Nouvelle connexion") != -1 || 
        message.indexOf("Deconnexion") != -1 ) {
        //Il faut rafraîchir la liste des utilisateurs
        load_liste_utilisateurs();
    }
    //Dans tous les cas, il faut scroller le plus bas possible
    //la conversation
    $('#conversation_main').scrollTop(9999999);
}

```

Listing 14. Envoie d'un nouveau message au serveur

```

function envoie_message() {
    //On récupère le message
    var texte = $('#message_text').val();
    //Si l'utilisateur est connecté
    if ( GLO_pseudo != null ) {
        //Si le texte est non vide
        if ( texte.length > 0 ) {
            //On envoie le message
            $.post('server-in.php?cas=user_message&pseudo='+GLO_pseudo+urlend(), {
                message: texte
            }, function() {
                //Au retour, on vide la zone de saisie pour
                //le prochain message
                $('#message_text').val('');
            });
        }
    } else {
        erreur("Vous devez vous connecter pour poster un message");
    }
}

function test_envoie_message(evt) {
    evt = evt || event;
    if ( evt.keyCode == 13 ) {
        envoie_message();
    }
}

```

la fonction `eval` pour transformer les données texte en code JavaScript exécuté. La fonction du Listing 10 nous permet d'effectuer cette transformation.

Pour gérer ces messages d'erreur ou de confirmation renvoyés par le serveur, nous avons deux fonctions similaires, dont une est présentée dans le Listing 11, qui pointent chacune vers l'élément HTML correspondant. Nous utilisons ici `jQuery` pour gérer l'apparition

lente de la zone et sa disparition rapide après 5 secondes d'affichage.

La fonction `ok()` fonctionne sur le même principe.

Après le chargement initial de notre application, le client doit commencer à écouter le serveur en vue d'être notifié du prochain message ou de la prochaine connexion. Le client doit donc lancer une requête AJAX vers `server-out.php`. C'est le sujet de la fonction `start_`

écoute présentée dans le Listing 12. Cette requête sera soit traitée par le serveur dès qu'un nouveau message arrivera, soit terminée au bout des 50 secondes de délai. Dans les deux cas, le client relancera la même requête pour le prochain message.

Suite à la réception d'un message de la part du serveur, j'ai identifié le besoin de réaliser certaines actions client en fonction de la nature du message, c'est l'objet du Listing 13. Pour des raisons de simplicité, la nature du message est contenue dans le message lui-même, mais ça pourrait être un paramètre supplémentaire.

Maintenant que nous avons défini les deux fonctions qui doivent être appelées au chargement initial, voilà à quoi ressemble `window.onload`:

```

window.onload = function() {
    load_liste_utilisateurs();
    start_ecoute();
}

```

Enfin, la fonction principale, présentée dans le Listing 14, permet d'envoyer un nouveau message. Il y a deux façons de poster un message, soit en cliquant sur le bouton `Send`, soit en appuyant sur `Entrée`. On va donc retrouver deux fonctions.

La fonction `test_envoie_message` n'est appelée que depuis le champ de saisie d'un message.

```

<INPUT TYPE="text" id="message_text"
    ONKEYPRESS="test_envoie_
    message(arguments[0]);" />

```

Alors que la fonction `envoie_message` est appelée directement quand l'utilisateur clic sur le bouton d'envoie.

```

<INPUT TYPE="button" VALUE="SEND"
    id="message_button" ONCLICK=
        "envoie_message();" />

```

Conclusion

Le cas pratique que nous venons de voir simule un envoi de données du serveur vers le client, à l'initiative du serveur. C'est une technique utilisée notamment par le site <http://www.meebo.com>, qui se veut un aggrégateur de messageries instantanées sur le web. Contrairement au `ping` client/serveur, cette méthode a pour principal intérêt de n'occuper les processus client et serveur que lors des nouveaux messages ou connexions. Vous pouvez retrouver cette messagerie web à l'adresse <http://labs.lucca.fr/exemples/web-messenger.html>.

NICOLAS FAUGOUT

Diplômé des Mines de Douai, Nicolas Faugout exerce son activité au sein de la société LUCCA (www.lucca.fr). Après s'être formé aux logiciels et aux techniques de développement LUCCA, Nicolas a pris en charge des formations sur l'utilisation de la méthode de développement AJAX dont il est devenu un expert.

Sur Internet

- <http://labs.lucca.fr/exemples/web-messenger.html> – L'exemple de la messagerie web disponible en ligne,
- <http://www.lesingleton.com/projects/code/phpapp/> – Librairie PHP qui permet de gérer facilement des variables d'application,
- <http://www.meebo.com/> – Aggrégateur de messagerie web,
- <http://jquery.com/> – Site officiel de la librairie JavaScript jQuery.

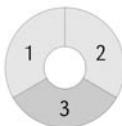
Creation de template Joomla! 1.5

Le CMS Joomla! connaît un réel succès pour tous ceux qui souhaitent créer un site Internet sans partir de zéro. Un des avantages de Joomla! est de pouvoir personnaliser et changer l'habillage graphique du site facilement.

Cet article explique :

- Vous apprendrez à créer et personnaliser votre premier template Joomla! 1.5.

Niveau de difficulté



En effet le système de gestion de contenu est pourvu d'un moteur de template qui lui permet de dissocier la partie technique de la partie graphique. Ce moteur associé à des normes rigoureuses de codage permet d'offrir des capacités de personnalisations graphiques étonnantes.

La création de site Internet avec le CMS Joomla! rencontre un vif succès en partie grâce à sa capacité d'évolution et d'adaptation. On compte à ce jour plus de 5000 extensions de tout type : *composants*, *modules*, *plugins*, *langues* et *templates*. Nous allons nous intéresser à ces derniers dans cet article. En effet, le terme de template est un mot anglicisme utilisé généralement en informatique pour désigner un modèle de conception de logiciel ou de présentation de données. Pour utiliser des termes français, on peut parler de gabarit, de thème ou de charte graphique. Dans le milieu du CMS Joomla!, le template est une extension qui permet de séparer le fond (contenu informationnel) de la forme (la manière dont il est présenté). Un template Joomla! agit comme un modèle dans lequel seulement certains éléments, comme le contenu sont modifiables. Ce concept permet de faciliter la création et la mise à jour d'un site tant sur le contenu que sur la présentation graphique. Grâce à ce système d'extension, les templates peuvent être installés en nombre illimité et la charte graphique d'un site Internet peut être complé-

Ce qu'il faut savoir :

- Bonnes connaissances en XHTML, CSS et PHP.
- Connaissance dans l'administration du CMS Joomla!.

template *PatTemplate* (<http://trac.php-tools.net/patTemplate>) ce qui permet de bénéficier directement des fonctionnalités reconnues de ce moteur. Cette intégration rend un template Joomla! plus difficile à créer pour un débutant mais donne beaucoup plus de possibilités une fois les bases maîtrisées.

Dans cet article, nous aborderons toutes les étapes de la création d'un template Joomla! et nous nous attacherons plus particulièrement aux aspects techniques que propose le système.

Processus de création

La création d'un template Joomla! suit un processus toujours quasi similaire. Ses étapes se

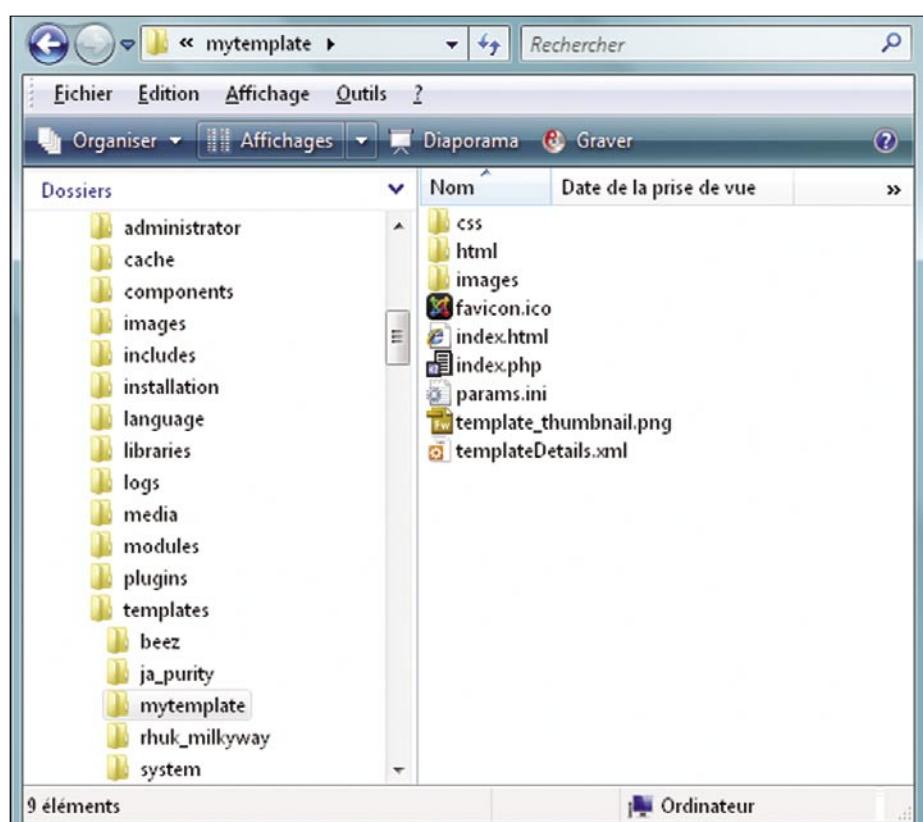


Figure 1. Liste des fichiers et répertoires d'un template Joomla!

distinguent souvent par l'intervention de compétences différentes. Lorsque les moyens le permettent, il est préférable de confier chaque étape de la création à la personne ayant les compétences les plus appropriées. Bien entendu, il est possible pour une seule et même personne de réaliser toutes ces étapes.

Création graphique

La création de la partie graphique consiste à créer une identité visuelle et ensuite la charte graphique qui en découle. Cette opération est généralement réalisée par un infographiste.

La charte graphique est un document qui définit l'ensemble des règles fondamentales d'utilisation des signes graphiques qui constituent l'identité visuelle. Ce document contient le *logotype*, les polices de caractères, les jeux de couleurs, des éléments graphiques...

Dans le cadre d'un projet de création de site, ce document est complété par une représentation graphique des différentes pages du site afin de définir la position des différents éléments. Typiquement, on retrouve une représentation graphique de la page d'accueil et une de la page de contenu.

Adobe Photoshop et l'aboutissement de cette création graphique se concrétise par un ou plusieurs fichiers au format PSD. Ces fichiers contiennent les différentes représentations graphiques des pages du site. Le système de calque proposé par le format PSD permet d'isoler des éléments graphiques qui serviront à faciliter la découpe et l'intégration de la maquette HTML.

Création de la maquette HTML

Cette étape relève souvent d'un intégrateur HTML qui saura utiliser des méthodes de codage optimisées (exemple : *tableless*, *xhtml*, *div*, *css*, ...) et s'assurer de la compatibilité avec les différents navigateurs.

Avant de réaliser le template Joomla!, il est conseillé de créer une maquette en HTML de la représentation graphique. Cette maquette servira de base de travail à l'intégration du code PHP spécifique de Joomla!. Pour ce faire, il faut effectuer une découpe du fichier PSD et extraire certains éléments graphiques au format GIF, JPG ou PNG. Ces formats sont plus appropriés pour le web car ils sont plus légers et donc plus rapides à charger. La difficulté de cet opération réside dans le choix des éléments à extraire, il faut alors se projeter sur la méthode de conception de la maquette HTML et jouer avec la CSS afin d'en optimiser le découpage. Par exemple, pour un arrière plan (*background*) de couleur unie, la découpe d'une image n'est pas nécessaire car on peut utiliser la propriété *background-color* de la CSS. Si l'arrière plan possède un motif répété, il n'est pas nécessaire de découper toute la partie, seul un bout suffit car on peut utiliser la propriété *repeat* de la CSS pour faire répéter plusieurs fois l'image.

Après la découpe, on obtient plusieurs images qui serviront à créer la page en HTML.

Listing 1. Représentation de la structure du site en code HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr-fr" lang="fr-fr" dir="ltr" >
  <link href="/joomla158/templates/montemplatejoomla/css/template.css"
        rel="stylesheet" type="text/css" />
</head>
<body>
  <div id="Partie1">Partie entête du site<br /><br />
    <div id="Position1">Logo</div>
    <div id="Position2">Champ de recherche</div>
  </div>
  <div id="Partie2">Partie principale du site<br /><br />
    <div id="Position3">Fil d'Ariane</div>
    <div id="Position4">Colonne de gauche</div>
    <div id="Position5">Contenu principal</div>
    <div id="Position6">Colonne de droite</div>
  </div>
  <div id="Partie3">Pied de page du site<br /><br />
    <div id="Section7">Copyright</div>
  </div>
</body>
</html>
```

En cohérence avec la structure d'un template Joomla!, il convient de créer un répertoire image pour y stocker toutes les images. On peut bien entendu classer les images en créant des sous répertoires. Il faut également créer un répertoire CSS pour y stocker les feuilles de style. Par convention de nommage, le fichier HTML portera le nom *index.html*.

Transformation de la maquette en template Joomla!

Cette étape fait intervenir du code PHP, il sera donc plus aisé pour un développeur PHP d'intégrer du code Joomla!.

Le but de cette étape est de rendre compatible la maquette HTML avec le CMS Joomla!. Pour cela, Joomla! dispose de plusieurs codes appelés JDOC, suite au nom donné aux balises. Le travail consiste à identifier les zones de contenu dynamique et de placer les balises JDOC avec des noms de positions appropriés.

L'intégration des balises JDOC permet par la suite d'afficher les différents contenus du site (article, module, composant, ...). La difficulté est de bien placer la balise au bon endroit sans que l'affichage graphique en soit perturbé. Il faudra parfois modifier et ajuster le code HTML de la maquette initiale pour obtenir le résultat voulu.

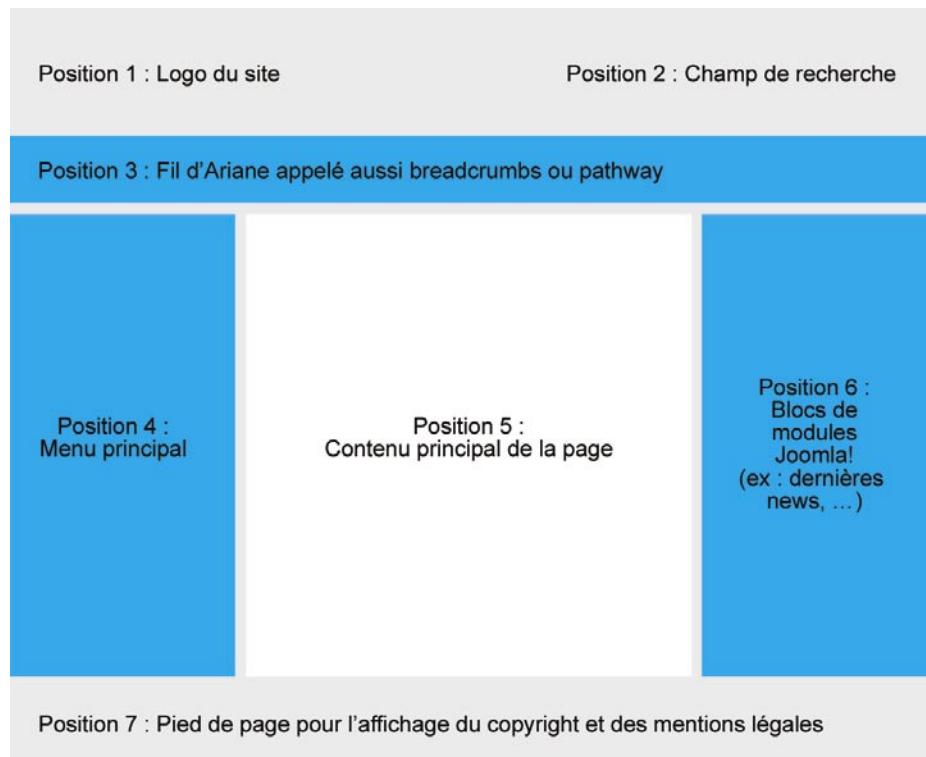


Figure 2. Représentation visuelle de la structure

Listing 2: Code source complet du fichier index.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
           "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr-fr" lang="fr-fr" dir="ltr" >
  <head>
    <jdoc:include type="head" />
    <link href="/joomla158/templates/montemplatejoomla/css/template.css"
          rel="stylesheet" type="text/css" />
  </head>
  <body>
    <div id="Partie1">Partie entête du site<br /><br />
      <div id="Position1">
        <jdoc:include type="module" name="header" />
      </div>
      <div id="Position2">
        <jdoc:include type="module" name="search" />
      </div>
    </div>
    <div id="Partie2">Partie principale du site<br /><br />
      <div id="Position3">
        <jdoc:include type="module" name="breadcrumbs" />
      </div>
      <div id="Position4">
        <jdoc:include type="module" name="left" />
      </div>
      <div id="Position5">
        <jdoc:include type="component" />
      </div>
      <div id="Position6">
        <jdoc:include type="module" name="right" />
      </div>
    </div>
    <div id="Partie3"> Pied de page du site<br /><br />
      <div id="Section7">
        <jdoc:include type="module" name="footer" />
      </div>
    </div>
  </body>
</html>

```

Listing 3. Liste des propriétés et des méthodes de l'objet \$this

```

$this->template : contient le nom de la template en cours.
$this->getType() : retourne le type du document (error, feed, html, pdf, raw).
$this->getHeadData() : retourne un tableau contenant les entêtes HTML.
$this->getMetaData(name, http_equiv = false) : retourne la valeur d'une balise meta.
$this->getCharset() : retourne le jeu de caractères du document.
$this->getLanguage() : retourne le language du document.
$this->getDirection() : retourne le sens de lecture du document.
$this->getTitle() : retourne le titre du document.
$this->getBase() : retourne l'adresse de base du document.
$this->getDescription() : retourne la description du document.
$this->getLink() : retourne l'adresse complète du document.
$this->getGenerator() : retourne la chaîne d'identification du générateur du document.
$this->getModifiedDate() : retourne la date de dernière modification du document
$this->countModules() : retourne le nombre de modules affectés à un ou plusieurs
                      emplacements donnés.
$this->params->get("theme") : retourne la valeur du paramètre appelé theme.

```

Grâce aux codes de condition PHP, on peut également gérer plusieurs types de présentation sur un template, c'est-à-dire qu'un même template peut contenir à la fois la page d'accueil et une page de contenu type.

À la place des balises JDOC, le CMS Joomla! génère en sortie, du code HTML qui s'ajoute au code HTML du template. Ce code HTML généré contient des styles CSS propres à Joomla!. Il est indispensable alors d'adapter ces styles à la charte graphique définie en amont. L'intégration du code PHP transformera le fichier *index.html* en fichier *index.php*.

Composition d'un template

Nous allons voir de quoi est constitué un template Joomla! 1.5. Pour cela, il faut se situer dans le dossier template à la racine du CMS Joomla!. On remarque plusieurs répertoires qui correspondent chacun à un template. Il suffit d'aller voir dans un répertoire existant, par exemple *rhuk_milkyway*.

```

templates/[NomDuTemplate]/
templates/[NomDuTemplate]/css/
templates/[NomDuTemplate]/images/

```

Le nom du template ne doit pas comporter d'accents, d'espaces ni de caractères spéciaux. Car cela peut provoquer des erreurs pendant l'installation du template. Vous trouverez ci-dessous la description des fichiers trouvés dans ce répertoire (voir Figure 1). Il faut noter que certains templates peuvent contenir plus de fichiers et de répertoires :

- Un fichier *index.php*, qui contient la structure du site c'est-à-dire l'organisation spatiale des contenus, menus, fonctionnalités, etc...
- Un fichier *templateDetails.xml*, qui est utilisé pour l'installation du template et aux paramètres (pour les templates paramétrables). Il sert également à l'identifier pour le retrouver dans la liste des templates utilisables, côté administration.
- Un répertoire *images*, dans lequel on trouve les images utilisées par le template (fonds, bordures, logo).
- Un répertoire *css* dans lequel sont rangées la ou les feuilles de styles du template.
- Un fichier *favicon.ico* qui sert à définir l'icône du site, et un fichier *template_thumbnail.png* qui sert d'image de prévisualisation du template dans l'administration.
- Un dossier *html*, qui peut contenir plusieurs sous dossiers. Il s'agit d'une nouvelle fonctionnalité de Joomla! 1.5 qui propose le concept de *view* permettant l'adaptation des données HTML en sortie sans modification des données de départ. Ainsi on peut changer l'apparence graphique des composants et modules de Joomla! sans en impacter les fichiers sources.

Listing 4. Code généré par le style table

```
<table cellpadding="0" cellspacing="0"
  class="moduledtable_menu">
  <tr>
    <th valign="top">Main Menu</th>
  </tr>
  <tr>
    <td>
      <ul class="menu">
        <li><!-- various menu items --></li>
      </ul>
    </td>
  </tr>
</table>
```

Listing 5. Code généré par le style rounded

```
<div class="module_menu">
  <div>
    <div>
      <div>
        <h3>Main Menu</h3>
        <ul class="menu">
          <li><!-- various menu items --></li>
        </ul>
      </div>
    </div>
  </div>
```

- Un fichier *index.html* pour interdire l'accès direct et le listing de ce répertoire.

En détaillant chaque point, nous allons créer une première structure de site avec notre propre template Joomla!.

Structure du site

Nous considérons que nous avons une structure de site simple. Afin de mieux structurer les données dynamiques qui s'afficheront sur le site, il faut définir les différentes positions des données dans une matrice (voir Figure 2). Cette structure nous donne le code HTML visible dans le Listing 2.

Nous allons créer une arborescence qui nous servira d'espace de travail pour créer notre template Joomla!. Dans le répertoire templates de Joomla!, nous allons créer un répertoire appelé *montemplatejoomla*, il s'agit du nom de notre template. A l'intérieur de ce répertoire nous allons créer un sous répertoire css et images. Dans le répertoire css, nous créons un fichier *template.css* que nous laissons vide pour l'instant. Le répertoire images sera vide dans notre exemple mais libre à vous d'y insérer des images et logos pour illustrer votre site. Puis, nous copions le contenu du code HTML présenté dans le Listing 2 dans un fichier *index.php* que nous créons dans le répertoire *montemplatejoomla*.

Création du fichier index.php

Une fois le fichier *index.php* créé, il contient déjà du code HTML et nous allons y ajouter

Listing 6. Code généré par le style horz

```
<table cellspacing="1" cellpadding="0" border="0" width="100%">
  <tr>
    <td valign="top">
      <table cellpadding="0" cellspacing="0" class="moduledtable_menu">
        <tr>
          <th valign="top">Main Menu</th>
        </tr>
        <tr>
          <td>
            <ul class="menu">
              <li><!-- various menu items --></li>
            </ul>
          </td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```

Listing 7. Code généré par le style xhtml

```
<div class="moduledtable_menu">
  <h3>Main Menu</h3>
  <ul class="menu">
    <li><!-- various menu items --></li>
  </ul>
</div>
```

Listing 8. Code généré par le style none

```
<ul class="menu">
  <li><!-- various menu items --></li>
</ul>
```

le code PHP propre à Joomla! pour la transformation en *Template Joomla!*. Dans un premier temps et par mesure de sécurité, il faut protéger l'accès direct au fichier *index.php* en intégrant le code PHP suivant à la 1ere ligne du fichier :

```
// Accès direct interdit
<?php defined( '_JEXEC' ) or die(
  'Restricted access' ); ?>
```

Maintenant intéressons nous à la partie *HEAD* du fichier, cette partie est générée dynamiquement par le CMS Joomla! et les paramètres sont modifiables dans l'interface d'administration de Joomla!. Pour ce faire, Joomla! dispose de plusieurs balises JDOC de type différent. Dans notre cas il suffit de placer un JDOC de type head afin que tout le contenu des *HEAD* soit généré automatiquement. Cette balise ne doit apparaître qu'une seule fois dans le code source. Ci-dessous le code à insérer :

```
// Insertion des métadonnées du site
<jdoc:include type="head" />
```

Le contenu principal est la zone où s'afficheront les composants et les articles du site. En effet, un article est affiché à l'aide du composant *com_content*. Pour permettre l'affichage du contenu principal, on utilise le code suivant qui doit être unique dans la page :

```
// Insertion du contenu principal
<jdoc:include type="component" />
```

Ensuite, nous allons insérer les JDOC de type *module* qui permettent d'afficher les modules Joomla! dans les différentes positions du site. Ce type de JDOC contient plusieurs subtilités que nous allons voir en détail.

Tout d'abord, il est possible de spécifier à la balise JDOC un module ou plusieurs modules dans une position. Dans le cas du *fil d'Ariane*, nous savons qu'il s'agit uniquement de l'appel d'un seul module. Par conséquent, on peut remplacer le texte temporaire à l'intérieur des balises *<div id="position3">* par le code suivant :

```
// Insertion du fil d'ariane
<jdoc:include type="module"
  name="breadcrumbs" />
```

Dans cette balise JDOC, le type est *module* au singulier ce qui signifie qu'un seul module sera appelé. La propriété *name* permet d'appeler le module qui est affecté à la position *breadcrumbs*. L'affectation d'une position à un module est définie dans l'administration de Joomla!, dans extension puis gestion des modules. Le nom des positions est défini dans le fichier *templatedetails.xml*. Par défaut Joomla! en compte un certain nombre :

Listing 9. Corps de la fonction modChrome

```
<?php
    function modChrome_monStyle( $module, &$params, &$attribs ) {
        /* Code pour la sortie d'affichage du module */
    }
?>
```

Listing 10. Balise JDOC de type module avec utilisation d'attribut personnalisé

```
<jdoc:include type="modules" name="user1" style="monStyle" couleur="red" />
```

Listing 11. Code source complet de la fonction modChrome

```
<?php
    function modChrome_monStyle( $module, &$params, &$attribs ) {
        // Test si l'attribut couleur existe, sinon on affecte le bleu
        if (isset( $attribs['couleur'] )) {
            $background = $attribs['couleur'];
        } else {
            $background = 'blue';
        }
        //Affichage des balises div et récupération du suffixe de classe
        echo '<div class="' . $params->get( 'moduleclass_sfx' ) . '">';
        //Test si on autorise l'affichage du titre du module
        if ($module->showtitle) {
            echo '<h3>' . $module->title . '</h3>';
        }
        // On utilise la variable background contenant une couleur pour
        // définir le nom de la classe css à utiliser
        echo '<div class="' . $background . '">';
        echo $module->content;
        echo '</div>';
        echo '</div>';
    }
?>
```

- **left** : à gauche du contenu principal `main`,
- **right** : à droite du contenu principal `main`,
- **top** : en haut du contenu principal `main` (module de news, ou parfois menus horizontaux),
- **bottom** : en bas du contenu principal `main`,
- **inset** : modules placés dans le secteur d'insertion,
- **banner** : bannière,
- **header** : entête de la page,
- **footer** : pied de page,
- **newsflash** : affichage des *Newsflashes*,
- **legals** : informations légales,

Listing 12: Le code source généré par le style `monStyle`

```
<div>
    <h3><!--Titre du module --></h3>
    <div class="red">
        <!-- Contenu du module -->
    </div>
</div>
```

- **breadcrumb** : pour publier le module `breadcrumb` (chemin de navigation ou fil d'ariane),
- **menu** : menu. `navigation`,
- **user1** : position libre par défaut : *module Latest News*,
- **user2** : position libre par défaut : *module Popular*,
- **user3** : position libre par défaut : *module Top Menu*,
- **user4** : position libre par défaut : *module Search*,
- **user5** : position libre,
- **user6** : position libre,
- **user7** : position libre,
- **user8** : position libre,
- **user9** : position libre,
- **advert1** : pour les publicités. Certains les utilisent pour les bannières,
- **advert2** : idem,
- **advert3** : idem,
- **icon** : position certainement destinée à des icônes,

- **debug** : emplacement destiné aux informations renvoyées par le débuggeur pour la mise en place du template. Cette position est absente dans de nombreux templates. Il est conseillé de le placer le plus bas possible de la page afin de perturber le moins possible l'affichage graphique.
- **syndicate** : dans le template `rhuk_milkyway`, position utilisée pour afficher le flux RSS.

Plus 15 positions dont le nom peut être choisi par le concepteur du template. Il est aussi possible de placer plusieurs `modules` à la suite avec une seule balise JDOC. En effet, il suffit d'utiliser le type `modules` au pluriel. Prendons le cas de la position 6 de notre structure, il s'agit du contenu de la colonne de droite où plusieurs modules peuvent être affichés l'un en dessous de l'autre. Ainsi tous les modules qui auront la position `right` seront affichés dans la colonne de droite. Le code ci-dessous est à placer entre la balise `<div id="position6">`:

```
// Insertion des modules de la colonne de
// droite
<jdoc:include type="modules" name="right" />
```

Après avoir placé toutes les balises JDOC dans le template nous obtenons un résultat similaire au Listing 4. Pour des besoins spécifiques, il est possible d'ajouter du code PHP. En effet, il existe un objet global nommé `$this`. Vous trouverez en listing 5, la liste des propriétés et des méthodes disponibles

Personnalisation des modules

Il existe un paramètre intéressant associé au type `module` ou `modules`. Il s'agit du style qui permet de personnaliser l'affichage d'un module. En effet lorsqu'un module est chargé avec le template, il est encapsulé dans du code HTML qui permet de donner un aspect graphique particulier autour du module, par exemple un encadré de couleur noir. Par défaut, Joomla! propose 6 styles différents, par exemple :

```
<jdoc:include type="modules"
    name="right" style="table" />
```

Le style `table` est celui par défaut, c'est-à-dire que si le paramètre `style` n'est pas défini, c'est celui là qui sera pris en compte. Ce style génère un tableau HTML et les modules s'affichent l'un en dessous de l'autre de façon verticale. Le code généré est visible dans le Listing 6. La figure permet de se rendre compte du rendu graphique.

Le style `rounded` permet quant à lui d'afficher des coins arrondis. Le code généré est visible dans le Listing 7 et le rendu graphique est visible à la Figure 4. Le style `horz` permet

Listing 13. Liste de toutes les classes css spécifiques à Joomla!

```

/* CSS GLOBALE */
body {}
p {}
h1 {}
h2 {}
h3 {}
a:link, a:active, a:visited {}
a:hover {}

/* CSS SPECIFIQUE DE JOOMLA */
.article_separator {}
.adminform {}
.author {}
.bannerfooter {}
.bannergroup {}
.bannerheader {}
.banneritem {}

.blog_more {}
.blogsection {}
.breadcrumbs {}
.button {}
.buttonheading {}
.clr {}
.componentheading, .contentheading {}
.content_email {}
.content_rating {}
.content_vote {}
.contentdescription {}
.contentpagetitle {}
.contentpane {}
.contentpanopen {}
.contenttoc {}
.createdate {}
.created-date {}
.date {}
.input {}
.inputbox {}

.intro {}
.latestnews {}
.loclink {}
a.mainlevel:link, a.mainlevel:active, a.mainlevel:visited {}
a.mainlevel:hover {}
.message {}
.metadata {}
.modifydate {}
.module {}
.moduletable {}
.mosimage {}
.mosimage_caption {}
.mostread {}
.newsfeed {}
.outline {}
.pagenav {}
.pagenav_next {}
.pagenav_prev {}
.pagenavbar {}
.pagenavcounter {}
.pathway {}
.pollstableboarder {}
.read {}
.search {}
.searchintro {}
.sections {}
.sectiontable_footer {}
.sectiontableentry {}
.sectiontablefooter {}
.sectiontableheader {}
.small {}
.smalldark {}

a.sublevel:hover {}
.title {}
.wrapper {}

```

d'afficher les modules l'un à côté de l'autre de façon horizontale. Le Listing 8 donne une idée sur la méthode d'affichage grâce à l'encapsulation de plusieurs tableaux HTML. Le style `xhtml` affiche le module dans une balise `div` conforme aux normes XHTML. Le code minimaliste généré est visible dans le Listing 9.

Le style `none` affiche le module sans aucun code de formatage autour. Le Listing 10 montre bien que seul le code généré par le module lui-même est affiché. Il n'est donc pas possible d'encadrer le module avec une classe `css` ou un tableau (voir Figure 5). Le style `outline` est seulement utilisé pendant le développement car il permet de pré-visualiser la position des modules (voir Figure 6). Il y a aussi une autre méthode pour afficher la position des modules de l'ensemble du site en tapant dans l'url : `index.php?tp=1` (voir Figure 7).

Finalement, si aucun style ne vous convient, vous pouvez toujours créer votre propre style. Joomla! propose une fonction appelée `modchrome` qui permet de programmer soit même le style de formatage que l'on souhaite donner pour chaque module. Cette fonction doit se trouver dans `templates/montemplatejoomla/html/modules.php`.

Dans notre template, nous devons créer un sous-répertoire `html` et créer le fichier `modules.php` vide pour l'instant. Ce fichier doit obligatoirement contenir la fonction suivante. Il faut noter que le nom de la fonction est composé du mot `modChrome_` suivi du nom du style personnalisé que l'on souhaite donner. Dans le fichier `index.php`, c'est ce nom qu'il faudra spécifier au paramètre `style` du JDOC de type module ou modules.

Par exemple :

```
<jdoc:include type="modules"
  name="right" style="monStyle" />
```

Cette fonction prend 3 arguments qui sont automatiquement gérés par le framework Joomla!. Parmi ces arguments, on retrouve l'objet module qui contient notamment le titre et le contenu du module, l'objet `params` qui contient les paramètres du module et ceux définis dans l'administration, des attributs qu'on peut ajouter en complément dans la balise JDOC.

`modChrome`, il faut éditer le code qui servira à l'affichage du formatage. Prenons par exemple le code du Listing 13. Ce code permet d'afficher le module dans des balises `div` et de

personnaliser la classe `css` qu'on définit au préalable dans la balise JDOC. Le code généré par la fonction est visible dans le Listing 14.

La feuille de style css

La `css` est contenue dans le répertoire `css` du template. Pour respecter la norme de Joomla!, il faut créer un fichier appelé `template.css` dans ce répertoire. Ce fichier `template.css` devra contenir toutes les classes spécifiques à Joomla!. En effet, lorsque les modules et les composants génèrent l'affichage, le code source intègre des classes bien spécifiques qu'on peut personnaliser à sa guise. Ces classes sont utilisées pour gérer l'aspect graphique des boutons, des champs, des libellés, des titres, des tableaux, de certaines icônes, etc... Ces classes spécifiques sont listées dans le Listing 15.

Le fichier `template.css` devient rapidement très volumineux dès que l'on souhaite personnaliser un peu son template. Il est donc conseillé d'utiliser un autre fichier comme `layout.css` pour stocker les classes provenant de la structure du template. Et un autre fichier comme `typographie.css` pour stocker les classes définissant la charte éditoriale et la typographie.

Listing 14. Exemple de fichier templatedetails.xml

```

<!-- Déclaration du fichier xml -->
<?xml version="1.0" encoding="utf8"?>
<!-- Version de Joomla compatible
     et le type d'extension -->
<install version="1.5" type="template">
    <!-- Nom du template
    <name>greenced</name>
    <!-- Version du template
    <version>1.0.0</version>
    <!-- Date de création du template
    <creationDate>12/11/08</creationDate>
    <!-- Nom de l'auteur
    <author>Panya Sayavongsa</author>
    <!-- Adresse mail de l'auteur
    <authorEmail>panya@sayavongsa.com</authorEmail>
    <!-- Site de l'auteur
    <authorUrl>www.monsite.com</authorUrl>
    <!-- Message de copyright -->
    <copyright></copyright>
    <!-- Type de licence
    <license>GNU/GPL</license>
    <!-- Description du template visible dans
         l'administration
    <description>Description de mon template</description>
    <!-- Liste et chemin des fichiers contenu dans le template
    <files>
        <folder>images</folder>
        <folder>css</folder>
        <filename>index.php</filename>
        <filename>templateDetails.xml</filename>
        <filename>template_thumbnail.png</filename>
    </files>
    <!-- Liste des positions pour les modules du template
    <positions>
        <position>user1</position>
        <position>user2</position>
        <position>user3</position>
        <position>left</position>
        <position>right</position>
        <position>footer</position>
    </positions>
</install>

```

Listing 15. Code complémentaire pour l'ajout d'un paramètre theme dans le template

```

<param name="theme" type="list" default="rouge"
    label="Theme" description="Theme couleur du template">
    <option value="rouge">theme rouge</option>
</param>

```

Listing 16. Code permettant d'appeler la css en fonction du thème sélectionné.

```

<link href="/joomla158/templates/montemplatejoomla/css/
    <?php echo $this->params->get("theme"); ?>.css"
    rel="stylesheet" type="text/css" />

```

Main Menu

- [Home](#)
- [Joomla! Overview](#)
- [Joomla! License](#)
- [More about Joomla!](#)
- [FAQ](#)
- [The News](#)
- [Web Links](#)
- [News Feeds](#)

Figure 3. Rendu graphique du module généré par le style table**Les images**

C'est dans le sous-répertoire *templates/montemplatejoomla/images* que vous pouvez réunir les fichiers graphiques référencés par votre template. L'installateur copiera ces fichiers dans le sous-dossier images.

Il est possible d'ajouter un *favicon*, c'est-à-dire une icône qui sera affichée dans la barre d'url ou dans la liste des liens favoris du navigateur. Le fichier *favicon.ico* doit être copié à la racine du template. Puis dans l'administration du site Joomla!, il est possible de préciser le nom du fichier *favicon*. Cependant, il n'est pas nécessaire de définir le *favicon* dans le fichier *index.php* du template, car la balise JDOC de type *head* s'occupe de générer automatiquement le code qu'il faut. Pour pré-visualiser le template dans l'administration, on peut ajouter un fichier *template_thumbnail.png* à la racine du template Joomla!. Celui-ci sera lu par le système de prévisualisation des templates (voir Figure 9).

Personnalisation des composants et modules

Pour personnaliser le code HTML en sortie d'affichage des modules et des composants sans modifier les codes sources d'origine, Joomla! propose un système qui consiste à externaliser dans le template, le code HTML en sortie. Par exemple le composant content, qui permet d'afficher les articles, génère toujours des tableaux HTML. Cette méthode

Main Menu

- [Home](#)
- [Joomla! Overview](#)
- [Joomla! License](#)
- [More about Joomla!](#)
- [FAQ](#)
- [The News](#)
- [Web Links](#)
- [News Feeds](#)

Figure 4. Rendu graphique du module généré par le style rounded



Figure 5. Rendu graphique du module généré par le style none

tient du code source de son ancêtre Mambo, le CMS à l'origine de Joomla!.

Pour résoudre le problème de génération des tableaux HTML en sortie, il faudrait modifier le code source d'origine de Joomla! directement dans le répertoire *components/com_content*. Le problème serait dans ce cas résolu. Cependant, ces modifications auront une forte probabilité d'être écrasées à la prochaine mise à jour de Joomla!.

Pour résoudre cette problématique, Joomla! version 1.5 propose le concept de vue (*View*) issu du modèle MVC (*Model View Controller*). C'est-à-dire que pour chaque composant est associé un sous répertoire supplémentaire appelé *views* dans lequel se trouvent les différentes vues destinées à la présentation des données du composant ou du module. Dans le composant *com_content*, il existe 5 sous répertoires appelés *archive*, *article*, *category*, *frontpage* et *section*. Chacun d'eux contient des fichiers qui définissent le formatage et l'apparence visuelle du contenu. Chacun des 5 sous répertoires possède un répertoire appelé *tmpl* (voir Figure 8). Ce répertoire contient plusieurs fichiers chargés de contrôler la présentation.

Dans notre cas, il faut donc créer un répertoire *html* à la racine du template *montemplatejoomla* et créer un sous répertoire appelé *com_content* (ou le nom d'un autre composant ou module dont on souhaite en modifier l'affichage).

Ensuite comme base on peut récupérer les vues d'origines du composant dans le répertoire *components/com_content/view/article tmpl*. Dans ce répertoire, il suffit de copier l'ensemble des sous répertoires vers le nouveau répertoire que



Figure 6. Rendu graphique du module avec le style outline

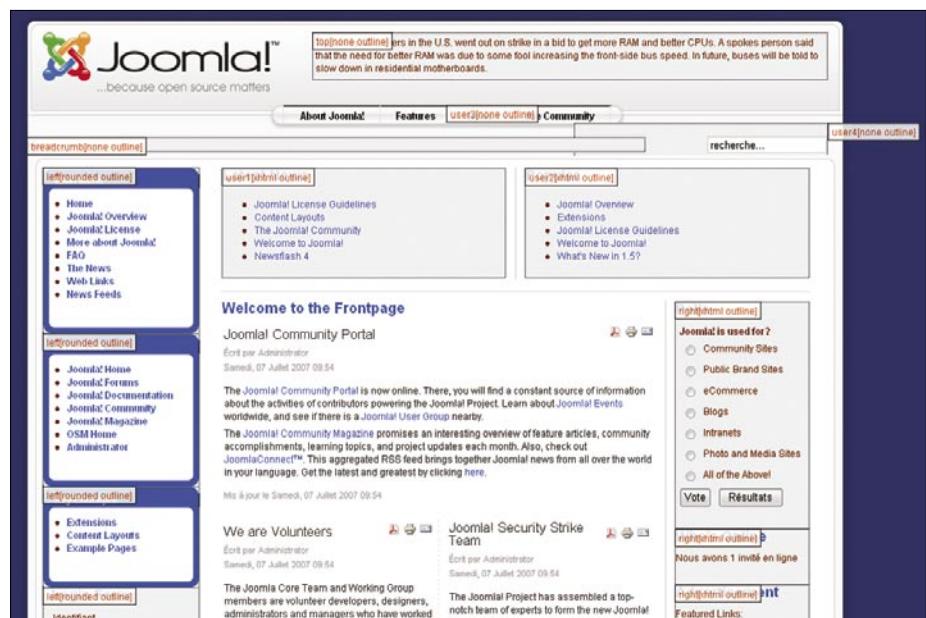


Figure 7. Affichage de la position des modules

nous venons de créer. Puis nous allons éditer les fichiers du sous répertoire article.

En modifiant le code source des fichiers du sous répertoire article situé dans le template, Joomla! va faire automatiquement une surcharge (*override*). C'est-à-dire que si des fichiers vus existent dans le template, le framework Joomla! utilisera ceux là plutôt que ceux du noyau.

Paquetage d'un template pour l'installation

Joomla! propose un système d'installation automatique pour les templates. Afin de rendre pos-

sible l'installation du template à partir de l'interface d'administration, il faut le préparer pour l'installation et le compresser dans un fichier *zip*.

Dans un premier temps, nous allons créer l'image de prévisualisation du template qui doit se nommer *template_thumbnail.png* et être placée à la racine du template. Le fichier doit respecter les dimensions 227x162px. L'image de prévisualisation s'affiche lorsqu'on passe la souris sur la ligne correspondante au template à activer (voir Figure 9). Si vous ne disposez pas encore d'une capture écran du template, vous pouvez créer une image blanche en attendant et refaire une capture du template lorsqu'il sera

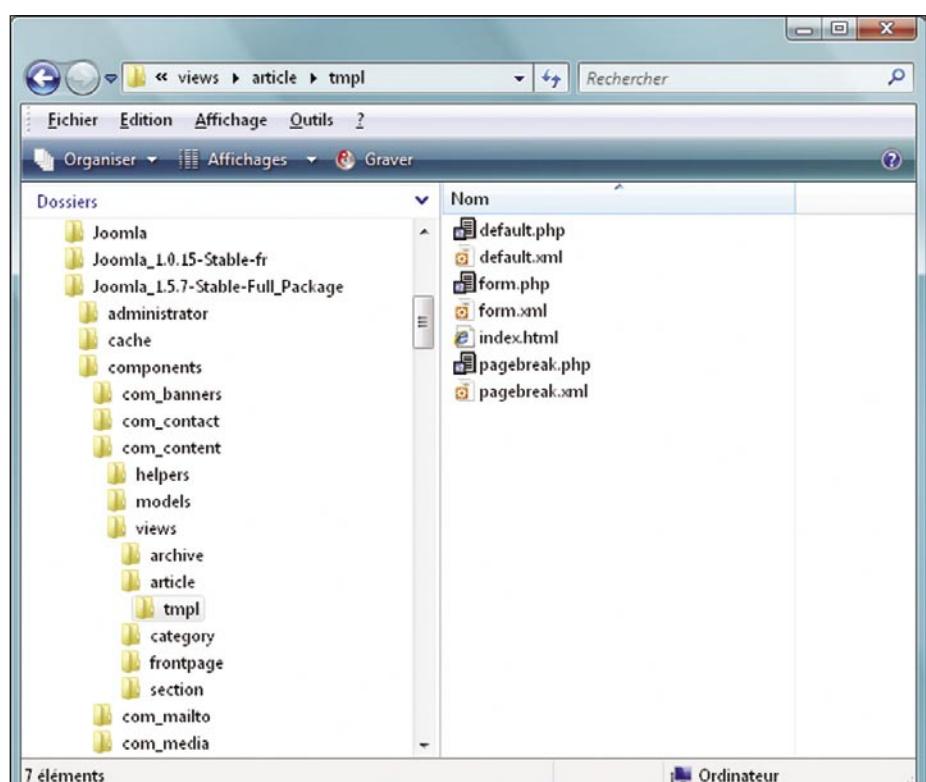


Figure 8. Arborescence du répertoire views

The screenshot shows the Joomla! Template Manager. At the top, there are buttons for 'Default', 'Edit', and 'Help'. Below is a table with columns: #, Template Name, Default, Assigned, Version, Date, and Author. The table lists four templates: 'beez', 'JA_Purity', 'panfolio', and 'thuk_milkyway'. The 'thuk_milkyway' row is highlighted. A preview window on the right shows a sample Joomla! site using the 'thuk_milkyway' template. The status bar at the bottom says 'Joomla is Free Software released under the GNU/GPL License.'

Figure 9. Gestion des templates dans la partie administration de Joomla!

installé sur le site puis remettre à jour l'image de prévisualisation.

Ensuite, il est indispensable de créer le fichier `templatedetails.xml` qui contient les instructions interprétées par Joomla! durant l'installation du template. Ce fichier contient les informations sur le template, la liste des fichiers et la liste des positions. Après avoir compressé tout le contenu du template dans un fichier `zip` portant le même nom, il est maintenant prêt à être installé sur votre site Joomla!.

Installation d'un template

Une fois le paquetage du template réalisé, le CMS Joomla! propose 2 façons de l'installer. La première consiste à copier les fichiers dans un répertoire `tmp` situé à la racine de Joomla!. D'aller dans l'interface d'administration (menu `Extensions > Installation`, voir Figure 10). Dans la partie `Install from Directory`, il suffit de cliquer sur le bouton `install` pour que l'installateur de Joomla! fasse son travail et exécute l'installation du template dans le bon répertoire.

La seconde méthode est beaucoup plus simple et représente un des avantages majeurs de Joomla!. Il s'agit de l'installation avec le fichier `zip` du template que nous avons créé avec le package.

Toujours sur la page d'installation des extensions, dans la partie `Upload Package File`, il suffit de cliquer sur parcourir et d'aller chercher le template archivé au format `zip`. Ensuite on clique sur `upload file and install`. Grâce au fichier `XML` `templatedetails.xml`, le système d'installation de Joomla! va reconnaître le type d'extension qui dans notre cas est un template et va automatiquement lancer la procédure d'installation adaptée.

Lorsque le message de confirmation apparaît, nous pouvons aller vérifier le bon fonctionnement du template en allant sur le menu `Extensions > Gestion des templates` (voir Figure 9). Pour l'activer, il faut sélectionner le template voulu et cliquer sur défaut représenté par une icône en forme d'étoile, il est alors visible sur l'url de votre site.

Paramétrage d'un template

Il est possible d'aller encore plus loin dans la personnalisation du template en lui donnant

- `name` : doit contenir le nom du paramètre tel que vous l'appellerez depuis votre Template (sans espace ni caractères spéciaux).
- `type` : le type de champ qui sera proposé à l'utilisateur pour saisir ou sélectionner l'information.

Les types de paramètre sont les mêmes pour les Templates et les Modules, à savoir :

- `category` : liste déroulante dynamique qui contient les catégories du site et retourne leur identifiant. Nécessite un attribut additionnel `section="nomDeLaSection"` qui indique dans quelle section on liste les catégories.
- `imagelist` : liste des images situées dans le dossier spécifié dans l'attribut `directory="chemin/relatif"`. Nécessite donc un nouvel attribut `directory`.
- `list` : liste déroulante. Nécessite un jeu de balises `<option>` en complément.
- `menuitem` : liste déroulante dynamique qui contient l'arborescence des menus du site et retourne leurs identifiants.
- `mos_menu` : liste déroulante dynamique qui contient les différents modules de menu du site.
- `radio` : boutons radio. Nécessite un jeu de balises `<option>` décrites plus bas.
- `spacer` : ligne de séparation entre deux groupes de paramètres.
- `text` : champ texte libre.
- `textarea` : champ de texte multi-lignes. Nécessite les attributs `cols` et `rows` qui, comme dans une balise HTML `<textarea>` indiquent le nombre de lignes et de colonnes de ce champ.

Une fois le paramètre défini, nous allons l'appeler dans le fichier `index.php` afin de pouvoir changer la css dynamiquement. Sachant que nous disposons déjà d'une css appelée `template.css`, nous allons créer des fichiers css complémentaires qui définiront les couleurs. Ces fichiers porteront le nom de `rouge.css` et `vert.css`. Bien entendu, il ne faut pas définir les couleurs dans le fichier `template.css` pour que cette méthode fonctionne.

Dans le fichier `index.php` et dans la balise

The screenshot shows the Joomla! Extension Manager. At the top, there are tabs for 'Install', 'Components', 'Modules', 'Plugins', 'Languages', and 'Templates'. The 'Install' tab is selected. Below is a section for 'Upload Package File' with a 'Package File:' input field, a 'Parcourir...' button, and a 'Upload File & Install' button. There are also sections for 'Install from Directory' (with 'Install Directory:' input field and 'Install' button) and 'Install from URL' (with 'Install URL:' input field and 'Install' button). The status bar at the bottom says 'Joomla is Free Software released under the GNU/GPL License.'

Figure 10. Page d'installation des extensions



Figure 11. Interface de paramétrage du template

HEAD, nous allons donc ajouter le code suivant dans la balise HEAD.

Maintenant, il ne manque plus qu'à tester le bon fonctionnement du thème en allant dans l'administration puis dans le menu *extensions > template manager*. Sélectionnez votre template et cliquez sur l'icône *edit*. Vous devrez voir apparaître dans la zone parameters à droite le paramètre theme que nous venons de créer (voir Figure 11). En sélectionnant le thème vous affectez automatiquement la valeur de la variable theme définie dans le template. Le principe du template club est de proposer tous les mois un ou deux nouveaux templates. En contrepartie, vous payez un abonnement pour accéder aux templates déjà réalisés et ceux à venir.

Conclusion

La création d'un template Joomla! est une opération délicate qui peut devenir très complexe. La complexité réside surtout dans l'intervention de plusieurs compétences durant les différentes étapes de création. Puis, tout dépend de la finition et de la qualité que l'on souhaite donner au résultat final. Il faut également savoir qu'un template Joomla! est loin de donner un site finalisé. En effet, le paramétrage, l'organisation du contenu, la création des types de page, l'intégration et la mise en page des articles, ainsi que l'ajout de modules et les composants supplémentaires impliquent nécessairement des adaptations graphiques et/ou techniques supplémentaires. Malgré ces différentes adaptations, la création d'un template Joomla! reste un moyen simple et rapide de mettre un site en ligne tout en disposant d'un design personnalisé et des fonctionnalités du CMS Joomla!.

PANYA SAYAVONGSA

De formation ingénieur en informatique, Panya Sayavongsa est consultant indépendant dans les nouvelles technologies du web. Il intervient plus particulièrement dans l'intégration des systèmes de gestion de contenu comme le CMS Joomla! mais aussi dans les problématiques d'intégration de site web 2.0. Pour plus d'informations, vous pouvez consulter son site : www.panfolio.com ou le contacter sur contact@panfolio.com.

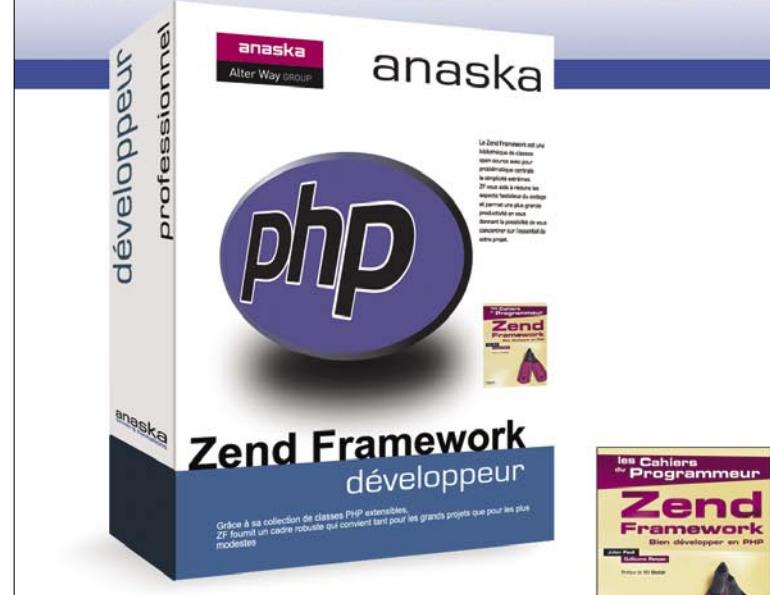
Sur Internet

- www.Joomla!france.org – Site francophone officiel de Joomla!,
- www.aide-Joomla!.com – Site français contenant plusieurs tutoriels Joomla!,
- www.Joomla!rt.com – Template club pour CMS Joomla!,
- www.rockettheme.com – Template club pour CMS Joomla!,
- www.Joomlajunkie.com – Template club pour CMS Joomla!,
- www.shape5.com – Template club pour CMS Joomla!,
- www.templateplazza.com – Template club pour CMS Joomla!,
- www.gavickpro.com – Template club pour CMS Joomla!,
- www.Yootheme.com – Template club pour CMS Joomla!,
- www.Joomlacssguide.com – Référence toutes des classes de css pour template Joomla!.



PHP > Framework

Formation au Zend Framework



"Une formation mise en place par l'auteur du livre Zend Framework"



Au Programme des 4 jours :

- * Introduction au Zend framework
- * Installation et configuration
- * La Programmation Orientée Objet
- * Composants du noyau
- * Gestion des bases de données
- * Modèle Vue Contrôleur (MVC)
- * Sécurisation et gesticions de droits
- * Fonctionnalités avancées du Zend framework

Prochaines sessions

Paris 2009
19/01, 09/02, 09/03
06/04, 04/05, 02/06

Lyon 2009
09/02, 09/03, 06/04
04/05, 29/06

Tarif
1700 € HT

LE SPECIALISTE DE LA FORMATION POUR L'OPEN SOURCE

Informations
01 45 28 09 82
www.anaska.com

anaska
Alter Way GROUP

AUTHORIZED
MySQL
EDUCATION CENTRE

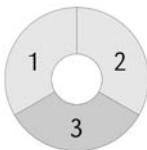
Barre d'avancement de téléchargement en PHP

Depuis l'arrivée de la version 5.2 de PHP, il est désormais possible, à l'aide de quelques extensions, de mettre en place une barre d'avancement de téléchargement. Vous allez voir comment dans cet article.

Cet article explique :

- Le principe de fonctionnement d'une barre d'avancement de téléchargement en PHP.
- La configuration serveur requise pour une telle utilisation.
- Le fonctionnement du module Zend ProgressBar.
- Quelques bases du framework Javascript Moo-tools pour le traitement côté client.

Niveau de difficulté



L'envoi de fichier sur un serveur par l'intermédiaire de PHP a toujours été une tâche pénible et hasardeuse pour l'utilisateur : pas de trace de l'avancement de la tâche, aucune idée du débit auquel le fichier est envoyé, et encore moins du temps restant... L'utilisateur finit par s'impatienter ou, pire, annuler l'envoi. Heureusement, depuis la version 5.2 de PHP, il est désormais possible, et même relativement simple, de mettre en place une barre d'avancement de téléchargement utilisant uniquement PHP côté serveur.

Méthodologie

Afin de concevoir une barre d'avancement de téléchargement, nous allons avoir besoin de plusieurs éléments :

- un formulaire d'envoi de fichier HTML,
- un script PHP pour recevoir notre fichier,
- un script PHP pour suivre l'évolution du téléchargement, et tenir informé l'utilisateur,
- un script JavaScript pour récupérer les informations sur le téléchargement et actualiser la barre d'avancement.

Ce qu'il faut savoir :

- Être familier avec l'envoi de fichier en PHP.
- Savoir manipuler le DOM en JavaScript.
- Être à l'aise avec la POO.
- Savoir configurer son environnement de développement.

Nous allons voir ensemble les extensions dont nous avons besoin pour concevoir cette barre d'avancement.

Extensions PHP

Une des extensions PHP nous permettant de récupérer les informations sur l'avancement d'un envoi de fichier, et celle que nous allons utiliser dans cet article, est l'extension *PHP APC*. Il s'agit à l'origine d'une extension utilisée pour la mise en cache de données. PHP 5.2 ayant apporté le support de la *RFC 1867*, à savoir la norme régissant l'envoi de fichier via le protocole *HTTP*, l'extension *APC* permet depuis la récupération d'un tableau d'information sur un téléchargement en cours.

Afin de bénéficier de l'extension *APC*, il va falloir l'installer sur votre système et modifier le fichier *php.ini* pour que les lignes du Listing 1 y soient présentes. Pour vérifier que votre installation est correctement configurée, n'hésitez pas à utiliser *phpinfo()*. Si une section *APC* est présente, et que les valeurs *apc.enabled* et *apc.rfc1867* sont à *on*, alors votre environnement est prêt à être utilisé.

Extension *upload_progress*

Une autre extension PHP, *upload_progress*, permet de récupérer, de façon similaire à l'extension *APC*, les informations concernant un téléchargement en cours via la fonction *uploadprogress_get_info*. L'extension *APC* étant plus répandue, et semblant plus mature également, elle a été retenue pour cet article. Toutefois, pour un projet ayant besoin d'une barre d'avancement mais n'utilisant pas les fonctionnalités de cache *APC*, il serait peut-être plus logique d'utiliser une extension dédiée à ce seul usage. Ici encore, il est surtout question de préférences et de tests pour s'assurer de quelle solution correspond le mieux à nos besoins.

Technologies employées

Afin de concevoir notre barre d'avancement de téléchargement, nous n'avons besoin que de notre extension *APC*. Toutefois, à l'heure actuelle, aucun projet aurait intérêt à se passer d'un framework JavaScript solide dès qu'il s'agit de manipulation du DOM et de requête *AJAX*. Pour cette raison, nous utiliserons ici le framework *Mootools*, pour des raisons de préférences personnelles et également car il s'agit d'un framework dont on parle plus rarement que les plus répandus (*Prototype*, *JQuery...*), ce qui fera une bonne occasion de le découvrir.

Depuis la sortie de la version 1.7, le Zend Framework inclut le discret module *zend_progressBar*. Si ses méthodes sont totalement documentées dans la documentation officielle, il est très malaisé, à l'heure où sont écrites ces lignes, de trouver des exemples concrets de son utilisation. Nous allons donc utiliser ce module pour concevoir notre barre d'avancement, et vous vous rendrez vite compte que l'on aura sans grand effort quelques bases pour concevoir des barres d'avancement très facilement réutilisables, en découpant totalement le suivi de la barre d'avancement et son affichage.

Mootools

Mootools (pour *My Object-Oriented Tools*) est un framework Javascript entièrement modulaire. Pour tester cet exemple, vous pouvez (et il est même préférable) vous contenter de télécharger la version complète du framework plutôt que de sélectionner vous-même les modules

à intégrer. Je vous recommande, à des fins de test, de prendre la version non compressée afin de pouvoir au besoin regarder au cœur du code source. Étudier un à un les modules de Mootools n'entre pas dans le cadre de cet article, aussi je vous renvoie à la documentation officielle où vous trouverez tout ce qui vous sera nécessaire.

Zend_ProgressBar

Le module `zend_ProgressBar` est constitué de deux entités distinctes :

- une classe `Zend_ProgressBar` que l'on tiendra informée de l'évolution du téléchargement,
- une classe `Zend_ProgressBar_Adapter`, qui traite les informations fournies à l'objet `Zend_ProgressBar`.

Il y a pour le moment trois *adapters* fournis avec le Zend_Framework :

- `Zend_ProgressBar_Adapter_Console`, permettant de créer des barres d'avancement en ligne de commande, et qui sort totalement du cadre de l'article.
- `Zend_ProgressBar_Adapter_JsPush`, qui fait s'exécuter à chaque mise à jour une fonction de `callback` devant traiter l'information côté client. Nous allons utiliser cet adapter car il est le plus rapide à mettre en place, mais pas nécessairement le plus adapté à une barre d'avancement de téléchargement. Un usage intéressant pour `Zend_ProgressBar_Adapter_JsPush` serait, par exemple, de tenir informé de l'avancement de l'exécution d'un seul gros script PHP (par exemple un traitement d'image par lot).
- `Zend_ProgressBar_Adapter_JsPull` qui envoie au navigateur les informations de téléchargement au format `JSON`. Cet *adapter* est le plus adapté pour une barre d'avancement, mais sa mise en place un peu moins rapide. Un exercice intéressant consisterait à réécrire le code de cet article en utilisant `Zend_ProgressBar_Adapter_JsPull`.

Côté Serveur

Pour le développement de notre barre d'avancement côté serveur, nous allons avoir besoin de trois scripts PHP :

- un simple formulaire HTML que nous mettrons dans le fichier `index.php`,
- un script de traitement du fichier téléchargé, que nous appellerons `upload.php`,
- le script permettant de mettre à jour notre barre d'avancement, `progress.php`.

Le formulaire

Nous allons utiliser un formulaire HTML d'envoi de fichiers classique et sans surprise, que

Listing 1. Configuration du fichier `php.ini`

```
extension=php_apc.dll # ou apc.so sous système UNIX
apc.rfc1867 = on
```

Listing 2. Formulaire d'envoi (fichier `index.php`)

```
<?php
echo '<?xml version="1.0" encoding="UTF-8" ?>';
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" \
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Tutoriel Zend_ProgressBar</title>
<link href="/styles/style.css" media="screen" rel="stylesheet"
      type="text/css" />
<script type="text/javascript" src="/scripts/library/mootools.js">
</script>
<script type="text/javascript" src="/scripts/progress.js">
</script>
</head>
<body>
<form action="upload.php" method="post" id="form_upload" enctype="multipart/
form-data">
<label for="fichier">Fichier :</label>
<input type="hidden" name="APC_UPLOAD_PROGRESS" value="<?php echo
      sha1(uniqid(rand())); ?>" id="APC_UPLOAD_PROGRESS" />
<input type="file" name="fichier" id="fichier" />
<input type="submit" name="upload" id="upload" value="Envoyer" />
</form>
<div id="progress_bar" class="invisible">
<p>Envoi en cours : <span id="percent">0</span> %</p>
<div class="container">
  <div class="bg_red" id="jauge"></div>
</div>
<p>Temps écoulé : <span id="time_taken">0</span>
  seconde(s)</p>
<p>Temps restant : <span id="time_remaining">0</span>
  seconde(s)</p>
</div>
</body>
</html>
```

Listing 3. Feuille de style (fichier `styles/style.css`)

```
.invisible {
  display: none;
}

.container {
  width: 200px;
  height: 15px;
  border: 1px black solid;
}

.bg_red {
  background-color: red;
  width: 0%;
  height: 100%;
}
```

vous retrouverez dans le Listing 2. Une seule chose à retenir ici. Pour fonctionner, l'extension APC doit connaître l'identifiant unique de chaque téléchargement. La création d'un tel identifiant est à la charge du développeur, et il doit le préciser par l'intermédiaire d'un champ *hidden* nommé `APC_UPLOAD_PROGRESS`. Attention : ce champ doit impérativement être placé avant le champ *input file* concerné. Le nom de ce champ peut au besoin être modifié via la directive `apc.rfc1867_name` dans votre `php.ini`. Pour générer notre identifiant unique, nous utiliserons la fonction PHP `uniqid()`.

Le Listing 2 peut se passer de commentaire tant il est simple, on notera uniquement l'instruction suivante : `echo sha1(uniqid(rand()))`, qui nous permet de générer notre identifiant unique de téléchargement, après un hachage `sha1`. On aurait pu se dispenser du hachage, mais de ce fait, ça permet de brouiller

(un peu) les pistes sur le comment est généré cet identifiant, et ce n'est pas plus mal. Notre rôle maintenant va être d'envoyer cet identifiant au script chargé de nous donner les informations d'avancement du téléchargement. Nous en reparlerons quand nous traiterons du développement de la partie client. Vous trouverez dans le Listing 3 le (bref) contenu du fichier CSS, à titre indicatif.

Réception du fichier

Très peu de chose à dire ici. Nous nous contenterons du minimum syndical, tel que vous pouvez le constater dans le Listing 4 : nous mettons le fichier téléchargé dans un dossier *documents*, situé un niveau en dessous

Il va de soi que cette partie a été allégée pour les besoins de l'article, dont l'intérêt est ailleurs, mais qu'il ne saurait être utilisé tel quel dans un environnement de production...

Listing 4. Script de réception du fichier `upload.php`

```
<?php
move_uploaded_file($_FILES['fichier']['tmp_name'],
    '../documents/' . $_POST['APC_UPLOAD_PROGRESS']);
?>
```

Listing 5. Script de la barre d'avancement, utilisant Zend_ProgressBar (fichier `progress.php`)

```
<?php
ob_start();
set_include_path(get_include_path() . PATH_SEPARATOR . 'library/');
require_once 'Zend/Session/Namespace.php';
require_once 'Zend/ProgressBar.php';
require_once 'Zend/ProgressBar/Adapter/JsPush.php';
$upload_id = $_POST['upload_id'];
$download_info = apc_fetch('upload_' . $upload_id);
if ($download_info === false) {
    $bytes_total = 1;
    $bytes_uploaded = 0;
    $done = false;
} else {
    $bytes_total = $download_info['total'];
    $bytes_uploaded = $download_info['current'];
    $done = $download_info['done'];
}
$adapter = new Zend_ProgressBar_Adapter_JsPush();
$adapter->setFinishMethodName('Zend_ProgressBar_Finish');
$progress_bar = new Zend_ProgressBar($adapter, 0,
    $bytes_total, 'download_progress_'
    . $upload_id);
if ($done) {
    $progress_bar->finish();
    $ns_download_progress = new Zend_Session_Namespace
        ('download_progress_' . $upload_id);
    $ns_download_progress->unsetAll();
} else {
    $progress_bar->update($bytes_uploaded);
}
?>
```

La barre d'avancement

Voilà le cœur de notre composant côté serveur ! C'est ici que tout ou presque va se passer. Vous trouverez le code de ce script dans le Listing 5.

Tout d'abord, expliquons succinctement cette première ligne énigmatique, `ob_start()`. Nous la plaçons ici pour éviter que la mise à jour de notre barre d'avancement ne provoque une *notice* désagréable. Je vous invite à lire l'en-cadré intitulé *Zend_ProgressBar_Adapter_JsPush et bufferisation de sortie* pour plus de détail.

Le reste du code est, je pense, assez simple à déduire. On suppose que le dossier *Zend* du Zend Framework est situé dans le dossier *library/*, aussi nous l'incluons dans l'*include path*. Nous utilisons ici les classes *Zend_Session_Namespace*, *Zend_ProgressBar* et *Zend_ProgressBar_Adapter_JsPush* aussi nous appelons les directives `require_once` appropriées.

La fonction `apc_fetch()` permet de récupérer des données mises en cache avec l'extension APC. En ce qui nous concerne, il s'agit des données concernant le téléchargement ayant pour identifiant `$_POST['upload_id']` précédé par le préfixe `upload_`. Ce préfixe peut être modifié par la directive `apc.rfc1867_prefix` dans votre `php.ini`. Dans le cas qui nous concerne, la fonction `apc_fetch()` retourne `false` quand aucun téléchargement ayant cet identifiant n'est trouvé, ce qui peut arriver le temps que l'envoi soit bien pris en compte par le serveur, et un tableau dans le cas contraire. Le tableau propose les informations suivantes :

- `total` : nombre total d'octets du fichier en cours d'envoi,
- `current` : octets déjà téléchargés,
- `filename` : valeur du champ *input file*, soit le chemin complet du fichier,
- `name` : attribut *name* du champ *input file*,
- `done` : indique si l'envoi est terminé (1) ou non (0).

Les valeurs suivantes sont rajoutées une fois l'envoi terminé (`done == 1`) :

- `temp_filename` : chemin du fichier temporaire,
- `rate` : le débit moyen du transfert, en *bits* par seconde,
- `cancel_upload` : retourne le code d'erreur PHP survenu durant l'envoi. Si il vaut 0 (soit la constante `UPLOAD_ERR_OK`), tout s'est déroulé correctement.

Nous gérons ici le cas où `apc_fetch()` retourne `false`, car il est très probable que cela arrive lors des premiers appels. Dans ce cas nous spécifions manuellement une valeur pour les informations d'envoi, de sorte à refléter une progression de zéro pourcent.

Une fois obtenues les informations, nous créons notre *adapter* en appelant son constructeur,

Listing 6. Création de notre objet de requête AJAX (fichier *scripts/progress.js*)

```
var progressBarRequest = new Request({
  url: 'progress.php',
  evalScripts: true
});
```

Listing 7. Création des événements JavaScript (suite du fichier *scripts/progress.js*)

```
window.addEvent('domready', function() {
  var iFrameElement = new Element('iframe', {
    name: 'upload_iframe',
    'class':'invisible'
  });
  iFrameElement.inject(document.getElement('body'));
  $('form_upload').set('target', iFrameElement.get('name'));
  $('form_upload').addEvent('submit', function() {
    progressBarRequest.send('upload_id=' + $('APC_UPLOAD_PROGRESS').get('value'));
  });
});
```

Listing 8. Fonctions de callback appelées par Zend_ProgressBar (suite et fin du fichier *scripts/progress.js*)

```
function Zend_ProgressBar_Update(data) {
  $('progress_bar').setStyle('display', 'block');
  $('percent').set('html', data.percent.round());
  $('time_taken').set('html', data.timeTaken);
  $('time_remaining').set('html', data.timeRemaining);
  $('jauge').setStyle('width', data.percent + '%');
  progressBarRequest.send.delay(200, progressBarRequest, 'upload_id=' + $('APC_UPLOAD_PROGRESS').get('value'));
}

function Zend_ProgressBar_Finish() {
  $('percent').set('html', '100');
  $('jauge').setStyle('width', '100%');
  alert('Téléchargement terminé !');
}
```

et nous lui spécifions une méthode d'appel lorsque le téléchargement sera fini via l'instruction `$adapter->setFinishMethodName('Zend_ProgressBar_Finish')`. Nous aurions pu définir la méthode de mise à jour via la méthode `setUpdateMethodName($name)` mais une valeur par défaut (`zend_ProgressBar_Update`) existe déjà, et nous l'utiliserons ici. Ces méthodes JavaScript seront donc appelées à chaque mise à jour de la barre (`zend_ProgressBar_Update`), ainsi qu'à la fin du téléchargement (`zend_ProgressBar_Finish`). Il faudra bien sûr les implémenter, ce que nous ferons tout à l'heure.

Nous créons ensuite notre objet `zend_ProgressBar` en lui passant quatre arguments:

- notre objet *adapter*,
- la valeur minimum de notre barre d'avancement, soit zéro,
- la valeur maximum de notre barre d'avancement, soit `$bytes_total`,

dant de la sorte on s'épargne bien des *bugs* très difficiles à débusquer !

Nous testons ensuite la variable `$done` pour agir en conséquence si le téléchargement est terminé ou non. Si il n'est pas terminé on met à jour la barre d'avancement en passant le nombre d'octets envoyés à la méthode `update($current_value)`.

Si l'envoi est terminé, nous appelons simplement la méthode `finish()`. En fait, nous faisons un peu plus que ça... Nous créons un objet `Zend_Session_Namespace` pour pouvoir supprimer les informations résiduelles en session. De la sorte, si le même téléchargement, avec le même identifiant, est renvoyé une seconde fois (ce qui normalement ne devrait pas arriver...), il reprendra de zéro, et non pas de la précédente valeur. Je vous renvoie à la documentation du Zend Framework pour d'avantage de renseignements sur la manipulation des objets `Zend_Session_Namespace`, mais le code devrait être relativement parlant. Pour schématiser, les lignes suivantes :

```
$ns_download_progress = new Zend_Session_Namespace('download_progress_' . $upload_id);
$ns_download_progress->unsetAll();
```

Ont un effet similaire à :

```
unset($_SESSION['download_progress_' . $upload_id]);
```

Rassurez-vous, le plus gros est passé. Il ne nous reste désormais qu'à implémenter nos deux fonctions `zend_Progress_Update` et `zend_Progress_Finish`, et à s'assurer que la barre d'avancement soit régulièrement mise à jour.

Côté Client

Pour le développement de notre barre d'avancement côté client, nous allons centraliser tout le code dans un seul fichier : *scripts/progress.js*. Ce script se base sur le framework Mootools, que nous trouverons dans le fichier *scripts/library/mootools.js*. Nous avons pris le parti ici de séparer totalement le HTML du code JavaScript, aussi, toute le code JavaScript utilisé se trouvera dans le seul fichier *scripts/progress.js*, et nous n'aurons pas besoin de revenir sur le formulaire HTML pour le modifier.

Quelques bases de Mootools

Avant de se lancer directement dans le code, posons quelques bases de l'utilisation du framework Mootools. Cet article n'a pas pour vocation de vous apprendre à utiliser ce framework, mais de vous faire comprendre le peu de code que nous avons écrit avec celui-ci.

Mootools réécrit nombres de classes natives du Javascript. Nous allons ici manipuler la classe `Element`, incontournable dès qu'il s'agit de manipuler le DOM. La fonction

`$(id_element)` est utilisée pour récupérer un élément depuis son `id`. Il s'agit d'une amélioration de la méthode `Javascript document.getElementById(id_element)`, à la différence que la fonction `$(id)` peut recevoir sans distinction l'`id` d'un élément ou l'élément lui-même, ce qui est pratique pour permettre de traiter les deux cas sans s'embarrasser de conditions sur le type.

Mootools fournit aux éléments des accesseurs `set(propriete, valeur)` et `get(propriete)`, permettant de récupérer ou de modifier une propriété d'un élément. Quelques exemples :

```
element.get('id');
// == element.id
element.set('html',
  '<b>Bonjour</b>');
// == element.innerHTML =
'<b>Bonjour</b>'
```

Il fournit également des accesseurs pour modifier des propriétés CSS, tels que `setStyle(propriete_css, valeur)` et `getStyle(propriete_css)`. Quelques exemples :

```
element.setStyle('visibility',
  'hidden'); // ==
element.style.visibility =
'hidden'
element.getStyle('visibility'); // ==
element.style.visibility
```

Il est recommandé d'utiliser les accesseurs lorsqu'un framework JavaScript vous le propose, car ils permettent de s'affranchir des éventuelles différences entre les navigateurs. Dans le cas de Mootools, ils proposent également des raccourcis utiles (voir le `set('html', [...])` plus haut).

Je n'irais pas plus loin sur les bases de Mootools, et entrons maintenant dans le vif du sujet. Nous allons déjà instancier un objet `Request`, qui servira à envoyer une requête AJAX au serveur, tel qu'indiqué dans le Listing 6.

Plutôt parlant, non ? Nous passons à l'objet `Request` un objet ayant deux propriétés : `url` qui sera l'URL du script PHP à appeler, ainsi qu'une option `evalScripts` que nous passons à `true`. Cette dernière option permettra d'interpréter le code JavaScript présent dans la réponse du serveur qui, dans notre cas, appellera une des deux fonctions de `callback Zend_Progress_Update` ou `Zend_Progress_Finish`. Par défaut, la requête est en mode `post`, nous n'avons donc pas à le préciser ici. N'hésitez pas à consulter la documentation de l'objet `Request` sur le site officiel de Mootools pour voir toutes les options possibles.

Les événements

Maintenant que nous avons notre requête, il va falloir l'appeler au moment voulu, à savoir lorsque le formulaire sera validé (soit sur l'événement `onSubmit`). Regardons ensemble le Listing 7, où nous nous chargeons de gérer les événements de notre page.

Nous commençons par ajouter un événement à l'objet `window`. Le premier argument, `'domready'`, représente en fait l'événement `onDomReady`. Nous pouvons en effet nous passer du `on` devant les noms d'événements avec Mootools. Cet événement est un équivalent de `onLoad`, à la différence qu'il est déclenché une fois que le DOM est prêt (d'où le nom), mais avant le chargement des objets externes (images, etc ...). Il est de ce fait préférable de l'utiliser par rapport à `onLoad` dans le cas présent, et dans l'écrasante majorité des autres cas.

Si nous sautons quelques lignes, nous voyons que nous modifions l'événement `onSubmit` de notre formulaire pour que notre requête soit envoyée :

```
$(form_upload).addEvent
  ('submit', function() {
    progressBarRequest.send
    ('upload_id=' + $('APC_UPLOAD_
      PROGRESS').get('value'));
  });
```

Comme nous pouvions nous en douter, nous envoyons au serveur la valeur du champ `d'id APC_UPLOAD_PROGRESS`.

Mais, quelles sont ces quelques lignes de code où il est question d'`iframe` et d'`inject` ? C'est simple, si nous laissons les choses telles quelles, lorsque nous validerons notre formulaire, le navigateur chargera la page précisée dans sa propriété `action` (à savoir `upload.php`). En chargeant une autre page, le navigateur va décharger celle-ci, et nous n'aurons donc pas de barre d'avancement...

AJAX ne permettant pas, pour des raisons de sécurité, d'envoyer via une requête le contenu d'un fichier sur le serveur, nous allons devoir utiliser une petite ruse : faire valider le formulaire (attribut `target`) dans un `iframe` invisible. De la sorte, nous ne quitterons pas notre page, et notre barre d'avancement pourra faire son travail. Comme il serait disgracieux d'avoir un `iframe` utilisé uniquement pour des besoins techniques dans notre code HTML, le parti a été pris ici de générer cet élément dynamiquement.

Nous donnons à notre `iframe` un nom (`upload_iframe`) et nous passons ce même nom à l'attribut `target` de notre formulaire. Attention : l'attribut `target` demande un nom, pas un `id`. Notons également que nous lui donnons la classe `invisible`, et vous remarquerez aussi que si nous ne mettons pas le nom des propriétés de nos objets JavaScript entre guillemet, nous le faisons pour la propriété `class`. La raison est simple : `class` étant un mot réservé en JavaScript, l'utiliser dans ce contexte sans le protéger par des guillemets provoquerait une erreur.

Il ne suffit cependant pas de créer un objet `Element`, il faut aussi l'insérer dans notre page, ici entre les balises `body`. C'est la raison d'être de cette ligne de code :

```
iFrameElement =
  inject(document.createElement
  ('body'));
```

La méthode `getElement(tag_name)` est un raccourci Mootools bienvenu, puisqu'il retourne la première occurrence d'une balise passée en argument. Nous retrouvons de cette manière facilement l'élément `body`,

Zend_ProgressBar_Adapter_JsPush et bufferisation de sortie

L'appel à la fonction PHP `ob_start()` permet de stocker les sorties (`echo`, etc.) vers le navigateur dans un tampon (ou *buffer* de sortie) à partir de son appel. Ce tampon pourra être vidé (et envoyé au navigateur) par, par exemple, un appel à la fonction PHP `ob_flush()`. Comme notre adapter `Zend_ProgressBar_JsPush` a été conçu pour envoyer des informations en temps réel au navigateur, et qu'il est compatible avec une utilisation du module MVC du Zend Framework (lequel utilise une mise en tampon des sorties vers le navigateur via `ob_start()`), nous reproduisons ici ce comportement pour éviter une *notice* désagréable de la part de PHP. Cette *notice* est lancée en cas d'un appel de la fonction `ob_flush()` sans mise en tampon préalable. Ce cas de figure concerne uniquement l'utilisation de `Zend_ProgressBar_Adapter_JsPush`, et je vous invite à regarder le code source de la classe (relativement simple) pour vérifier par vous-même.

Barre d'avancement et hébergement mutualisé

La mise en place d'une barre d'avancement demandant l'utilisation d'extensions particulières, rarement mises à disposition sur des serveurs mutualisés, il y a de fortes chances que vous soyez contraint d'utiliser un serveur dédié pour ce genre de fonctionnalités (ou dans le meilleur des cas vous serez très restreint dans le choix d'un hébergeur mutualisé). Cela peut paraître de prime abord extrêmement restrictif, mais un site proposant d'envoyer des fichiers assez lourds pour nécessiter une telle barre de chargement aurait dans tous les cas du mal à se contenter d'un serveur mutualisé.

Terminologie

- JSON, pour *JavaScript Object Notation*, est un format de données très commode à utiliser pour les interactions AJAX entre JavaScript côté client et un autre langage côté serveur. Il s'agit de la représentation littérale d'un objet JavaScript. De nombreux outils permettent de transformer une chaîne de caractère au format JSON en objet JavaScript, de même qu'il existe de nombreux outils permettant de transformer un objet PHP en chaîne de caractère au format JSON. Mootools et Zend Framework disposent l'un et l'autre d'outils de ce genre.
- MVC, pour *Modèle-Vue-Contrôleur*, est une méthode de conception, aujourd'hui très répandue sur les gros projets Web, visant à séparer l'application en trois parties : l'extraction et modèle des données (modèle), l'affichage et la présentation à l'utilisateur (vue) et le traitement des données et gestion des événements (contrôleur). De nombreux frameworks PHP (Zend Framework, symphony, etc.) permettent de déployer simplement des applications basées sur une méthode MVC.

et nous injectons notre iframe dedans via sa méthode `inject(element_dans_lequel_injecter)`.

Nos événements sont prêts à être utilisés, il ne reste que nos deux fonctions de *callback*.

Les fonctions de *callback*

Voyons directement le Listing 8 pour notre fonction `zend_ProgressBar_Update(data)`. L'argument passé à la fonction est un objet proposant les propriétés suivantes :

- `current` : la valeur courante de la barre d'avancement, ici le nombre d'octets envoyés,
- `total` : la valeur maximum de la barre d'avancement, ici le poids du fichier en octets,
- `percent` : pourcentage d'accomplissement de la barre d'avancement, non arrondi,
- `timeTaken` : le temps écoulé en secondes,
- `timeRemaining` : le temps restant en secondes,
- `text` : un texte optionnel pouvant être passé à l'objet `zend_ProgressBar`, que nous n'utilisons pas ici.

Cet objet est généré automatiquement par `l'adapter Zend_ProgressBar_Adapter_JsPush`.

Le code de notre fonction `zend_ProgressBar_Update` est très simple, du moins pour ses cinq premières lignes. Nous affichons le bloc de la barre d'avancement pour qu'elle apparaisse à l'utilisateur, nous modifions le pourcentage

en l'arrondissant (remarquez l'ajout de Mootools qui propose une méthode `round()` pour les objets de type numérique). Nous utilisons la propriété `percent` pour modifier la taille de notre barre d'avancement, qui est en pourcentage, grâce à un des accesseurs `setStyle...` Quoi de plus intuitif ?

La dernière ligne est la plus ardue à comprendre. Mootools modifie également la classe `Function` du JavaScript et lui ajoute entre autres choses une méthode `delay(delay_en_ms[, bind[, args...]])`, qui permet de retarder l'exécution de la fonction. Le premier, assez parlant, est le temps à attendre avant l'appel, en millisecondes. Le second argument, optionnel, est un objet à rattacher à la fonction. Nous n'irons pas plus loin sur le mécanisme de *binding* propre à Mootools, car ce sujet est très vaste. Nous passons ici notre objet de requête. Les arguments suivants sont en fait les arguments de votre fonction, ici ce sera l'argument de la méthode `send`, à savoir les données à envoyer au serveur (l'identifiant du téléchargement).

En procédant de la sorte, nous sommes assurés de ne pas envoyer de requête tant que le serveur n'a pas répondu à la précédente, et nous attendons 200 millisecondes entre chaque envoi. Je vous conseille de faire de nombreux tests pour trouver le bon compromis permettant une actualisation fluide de la barre d'avancement sans surcharger d'appel votre serveur.

Voyons maintenant notre fonction `zend_ProgressBar_Finish()`, qui, elle, ne prend

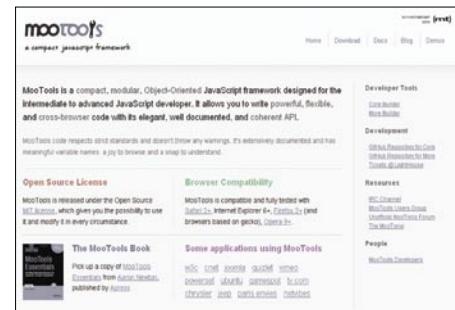


Figure 1. Site officiel du framework Mootools

pas d'argument. Nous nous contentons ici de mettre la barre à 100%, et d'avertir l'utilisateur de la fin de l'envoi. C'est bien évidemment très insuffisant... Mais cela devrait vous permettre de comprendre le fonctionnement global de `zend_ProgressBar_Adapter_JsPush`. Si vous voulez démontrer une des faiblesses de l'application, essayez par exemple d'envoyer un fichier, et de stopper l'envoi en plein milieu (bouton stop de votre navigateur). Le message *Envoi terminé !* s'affiche, et la barre d'avancement est passée à cent pour cent. Pas très parlant pour l'utilisateur !

Conclusion

Il est évident que l'exemple que nous avons vu est totalement insuffisant. Aucun contrôle n'a été fait sur le fichier reçu, ce qui est à la fois peu intuitif pour l'utilisateur en cas d'erreur lors de l'envoi, et surtout il s'agit d'une énorme faille de sécurité. De même, la présentation aurait gagné à être plus fouillée, par exemple en proposant un temps non plus en secondes, mais de façon formatée en affichant les minutes, voire les heures restantes.

Rien ne vous empêche, cependant, de créer vos propres classes filles de `zend_ProgressBar` pour envoyer plus d'informations au navigateur (comme par exemple le *bitrate* moyen, calculé à partir de la valeur courante et du temps écoulé), et de développer de véritables classes réutilisables en JavaScript afin de disposer de barres d'avancement à la fois prêtes à l'emploi et hautement personnalisables. Vous trouverez l'application sur le CD, aussi n'hésitez pas à voir de quoi il en retourne. Et rien ne vous empêche non plus de partir de celle-ci pour créer votre propre barre d'avancement...

Sur Internet

- <http://fr3.php.net/apc> – Documentation de l'extension PHP APC,
- <http://fr.php.net/manual/fr/install.pecl.intro.php> – Documentation sur l'installation des extensions via PECL,
- <http://framework.zend.com/> – Site officiel du Zend Framework,
- <http://fr.php.net/manual/fr/ref.outcontrol.php> – Documentation sur les fonctions PHP de bufferisation de sortie,
- <http://fr3.php.net/manual/fr/features.file-upload.errors.php> – Codes d'erreur PHP lors de l'envoi d'un fichier.
- <http://www.mootools.net/> – Site officiel du framework Mootools.

LUCAS CORBEAUX

Développeur essentiellement Web depuis plus de trois ans, l'auteur nourrit une certaine curiosité vis à vis des technologies informatiques en général et du Web en particulier. Après avoir collaboré avec la Web Agency Rage Conception, il travaille désormais à la conception de Buzzic.com, un site de diffusion musicale en ligne. Vous pouvez le contacter à l'adresse email : lucas.corbeaux@gmail.com.

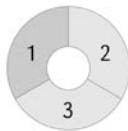
Créer votre blog avec Drupal

Drupal est un CMS de plus en plus utilisé à un niveau professionnel. Complet, il bénéficie d'un support communautaire important et de modules aboutis, y compris dans sa version de base. Nous allons voir dans cet article, comment l'utiliser au mieux.

Cet article explique :

- L'installation du CMS Drupal.
- Son utilisation basique.
- L'installation d'un thème et d'un module.

Niveau de difficulté



Drupal est un CMS, ou système de gestion de contenu, apparu en 2001. Il permet de déployer rapidement des sites web dynamiques. Il peut aussi bien être utilisé tel quel, en tant que moteur de contenu, que personnalisé par le biais de modules et thèmes. Il est de plus porté par une communauté active qui met à disposition nombre de contributions. Enfin un système d'API permet aux utilisateurs de développer des modules pour leurs besoins spécifiques. Dans cet article, nous allons voir comment déployer un blog en utilisant les modules de base. Nous verrons aussi comment installer un thème et un module.

Pourquoi choisir Drupal ?

Le choix d'un CMS se fait avant tout en fonction des besoins des sites que vous comptez mettre en place avec lui. Nous avons pris ici l'exemple d'une plate-forme de blog. Il existe d'autres applications, spécifiquement dédiées à ce besoin. Les plus connues sont *DotClear* ou *Wordpress*. L'avantage de Drupal, est de proposer une étendue des contenus administrables plus importante que ces applications. Ainsi dans le cadre d'un portail, par exemple un site présentant un projet communautaire avec de la documentation, des images et le blog des participants, Drupal sera plus adapté. Par rapport aux autres CMS, Drupal est très modulaire

Ce qu'il faut savoir :

- Notions d'utilisation d'un blog.
- Quelques notions d'HTML et de mise en page.

et dispose d'une base importante de modules déjà existants, y compris pour la branche la plus récente. En dépit de ce choix de construction, Drupal est utilisable directement avec le *packaging* de base. En revanche, le CMS reste encore un peu lourd et sa prise en main, ainsi que celle de ses modules les plus importants, nécessiteront d'abord des essais sur un site local.

Installation du CMS

La première étape consiste à télécharger la dernière version de Drupal et l'installer. Il est à noter que l'équipe de développement maintient deux branches en parallèle, à savoir actuellement les branches 5.X et 6.X. La branche 7.X est encore en développement. Les deux branches sont mises à jour régulièrement et les versions sortent en parallèles. Il est recommandé, pour le déploiement d'un nouveau site, de télécharger les dernières versions. Le choix entre la version 5 et 6 se fera en fonction de la compatibilité des modules que vous désirez déployer. En effet certains modules n'ont pas encore été portés dans la branche 6.X. Pour la suite de notre article, nous allons utiliser la dernière version de la branche 6.X, à savoir, au moment de l'écriture de ces lignes, la 6.8. Vous pouvez aussi, si vous le désirez, télécharger l'archive de tra-

duction permettant d'obtenir la version francophone de Drupal : <http://drupal.org/project/fr>.

Mise en place de la plate-forme

Vous aurez besoin, pour installer l'application, d'une plate-forme LAMP ou WAMP, telle que *WampServer*. Il vous faudra aussi un décompresseur sachant ouvrir les fichiers .tar.gz car tous les modules et archives Drupal sont enregistrés sous ce format. Sous Linux pas de soucis. Sous Windows, je vous conseille 7-zip, <http://www.7-zip.org/>, qui se substituera bientôt à votre *WinZip* favori. Il vous faudra créer une base de données où seront stockés les contenus et tables propres au CMS. Pour cela utilisez *phpMyAdmin* afin de créer une base. Idéalement, il serait mieux de créer un utilisateur propre à cette base. Drupal peut être installé sur MySQL ou PostgreSQL. Déployez l'archive Drupal sur le répertoire *www* de votre serveur. Vous obtiendrez un répertoire *drupal-6.8* que vous pouvez renommer en *drupal* pour plus de commodités. Si vous avez téléchargé l'archive de traduction, déployez la à l'intérieur de ce répertoire. Dernier point, Drupal peut gérer plusieurs sites avec plusieurs configurations différentes. Il vous faudra initialiser le fichier du site par défaut. Pour cela, rendez-vous dans le sous-répertoire *sites/default* du répertoire *Drupal* et copiez le fichier *default.settings.php* dans un fichier *settings.php*. Votre plate-forme est maintenant prête pour l'installation de Drupal.

Étapes d'installations

Comme la plupart des applications web actuelles, Drupal dispose d'une interface d'installation.

Identifier votre administration

Contrairement à de nombreux CMS, Drupal ne fait pas de différence entre le *back office* et le *front office*. Vous basculez de l'un à l'autre selon les pages que vous fréquentez. Si vous désirez néanmoins identifier l'une de l'autre, il est possible d'utiliser deux thèmes différents. Pour cela il suffit de se rendre dans la rubrique Thème de l'administration de la rubrique Configuration du site et de changer l'option thème de l'administration.

Elle est accessible depuis votre navigateur, en pointant sur le répertoire où vous avez déployé votre application ; pour ma part <http://localhost/drupal>. Si vous avez mis en place les fichiers de traduction, alors un écran vous invitant à choisir la langue d'installation s'affichera, comme vous pouvez le voir dans la Figure 1. Il est recommandé de choisir votre langue natale pour plus de facilités. La deuxième étape (Figure 2) consiste à entrer les informations de bases de données : nom de la base, nom d'utilisateur et mot de passe. Si vous le désirez, l'interface vous permet d'indiquer l'adresse et le port du serveur, ainsi qu'un préfixe de table, très pratique si vous êtes sur un hébergement mutualisé. Une fois cette étape validée, Drupal se charge de l'installation des modules et des traductions des différentes interfaces. Attention cette opération est parfois très longue, de l'ordre d'une minute. Pour éviter les erreurs vous pouvez modifier, si possible, votre fichier `php.ini` et augmenter la variable `max_execution_time`. L'étape suivante (Figure 3) vous permet de configurer le site et d'en indiquer le nom et le mot de passe de l'administrateur. Une fois ceci validé, votre site est prêt.

Les différentes notions Drupal

Nous allons, au cours de ce chapitre, définir les différentes notions et termes utilisés par Drupal, à savoir les noeuds, les blocs et les modules.

Les noeuds

Tous les contenus gérés par Drupal sont stockés dans des objets appelés noeuds ou *node*. Dans sa forme la plus simple, un noeud possède un titre et un élément de type texte. Il est possible d'étendre un noeud et de créer un nouveau type de contenu en ajoutant des champs, ceci à l'intérieur même du CMS ou bien par l'utilisation de modules de création de contenu comme CCK.

Drupal fournit deux types de base : la page et l'article. L'activation de modules permet de mettre à disposition de vos contributeurs de nouveaux types de contenus, et d'en créer par vous-même.

Les blocs

Les blocs sont des éléments visuels contenant formulaires ou informations spécifiques. Ils sont utilisés pour afficher les éléments de navigation du site, les formulaires d'inscription. Chaque bloc peut être configuré individuellement et affiché selon un certain nombre de critères.

Les modules

Les modules sont des briques logicielles qui permettent d'étendre les possibilités du CMS. Drupal entrée ou *hook*. Les modules peuvent fournir de nouveaux types de contenus, types de blocs ou de nouvelles fonctionnalités.

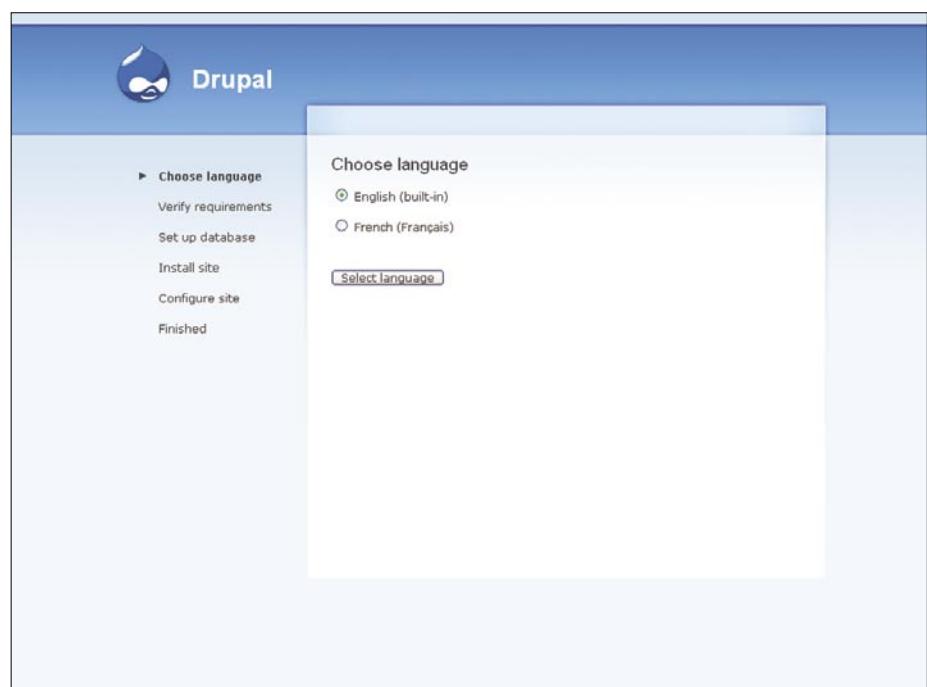


Figure 1. Installation Drupal : choix de la langue

Administration du site

Après cette présentation et l'installation du CMS, nous allons voir comment l'administrer. Nous allons ainsi voir comment activer des modules, comment ajouter du contenu et créer des utilisateurs.

Activation d'un module

Le but de notre installation est de propulser un ou plusieurs blogs à l'aide de Drupal. Pour cela, Drupal fournit un module `Blog` directement dans son archive de base. La communauté développe de nombreux modules, les plus populaires sont intégrés officiellement

dans les distributions du CMS, c'est le cas, entre autre, du module `Blog`. Pour l'activer rendez-vous dans la partie *Administrer, Constructions du sites, modules*. L'activation se fait en cochant les modules à activer, comme on peut le voir sur la Figure 4. Drupal gère les dépendances entre les modules, empêchant la désinstallation d'un module dont dépend un autre.

Une fois le module activé, vous pouvez ajouter un nouveau type de contenu `Billet de blog`. Si vous voulez ajouter un module en dehors des modules de bases, vous devrez le déployer dans le répertoire `modules` de Drupal.



Figure 2. Installation Drupal : configuration de la base de données

Figure 3. Installation Drupal : configuration du site

Éditez vos pages

Vous pouvez maintenant ajouter vos billets. L'éditeur est simple, comme le montre la Figure 5. Vous pouvez néanmoins l'améliorer en installant un éditeur *WYSIWYG* tels que *Tiny MCE* ou *FCKEditor*. Pour plus d'informations rendez-vous sur la page <http://drupal.org/project/wysiwyg>. À noter que de base vous pouvez utiliser, pour les textes de vos contenus, deux syntaxes : *full HTML* et *filtered HTML* qui permet de ne conserver que les balises HTML essentielles. Étant sous le compte administrateur, vous pouvez configurer un certain nombre d'options de publication et paramétriser les commentaires. Vous pouvez avoir un aperçu de votre billet et l'enregistrer. Une fois validé, votre billet sera, par défaut, publié et affiché sur la page d'accueil. Vous pouvez, là aussi, contrôler ceci dans la rubrique *Options de publication*. Votre blog est maintenant alimenté et la page d'accueil en sera modifiée. Il est temps d'ouvrir votre espace à d'autres contributeurs.

Ajoutez et configurez des utilisateurs

L'avantage d'utiliser Drupal comme plate-forme de blog, est de pouvoir créer un portail et pouvoir permettre à plusieurs contributeurs de travailler sur votre site. De même vous pouvez leur attribuer un espace de blog et contributions. Pour créer un utilisateur, rendez-vous dans la catégorie *Administrer*, *Gestion des utilisateurs*, *Utilisateurs* et cliquez sur "Ajouter un utilisateur". Renseignez les différents champs : nom d'utilisateur, adresse e-mail, mot de passe, le statut et la langue. Il est à noter que vous pou-

vez permettre aux visiteurs de créer un compte sur votre site. En vous rendant dans la rubrique *Paramètres des utilisateurs* de la catégorie *Paramètres des utilisateurs*, vous pourrez configurer les différents droits de vos utilisateurs.

Une fois l'utilisateur créé, celui-ci peut se connecter à votre site et bénéficier de droits que

vous pouvez gérer par l'intermédiaire de rôles qui lui sont attribués. Par défaut Drupal propose deux rôles de base, comme on peut le voir sur la Figure 6 :

- utilisateur anonyme, qui prend en compte tous les visiteurs du site
- utilisateur authentifié, qui permet d'identifier les utilisateurs connectés.

Il est à noter qu'il n'existe pas de rôle administrateur. Vous pouvez en revanche donner des droits d'administration à vos utilisateurs, mais seul le premier utilisateur enregistré lors de l'étape d'installation est administrateur.

La rubrique *Rôles* vous permet d'ajouter des rôles et de configurer les rôles déjà existant. Pour permettre à vos utilisateurs de bloguer, vous pouvez tout d'abord cliquer sur *modifier les droits d'accès* de la ligne utilisateur identifié. Vous vous retrouvez alors devant une page identique à celle Figure 7. Cliquez ensuite sur les options que vous voulez allouer, dans mon cas *créer des billets de blogs*, *supprimer ses propres billets de blog*, *modifier ses propres billets de blogs*. Enfin, cliquez sur le bouton *enregistrez les droits d'accès* pour que vos modifications soient prises en compte. Vous pouvez aussi, afin de gérer plus finement vos droits, créer un nouveau rôle *contributeurs de blogs* ayant les droits mentionnés auparavant et ajouter ce rôle individuellement à chaque contributeur, en passant par la rubrique *Utilisateur*. Attention à ne pas aller trop finement dans la gestion des droits, où vous vous perdriez dans une liste interminable de rôles. Attention aussi à bien repartir la

Figure 4. Configuration Drupal : activation d'un module

liste des droits quand vous installerez un nouveau module, ceux-ci fournissant de nouvelles fonctionnalités. Pour votre plate-forme, je vous recommande d'ajouter encore un administrateur de blog qui aura le droit de supprimer et d'éditer les billets des autres contributeurs. Votre plate-forme de blog est maintenant prête et peut accueillir de nouveaux rédacteurs.

Personnalisation de l'aspect du site

Votre site est désormais pleinement fonctionnel. Le problème est que son aspect est encore celui de centaines d'autres, c'est à dire le thème de base de Drupal. Nous allons voir maintenant comment personnaliser votre site en quelques lignes.

Qu'est ce qu'un thème?

Comme nous avons pu le voir auparavant, Drupal a une notion particulière et spécifique du contenu, structuré de la façon la plus simple possible. Ces contenus ne s'occupent absolument pas de la présentation. Celle-ci est laissée au soin des thèmes contenant des feuilles de styles et des *templates*. Les avantages sont nombreux. Un administrateur peut changer radicalement la mise-en-page de son site sans repasser sur ses contenus et en respectant l'accessibilité de son site. Drupal offre différents mécanismes comme la possibilité d'étendre un thème en sous-thèmes, permettant de ne développer que les points qui diffèrent d'un thème à l'autre. Par exemple le thème *Minelli* est un sous-thème de *Garland*. Autre point intéressant, en association du module *Color*, le schéma de couleur d'un thème peut-être modifié en quelques clics.

Changez les couleurs

Nous allons dans un premier temps personnaliser notre thème par défaut, qui fait partie des thèmes *recolorables*. Pour cela allez dans la rubrique *Thèmes* de la catégorie "Construction du site" (Figure 8). Si ce n'est déjà fait activez le thème *Garland* en cochant la case et en enregistrant les modifications. Ensuite cliquez sur le lien "Configurez" pour vous rendre sur les options du thème. Celle-ci sont diverses et permettent une personnalisation simple mais déjà intéressante. Vous pouvez ainsi activer ou désactiver l'affichage des éléments de votre site, ou encore mettre à jour le logo affiché sur vos pages. Si le module *color* est installé, vous pouvez choisir un jeu de couleurs parmi ceux proposés, comme le montre la Figure 9. Validez et votre site s'affichera sous ses nouvelles couleurs.

Changez de thème

Pour aller plus loin, nous allons maintenant voir comment installer un thème. Tout d'abord, il convient d'en choisir un dans l'interminable liste qui s'offre à vous sur <http://drupal.org>

Figure 5. Configuration Drupal : édition d'un billet de blog

Figure 6. Configuration Drupal : page d'ajout et édition des rôles

Figure 7. Configuration Drupal : gestion des droits d'accès

The screenshot shows the 'Themes' configuration page in Drupal 6.8. It lists six themes: Bluemarine, Chameleon, Garland, Marvin, Minnelli, and Pushbutton. Each theme is shown with a preview image, a brief description, and configuration options. The 'Garland' theme is selected as the default theme.

Figure 8. Configuration Drupal : Configuration des thèmes

project/Themes. Attention à bien prendre un thème compatible avec votre version de Drupal. Si vous désirez personnaliser votre thème, je vous invite à télécharger le Zen thème qui est une base simple et appropriée pour cela. Une fois téléchargé, créez un répertoire themes sous l'arborescence site/default et déployez-y le thème que vous venez de télécharger. En revenant

sur la page des thèmes, vous verrez alors que le thème et ses thèmes dérivés ont été ajoutés. Vous pouvez désormais activer le thème Zen et choisir de l'utiliser par défaut.

Accéder à vos contenus

Votre contenu est maintenant en place, celui de vos contributeurs aussi, et vous voilà avec de Me-

Sur Internet

- Le site de Drupal : <http://www.drupal.org/>
- Présentation de drupal sur framasoft : <http://www.framasoft.net/article4124.html>
- Le site de la communauté francophone Drupal : <http://www.drupalfr.org/>
- Liste des modules indispensables de Drupal : http://www.ineation.com/drupal_modules_partie1

nus de la catégorie construction du site et d'en ajouter un. Le menu se compose ensuite en ajoutant les éléments un par un. Vous pouvez utiliser les raccourcis Drupal. Ainsi le lien vers le blog sera blog, celui vers le premier blog sera blog/1 et ainsi de suite. Une fois votre menu créé, vous pouvez l'intégrer en tant que bloc sur votre site.

Pour aller plus loin, il existe plusieurs possibilités. La première, incluse dans Drupal, est d'utiliser le module taxonomie afin de créer des listes de termes, appelés aussi tags, que vous pourrez ensuite attribuer à vos différents contenus. Par exemple si, dans vos blogs, vous parlez de photos, vous pouvez créer le terme, l'associez à vos différents billets sur le sujet et ensuite créer un lien vers le terme dans votre menu. Dans Drupal aussi, vous trouverez le type contenu Book qui permet de structurer vos pages et les lier entre elles. C'est le contenu idéal pour mettre en place une documentation. Enfin, de nombreux modules existent pour personnaliser plus finement vos pages, tels que le module View permettant de créer, comme son nom l'indique, des vues regroupant plusieurs contenus au sein de blocs ou de pages.

Conclusion

Cet article constitue une première approche du CMS Drupal. Nous avons pu voir que sans éditer une ligne de code, il est possible de déployer très simplement un site avec des fonctions de bases. Bien entendu, certains aspects, tels que l'intégration de nouveaux contenus, de création de nouveaux blocs nécessiteront de mettre un peu plus les mains dans le cambouis, mais ceci est une autre histoire, et fera l'objet d'un autre article. En attendant je vous invite à explorer la galaxie Drupal en commençant par les liens fournis ci-dessous.

BRICE FAVRE

L'auteur travaille depuis plus d'une dizaine d'années dans le domaine de l'internet. Au sein de SQLI, il anime et participe à la cellule PHP Lyon et Suisse. Dans le cadre de ses travaux, il participe aux tests et à la capitalisation sur les développements PHP, aussi bien dans le cadre de projets que de déploiements d'applications. Contact : bfavre@sqlil.com.

Figure 9. Configuration Drupal : Choix des couleurs

The screenshot shows the 'Garland - Fluid Width' theme configuration page. It includes a color palette, a preview of the theme, and a sidebar for activating/deactivating page elements.

Vous ne savez pas comment gérer un site de e-commerce?

Vous ne voulez pas perdre d'argent?

Nous avons la solution.

eziBoutik

votre boutique e-commerce en version "full-services"

www.eziboutik.fr
info@eziboutik.fr

Design à la carte

Référencement

Gestion

Accompagnement

Sécurité

ADWORDS
QUALIFIED
COMPANY
Google

 Adhérent
AFNIC

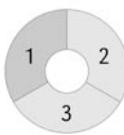
Network
Solutions



Hébergement E-Commerce

Cet article vise à présenter les différents points sur lesquels votre attention doit se porter lors du choix de votre futur hébergeur.

Niveau de difficulté



L'hébergement est une discipline extrêmement complexe du fait de la quantité d'éléments techniques à prendre en compte. Aucun aspect ne doit être mis de côté avant de sélectionner votre hébergeur.

Réseau

Le réseau est le lien fondamental qui relie l'acheteur à votre boutique. Maximiser la qualité de cette liaison revient à optimiser l'expérience utilisateur et à accroître vos chances de réaliser des ventes.

Rapidité

La rapidité d'affichage d'une page est une dimension primordiale du web. Plus la page est lente à charger, plus les conséquences seront nombreuses et gênantes. Une étude récente a ainsi révélé les déconvenues suivantes :

- *Amazon* : une augmentation de 100 millisecondes de la durée de chargement d'une page provoque une baisse des ventes de 1%,
- *Yahoo!* : une augmentation de 400ms entraîne une chute de trafic de 5 à 9%,
- *Google* : 500ms peuvent entraîner jusqu'à 20% de recherches en moins.

L'idéal est donc de réaliser des tests sur des sites gérés par l'hébergeur afin de valider le niveau de performance des infrastructures.

Capacité

En se lançant dans un projet web il est courant d'imaginer le pire (plantage, piratage, déni de service), il est plus rare de se préparer au meilleur. Quoi de mieux sur le web que de voir un afflux important de visiteurs sur son site. Inutile d'être un grand statisticien pour deviner que le chiffre d'affaire est toujours proportionnel au trafic. Une augmentation de trafic implique une consommation importante de bande passante. Il est par conséquent essentiel de vérifier que l'hébergeur dispose de tuyaux suffisamment larges pour absorber

toute augmentation soudaine de consommation de bande passante.

Qualité

La capacité absolue n'est pas tout. De nombreux hébergeurs annoncent des capacités de centaines de mégas octets de bande passante à des prix défiant toute concurrence. Cette bande passante correspond le plus souvent à une bande passante partagée par une multitude de clients. Dans une telle situation, un site qui consommerait soudainement toute la bande passante (attaque, DOS, téléchargements illégaux) mettrait en péril ses voisins.

Il est donc préférable de toujours choisir une bande passante dédiée que vous serez le seul à exploiter. Une capacité de 1Mbps est suffisante pour 99% des sites d'e-commerce.

Redondance

Il est essentiel de vérifier que la bande passante est fournie par plusieurs opérateurs (tels que *Colt*, *Level3*, *PSINet*, *Teleglobe*, etc.). On parle alors de bande passante *multi-homée*. Plus le nombre d'opérateurs est important moins vous prenez de risques en cas de défaillance de l'un d'entre eux. Le trafic passant par un opérateur qui viendrait à *tomber* serait alors instantanément réparti sur les autres. Les clients les plus exigeants iront même jusqu'à vérifier si France TELECOM fait partie de ces opérateurs (en terme technique, l'hébergeur dispose-t'il d'un lien de *peering* avec FT?). La logique est ici la suivante : *Orange* (France TELECOM) est le principal fournisseur d'accès ADSL en France. Ainsi, si votre site réside sur les mêmes tuyaux que ces millions d'internautes français, la vitesse de chargement sera alors accrue.

Sécurité

Point essentiel de tout projet web, la sécurité revêt une dimension encore plus importante dans le cadre d'une boutique en ligne.

Cryptage

La quasi totalité des systèmes de paiement électronique fonctionne en France de la même manière. Une fois la commande préparée (produits sélectionnés, adresse et mode de livraison renseignés, etc.), l'internaute finalise sa commande en basculant sur le serveur de la banque. Il renseigne alors

son numéro de carte bleue ainsi que la date de validité. Cette page, située sur le serveur de la banque, est toujours protégée par un cryptage de type SSL. Ce dernier empêche tout pirate d'analyser et de capturer les informations envoyées à la banque. Ce niveau de sécurité est ici indispensable du fait de la confidentialité extrême de ces informations.

Ce niveau de protection a-t'il besoin d'être présent sur tout le site ? Rien n'est moins sûr. En effet, que risquerait un client qui se ferait usurper son identité ? Que le pirate puisse mettre des produits dans son panier, qu'il puisse afficher une facture ... la belle affaire.

Certains systèmes de paiement sécurisés tels que PAYBOX permettent en revanche au commerçant de collecter les informations bancaires (numéro de carte bleue, date d'expiration) au sein même du site. Dans un tel cas la protection SSL devient obligatoire. Le choix du SSL rend l'option d'un hébergement mutualisé peu envisageable (il conviendrait en effet que chaque site du serveur dispose d'une adresse IP qui lui soit propre). De plus, le choix du cryptage SSL impose d'acheter un certificat auprès d'un organisme tel que *Verisign* ou *Thawte*.

Ne choisissez pas les certificats dont le prix est trop bas (quelques euros). Il est probable en effet que de tels certificats ne soient pas compatibles avec tous les navigateurs (notamment IE6). Un internaute disposant d'un navigateur incompatible verrait alors apparaître une alerte. Des termes tels que *sécurité*, *non compatibilité*, *danger* risqueraient alors de l'effrayer et pourraient avoir pour conséquence fâcheuse de faire fuir ce précieux prospect.

Ne tombez cependant pas dans l'excès de zèle inverse en souscrivant à des offres trop onéreuses uniquement car ces dernières contiennent le terme *e-commerce* dans leur intitulé. Il s'agit la plupart du temps d'offres purement marketing qui ne vous apportent rien si ce n'est d'alléger votre portefeuille. Un certificat *web* standard est largement suffisant.

Audit

Il est essentiel de surveiller le bon fonctionnement de votre site. Les fichiers de *logs* sont un excellent outil pour savoir si votre site fait l'objet d'attaques ou pour détecter d'éventuels comportements anormaux. Il est important de valider que l'hébergeur sera bien en mesure de mettre à disposition

SOLUTION E-COMMERCE ET SITE WEB

www.horra.fr



→ La solution e-commerce complète pour répondre à tous vos besoins Internet et assurer le succès de votre e-stratégie.

Quelques réalisations :

www.groupe-eurosign.com/aprintplus/



Site e-commerce

Conception de toutes communications visuelles : plaques signalétiques, autocollants, découpes lettres, vinyles, impressions numériques, gravures...

www.jcomcreation.com/OsCom/



Site d'annonces automobiles :

Annonces automobiles toutes marques.

www.jcomcreation.com

Site flash :

Site vitrine avec intégration galerie photos .



www.horra.fr



Site vitrine flash :

Site vitrine avec graphisme personnalisé.

Pour tous renseignements :
08 25 18 79 95 / 06 13 91 89 22
www.horra.fr / contact@horra.fr

ces fichiers. Les *logs* sont généralement divisés en deux fichiers : un fichier pour les accès aux pages et un autre pour les erreurs. C'est ce dernier qu'il conviendra de contrôler le plus régulièrement possible.

Technologie

Bien que la compétence première demandée à un hébergeur soit de savoir gérer ses machines, il est important de vérifier qu'il ne sera pas désemparé face à la technologie choisie pour la réalisation du projet.

Compétences

Les systèmes de paiement électronique reposent généralement sur l'exécution de binaires. Tout langage de programmation est donc envisageable. Il peut être intéressant de vérifier que l'hébergeur a de l'expérience avec le langage choisi. En effet, en cas de souci, cet ultime recours pourrait faire la différence.

Les complications les plus courantes lors de la mise en place d'un système de paiement sécurisé sont liées généralement à des problématiques de :

- chemin (relatif, absolu),
- de droits (lecture, exécution),
- d'inclusion (include de fichiers PHP).

Ces notions sont intimement liées au système d'exploitation et au paramétrage donc à l'hébergement. De la même manière, il peut être judicieux de savoir si l'hébergeur sélectionné dispose d'une expérience avec le système de paiement retenu.

Souplesse

De plus en plus de solutions de paiements (*Google Checkout*, *Paypal*, *eBay*) mettent à disposition des API qui permettent à vos scripts de communiquer directement avec leur plateforme. Les protocoles d'échange reposent la plupart du temps sur le langage XML. Il est important de contrôler que l'environnement PHP de votre futur serveur dispose bien de tous les modules requis (ex. DOM, XML-RPC, SOAP, cURL, SSL). Plus généralement, il est essentiel de vérifier que l'hébergeur aura la souplesse de vous installer un module le jour où vous en aurez besoin. Les hébergeurs ont en effet souvent la fâcheuse habitude de ne pas vouloir s'éloigner des installations standards ce qui se révèle finalement extrêmement contraignant pour le client.

Sauvegardes

Les données stockées dans le cadre d'une boutique représentent un véritable trésor. Les coordonnées de vos clients, les commandes et autres factures sont au cœur de votre business. Il est important de vérifier qu'une sauvegarde est bien prévue par votre hébergeur. Il est également essentiel de contrôler que ces sauvegardes sont bien stockées sur un serveur différent du serveur de production; il convient en effet d'éviter tout drame en cas de crash machine. Le nombre de sauvegardes est un élément supplémentaire à contrôler. Il n'est pas rare en effet de ne s'apercevoir d'un problème qu'après quelques jours (notamment si ce plantage arrive un week-end). Il faut donc s'assurer que l'hébergeur est en mesure de restaurer une sauvegarde de plus de trois jours.

Conclusion

Une fois ces différentes considérations prises en compte, le grand conseil reste de toujours privilégier un hébergeur à taille humaine. Il est essentiel de ne pas être considéré comme un simple *numéro d'incident*. Un hébergeur dont la taille vous permet d'être en rapport direct avec l'administrateur système en charge de votre dossier n'a pas de prix. C'est cette dimension qui vous permettra d'obtenir cette réactivité si indispensable sur le web.

FRANÇOIS-XAVIER

François-Xavier Bois est le co-fondateur de Kernix Software (<http://www.kernix.com>), société spécialisée dans le développement et l'hébergement d'applications web. Il est le créateur du framework de développement KWO. Kernix exploite notamment ce framework pour réaliser des sites marchands et des réseaux sociaux sur mesure. François-Xavier Bois est l'auteur de plusieurs ouvrages consacrés à PHP et MySQL.

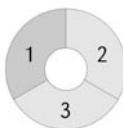
Quand l'Open Source révolutionne la distribution

À l'heure où le e-commerce est en plein essor, les sites marchands doivent être à la pointe de l'innovation pour conquérir les marchés et conserver leurs marges. La rencontre entre CMS et solution ERP matures constitue indéniablement l'une des clés majeures pour permettre aux cyber-marchands de développer de nouveaux avantages concurrentiels.

Cet article explique :

- Cet article traite de plates-formes e-commerce haut de gamme et à forte valeur ajoutée, mettant en jeu des solutions hétérogènes mais néanmoins complémentaires, afin d'obtenir une solution de distribution globale verticalisée automatisant les interactions entre support de vente en ligne (portail e-commerce) et gestion courante de l'activité (logiciel de gestion, ERP).

Niveau de difficulté



Ce qu'il faut savoir :

- Notions : process e-commerce et logistiques avancés.

- risques et complexité liés à l'acheminement et à l'importation des produits,
- assurance qualité et respect des réglementations Européennes,
- mais surtout : allongement des délais de livraison et des volumes minimum de commande, qui oblige les distributeurs à augmenter la taille de leurs stocks pour ne pas impacter leur taux de service.

Leur conviction est que, dans le cadre d'actions de *sourcing*, le stock du grossiste traditionnel doit être remplacé par un stock virtuel *en mouvement* entre l'usine lointaine et le magasin. Que les produits soient dans l'usine, dans un bateau, en douane ou dans un camion, le grossiste nouvelle génération doit être en mesure de les localiser, de les comptabiliser et de les vendre à tout moment et quel que soit leur position dans la chaîne logistique, en s'engageant sur les quantités et sur les délais.

C'est grâce à ce nouveau type d'acteur global, jouissant d'implantations fortes des deux côtés de la chaîne Europe/Asie par exemple, que les distributeurs peuvent avoir recours au *sourcing* avec des commandes plus petites, plus fréquentes, et offrant la même sérénité que celle offerte par les grossistes traditionnels, mais en continuant à bénéficier des meilleurs produits au meilleur prix.

Pour que ce concept de distribution nouvelle génération fonctionne, ces grossistes ont besoin d'un système d'information flexible et puissant capable de :

- prendre un maximum de petites commandes de façon automatisée via un portail internet,
- fournir une traçabilité détaillée des produits dans la chaîne logistique, jusqu'à leur réservation à la volée,
- moduler les prix de vente en fonction de la position des produits dans la chaîne logistique.

e Z publish est un gestionnaire de contenu *Open Source*, développé en PHP, s'appuyant sur une base de données MySQL. eZ publish est une solution personnalisable à volonté permettant la création d'un site internet évolutif, adaptée à la publication sur Internet, au e-commerce ou encore à la mise en place d'intranet. eZ publish est un véritable framework applicatif qui permet la réalisation de projets complexes. Sa structuration et son puissant modèle de contenu orienté objet, permettent de gérer les problématiques de gestion de contenu les plus ardues. Dans le cadre d'une plate-forme e-commerce verticalisée, il est utilisé comme front end du site internet de vente en ligne, permettant de présenter les produits, leurs prix, permettre aux internautes d'ouvrir un compte client, de commander et de suivre le traitement de leurs commandes en temps réel.

Open ERP est développé en python et s'appuie sur une base de données PostgreSQL. C'est un logiciel *Open Source* de gestion commerciale et comptable très complet, qui dispose de nom-

breux modules pour aider les entreprises à mieux gérer leurs ressources, qu'elles soient financières, matérielles ou humaines. Il est notamment utilisé pour la gestion des stocks produits, le suivi en temps réel du circuit logistique et des chaînes de production, les relations clients par le biais d'un module CRM avancé, ainsi que la gestion commerciale et financière jusqu'à l'édition d'un plan comptable général. Il s'agit d'une solution modulaire capable de s'adapter à tout type d'entreprise de petite et moyenne taille.

Un concept innovant !

Révolutionner la distribution grâce à l'Open Source, telle est l'ambition de clients spécialistes du *sourcing* qui s'appliquent avec enthousiasme à créer et développer des concepts de distribution multi-canal innovant autour de solutions exclusivement Open Source.

Le développement explosif en Europe du *sourcing* vers les pays à bas coûts, offre aux distributeurs de nouvelles solutions d'approvisionnement. Si le *sourcing* offre des avantages indéniables comme la réduction spectaculaire des coûts d'achat, l'accès à de nouveaux produits ou la possibilité de développer leurs propres marques, le recours à ce mode d'approvisionnement représente néanmoins pour les distributeurs un challenge, parfois insurmontable, qui remet en cause leur mode de fonctionnement et leur organisation :

Seul un ERP Open Source peut permettre de développer rapidement les fonctionnalités innovantes dont ce type de clients a besoin. C'est l'arrivée à maturité du projet Open ERP qui a rendu le projet possible, par sa gestion des ressources de l'entreprise hautement modulaire et facilement personnalisable :

- gestion des fichiers et de la relation client,
- actions de prospection et de fidélisation,
- organisation optimisée de la logistique,
- automatisation et suivi des achats,
- gestion des processus de vente du devis à la facture,
- analyse stratégique des données de l'activité,
- gestion comptable et financière.

Le système implémenté permet une gestion fine des stocks en flux tendu, tout en continuant à offrir aux clients une disponibilité optimale sur les produits du catalogue. Il permet de garantir en temps réel au client le meilleur couple PRIX / DELAI en fonction de la position de chaque produit au sein de la chaîne logistique. Pour atteindre cet objectif, *les produits sont mis en vente, non pas lorsqu'ils rentrent en stock, mais dès que l'ordre de fabrication est lancé auprès du fournisseur*. Le grossiste nouvelle génération dispose de toute la durée de fabrication et d'acheminement de

la marchandise pour la vendre. Grâce à un procédé de réservation en temps réel des produits *en mouvement* dans la chaîne logistique, le distributeur est capable de calculer les délais de livraison et de s'engager auprès de ses clients tout en leur faisant profiter de conditions tarifaires d'autant plus avantageuses que la commande est passée tôt dans la chaîne logistique. Le résultat est une efficacité inégalée dans la conduite des actions de vente et dans le pilotage de l'organisation logistique :

- Vente en *push* via des campagnes de marketing efficaces afin de recruter rapidement des clients sur les produits à l'approche,
- Organisation logistique intégrée afin d'acheminer les produits déjà vendus directement de l'usine au client sans stockage intermédiaire.

Des fonctionnalités avancées pour un système souple et puissant

La version du connecteur actuellement disponible entre Open ERP et eZ publish, deux solutions Open Source matures et leader dans leurs domaines respectifs, offre à ce jour le panel de fonctionnalités suivantes :

- création et mise à jour automatique des produits et quantités disponibles, saisis dans Open ERP et importés dans eZ publish (en temps réel via des mises à jour unitaires ou en différé via des mises à jour par lots),
- synchronisation en mode connecté de l'état des stocks et des conditions tarifaires de Open ERP vers eZ publish afin de calculer en temps réel les meilleures conditions, pour un client donné et en fonction de la position d'un produit dans la chaîne logistique,
- *import bi* directionnel et la mise à jour des bases de données clients entre eZ publish et Open ERP : création automatique des comptes clients depuis eZ publish vers Open ERP et vice et versa, rattachés ou non à une société d'appartenance, des adresses associées (facturation, livraisons), de leurs comptes comptables respectifs avec rattachement à chaque compte client de ses commandes et factures. Open ERP offre des fonctionnalités poussées de CRM et eZ publish offre un portail de *self care* aux clients internautes,
- interconnexion des formulaires de contact depuis eZ publish vers le module de gestion des cases (CRM) d'Open ERP :

P U B L I C I T É



Gérez vos campagnes de NetLinking avec Yooda Submit



Améliorez le Page Rank et le Positionnement de vos pages !



Performant

Soumettez manuellement votre site dans un nouvel annuaire en moins de 2 minutes !



Intelligent

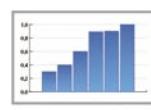
Disposez d'un Agent qui vous trouve des annuaires qualifiés sur votre thématique.

Téléchargez votre campagne d'essai sur www.yooda.com



Collaboratif

Bénéficiez d'une base d'annuaires enrichie et qualifiée par l'ensemble des utilisateurs.



Professionnel

Disposez enfin, d'un logiciel de soumission adapté au métier du référencement.

Yooda Submit trouve des annuaires sur votre thématique et vous accompagne dans la mise en place de liens de qualités.

- toute demande de devis ou d'information postée sur un formulaire disponible sur le site est automatiquement transmise et archivée au sein du module de CRM d'Open ERP, et rattachée automatiquement au compte client s'il existe dans la base (à partir de la reconnaissance de l'email de réponse saisi dans le formulaire). Toutes les demandes qui sont faites et les réponses retournées depuis Open ERP à travers l'email gateway sont ainsi historisées et conservées,
- import bi directionnel des commandes multi paiement et multi transporteurs entre eZ publish et Open ERP : en vue d'un traitement aval automatique : ordre logistique, facturation, comptabilité, gestion des stocks,...
- création automatique des factures au sein d'Open ERP, avec les frais de ports afférents, calculés au coût le plus avantageux, et tenant compte des remises attribuées aux clients disposant de codes promotion au sein du module de gestion de remises d'eZ publish,
- validation des ordres d'approvisionnement entraînant la validation automatique de la demande d'expédition des colis,
- prise en compte de la taxe relative à l'éco-contribution pour tous les produits électroniques,
- gestion de colisages multiples en s'appuyant sur les fonctionnalités d'Open ERP : à la pièce, au carton, à la palette ... en fonction des quantités commandées par le client et des conditionnements proposés par les fournisseurs,
- calcul des frais de port en fonction du poids de la commande, mais également de ses dimensions physiques (volume, encombrement ... relatifs aux quantités commandées et au conditionnement proposé en conséquence), du statut du client (professionnel / particulier), de sa localisation géographique ...,
- génération le cas échéant des bordereaux d'expédition, grâce à une interconnexion avec les logiciels *Expéditor Inet*, *UPS* ou *Chronopost* notamment,
- réimport des numéros de colis créés (compatible avec le scanner d'une douchette de prix), associés avec le numéro de facture généré par Open ERP,
- mise à jour des stocks disponibles,
- validation définitive de la commande avec changement d'état (*Livrée*) dans le process d'achat d'eZ publish, et le rapatriement du numéro de facture et de colis au sein du compte du client,
- gestion des règlements des commandes : paiement par carte bancaire en 1 ou 3 fois, abonnement, plan de financement, contre remboursement, mandat, virement bancaire, par encours pour les clients professionnels en compte. Les diverses modalités de règlements sont gérées à la fois dans eZ publish et dans Open ERP : des marqueurs sur le nom des commandes sont générés à la volée de telle sorte de pouvoir facilement identifier chacun d'entre eux pour le traitement informatique (*workflow* adapté), effectuer la réconciliation des comptes bancaires et en extraire in fine aisément des statistiques de ventes selon les différentes méthodes, zones géographiques, types de clients, de produits.... Les encours clients professionnels sont plafonnés et vérifiés en temps réel au moment de la validation de la commande,
- paiement multi devises : euro / dollar,
- workflow* d'annulation de commande / délai de paiement dépassé, avec le lettrage comptable associé dans Open ERP (annulation d'écriture), restitution des stocks disponibles..., dans Open ERP avec mise à jour en temps réel dans eZ publish,
- gestion des remises et codes promotion qui ne s'appuie uniquement désormais que sur le module de *price lists* proposé par Open ERP. Pour chaque commande, eZ publish propose au client le meilleur prix possible. Pour ce faire, il s'appuie sur le résultat du calcul combinatoire effectué par Open ERP tenant compte :
 - des quantités commandées (système de prix dégressif) ou du délai restant avant arrivée de la marchandise en France (vente éclair à prix cassés),
 - des codes promotions génériques (attribués à un produit donné, une gamme ou une typologie de clients spécifiques : grossistes, apporteurs d'affaires et revendeurs) et des remises personnalisées (bons clients),
 - de packs promotionnels de produits.

Fonctionnement général de la plate-forme

Ce connecteur pleinement bidirectionnel permet de profiter intégralement des possibilités de chacune des deux solutions : d'un côté la gestion de contenu éditorial, la vente et le suivi en ligne des commandes avec eZ publish, et de l'autre, tout ce qui a trait aux opérations logistiques et financières, ainsi qu'au management de la relation client (gestion des contacts, tickets et notes, devis, mailings ...) avec Open ERP.

Open ERP centralise désormais les bases de données relatives aux clients et aux produits et renvoie ses informations vers eZ publish en mode connecté. La création et la mise à jour des fiches produit s'effectue dans Open ERP (dénomination, référence, code EAN, fournisseur, prix, quantités disponibles ...). à l'exception du contenu éditorial (descriptif détaillé, photos, animations ...), disponible dans eZ publish.

Un *workflow* de validation permet de ne publier que les produits dont l'ensemble des informations descriptives ont été renseignées. Les fiches produit sont créées dans Open ERP par le service sourcing qui renseigne toutes les conditions d'achat (informations fournisseur, coûts, délai d'approvisionnement, conditionnement ...).

Un numéro de référence unique est automatiquement assigné à chaque produit par Open ERP en respect de la normalisation EAN13. Le service marketing complète les fiches produit dans Open ERP avec les dénominations commerciales, les argumentaires de ventes et applique une stratégie de prix via l'assignation de listes de prix qui peuvent être propres à un client ou à un groupe de clients.

Les fiches produits sont validées au sein d'Open ERP avant d'être exportées automatiquement vers le catalogue produit d'eZ publish. L'équipe en charge du site e-commerce complète, au niveau d'eZ publish, le contenu éditorial des fiches produit (photos des produits, descriptifs détaillés, options, produits associés, accessoires ...) avant de valider leur publication en ligne.

La gestion des stocks est quant à elle exclusivement confiée à Open ERP. Les stocks sont tenus à jour dans Open ERP, et les délais de livraison calculés en temps réel en fonction du statut de chaque produit dans la chaîne logistique. Les produits peuvent être localisés dans des stocks réels ou dans différents stocks virtuels (en cours de transit dans la chaîne logistique) : en cours de fabrication usine, en stock départ usine (dépôt étranger), en cours de transport...

Les produits sont affectés à des unités logistiques individuellement répertoriées et des jalons de temps sont introduits permettant de disposer de différents stocks virtuels en cours

Sur Internet

- <http://www.ez.no/> – Site officiel de la société norvégienne eZ systems, éditeur d'eZ publish,
- <http://www.internethic.com/solutions/ezpublish/> – Pour davantage d'informations en français sur les fonctionnalités offertes par la solution eZ publish,
- <http://www.openerp.com/> – Site officiel de la société belge Tiny SPRL, éditeur d'Open ERP,
- <http://www.internethic.com/solutions/open-erp/> – Pour davantage d'informations en français sur les fonctionnalités offertes par la solution Open ERP,
- http://www.ez.no/community/contribs/applications/TinyERP_connector_1/ – Pour télécharger la v1 du connecteur eZ publish-Open ERP,
- <http://www.bainetspa.fr/> – Site e-commerce de la société Prodig, premier client utilisant le connecteur eZ publish-Open ERP v2.

de fret, et par là même d'afficher des délais de livraison les plus proches de la réalité, ainsi que le delta différentiel entre prévision et date réelle de livraison.

Lors d'une commande passée en ligne, eZ publish s'assure en temps réel au cours de la cinématique d'achat de la disponibilité des produits commandés, en interrogeant Open ERP en mode connecté. Les stocks sont réservés en ligne dans Open ERP dès validation du panier puis décrémentés après confirmation du paiement. Un process de *workflow* d'annulation de commandes ou de délai de paiement dépassé a également été développé afin que les produits des commandes annulées soient remis dans le circuit logistique, au sein du stock virtuel de destination.

Conclusion

Avec ce type de plate forme automatisée et verticalisée, de jeunes pousses sont parfaitement à même de rivaliser avec de grandes enseignes, avec des moyens humains considérablement moindres et une politique qualité équivalente si ce n'est encore meilleure. Certaines d'entre elles génèrent déjà plusieurs centaines de milliers d'euro de CA mensuel avec un taux de croissance moyen de 25 à 30% chaque mois.

Avec des équipes de moins de 10 personnes en moyenne pour gérer l'intégralité de leur ac-

tivité (marketing / communication, gestion des fournisseurs, logistique et expéditions, relation client...), ces distributeurs nouvelle génération expédient plusieurs dizaines de milliers de commandes annuelles, à destination d'autant de clients.

En parallèle, certains grands distributeurs nationaux ont également manifesté leur intérêt pour la mise en place d'une plate-forme globale de ce type leur permettant de gérer, via une interface unique, les activités (stocks, clients, vente en ligne / vente directe via caisses enregistreuses, CA, marges...) de leur Groupe ainsi que celles de leur réseau de franchises.

Afin d'aller encore toujours loin dans la satisfaction des exigences spécifiques de ces distributeurs, un certain nombre d'évolutions majeures doivent être apportées à la solution d'ici à l'été 2009. Il s'agira notamment d'améliorer les process logistiques (gestion de stocks physiques multiples – entrepôts, showrooms..., gestion de la vente en contre marque / marque blanche), l'ergonomie générale du site (via un moteur d'indexation et de navigation contextuelle) et d'optimiser les temps de traitement informatique des opérations, certains sites connaissant désormais un trafic mensuel de plusieurs centaines de milliers d'internautes.

En conséquence, la 3ème génération de ce connecteur devrait être prochainement conçue sous licence Open Source GNU/GPL, afin d'en faire profiter les communautés de ces deux applications et favoriser encore davantage son essor.

CYRIL FOUILLOT, SAMUEL PACCOUD

Cyril Fouillot est co-fondateur de la société Internethic (<http://www.internethic.com>), société spécialisée dans l'intégration sur mesures de solutions Open Source, et co-fondateur de la société Atomes (<http://www.atomes.com>), société spécialisée dans l'hébergement managé de solutions Open Source, notamment d'applications Java/J2EE et de plates-formes conçues sous eZ publish et Open ERP. Internethic est une SSL marseillaise, contributrice officielle de la solution eZ publish depuis 2002 et reconnue sur le plan international comme disposant d'une des plus fortes expertises techniques sur cette solution. Internethic est partenaire officiel des solutions eZ publish et Open ERP et apporte à leurs communautés respectives une solution e-commerce verticalisée (CMS – ERP), en collaboration étroite avec les éditeurs.

Samuel Paccoud est co-fondateur et responsable de développement du Groupe Prodigg. Dans ce cadre, il assure notamment la promotion et l'animation des sites des filiales du Groupe, dont www.bainetspa.fr, premier site e-commerce à avoir utilisé la solution verticalisée eZpublish – Open ERP.

P U B L I C I T É



Logiciel SeeUrank 3

Analyse & suivi de référencement

Test gratuit
30 jours

- Analysez votre référencement
- Automatisez vos audits
- Editez vos rapports personnalisés



- 1 Analyse du suivi de positionnement
- 2 Relevé des backlinks
- 3 Analyse concurrentielle des sites
- 4 Densité des mots clés / Analyse des liens & des images
- 5 Pages connues dans les moteurs
- 6 Suggestion de mots clés

Bien choisir sa boutique en ligne

Vendre en ligne c'est urgent, simple et très rentable si vous faites les bons choix. Voici un panorama des types de solutions existantes, et des critères à prendre en compte pour choisir et passer à l'action. Suivez le guide...

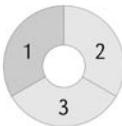
Cet article explique :

- Les types de solutions existantes de boutique en ligne.
- Les critères à prendre en compte pour faire un bon choix.

Ce qu'il faut savoir :

- Rien de particulier si vous optez pour une solution commerciale.
- Faire un site web sinon.

Niveau de difficulté



Pour commencer, nous allons vous aider à choisir pour quel type de solution de vente opter. Ensuite, nous allons vous présenter les critères qui vont vous permettre de faire un choix précis.

Panorama des types de solutions existantes

Débutons donc par défricher les possibilités qui s'offrent à vous, pour créer votre boutique en ligne. Les possibilités sont nombreuses, on peut les synthétiser en trois catégories : développer soi-même, prendre une solution Open Source gratuite, ou une solution commerciale clé en main.

Développer soi-même

Dans ce magazine nous avons l'amour infini de la programmation, alors commençons par étudier le fait de développer soi-même sa solution E-Commerce. Il y a quelques années, le cahier des charges d'un développement-maison d'une solution de vente en ligne aurait été assez succinct, et sa réalisation possible en moins d'un mois. Il se serait résumé à mettre en place un système de panier, de demande de coordonnées et d'envoi à la plate-forme de paiement.

Aujourd'hui le marché devient plus mature, la concurrence entre sites e-commerce plus forte, et vendre en ligne requiert des fonctionnalités de plus en plus évoluées : *URL-rewriting*,

coupons, remises à critères multiples, cadeaux, comptes client, suivi des colis en ligne, différenciation des familles de clients, exportation vers les comparateurs de prix, stocks, factures, mises en avant d'articles, la liste est encore longue...

Il convient donc d'accepter que ce genre de développements va vous nécessiter beaucoup d'investissement soit en temps soit en argent. Mais cela peut être indispensable si vous avez des besoins très particuliers non couverts par des solutions existantes. Développer soi-même peut donc être intéressant si vous avez beaucoup de temps libre ou beaucoup de moyens financiers. Mais disons le tout net, cette approche est rédhibitoire pour la majorité des futurs e-commerçants car difficilement rentable.

Solution Open Source gratuite

Une idée a priori bonne est donc de s'orienter vers le gratuit. Cela semble idéal : aucun coût, une communauté souvent active, des contributions parfois riches et variées. De façon pratique, il y a cependant quelques inconvénients :

- il va quand même falloir prévoir un peu de budget pour l'hébergement, voire beaucoup de budget pour des solutions lourdes,
- vous allez sûrement passer de nombreuses heures à mettre en place votre boutique, à en faire le paramétrage et cela n'est souvent pas une mince affaire. Face à vos nombreuses questions, vous allez épucher les forums dédiés à la solution retenue, ce qui va vous occuper un bon moment,
- il va vous falloir devenir spécialiste *ad-hoc* de la solution, pour résoudre les soucis rencontrés ou parvenir à vos objectifs de fonctionnalités (mise en place du paiement CB, personnalisation graphique, frais de port parfois pas tout à fait adapté au marché européen, etc,

• assistance : disons le tout net : vous allez être seul, vraiment seul. Hourra, il y a probablement une communauté active et très sympathique, mais vous n'aurez pas un numéro de téléphone à appeler avec quelqu'un qui saura répondre en 2 minutes à tous vos problèmes. Il vous faudra parcourir les forums, appliquer des éventuels *patches* à divers étages de votre édifice, et éventuellement *bidouiller* dans le code source, en espérant ne pas avoir à appliquer de futures mises à jour qui ruinerait tout cela.

Conclusion : l'Open Source est une réelle première étape vers le E-Commerce, mais il faut être conscient à l'avance que cela va vous nécessiter un bon nombre d'heures et des compétences techniques. Votre temps c'est de l'argent (pendant ce temps là vous n'avez rien vendu n'est-ce pas?), alors n'auriez vous pas gagné plus d'argent dans votre tirelire en passant par une solution commerciale clé en main ?

Solution commerciale clé en main

A priori, la solution commerciale n'est pas votre amie, fidèle lecteur. Mais penchons-nous dessus malgré tout. Cela a un coût... Oui mais on trouve maintenant des solutions éprouvées pour une somme mensuelle souvent modique et vous allez pouvoir commencer à vendre en ligne très rapidement car vous allez avoir affaire à un spécialiste dont *lesupport-a-réponse-a-tout-merci*. La rapidité de mise en œuvre, l'assistance, des nouvelles fonctionnalités régulières pour rester à la pointe sans effort, voilà qui va probablement largement compenser le prix, et être beaucoup plus rentable que les heures de travail des autres solutions.

Attention cependant : la qualité des prestataires est très inégale, un comparatif rigoureux s'impose. Le prestataire a-t-il une réelle expérience ? Des



Raynette

Solution E-commerce d'Excellence clé en main

Retrouvez-nous sur

▶ www.raynette.fr

C'est la crise, vendez en ligne !



Boutique à la carte

Votre boutique adaptée à votre budget : vous ne prenez que les options nécessaires

- Paiement CB
- Hébergement
- Assistance
- Mises à jour

- Pack FIRST
- +
- Optimisation du référencement
- Promotions
- Nouveautés
- Livraisons
- Suivi Colissimo
- Remises, coupons cadeaux, prix dégressifs

- Pack PRO
- +
- Factures
- Catégories multiples d'article
- Coups de cœur
- Recommandation d'articles
- Envoi de SMS

Boutique à la carte

Formule de base incluant paiement CB hébergement, assistance mises à jour régulières + Votre sélection de modules optionnels.

Modules disponibles

Promotions
Nouveautés
Optimisation du référencement
Livraisons
Suivi Colissimo
Remises, coupons, cadeaux
Articles coups de cœur
Recommandation d'articles
Catégories multiples d'article
Stocks
Stats multiples de ventes

Envoi de SMS
Factures clients
Marques
Familles clients
Familles d'articles
Articles: délais de disponibilité
Articles: modèles d'options
Fournisseurs
Ecotaxe (DEEE)
Export vers les comparateurs de prix

Importation d'articles
Vente d'articles à télécharger
Liste des visites en cours
Monnaie (au choix)
Langues (au choix)
Suivi Chronopost / UPS / Fedex
Paiements par CB (banques au choix)
Paiement Paypal
Paiements par virement
Paiements avec 1EURO.COM
Licence 5 à 50 administrateurs

► Un accompagnement dans votre développement, en vous apportant des services de qualité irréprochable et d'une fiabilité absolue.

Spécialiste E-Commerce depuis 1998 sur Internet, plus de 1600 clients nous font confiance. Notre expérience et notre savoir-faire sont à votre service pour mettre à votre disposition un service complet, des conseils d'expérience, et une boutique en ligne fiable, puissante et intuitive.

Notre mission : vous permettre par nos produits et services, d'accéder à vos objectifs de croissance et ainsi aider à votre succès sur Internet.

références dignes de ce nom ? Le support technique est-il réactif et performant ? Les ajouts de fonctionnalités sont-ils réguliers ? Une réponse négative à l'une de ces questions élimine tout intérêt d'une solution commerciale face à l'Open Source.

Synthèse des types de solutions existantes

Vive le développement-maison ! On aime cela, mais le travail à faire est titanique pour correspondre aux attentes de votre public, et arriver à la cheville de vos concurrents existant et à venir. À choisir, si vous avez énormément de temps ou des besoins très particuliers non disponibles ailleurs.

Vive les solutions Open Source gratuites ! Elles sont souvent assez complètes si récentes, mais elles vont vous coûter un investissement personnel fort. À choisir si vous avez le temps et les compétences.

Vive les solutions commerciales clé en main ! Toutes prêtes, rassurantes, et avec assistance. À choisir si vous voulez faire efficace. Mais attention on y trouve le meilleur comme le pire, prenez le soin de bien comparer. Dans ce panorama, vous trouverez sûrement votre bonheur en fonction de vos besoins.

Des critères de choix

Maintenant que vous avez les éléments pour savoir quel type de solution prendre, il vous faut choisir une offre précise. Voici les critères les plus importants pour faire votre choix.

Le coût & le temps

Cela n'a l'air de rien, mais le temps que vous allez passer à préparer votre boutique va vous coûter plus que vous ne pensez, car c'est du temps perdu à ne pas vendre vos produits/services, et donc des sous dont vous ne verrez jamais la couleur. Le temps passé dépendra en bonne partie de l'assistance disponible ou pas.

Efficacité : le référencement

On aurait pu mettre ce critère en premier, car sans trafic, point de ventes. Pour maximiser vos chances d'obtenir un bon référencement, voici les points importants :

- Un nom de domaine différent par langue, c'est beaucoup mieux.
- *URL-rewriting* : pour obtenir des URL *statiques* propres que les moteurs de recherche classent généralement beaucoup mieux que les URL dynamiques contenant quantité de paramètres.
- Possibilité de modifier les meta tags (titre, description, mots-clés) pour chaque page, catégorie, article.
- Génération d'un *sitemap* si possible (fichier listant toutes les URL de votre site, à transmettre à Google pour un référencement plus rapide et complet).

La modularité

Est-il utile que vous ayez dans l'administration de votre boutique des options partout, des

champs à remplir à profusion et autres gadgets qui vous sont personnellement inutiles ? Cela risque de vous noyer au début, et vous y verrez moins clair dans vos tâches quotidiennes ensuite. Si possible, choisissez une solution où vous pouvez n'avoir que les fonctionnalités dont vous avez besoin (et les autres pouvant être ajoutées au fur et à mesure de votre évolution), ce sera bien plus sympathique et efficace.

La montée en puissance

Un site qui se met à bloquer car la solution E-Commerce devient vite trop lourde, ou parce que l'hébergement est trop limité, et c'est la catastrophe assurée. Assurez-vous que le jour prochain où vous aurez moults visiteurs et commandes, votre boutique tiendra le choc victorieusement.

L'évolution

Pour l'instant vous démarrez votre activité de vente en ligne, vos besoins sont donc probablement réduits, mais cela ne va pas durer. Gouverner (votre boutique) c'est prévoir. Assurez-vous que la solution choisie a des possibilités d'évolution de préférence vastes. Ainsi votre ascension (fulgurante bien entendu) ne se trouvera pas entravée.

Les moyens de paiement disponibles

Vérifiez dès le départ que la solution retenue permet de faire des paiements CB avec votre banque habituelle. On rencontre surtout ce genre de surprises avec des solutions Open Source d'origine anglophone (voire même des solutions commerciales bien de chez nous, sic!).

L'assistance

Le Saint-Graal de l'efficacité de votre activité, qui vaut son pesant d'or. On ne se rend compte de son importance qu'au moment où on se retrouve nez à nez avec sa boutique en ligne bloquée qui ne veut rien savoir, et un superbe chiffre d'affaires égal à zéro. Soit la solution retenue a une assistance rapide et efficace, soit vous avez une connaissance ou un prestataire qui pourra vous assurer cela. À vérifier dès le départ.

Efficacité : les leviers marketing

Vous allez apprécier (ou pester si vous ne les avez pas) les possibilités :

- d'envoyer un mailing personnalisé à vos clients,
- de créer des promotions avec prix barrés (sinon quel intérêt de faire des promotions),
- d'afficher éventuellement le prix conseillé par le fabricant de vos articles, et votre petit prix à vous avec l'économie réalisée par le visiteur (effet garanti...),
- de créer des coupons de réductions pour relancer des clients qui ne commandent plus chez vous depuis trop de temps,
- de pouvoir exporter vos articles chez des comparateurs de prix. Cela peut apporter un volume de ventes conséquent,

- de créer des groupes/familles de clients (revendeurs) pour qu'ils aient des prix différents,
- d'offrir un cadeau ou les frais de port à partir d'un certain montant de commande. Cela aide à augmenter le panier moyen. Et si c'est bon pour le panier moyen, c'est bon pour vous,
- d'établir des règles de remise diverses et variées.

Les outils marketing sont ce qui évolue le plus dans la vente en ligne, il est de bon ton de vérifier si la solution retenue évolue régulièrement sur ce point.

L'ergonomie

Si votre boutique fonctionne bien, autant vous dire que vous allez passer du temps dans votre administration de boutique, à traiter les commandes. Il est donc préférable que ce soit bien organisé, clair, et automatisé, vous gagnerez du temps.

Les possibilités de design

Forcément, vous avez soit une idée bien précise de l'aspect que vous voulez donner à votre boutique en ligne, soit les exigences vont arriver au fur et à mesure. Il y a 2 écoles pour la personnalisation graphique : les kits graphiques tout prêts où les possibilités de personnalisation sont forcément limitées (mais plus rapides à faire), et les *templates* (pages HTML modèles) entièrement modifiables qui permettent d'exploiter au maximum les possibilités de design et d'éviter que votre boutique ressemble un peu trop à celle du voisin. Chaque système a ses avantages et ses inconvénients, l'important est de savoir que ces deux choix existent.

Conclusion

En ces temps de crise, le e-commerce ne connaît pas la crise. Vendre en ligne et en direct ce sont des marges plus confortables, de la trésorerie, et un investissement qui peut être minime. Vendre en ligne dès maintenant, c'est prendre des places avant vos concurrents (il est grand temps de s'y mettre).

Ne reste plus qu'à bien choisir votre boutique en ligne, l'important étant d'opter pour une solution qui répondra aujourd'hui et demain à vos besoins et impératifs. Les solutions e-commerce évoluent vite, le choix est maintenant large, et cet article vous permettra de vous faire une opinion personnelle sur les possibilités que vous rencontrerez. En vérifiant les critères qui vous sont importants parmi la liste ci-dessus, vous trouverez votre bonheur, le succès qu'on vous souhaite, et une place au soleil (qu'on vous souhaite également !).

SÉBASTIEN JONCHERAY

L'auteur est le fondateur de la société RAYNETTE (www.raynette.fr), spécialisée e-commerce depuis 10 ans, avec 1600 clients. Ingénieur de formation, il privilégie la qualité technique de ses solutions et a une vaste expérience du e-commerce. Il est par ailleurs l'auteur technique du moteur de recherche Mozbot.fr en partenariat contractuel avec Google, avec ses comparses et amis de Abondance.com et Brioude Internet Référencement.

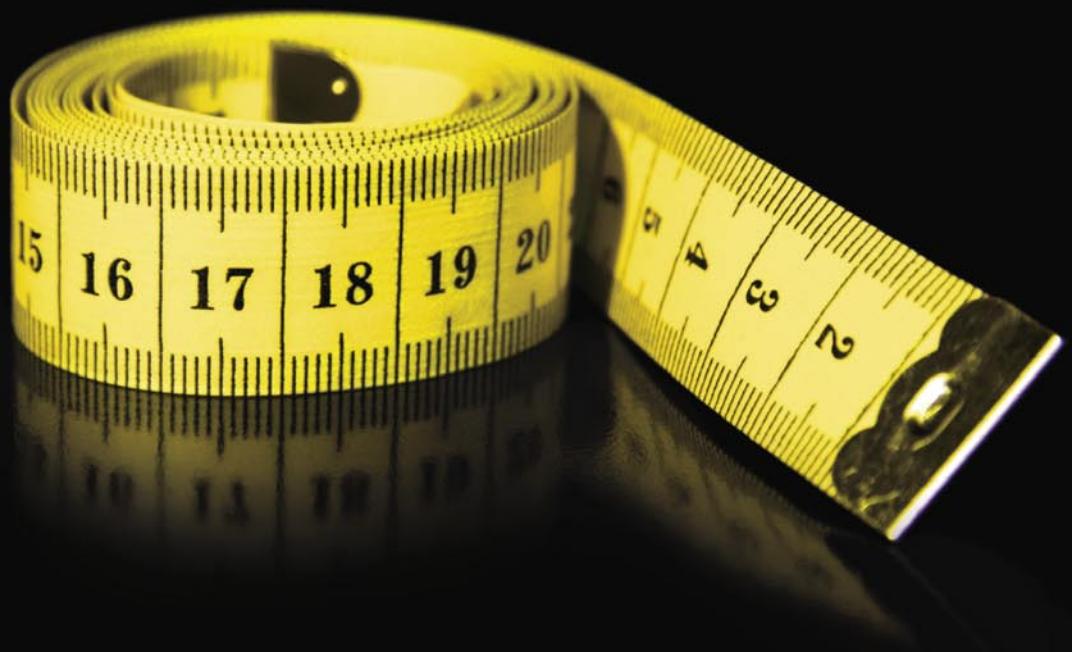
Un projet Web d'envergure ?

Site communautaire, boutique e-commerce, intranet, extranet...



Optez pour le sur mesure !

Web design, développement, hébergement, e-marketing...



Kernix est une société d'experts en développement d'applications en ligne. Nous proposons des solutions performantes et accompagnons nos clients tout au long de leur projet web : réflexion et stratégie en amont ; web design, ergonomie et développement ; hébergement et promotion en ligne ; maintenance et évolutions.

Pour en savoir plus, rendez-vous sur www.kernix.com

Créer sa boutique en ligne avec AMEN

Voilà, depuis quelques temps, que trotte dans ma tête, une envie folle de créer ma propre boutique en ligne. Mais avant de foncer tête baissée dans la mise en place de mon e-commerce, il m'a fallu répondre à plusieurs interrogations.

- Allais-je vendre un ou plusieurs produits ?
- Combien de produits vais-je proposer en ligne 10, 100, 1000... ?
- Comment intégrer une solution de paiement par CB ?
- La mise à jour des produits sera-t-elle fréquente ?
- Faudra-t-il que je gère mon stock depuis mon site ?
- Quels types de livraisons vais-je proposer ?
- Quelles seront les modalités de paiement ?
- Devrais-je gérer la TVA ?
- La vente se limitera-t-elle à l'hexagone ou bien à d'autre pays ?
- Etc...

Autant de questions qui doivent trouver impérativement une réponse, car ce sont d'elles que dépendra la réussite de mon projet. Les réponses ont été trouvées, certes après m'être creuser un peu la tête, reste à créer ma boutique en ligne, là plusieurs solutions s'offrent à moi :

- le faire moi-même de A à Z, c'est vrai que cette solution offre l'avantage que le site

soit le reflet exact de mon projet, par contre il n'est pas prêt de voir le jour même en prenant un logiciel dédié (Open Source ou propriétaire)... car voilà, pour cette option il faut être un *crack* de la programmation et je suis loin de l'être !

- le faire réaliser par une *Société de Services et d'Ingénierie Informatique*, une SSII pour faire plus court, dans ce cas j'explose mon budget,
- idem pour les agences Web, ces agences sont spécialisées dans le développement et la création de site Internet dans ce cas l'investissement de départ est trop important,
- reste alors les plates-formes e-commerce. Ces dernières proposent une solution *clé en main*, elles fournissent différents modèles de sites ainsi que l'hébergement de ce dernier. Il ne vous reste plus qu'à vous concentrer sur le contenu, le style. Parmi le nombre impressionnant de ces plates-formes, mon choix s'est porté sur le site *Amen.fr* <http://www.amen.fr>, ce site offre une multitude de services dont, entre autre, des hébergements professionnels dédiés ou privés, des noms de domaine, des courriers électroniques, et enfin chose qui m'intéresse des solutions e-commerce qui peuvent être testées gratuitement pendant 30 jours. Le site propose trois *packs* ; parmi ces derniers je suis sûr que vous trouverez votre bonheur, pour ma part j'ai je-

té mon dévolu sur le pack *e-commerce Premium*.

Je passe donc commande pour le pack *e-commerce Premium*, après avoir donné lors de ma commande le nom de domaine de mon site de vente en ligne, je reçois un mail m'informant que mon site est prêt à être en ligne, ainsi que l'identifiant et le mot de passe.

Je me connecte donc à l'adresse fournie, je tape mon identifiant ainsi que mon mot de passe.

Une page web s'affiche me proposant une collection importante de modèles triés pas type. Holà, je n'aime guère être sollicité à froid, je reviendrai sur les modèles ultérieurement.

Voilà, je clique sur *Ok* et je me retrouve sur le page d'administration de ma boutique.

La page se décompose en trois zones :

- Dans la partie haute on trouve différents onglets Commandes, Clients, Produits, Catégories, Style, Marketing, Paramètres, Aide,
- Dans la partie latérale gauche, on trouve les différentes opérations susceptibles d'être effectuées sur l'onglet sélectionné,
- Dans la partie à droite on trouve les données des différentes opérations.

Les onglets sont au nombre de 8 :

- *Commandes* : qui permet la gestion des commandes, des clients ainsi que la création

The screenshot shows the main administration dashboard of Amen.fr. At the top, there is a navigation bar with links for Commandes, Clients, Produits, Catégories, Style, Marketing, Paramètres, and Aide. The Amen logo is in the top right corner. The main content area has several sections:

- Milestones:** Includes links for Page d'accueil, Shop-Administrator, and Me déconnecter.
- Optimisation (Inactive):** Shows that the optimization is inactive.
- Menu contextuel:** Includes links for Assistant de création.
- Classeur:** Shows that the folder is empty.
- Favoris:** Shows a star icon.
- Historique:** Shows a history icon.
- Bienvenue à la page d'administration de votre boutique !** (Welcome to the administration page of your store!) with a help icon.
- Utilisez les modules suivants pour accéder aux fonctions de votre boutique.**
- Paramètres e-mail incomplets:** A warning message stating that the system cannot send emails because no sender address is specified in the email settings. It suggests entering a sender address.
- Utilisez les paramètres pour l'optimisation de la vitesse d'affichage.** A note stating that optimization is not activated for this store. It recommends configuring settings to increase page display frequency.
- Commandes:** Describes managing client commands and creating documents (invoices, spreadsheets, etc.).
- Produits:** Describes preparing product presentation using text and images, organizing products by type, price lists, and search statistics.
- Style:** Describes creating the store using different models, applying styles to images, colors, and navigation elements.
- Paramètres:** Describes configuring the system and command process according to the store's needs.
- Assistant de création:** Describes using the assistant to create the store in several steps.
- Clients:** Describes the client database containing all addresses and criteria for client groups or purchase volumes.
- Catégories:** Describes organizing products into categories, completing the site with promotional products, and setting up delivery conditions.
- Marketing:** Describes advertising offers on the Internet, using bulletin boards, coupons, and product portals.
- Aide:** Describes assistance, including first steps and an online help section.

Figure 1. Page d'administration

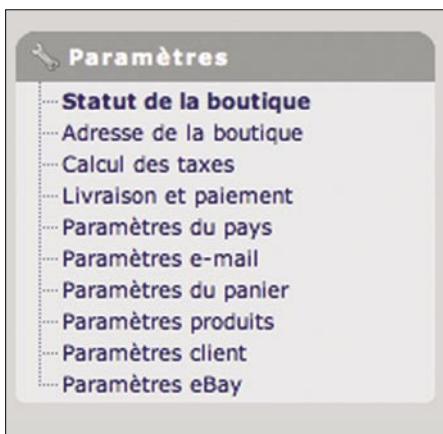


Figure 2. Les options de Paramètres

- des divers documents administratifs, par exemple les factures, bordereaux, etc,
- *Clients* : qui contient toutes les données d'accès des clients, permet également des recherches sur différents critères,
 - *Produits* : qui permet la gestion des produits, ainsi que la présentation des fiches produits à l'aide de texte et d'images,
 - *Catégories* : qui permet l'organisation des produits en catégories,
 - *Style* : qui permet la personnalisation de votre site à l'aide des différents modèles, ainsi que la création de modèles,
 - *Marketing* : qui permet d'annoncer ses offres sur Internet et de référencer sa boutique en ligne sur des moteurs de recherche,
 - *Paramètres* : qui permet de configurer le système (calcul des taxes, paramétrages des pays, les types de livraisons et de paiement, l'adresse et l'email de la boutique,
 - *Aide* : qui permet de bénéficier en permanence d'une assistance lors de votre travail
 - *Assistant de création* : si toutes ces options vous affolent, l'assistant prendra en charge entièrement la création de votre bouti-

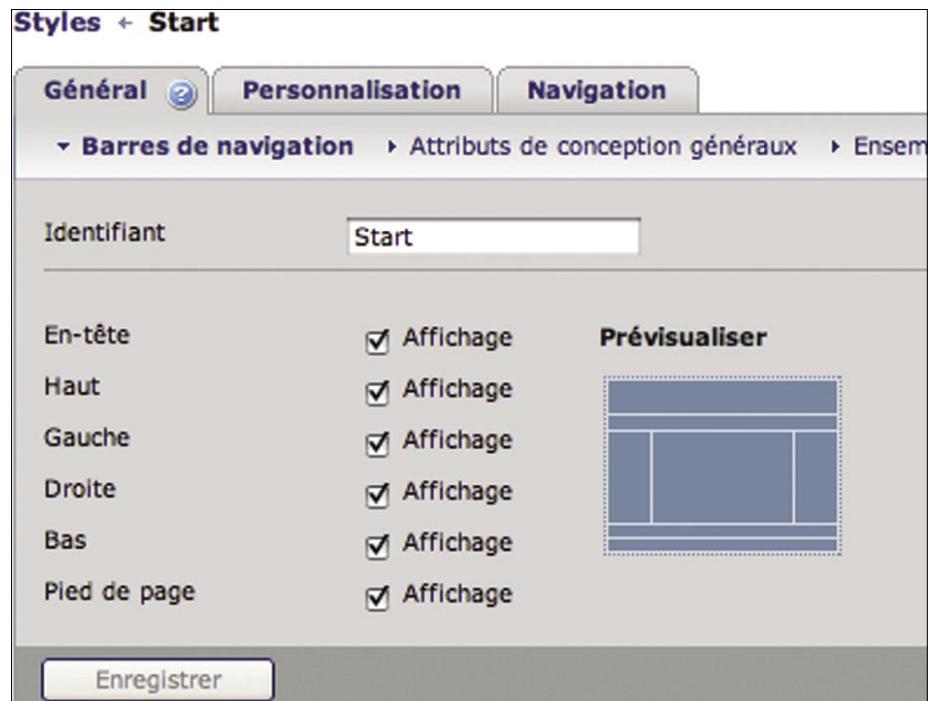


Figure 4. Affichage de différentes parties de la page

que. En quelques étapes et quelques clics de souris, votre boutique verra le jour et sera opérationnelle.

Voilà, je viens de survoler toutes les catégories de fenêtre, maintenant, il me reste plus qu'à trouver un style, un nom, un slogan, en résumé un leitmotiv représentatif de ma boutique. Je commence donc par lui donner une identité à l'aide de l'onglet *Paramètres*. Dans les options de ce dernier.

Je clique donc sur l'option *Statut de la boutique*, là je vais commencer par lui trouver un nom, un slogan, un message pour avertir quand la boutique sera fermée, un logo, paramétrier l'optimisation de l'affichage des pages. Ensuite dans *Adresse de la boutique*, je saisir le nom de la société, le responsable (moi entre autre) nom et prénom, l'E-mail, le téléphone, le Fax, l'adresse. Dans *Calcul des taxes* les taxes. Dans *Livraison et paiement* je saisir les différents modes de livraisons ainsi que les modes de paiement, et ainsi de suite pour les paramètres du pays, de l'e-mail, du panier, des produits, des clients, de eBay.

Voilà, ma boutique a son identité, il me reste plus qu'à lui trouver un style, et là j'ai l'embarras du choix, il y a qu'à en juger par l'abondance de modèles proposés dans l'onglet *Style*.

Ces derniers sont classés par *Branches* (*Technologie, Santé, Livres, Informatique*, etc...) ou bien par *Apparence et convivialité* (*Elégant, Sportif...*).

Ensuite, il me suffit de déterminer si je veux afficher les différentes parties de ma page, en-tête, haut, bas, gauche, droite, bas et pied de page. Reste plus qu'à personnaliser tout ceci en y ajoutant, la police des textes, les images des dif-

férents fonds et ceci par le biais de l'onglet *Personnalisation*. Et par l'onglet *Navigation* ajouter ou supprimer les différents liens comme *Lien vers la page d'accueil*, *Lien vers les coordonnées*, *Lien vers les conditions générales de vente*, etc..., mais également ajouter des liens qui ne figurent pas dans la liste.

Voilà, ma boutique est créée, reste maintenant à mettre en place ma vitrine et mon stock, pour cela l'onglet *Produits* me sera d'un grand secours, il me permettra de saisir le descriptif de mes produits, le nombre de pièces en stock, les seuils d'alerte, leur prix catalogue, leurs dimensions et leurs poids, les adresses des fournisseurs. Je pourrai également créer des lots de produits. À tout moment, je peux revenir sur les produits et en modifier les caractéristiques.

Mes produits sont maintenant dans ma boutique, reste plus qu'à faire la promotion de cette dernière, c'est là qu'intervient l'onglet *Marketing*, il me permet de créer des bulletins d'information que je pourrai envoyer par courrier électronique à mes futurs clients, créer des coupons pour les fidéliser, m'inscrire sur le célèbre moteur de recherche Google, évaluer le comportement de mes clients avec e tracker, faire du commerce avec eBay...

En conclusion, créer sa propre boutique en ligne est d'une simplicité impressionnante, quelques clics de souris et le tour est joué ! Enfin le e-commerce est à la portée de tous, néophytes comme experts ! La solution est riche en fonctionnalités, intuitive, simple à prendre en main, et le tout en mode sécurisé.

Voilà, tout est en place pour satisfaire le client, il n'y a plus qu'à attendre les commandes...

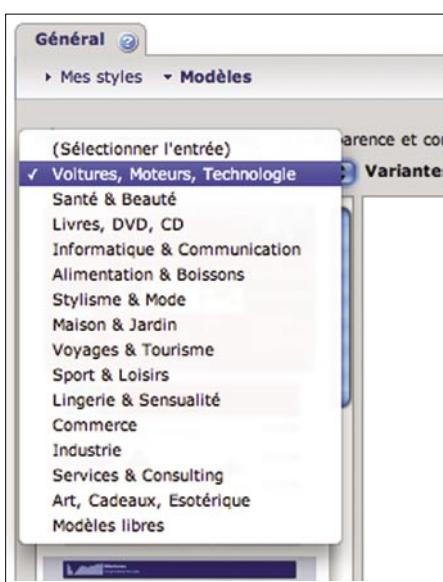


Figure 3. Classement de l'onglet Style

CakePHP

le framework pour développer rapidement vos applications

Démarré en 2005 lorsque Ruby On Rails devenait populaire, le projet CakePHP compte désormais de nombreux utilisateurs et une communauté toujours plus active. Le framework se veut avant tout simple et efficace, en se basant sur le paradigme des conventions plutôt que de la configuration.

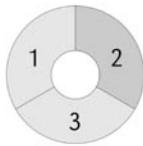
Cet article explique :

- Les principales conventions utilisées par CakePHP.
- La génération automatique des actions de base (CRUD) d'une application.
- Les opérations usuelles sur les modèles et leurs relations.

Ce qu'il faut savoir :

- Connaissance du langage PHP 4+.
- Notions de l'architecture MVC.

Niveau de difficulté



Voici une présentation des points clés de CakePHP permettant de se lancer dans son apprentissage.

CakePHP a été conçu afin de faciliter au maximum la vie du développeur qui l'utilise. Cependant il faut au préalable comprendre la philosophie du framework et la suivre au départ de manière presque aveugle. Ce n'est qu'une fois les principes clés assimilés et les applications de base réalisées avec succès que l'on pourra se rendre compte de la réelle valeur ajoutée de CakePHP et de tous ses composants. Voici donc les points qui vous permettront de débuter sans encombre.

Des conventions plutôt que de la configuration

Un des principaux atouts de CakePHP contribuant à la mise en place rapide et à la magie du framework est son fonctionnement par conventions. En effet, le framework ne nécessite que très peu de configuration comme nous le verrons plus tard et se base sur une découverte intelligente de l'application d'après certaines conventions.

Les conventions permettent également de sous-traiter certaines tâches classiques et rébar-

batives (intégrité des relations, récupération et associations des modèles, mise à jour de champs ...) à CakePHP qui est livré avec un bon nombre de fonctionnalités utiles.

Avantages des conventions

En tant que développeur, chacun a pu se créer ses propres règles de codage et façons de travailler au fur et à mesure de ses expériences. Il est ainsi vu comme difficile (ou d'un intérêt limité) de changer sa manière de coder afin de se plier à une technologie (en l'occurrence CakePHP), mais nous espérons que ce paragraphe et toutes les belles choses décrites ci-après vous feront tenter l'expérience : vous avez beaucoup à y gagner !

Les conventions sont tout d'abord le meilleur moyen de pérenniser un code. En effet, à l'heure actuelle le code d'une application est amené à être maintenu, réutilisé ou partagé avec d'autres développeurs. D'autre part, la lecture d'un code développé par une autre

personne peut s'avérer initialement difficile car la manière de développer (le style, l'organisation des fichiers ...) est rarement documentée.

C'est là tout l'intérêt des conventions ! Elles permettent d'avoir une base commune et d'éviter de sur-documenter une application afin de se focaliser sur la valeur ajoutée et les parties importantes du code. À l'échelle d'une entreprise, ces conventions de codage et de nommage sont bien souvent en place en interne ... avec CakePHP cependant, vos conventions seront désormais compréhensibles par tous !

Enfin, le respect des conventions permet une réduction non négligeable du code produit ce qui augmente la lisibilité de celui-ci. Finis les fichiers de configuration à rallonge, les bibliothèques de fonctions *pratiques* qui passent d'un projet à un autre, les bugs issus de méthodes de ces bibliothèques ... CakePHP est livré avec de nombreuses fonctionnalités pratiques, grandement testées et bien documentées !

Principales conventions

Nous détaillerons ci-après la plupart des conventions à suivre dans un projet CakePHP, dans un but de compréhension et d'apprentissage. Cette liste n'est cependant pas exhaustive, nous vous invitons à lire la documentation ou les articles sur le web afin de connaître tous les détails.

De base, l'anglais est préférable

CakePHP est livré avec des mécanismes de reconnaissance singulier / pluriel optimisés pour la langue anglaise (exemple : *people / person*). Ainsi, afin de ne pas avoir de mauvaise surprise il est recommandé d'utiliser des mots anglais. C'est également une bonne chose pour le partage de votre code !

Bien que la plupart des mots français soient gérés (exemple : *personnes / personne*), des cas propres à notre langue ne pourront être reconnus (exemple : *choux / chou ou bases_de_donnees / base_de_donnees*) dans la version actuelle du framework. Par un système de configuration d'inflexions il est néanmoins possible de tenir compte de ces cas particuliers et la communauté française travaille actuellement en ce sens afin de proposer une version francisée du framework.

Base de données et champs automagiques

La modélisation de la base de données est l'élément primordial d'un projet. Si les conventions sont respectées, il sera ensuite possible de générer la plus grande partie du code (CRUD) associé en quelques minutes ! Nous détaillerons ceci plus loin.

Concernant les conventions, elles sont les suivantes :

- le nom de la table doit être en minuscules et au pluriel (exemple : `my_messages`),
- un champ `id` devra être présent et pourra être de deux types : entier auto-incrémentable ou une chaîne de 36 caractères (`CHAR(36)`) auquel cas l'identifiant généré sera un UUID,
- une clé étrangère devra porter le nom du modèle référencé au singulier suivi de `_id` (exemple : `user_id`).

D'autre part on trouve des champs *automagiques*. On les nomme de cette manière car ces champs seront renseignés automatiquement par Cake lors de la sauvegarde d'informations (ils ne sont en aucun cas obligatoires !).

- le champ `created` de type `DATETIME` contiendra la date de création de l'élément,
- le champ `modified` ou `updated` contiendra la date de dernière modification de celui-ci,
- le champ `title` ou `name` sera utilisé comme intitulé de l'élément lors de son affichage sur le site. Dans le cas où aucun de ces champs n'existe, c'est l'identifiant qui sera utilisé,

Un dernier exemple un peu plus poussé : si un utilisateur est associé à plusieurs messages (c'est-à-dire que la table `messages` contient un champ `user_id`), il suffit d'ajouter un champ de type entier (`INT`) nommé `message_count` dans la table `users` afin que celui-ci soit automatiquement incrémenté et décrémenté ... et contienne à tout instant le nombre de messages associés à l'utilisateur.

Modèles et contrôleurs : nommage des classes

Afin d'utiliser votre base de données, vous aurez besoin d'une classe de modèle qui générera tous les aspects liés à votre modèle (validation, associations, actions métier ...) et une classe contrôleur chargée de gérer les événements et d'appeler les méthodes adéquates du modèle pour transmettre les informations utiles à la vue. Renseignez-vous sur le modèle MVC pour mieux comprendre ces concepts !

Pour CakePHP les classes de modèle doivent porter le nom de la table associée au modèle, au singulier et `CamelCased` (c'est-à-dire que l'on supprime les *underscores* du nom de fichier et on met la première lettre de chaque mot en

majuscule ... comme les bosses d'un chameau !). De plus, les modèles doivent hériter de la classe `AppModel`.

De même, les contrôleurs doivent se nommer comme les modèles mais au pluriel (également `CamelCased`), suivis du mot `Controller` et doivent hériter de la classe `AppController`.

Par exemple : la table `my_messages` aura pour modèle la classe `MyMessage` qui étende `AppModel` et le contrôleur sera `MyMessagesController` qui étende `AppController`.

Organisation des fichiers

Étant dans un framework MVC, les fichiers sont répartis dans 3 principaux dossiers à la racine de l'application : `models`, `views` et `controllers` ! Un dossier `webroot` contiendra toutes les ressources utilisées (javascript, css ...) et tiendra lieu de racine de documents. Les noms de fichiers doivent être en minuscule et les mots séparés par des *underscores* (`_`). Le nom du fichier correspond en général à la classe qu'il contient (modèle ou contrôleur), ou à la vue à laquelle il est associé.

Listing 1. Création de la base de données

```
CREATE TABLE 'users' (
  'id' INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  'name' VARCHAR( 100 ) NOT NULL ,
  'email' VARCHAR( 100 ) NOT NULL ) ;
CREATE TABLE 'messages' (
  'id' INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  'title' VARCHAR( 150 ) NOT NULL ,
  'content' TEXT NOT NULL ,
  'user_id' INT UNSIGNED NOT NULL ,
  'created' DATETIME NOT NULL ,
  'modified' DATETIME NOT NULL ) ;
```

Listing 2. Création du modèle user dans le fichier /models/user.php

```
<?php
class User extends AppModel {
  var $name = 'User';
}
?>
```

Listing 3. Création du modèle message dans le fichier /models/message.php

```
<?php
class Message extends AppModel {
  var $name = 'Message';
  var $belongsTo = 'User';
}
?>
```

Listing 4. Création du contrôleur user dans le fichier /controllers/users_controller.php

```
<?php
class UsersController extends AppController {
  var $scaffold; // Cette ligne permet de passer en mode prototypage
  // rapide pour ce contrôleur
}
?>
```

Listing 5. Création du contrôleur message dans le fichier /controllers/messages_controller.php

```
<?php
class MessagesController extends AppController {
  var $scaffold;
}
?>
```

Chargement automatique de fichiers

Ainsi, rien qu'en suivant les conventions précédentes et sans ligne de code supplémentaire votre application est prête à être exécutée ! (Note : en réalité il faudra simplement relier votre application au cœur de CakePHP et configurer la connexion à la base de données dans un fichier *database.php*). Voyons ce qui se passera à l'appel d'une page :

- appel de l'URL http://www.helloworld.com/my_messages/hello,
- exécution de la méthode `hello()` de la classe `MyMessagesController` située dans le fichier `/controllers/my_messages_controller.php`,
- le contrôleur fera sans doute appel à une méthode du modèle `MyMessage` située dans le fichier `/models/my_message.php` qu'il connaîtra automatiquement,
- le modèle interagira avec la table `my_messages` de la base de données et renverra des données au contrôleur,
- enfin le contrôleur transmettra ces informations à la vue pour un affichage de la page, ce qui affichera le contenu du fichier `/views/my_messages/hello.ctp`.

Nous voyons par ce simple exemple qu'il est ainsi possible de séparer clairement chaque partie de l'application. Ceci rend plus simple le travail collaboratif, la rédaction de tests unitaires et la recherche d'une ligne de code : suivant ce que l'on cherche on sait exactement dans quel fichier chercher. Note avancée : à chaque étape de ce processus il est possible d'intercaler des actions grâce à l'existence de méthodes de *callback*, pour une meilleure souplesse.

Si vous ne voulez pas des conventions

Il faut cependant noter que le suivi des conventions n'est pas obligatoire (de la même manière qu'une personne peut construire sa maison tout seul !). Les plus rebelles ou ceux qui n'ont pas la main sur les conventions de nommage utilisées, dans le cas de la reprise d'une application existante par exemple, trouveront donc une solution à leur situation.

Nous ne pouvons tout détailler ici, mais sachez qu'il est possible de modifier le comportement du framework et de se servir de sa puissance sur un code ne respectant pas les conventions ... il suffira simplement de rajouter un peu de configuration à votre application !

Le scaffolding ou prototypage rapide

CakePHP est avant tout une application facilitant la mise en place rapide d'une application (framework de RAD). Comme nous l'avons vu précédemment, une bonne modélisation de la base de données respectant certaines conventions peut vous faire gagner un temps précieux. Il est désormais temps de voir tout ce que CakePHP vous propose !

ID	Title	Content	User	Created	Modified	Actions
1	Hello world	Hello !	Pierre	2009-02-05 07:52:50	2009-02-05 07:52:50	View Edit Delete

Figure 2. Récapitulatif du message créé

Le prototypage rapide est une fonctionnalité permettant d'analyser votre base de données et de vous fournir des formulaires permettant toutes les actions CRUD (ajout, visualisation, mise à jour et suppression) d'un objet et de ceux qui y sont associés. Prenons l'exemple d'une base de données gérant des utilisateurs

et leurs messages de manière simpliste. Nous devrons donc créer 2 tables telles que décrit dans le Listing 1.

Le plus difficile est effectué, analysons désormais les modèles associés dans lesquels nous indiquons qu'un message appartient à un utilisateur. Cette étape est présentée dans les Listing

Listing 6. Méthode classique de création d'un nouvel objet en base de données

```
function add() {
    if (!empty($this->data)) {
        $this->Recette->create();
        if ($this->Recette->save($this->data)) {
            $this->Session->setFlash(__('The Recette has been saved', true));
            $this->redirect(array('action'=>'index'));
        } else {
            $this->Session->setFlash(__('The Recette could not be saved. Please, try again.', true));
        }
    }
}
```

Listing 7. Exemple de récupération de tous les utilisateurs existants

```
class RecettesController extends AppController {
    function add() {
        $utilisateurs = $this->Recette->Utilisateur->find('all');
        ...
    }
}
```

Listing 8. Récupération d'une entrée de la base de données et des informations associées

```
class RecettesController extends AppController {
    function view($id) {
        $this->Recette->id = $id;
        $this->Recette->recursive = 1;
        $recette = $this->Recette->read();
        ...
    }
}
```

Listing 9. Association du comportement Translate à un modèle et paramétrage

```
class Recette extends AppModel {
    var $actsAs = array(
        'Translate' => array(
            'titre' => 'Titres',
            'description' => 'Descriptions' ) );
    ...
}
```

Tableau 1. Relations possibles entre différents modèles

Relation	Nom de la relation	Exemple
1 à 1	hasOne	Un utilisateur a un profil
1 à n	hasMany	Un utilisateur peut être l'auteur de plusieurs recettes
n à 1	belongsTo	Une recette appartient à un utilisateur
n à n	hasAndBelongsToMany	Une recette est associée à plusieurs tags, et un tag est associé à plusieurs recettes

2 et 3, notez au passage la simplicité de définition des associations (Listing 3).

Enfin demandons à Cake de faire le travail de prototypage rapide, ceci depuis les contrôleurs associés aux *Utilisateurs* (Listing 4) et aux *Messages* (Listing 5).

C'est tout ! Désormais, en vous rendant sur les pages <http://www.votresite.com/users> et <http://www.votresite.com/messages> vous pourrez tout gérer. Les formulaires permettant les actions CRUD sont disponibles.

La Figure 1 est une capture d'écran du formulaire d'ajout d'un nouveau message. Nous voyons que la liste des utilisateurs existants (un utilisateur a été créé auparavant) s'affiche, afin de relier le message à son auteur. La Figure 2 quant à elle montre le résultat de la création : le lien avec l'utilisateur est bien sauvegardé et les champs automatiques ont été renseignés. Magique n'est-ce pas ?

CakePHP en ligne de commande

CakePHP offre une interface en ligne de commande permettant de réaliser certaines tâches très rapidement, comme la création du squelette d'application, la génération interactive des fichiers pour chaque entité (modèle, contrôleur et vues) ou encore l'extraction des chaînes *gettext* présentes dans tous les fichiers de l'application dans le cadre d'une application multilingue. Il est également possible de créer ses propres scripts exécutables via cette console, les *shells* et les *tasks*.

Squelette d'application

Après avoir ajouté l'exécutable php dans les variables d'environnement (sous Windows) et créé une nouvelle base de données, nous pouvons ouvrir un terminal (*Démarrer > Exécuter > cmd.exe*) et appeler la console :

```
cake bake
```

Le script nous propose de créer une nouvelle application : il suffit d'indiquer le répertoire de destination et la console crée toute l'arborescence de base, et génère une nouvelle clé de sécurité dans le fichier *config/core.php* qui servira lors de la mise en place d'un système d'authentification.

Le script nous demande ensuite les codes d'accès à la base de données, puis se termine. Nous pouvons nous rendre à la racine de l'application dans un navigateur, et nous voyons apparaître la page d'accueil de notre nouvelle application.

La console propose ensuite de définir les associations existant entre cette table et les autres tables de la base. Quatre types d'associations sont gérés par CakePHP : *hasOne* (possède un seul), *hasMany* (possède plusieurs), *belongsTo* (appartient à) et *hasAndBelongsToMany* (possède et appartient à plusieurs).

Si nous respectons les conventions de nommage des tables et des champs (nom des tables au pluriel, 'id' pour les clés primaires, nom de table au singulier suivi de '_id' pour les clés étrangères), CakePHP détecte seul les paramètres de chaque association, mais il reste tout à fait possible de créer les modèles à partir de tables venant d'une ancienne application qui ne respecte pas ces conventions. Notre premier modèle est créé !

Création d'un modèle

Création d'un modèle

Contrairement au *scaffolding*, qui ne crée pas les fichiers nécessaires dans l'application, la console permet de générer effectivement ces fichiers de manière interactive. Appelons à nouveau la console, de la même manière que plus haut :

```
cake bake
```

Cette fois la console a détecté que notre application existe, et nous propose une liste d'actions possibles : ajouter une configuration de base de données, créer un modèle, une vue, un contrôleur ou un nouveau projet. Nous choisissons de créer un modèle : la console propose la liste des tables de la base de données. Nous choisissons l'une des tables : la console demande si nous voulons définir des règles de validation pour les champs de cette table. Nous répondons oui : pour chaque champ, nous pouvons choisir parmi 28 règles prédéfinies (*notEmpty*, *alphaNumeric*, *email*, etc.), écrire notre propre expression régulière, ou bien ne pas définir de règle de validation.

Création d'un contrôleur

Passons maintenant à la génération d'un contrôleur : la console nous donne la liste des contrôleur possibles en fonction des tables de la base, puis propose de créer dans ce nouveau contrôleur les actions de base (*index*, *add*, *edit* et *delete*), et éventuellement les actions de base d'administration (*admin_index*, *admin_add*, *admin_edit* et *admin_delete*).

Création d'une vue

Terminons par la création des vues : CakePHP se charge de créer une vue pour chaque action du contrôleur. Nous pouvons à présent parcourir

The screenshot shows a web form titled "New Message". It has two input fields: "Title" and "Content". Below the "Content" field is a dropdown menu labeled "User" with "Pierre" selected. At the bottom of the form are several buttons: "Submit", "List Messages", "List Users", and "New User".

Figure 1. Formulaire de création de message généré par CakePHP

l'application dans un navigateur : tout est déjà prêt, il ne reste qu'à affiner la présentation des données ou certains traitements, mais l'application en l'état est totalement fonctionnelle, après avoir tapé une ligne de commande et répondu à quelques questions !

Les modèles

Les modèles, l'une des trois couches du modèle MVC, constituent le cœur de l'application. Ils encapsulent l'accès aux tables de la base de données et sont responsables de la plupart des manipulations de ces données : lecture, écriture, suppression et validation du format des données avant sauvegarde. Le framework est prévu pour pouvoir travailler avec un grand nombre de SGBD différents de façon transparente (MySQL 4 et 5, Postgres, Firebird, SQL Server, Oracle, SQLite, ODBC et ADODB).

Les modèles sont conçus dans l'optique d'une grande souplesse puisqu'il est possible d'utiliser un modèle sans table (gestion d'un formulaire de contact par exemple), que l'on peut spécifier des bases de données différentes d'un modèle à l'autre, et qu'un modèle peut manipuler des données issues d'une autre source qu'une table de base de données, par exemple une API externe. Il suffit pour cela de créer sa propre *datasource* et de brancher le modèle dessus.

Note : toutes les classes de modèles de l'application étendent la classe `AppModel`, il est donc facile de factoriser certaines actions redondantes sur plusieurs modèles à l'intérieur de cette classe mère.

Opérations usuelles sur les données

Examinons un contrôleur créé plus tôt via la console : nous avions demandé à CakePHP de créer pour nous les actions de base de manipulation de données ; penchons-nous dans le Listing 6 sur l'action `add()` permettant d'ajouter une entrée à la base de données.

À l'intérieur d'une action d'un contrôleur, les données issues d'un formulaire présent dans la vue de cette action sont toutes regroupées dans `$this->data`. La première étape vérifie donc si cette variable existe (le formulaire a été soumis) et dans ce cas, la méthode `create()` du modèle `Recette` est appelée : CakePHP prépare une enveloppe vierge qui va accueillir les données à sauvegarder. Il essaie ensuite de sauvegarder effectivement les données soumises, et redirige sur l'index du contrôleur en cas de succès (c'est-à-dire d'une part que les données ont passé tous les tests de validation, et d'autre part que la ligne a bien été ajoutée dans la base).

Relations entre modèles

Une des grandes forces de CakePHP réside dans la gestion automatique des relations entre les modèles. Les quatre relations possibles sont décrites dans le Tableau 1.

Une fois les relations définies dans le modèle, on peut accéder directement à l'un des modèles associé : si par exemple dans l'action `add()` du contrôleur des recettes, nous souhaitons obtenir la liste des utilisateurs nous utiliserons l'instruction présentée dans le Listing 7.

La gestion de ces relations permet aussi de rapprocher en un seul appel les données d'un ou plusieurs enregistrements du modèle ainsi que des modèles associés. La *profondeur* jusqu'à laquelle CakePHP doit parcourir les associations peut être réglée grâce à la variable de modèle `$recursive`.

Analysons l'action `view()` de notre contrôleur des recettes dans le Listing 8.

La variable `$recette` contiendra non seulement les données de la recette d'identifiant `$id`, mais aussi les informations sur l'utilisateur qui l'a écrite et les tags associés à la recette !

Ce formidable automatisme se retrouve également dans la sauvegarde des enregistrements, puisque l'on peut sauvegarder l'enregistrement d'un modèle et de ses modèles associés en un seul appel là aussi (méthodes `save()` ou `saveAll()`).

Les callbacks

Les *callbacks* sont des méthodes spéciales qui sont déclenchées automatiquement par le

Listing 10. Formulaire d'ajout d'un «message» dans le fichier «/views/messages/add.ctp»

```
<?php
    echo $form->create('Message');
?>
<fieldset>
    <legend><?php __('Add Message'); ?></legend>
    <?php
        echo $form->input('title'); // Affichera un "input text"
        echo $form->input('content'); // Affichera un "textarea"
        echo $form->input('user_id'); // Affichera un "select" contenant
                                        // les couples id/name des utilisateurs existants
    ?>
</fieldset>
<?php
    echo $form->end('Submit');
?>
```

Listing 11. Configuration du composant d'Authentification et restriction des pages d'administration

```
class AppController extends Controller {
    var $components = array('Auth');
    function beforeFilter() {
        $this->Auth->loginAction = '/users/login';
        $this->Auth->loginRedirect = '/admin/recettes/index';
        $this->Auth->logoutRedirect = '/';
        $this->Auth->loginError = "Identifiant ou mot de passe incorrects.";
        $this->Auth->authError = "Vous n'avez pas accès à cette page.";

        if(empty($this->params['prefix']) or $this->params['prefix'] != Configure::read('Routing.admin')) {
            $this->Auth->allow();
        }
    }
}
```

Listing 12 : Contrôleur basique de gestion des utilisateurs, tout est fait par Cake !

```
class UsersController extends AppController {
    function login() { }
    function logout() {
        $this->Session->setFlash("Vous êtes maintenant déconnecté.");
        $this->redirect($this->Auth->logout());
    }
}
```

framework lors de chaque appel à une page. On trouve par exemple les méthodes de modèle `beforeSave()` ou `afterFind()`, qui peuvent accueillir des traitements supplémentaires en plus des méthodes de base. D'autres callbacks sont disponibles dans les contrôleurs (`beforeFilter`, `beforeRender`...).

Extensions de modèle

Dans l'optique de produire du code le plus réutilisable possible, CakePHP permet d'étendre les fonctionnalités de base d'un modèle avec des *Behaviors* (*Comportements*), qui ajoutent des méthodes à un modèle existant ou utilisent les *callbacks* pour effectuer des traitements supplémentaires. CakePHP en fournit quelques uns, comme par exemple le comportement *Tree* qui permet de gérer une table comme une arborescence. La gestion des nœuds se fait en toute simplicité : la logique se trouve dans des *callbacks* du *Comportement*, appelés avant et après la sauvegarde classique d'un enregistrement.

Exemple de Behavior : Translate

Ce comportement, inclus dans le framework, permet de gérer une application multilingue. Il faut avant tout préparer la base de données à accueillir plusieurs langues : pour ce faire nous devons créer une table `i18n` (dont le schéma est fourni dans le répertoire `config/sql`) qui va contenir tous les champs textuels des autres tables, et ces autres tables ne doivent pas contenir de champs susceptibles d'être traduits.

Nous associons ensuite le comportement *Translate* au modèle en précisant quels champs textuels devront être traduits grâce à l'attribut `$actAs` du modèle, tel que présenté dans le Listing 9.

La gestion des traductions se fait ensuite de manière transparente : avant toute action sur un enregistrement, il suffit de préciser sur quelle(s) langue(s) on travaille, et *Translate* s'occupe du reste. Ainsi pour afficher une recette en anglais :

```
function view($id) {
    $this->Recette->locale = 'eng';
    $this->Recette->id = $id;
    $recette = $this->Recette->read();
    ...
}
```

Et CakePHP renvoie un enregistrement avec les champs `'titre'` et `'description'` traduits, comme si nous avions affaire à la table `recettes` avec les champs traduits en anglais, alors qu'en réalité ces deux champs n'existent pas dans la table `recettes` mais dans la table `i18n`.

Les vues

Les vues de CakePHP n'utilisent pas de langage spécifique, et sont donc du pur (x)HTML au milieu duquel on utilise le PHP classique pour afficher les données dynamiques. Il est cependant possible d'utiliser des moteurs de *templates* comme Smarty, mais c'est en pratique très peu utilisé à cause des nombreux *helpers* disponibles qui permettent de garder un code

propre, minimal, réutilisable et facile à prendre en main !

Différents éléments composant une vue

Dans un but de réutilisabilité du code et pour diminuer la redondance, les vues sont composées de plusieurs fichiers.

- *Layouts* : un fichier gabarit qui contient toute la structure commune aux pages du site (menus, en-tête et pied de page ...), il est donc très simple de changer radicalement de structure de page entre sections du site ou versions.
- *Elements* : certains éléments communs à plusieurs pages (exemple : formulaires de connexion) peuvent être mis sous forme d'un fichier séparé qui sera inclus dans chacune des pages, avec possibilité de passage de paramètres,
- *Corps de la page* : associés à chacune des actions, les fichiers de vue contiennent le code de la page strictement nécessaire à l'action demandée (formulaire, affichage des informations) ce qui permet de les réutiliser simplement d'un projet sur l'autre.

Des helpers pour se simplifier la vie

Comme vous commencez à le voir, le framework contient énormément de fonctionnalités n'ayant qu'un seul but : faciliter le développement en s'abstrayant des tâches rébarbatives !

Ainsi, CakePHP est livré avec des classes contenant des méthodes vraiment pratiques : les *helpers*.

P U B L I C I T É



Cognix Systems, SSII et Web Agency depuis 2002

Expertise autour des technologies Web sur PHP

Développeurs PHP, venez nous rejoindre !

Intégrer notre équipe **jeune et dynamique** présente sur Rennes pour le développement d'applications Internet/intranet/extranet.

recrutement@cognix-systems.com

02 99 27 75 92

Plus d'informations sur
www.cognix-systems.com



Le FormHelper

Afin de gérer simplement les formulaires affichés sur votre site et leur lien avec les données issues ou destinées à la base de données, une classe vous fournit un tas de méthodes *automatiques*. Voici dans le Listing 10, à titre d'exemple, le code permettant de générer le formulaire d'ajout de la Figure 1 (code généré automatiquement depuis la ligne de commande).

Nous pouvons voir dans cette portion de code que la création d'un formulaire se fait en définissant le modèle associé, et que Cake se chargera ensuite de déterminer le type d'élément de formulaire à afficher en fonction du type de données de la base de données : ici le champ `title` est un `VARCHAR`, content un `TEXT` et `user_id` une clé étrangère vers une entrée de la table `users`. Sachez également que vous pourrez insérer directement un type d'élément précis (`select`, `radio`, `checkbox` ...) par l'appel de la méthode correspondante du `helper` ... et qu'il existe beaucoup d'autres méthodes et options pour faire ce que vous voudrez !

Le HTMLHelper

De la même manière, une autre classe contient de nombreuses méthodes utiles pour vous faciliter l'affichage de code HTML. Quelques exemples : insérer des balises `doctype` conformes, un `charset` uniforme au site, des liens (vers des pages, images, fichiers) qui ne seront pas cassés par un changement de schéma d'URL *rewriting*, créer simplement des tableaux de données et bien d'autres choses.

CakePHP contient un système de gestion d'URL très fin (les routes) qui vous permettra de personnaliser entièrement vos urls en modifiant quelques lignes dans le fichier `/config/routes.php`. L'intérêt du *HTMLHelper* est qu'il se chargera de générer automatiquement un lien correspondant à ces paramètres afin de

pouvoir modifier très simplement votre système d'URL.

Le AjaxHelper

Basé sur la librairie `prototype`, le *AJAX Helper* vous permet de mettre en place des formulaires, liens, auto-complétion, glisser-déposer et autres effets en un seul appel de méthode. Il existe également un équivalent *jQuery*.

... et bien d'autres : RSS, JavaScript, RSS, *Paginator* (pour paginer simplement vos résultats), *Number*, *Session*, XML ... vous permettront de mettre en place en toute simplicité des fonctionnalités pratiques et répandues. Ceci est une liste non exhaustive se basant sur les *helpers* du cœur du framework : la communauté en propose beaucoup d'autres, et vous pouvez en toute simplicité créer les vôtres !

Des composants livrés avec le framework

Au même titre que les *Behaviors* sont des extensions de modèle, les *Components* (composants) sont des extensions de contrôleur. Parmi ceux livrés avec CakePHP (*Cookies*, *Sessions*, *Email*, *Security*, etc.), nous en aborderons deux : *Auth* et *ACL*.

Auth

Le composant *Auth* permet de mettre en place une authentification par `login` / mot de passe sur certaines parties du site. L'utilisation typique de cette fonctionnalité est la protection d'un espace d'administration du site, c'est-à-dire l'authentification requise pour les actions prefixées par `'admin_'`. Commençons par activer le routage automatique des actions réservées à l'administration en décommentant une ligne dans `config/core.php` :

```
Configure::write('Routing.admin', 'admin');

/admin/recettes/edit/12' renverra sur l'action
admin_edit($id = null) du contrôleur
```

`RecettesController`, avec comme paramètre `12`, l'identifiant de la recette à éditer.

Toujours dans l'esprit *convention au lieu de configuration*, CakePHP attend que les utilisateurs autorisés soient stockés dans une table `'users'` avec un champ `'user_name'` (le `login`) et un champ `'password'`. Ce dernier devra être de type `varchar (40)` pour stocker la clé de `hashage` du mot de passe, composée de la clé de sécurité générée lors de la création de l'application avec la console, suivie du mot de passe, le tout crypté en `SHA1` ce qui nous donne une chaîne de 40 caractères.

Afin de restreindre l'accès à toutes les actions de nos contrôleurs commençant par `admin_` aux seuls utilisateurs logués, nous allons placer la logique d'authentification dans la classe `AppController` (Listing 11) dont héritent tous les autres contrôleurs. Comme cette logique doit intervenir avant toute autre opération lors de l'appel à une action, nous utiliserons le `callback beforeFilter`, qui comme son nom l'indique est appelé avant l'action elle-même.

Il suffit de définir quelques propriétés du composant *Auth* (les URL de connexion, de déconnexion et de redirection en cas de succès, les messages d'erreur, la condition d'autorisation ou de refus) et l'accès à une action d'administration demande maintenant de s'identifier. Il ne nous reste qu'à créer les actions de `login` et `logout` dans le contrôleur des utilisateurs, `UsersController` présenté dans le Listing 12.

L'action `login` reste vide, CakePHP se charge lui-même de vérifier si le couple `login` / mot de passe, qui sera fourni par le biais du formulaire à créer dans la vue de cette action, correspond bien à un utilisateur de la table `users`. L'action `logout` ne fait que rediriger sur l'URL définie comme URL de déconnexion dans le `beforeFilter`, avec un message.

ACL

Le composant *ACL* (*Access Control List*) permet un contrôle poussé des autorisations d'accès en considérant d'une part des rôles (*aros*) et d'autre part des ressources (*acos*), avec des relations d'autorisation ou d'interdiction entre un rôle et une ressource donnés. Notons bien qu'il ne s'agit pas de la même chose que l'authentification vue plus haut, mais d'un processus de contrôle d'accès aux ressources une fois qu'un utilisateur a été reconnu.

La solution proposée par CakePHP offre une granularité importante, puisqu'il est possible qu'un rôle soit un utilisateur ou un groupe d'utilisateurs, et qu'une ressource soit un enregistrement précis d'une table, une action, un contrôleur complet ou même l'application dans son ensemble. Le framework gère ces données sous forme de deux arborescences, une pour les rôles et une pour les ressources, il est donc possible (et préférable !)

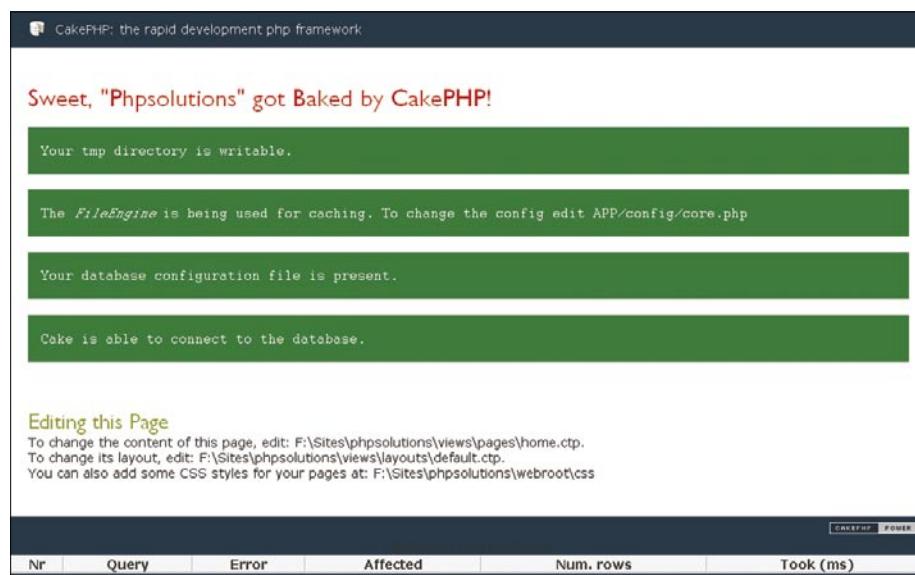
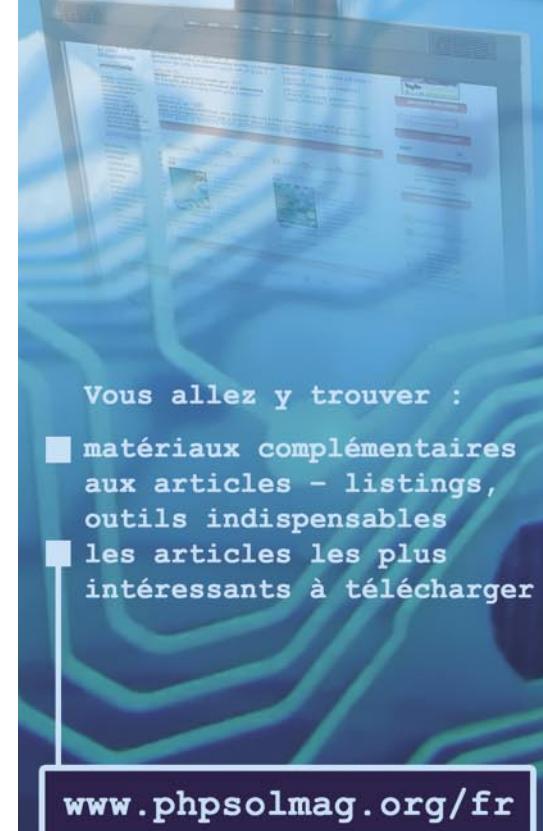


Figure 3. Aperçu de l'application créée en ligne de commande

Visitez notre site Internet



Sur Internet

- <http://www.cakephp.org/> – Site officiel du framework contenant entre autres la documentation et l'API,
- <http://www.cakephp-fr.org/> – Site de la communauté francophone principalement constitué d'un forum d'aide et de conseils,
- <http://www.formation-cakephp.fr/> – Blog francophone présentant clairement diverses fonctionnalités de CakePHP et leur mise en application,
- <http://www.planetcakephp.org/> – Site recensant les flux des différents blogs CakePHP.

de ne pas définir toutes les autorisations d'accès pour tous les rôles vers toutes les ressources, mais de définir une autorisation pour un groupe d'utilisateurs entier vers une ressource, et chaque membre du groupe héritera des mêmes droits. Il reste ensuite possible de créer une exception sur l'un des membres du groupe sans influer sur les autres membres.

Le composant fournit des méthodes pour d'une part autoriser ou interdire l'accès d'un rôle à une ressource, et d'autre part pour contrôler si une autorisation d'accès existe entre un rôle et une ressource donnés. De plus, il existe un comportement du même nom (ACL) qui permet de gérer automatiquement les rôles, par exemple si nous voulons créer un rôle à chaque création d'un utilisateur.

La communauté de CakePHP

CakePHP est un projet Open Source avec une communauté développée et très active. De nombreux blogs abordent diverses astuces ou donnent quelques conseils, et les ressources officielles sont de très bonne qualité. Voici donc les principales sources d'informations proposées par la communauté.

La documentation

La documentation (également appelée le *CookBook*) couvre la quasi-intégralité des sujets et est mise à jour / améliorée régulièrement. Chacun peut y contribuer en soumettant des modifications qui seront ensuite validées par les responsables. La traduction française est actuellement en cours (plus de 40% de traduction effectuée), c'est d'ailleurs l'action prioritaire de la communauté francophone <http://book.cakephp.org/fr>.

L'API

Basée sur la version actuelle du framework, l'API est automatiquement générée depuis le code de l'application et ses commentaires. Un moteur de recherche vous permet de trouver rapidement une méthode afin de voir précisément quelle est sa spécification, et même d'en lire le contenu. La navigation dans le code du cœur est ainsi très simple !

Forums d'aide

La communauté anglophone utilise un groupe Google afin de fournir de l'assistance aux développeurs, ce qui était jusqu'à récemment également le cas de la communauté francophone.

Depuis peu un forum a vu le jour, permettant de dialoguer, demander de l'aide sur l'utilisation du framework, de résoudre un problème insolvable ou bien pour la conception de votre application. Une équipe jeune et dynamique (<http://forum.cakephp-fr.org/>) vous y attend et vous aidera à faire vos premiers pas !

Blogs et autres ressources

De nombreux blogs anglophones et francophones traitent du développement CakePHP. En France on trouve notamment le blog <http://www.formation-cakephp.com> qui est une mine d'or de tutoriaux de qualité. Pour les blogs non francophones, le site <http://planetcakephp.org/> recense les flux des meilleurs blogs dédiés à CakePHP.

D'autre part, un channel IRC très actif (anglophone) vous permettra de dialoguer avec les développeurs du cœur du framework et d'autres utilisateurs pour avancer dans le développement de vos applications : #cakephp sur [irc.freenode.net](irc://irc.freenode.net).

Conclusion

Comme nous l'avons vu tout au long de cet article, CakePHP a été conçu de manière à permettre un développement rapide d'applications web fonctionnelles sous tous les aspects. Historiquement anglophone mais depuis quelques mois également francophone, la communauté ne cesse de s'agrandir. Vous pourrez ainsi débouter en vous basant sur de nombreux tutoriaux, avec de la documentation et de l'assistance à votre disposition.

Nous vous invitons à suivre les quelques exemples présentés ici, puis à découvrir les nombreuses autres fonctionnalités au fur et à mesure de vos besoins.

PIERRE-EMMANUEL FRINGANT, PIERRE MARTIN

Pierre-Emmanuel Fringant est webmaster indépendant à Caen dans le Calvados (Normandie). Après un DUT informatique obtenu à l'IUT de Ifs, il acquiert 6 ans d'expérience dans plusieurs entreprises (agence de communication, webagencies), puis se lance dans l'aventure indépendante début 2007.

Pierre Martin est élève ingénieur double compétence informatique/management en dernière année à l'INSA et ESC de Toulouse. Passionné de programmation, il s'est tourné vers le développement web lors de stages et par son activité de développeur web indépendant ces 2 dernières années.

Tous deux font partie de la communauté francophone du framework CakePHP.

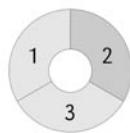
Introduction au partitionnement avec MySQL 5.1

Le 27 novembre 2008 après de longs mois d'attente, MySQL 5.1 est (enfin) sortie en Generally Available (GA). L'une des fonctionnalités phare de cette nouvelle mouture est la possibilité de partitionner ses tables.

Cet article explique :

- Comment manipuler le partitionnement avec MySQL 5.1

Niveau de difficulté



Ce qu'il faut savoir :

- Il vous sera utile d'avoir des connaissances techniques sur MySQL.

L'autre type de partitionnement est le partitionnement horizontal. Là, la division se fait au niveau des enregistrements, la table est moins haute, le but étant de chercher l'information dans un sous ensemble plus petit. En d'autres termes, il est plus facile de rechercher une aiguille dans une petite botte de foin que dans une grande. MySQL 5.1 implémente le partitionnement horizontal, c'est donc de ce type de partitionnement que traite cet article.

Certes, il était déjà possible, depuis la 3.23, de diviser ses tables manuellement avec le moteur de stockage *MRG_MYISAM*, certes la technologie partitionnement (algorithme *KEY*) existe depuis la 4.1 avec le moteur *NDB_Cluster*, mais découper ses tables simplement, par *range*, *list*, *hash* ou *key*, fusionner les partitions, les séparer, les sous-partitionner... tout ceci n'est bien possible que depuis la 5.1. La version de MySQL utilisée pour cette article est la 5.1.30 *Community Server*, pour *M.S. Windows*.

Principes

Le partitionnement est un type d'architecture qui consiste à diviser une table en plusieurs parties. Cette technique de conception a deux principaux objectifs : augmenter les performances et simplifier la gestion des données.

Augmenter les performances

Partitionner est d'autant plus utile que le volume de données est important. On peut voir les partitions comme des sous ensembles d'une même table, l'optimiseur de MySQL sait sur quelles partitions chaque enregistrement se trouve. Il se peut alors, si les données recherchées ne concernent pas toutes les partitions, que le temps de réponse d'une re-

Simplifier certaines tâches de maintenance des données

Si des données obsolètes se trouvent dans une même partition, effacer entièrement cette partition (*DROP PARTITION*) est plus rapide que d'effacer tous les enregistrements les uns après les autres (*DELETE FROM*). De plus, contrairement au moteur de stockage *MERGE* qui permet de faire du partitionnement manuel depuis la version 3.23.25, MySQL 5.1 permet de partitionner et de maintenir beaucoup plus simplement ses partitions.

Quand on parle de partitionnement il faut avoir à l'esprit qu'il y a deux façons de partitionner : horizontalement et verticalement. Le partitionnement vertical consiste à rendre une table moins large, en la divisant au niveau de ses colonnes. Cela peut être utile pour isoler des champs de type *BLOB* ou *TEXT*. Par exemple la table *client_infos* divisée en deux tables *client* et *client_extra* (Listing 1).

Partitionner une table

La partitionnement se définit au niveau de la structure de la table, lors de la création de cette dernière (*CREATE TABLE*) ou alors après, en modifiant la structure de la table (*ALTER TABLE*). MySQL propose quatre types de partitionnement :

- **RANGE** : permet de spécifier des intervalles de valeurs
- **LIST** : division des données sous forme de listes de valeurs
- **HASH** : utilisation d'une clé de *hashage* pour répartir les données de façon homogène
- **KEY** : utilisation également d'une clé de *hashage* pour répartir les données de façon homogène.

La syntaxe SQL de MySQL 5.1 a été enrichie pour l'implémentation du partitionnement avec l'apparition du mot clé (*partition*). Les clauses permettant de partitionner une table doivent être placées à la fin

Tableau 1. Fonctions date retournant un nombre entier

DAY()	LAST_DAY()	SECOND()	WEEK()
DAYOFWEEK()	MICROSECOND()	TIME_TO_SEC()	WEEKDAY()
DAYOFYEAR()	MINUTE()	TIMESTAMPDIFF()	WEEKOFYEAR()
EXTRACT()	MONTH()	TO_DAYS()	YEAR()
HOUR()	QUARTER()	UNIX_TIMESTAMP()	YEARWEEK()

du code SQL de création de la table, de plus chaque type de partitions à sa syntaxe propre (Listing 2).

Les contraintes

La clé de partition doit être de type entier ou composée d'une fonction qui retourne un entier ou une valeur `NULL`, cette limitation ne s'applique pas pour le partitionnement par `KEY` (Listing 3).

Si la table a un index unique ou une clé primaire, la clé de partition doit être au moins une partie de cet index unique (ou clé primaire) (Listing 4). Les tables de type merge, federated, cvs ne peuvent être partitionnées (Listing 5).

Le partitionnement par RANGE

La clause de partitionnement est `PARTITION BY RANGE` (colonne). Il faut ensuite donner un nom à la partition : `PARTITION nom_partition` et définir la borne maximale (non incluse) de l'intervalle : `VALUES LESS THAN (borne_max_partition)`. La borne maximale de toutes les partitions peut être définie par le mot clé : `MAXVALUE` (Listing 6).

Tout ID strictement inférieur à 64 est stocké dans la partition `p_id_moins_de_64`, Tout ID strictement inférieur à 128 est stocké dans la partition `p_id_moins_de_128`, etc...

Il faut également garder à l'esprit que l'ordre des partitions est important. En effet, la clause `MAXVALUE` ne peut être que la dernière de la liste et les bornes de partitions ne peuvent être définies que de la plus petite à la plus grande comme dans le Listing 7.

Sur le disque dur, il est possible de voir le résultat du partitionnement de la table employée. MySQL a physiquement divisé les fichiers `.MYD` (données) et `.MYI` (index) de cette table MyISAM en autant de partitions. On a donc 8 fichiers (4 partitions fois 2 fichiers). Le `.frm` (structure) lui est bien là et un fichier `.part` est également créé. Il contient les informations sur les partitions (Listing 8).

Le plus important dans le partitionnement est le choix du critère de partitionnement, c'est à dire la colonne choisie pour partitionner la table. Ce choix se fait principalement en fonction du type de requêtes exécutées sur la table. L'algorithme hash est très utile pour partitionner des dates. Mais les types `datetime`, `timestamp` de MySQL ne sont pas des entiers. Il faut donc utiliser une fonction sur cette colonne qui renvoie un entier ou la valeur `NULL`. Le tableau 1 liste quelques unes des fonctions date qui retournent un entier.

La fonction `YEAR()` est toute indiquée, elle extrait l'année d'une date et la retourne sous la forme d'un entier (Listing 9). Une autre fonction recommandée est `TO_DAYS()`. Elle

Listing 1. Partitionnement vertical

```
mysql> DESC client_infos;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int(11) | YES  | PRI | NULL    |       |
| nom   | char(45) | YES  |      | NULL    |       |
| photo | blob   | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.03 sec)

mysql> DESC client;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int(11) | YES  | PRI | NULL    |       |
| nom   | char(45) | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> DESC client_extra;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int(11) | YES  | PRI | NULL    |       |
| photo | blob   | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
```

Listing 2. Syntaxe de création d'une table partitionnée

```
CREATE TABLE nom_de_la_table ( <definition_de_la_table> )
  ENGINE = <moteur_de_la_table>
  [definition_de_la_partition]
  [definition_de_la_partition]
  PARTITION BY
    { [LINEAR] HASH(expr)
    | [LINEAR] KEY(column_list)
    | RANGE(expr)
    | LIST(expr) }
  [PARTITIONS nombre]
```

Listing 3. Le type de la colonne est important

```
CREATE TABLE inscrit_hash (
  ID int NOT NULL,
  inscription datetime NOT NULL
) ENGINE=MyISAM
PARTITION BY HASH (inscription) PARTITIONS 4 ;
ERROR 1491 (HY000) : The PARTITION function returns the wrong type
CREATE TABLE inscrit_hash (
  ID int NOT NULL,
) ENGINE=MyISAM
PARTITION BY HASH (YEAR(inscription)) PARTITIONS 4 ;
Query OK, 0 rows affected (0.17 sec)
CREATE TABLE inscrit_key (
  ID int NOT NULL,
  inscription datetime NOT NULL
) ENGINE=MyISAM
PARTITION BY KEY (inscription) PARTITIONS 4 ;
Query OK, 0 rows affected (0.09 sec)
```

Listing 4. La clé de partition doit être au moins une partie de la clé primaire

```
CREATE TABLE commande (
    ID_commande int NOT NULL AUTO_INCREMENT,
    ID_client smallint NOT NULL,
    PRIMARY KEY (ID_commande)
) ENGINE=MyISAM
PARTITION BY RANGE (ID_client) (
    PARTITION p0 VALUES LESS THAN (16384),
    PARTITION p1 VALUES LESS THAN (32768) );
ERROR 1503 (HY000): A PRIMARY KEY must include all
    columns in the table's partitioning function
CREATE TABLE commande (
    ID_commande int NOT NULL AUTO_INCREMENT,
    ID_client smallint NOT NULL,
    KEY (ID_commande)
) ENGINE=MyISAM
PARTITION BY RANGE (ID_client) (
    PARTITION p0 VALUES LESS THAN (16384),
    PARTITION p1 VALUES LESS THAN (32768) );
Query OK, 0 rows affected (0.11 sec)
```

Listing 5. Une table avec un moteur de stockage csv ne peut être partitionnée

```
CREATE TABLE journalisation (
    ID int(11) NOT NULL,
    evenement char(35) NOT NULL DEFAULT '',
    ) ENGINE=csv
PARTITION BY KEY (id) PARTITIONS 3 ;
ERROR 1572 (HY000): Engine cannot be used
    in partitioned tables
```

Listing 6. Table partitionnée par range

```
CREATE TABLE employe_range (
    ID tinyint unsigned NOT NULL,
    Name char(35) NOT NULL DEFAULT '',
    PRIMARY KEY (ID)
) ENGINE=MyISAM
PARTITION BY RANGE (ID) (
    PARTITION p_id_moins_de_64 VALUES LESS THAN (64),
    PARTITION p_id_moins_de_192 VALUES LESS THAN (192),
    PARTITION p_id_moins_de_256 VALUES LESS THAN (MAXVALUE));
```

Listing 7. L'ordre des partitions est important

```
CREATE TABLE employe (
    ID tinyint unsigned NOT NULL,
    Name char(35) NOT NULL DEFAULT '',
    PRIMARY KEY (ID)
) ENGINE=MyISAM
PARTITION BY RANGE (ID) (
    PARTITION p_id_moins_de_192 VALUES LESS THAN (192),
    PARTITION p_id_moins_de_64 VALUES LESS THAN (64),
    PARTITION p_id_moins_de_128 VALUES LESS THAN (128),
    PARTITION p_id_moins_de_256 VALUES LESS THAN (MAXVALUE));
ERROR 1493 (HY000): VALUES LESS THAN value must be strictly increasing
    for each partition
```

Listing 8. Vue disque dur d'une table partitionnée

```
employe_range#P#p_id_moins_de_128.MYD
employe_range#P#p_id_moins_de_128.MYI
employe_range#P#p_id_moins_de_192.MYD
employe_range#P#p_id_moins_de_192.MYI
employe_range#P#p_id_moins_de_256.MYD
employe_range#P#p_id_moins_de_256.MYI
employe_range#P#p_id_moins_de_64.MYD
employe_range#P#p_id_moins_de_64.MYI
employe_range.frm
employe_range.par
```

Listing 9. Partition par année avec la fonction year

```
CREATE TABLE inscription (
    ID mediumint unsigned NOT NULL
    AUTO_INCREMENT,
    date_inscription datetime,
    KEY (ID)
) ENGINE=MyISAM
PARTITION BY RANGE
(YEAR(date_inscription)) (
    PARTITION p1 VALUES
    LESS THAN (1970),
    PARTITION p2 VALUES
    LESS THAN (1980),
    PARTITION p3 VALUES
    LESS THAN (1990),
    PARTITION p4 VALUES
    LESS THAN (2000),
    PARTITION p5 VALUES
    LESS THAN (2010));
```

Listing 10. Partition par trimestres avec la fonction to_days

```
CREATE TABLE commande (
    id_commande mediumint
    unsigned
    NOT NULL AUTO_INCREMENT,
    date_commande date,
    INDEX idx (id_commande)
) ENGINE = MYISAM
PARTITION BY RANGE(
    TO_DAYS(date_commande) ) (
    PARTITION A2008T1 VALUES
    LESS THAN( TO_DAYS('2008-04-01') ),
    PARTITION A2008T2 VALUES
    LESS THAN( TO_DAYS('2008-07-01') ),
    PARTITION A2008T3 VALUES
    LESS THAN( TO_DAYS('2008-10-01') ),
    PARTITION A2008T4 VALUES
    LESS THAN( TO_DAYS('2009-01-01') ),
    PARTITION A2009T1 VALUES
    LESS THAN( TO_DAYS('2009-04-01') ));
```



**Mettez du piment
dans vos solutions Web !**

- › 1 pincée d'Open Source
- › 1 zeste de créativité
- › 1 cuillère à café d'interactivité
- › 1 concentré de technologie



Core-Techs
Ouverture facile !

Tél. 01 42 01 34 05

www.core-techs.fr

Portails Internet - Ecommerce - Intranet - Extranet - GED - CRM - Développements spécifiques

Listing 11. Table partitionnée par list

```
CREATE TABLE message_list (
    id_message int unsigned
        NOT NULL AUTO_INCREMENT,
    nature_message tinyint
        unsigned NOT NULL,
    KEY (id_message)

) ENGINE=InnoDB
PARTITION BY LIST (nature_message) (
    PARTITION p0 VALUES IN
        (0),
    PARTITION p1 VALUES IN
        (1, 4, 6, 7),
    PARTITION p2 VALUES IN
        (2, 5),
    PARTITION p3 VALUES IN
        (3, 8, 9, 10));

```

Listing 12. Table partitionnée par list avec valeur NULL

```
CREATE TABLE short_message_list (
    id_message int unsigned
        NOT NULL AUTO_INCREMENT,
    nature_message tinyint
        unsigned,
    KEY (id_message)
) ENGINE=Memory
PARTITION BY LIST
    (nature_message) (
    PARTITION p_undef
        VALUES IN (NULL),
    PARTITION p_faux VALUES IN (0),
    PARTITION p_vrai VALUES IN (1));
INSERT INTO short_message_list
    (nature_message) VALUES (0);
Query OK, 1 row affected
(0.00 sec)
INSERT INTO short_message_list
    (nature_message) VALUES (1);
Query OK, 1 row affected
(0.00 sec)
INSERT INTO short_message_list
    (nature_message) VALUES (NULL);
Query OK, 1 row affected
(0.00 sec)
INSERT INTO short_message_list
    (nature_message) VALUES (42);
Query OK, 1 row affected
(0.00 sec)
ERROR 1526 (HY000):
Table has no
partition for value 42
SELECT * FROM short_message_list;
+-----+-----+
| id_message | nature_message |
+-----+-----+
| 3 | NULL |
| 1 | 0 |
| 2 | 1 |
+-----+-----+
3 rows in set (0.02 sec)
```

Listing 13. Table partitionnée par hash

```
CREATE TABLE ville_fra (
    id_ville mediumint unsigned
        NOT NULL AUTO_INCREMENT,
    Population int unsigned
        NOT NULL DEFAULT '0',
    PRIMARY KEY (id_ville)
) ENGINE=Archive
PARTITION BY HASH (id_ville) PARTITIONS 4;
```

Listing 14. Table partitionnée par linear hash

```
CREATE TABLE ville (
    id_ville mediumint unsigned
        NOT NULL AUTO_INCREMENT,
    nom char(35) NOT NULL DEFAULT '',
    Population int unsigned
        NOT NULL DEFAULT '0',
    PRIMARY KEY (id_ville)
) ENGINE=Archive
PARTITION BY LINEAR HASH
    (id_ville) PARTITIONS 4;
```

Listing 15. Partitionnement par key avec un clé primaire

```
CREATE TABLE livre (
    isbn char(13) NOT NULL,
    titre char(255) NOT NULL DEFAULT '',
    auteurs tinyint unsigned NOT NULL,
    PRIMARY KEY (isbn)
) ENGINE=MyISAM
PARTITION BY KEY (isbn)
    PARTITIONS 3;
Query OK, 0 rows affected (0.09 sec)
```

Listing 16. Partitionnement par key sans spécifier de colonnes

```
CREATE TABLE livre3 (
    isbn char(13) NOT NULL,
    titre char(255) NOT NULL DEFAULT '',
    auteurs tinyint unsigned NOT NULL,
    PRIMARY KEY (isbn, titre)
) ENGINE=MyISAM
PARTITION BY KEY () PARTITIONS 7;
Query OK, 0 rows affected (0.09 sec)
```

Listing 17a. Partitionnement par key multi-colonne

```
CREATE TABLE livre2 (
    isbn char(13) NOT NULL,
    titre char(255) NOT NULL DEFAULT '',
    auteurs tinyint unsigned NOT NULL,
    PRIMARY KEY (isbn)
) ENGINE=MyISAM
PARTITION BY KEY (isbn, titre) PARTITIONS 5;
Query OK, 0 rows affected (0.09 sec)
ERROR 1503 (HY000): A PRIMARY KEY must include all
columns in the table's
partitioning function
```

Hébergement Pack Master

- Votre nom de domaine
 - Gestion multi-domaines
 - **1 Go d'espace disque !**
 - Boites emails illimitées
 - Bases de données Mysql
 - Langage html, php, perl
 - Support technique
 - Espace d'administration

68,00€
par an seulement⁽¹⁾

• TTC sans frais cachés

Autres services :

- **Création de site web,**
 - **Référencement,**
 - **Location de serveurs dédiés**



Clicasso.fr

salle hébergeur, votre partenaire

Contact : Informations légales | Voir toutes nos offres d'hébergement

Commandez à [www.domaine.com](#) [ok]

[Espace clients](#)

login : [Mot de passe](#)

[Nos offres](#)

- [Hébergement BASIC](#)
- [Hébergement PERSONNALISÉ](#)
- [Hébergement MASTER](#)
- [domaine INTER](#)
- [domaine FRANCE](#)
- [référencement](#)

Service commercial | Support technique

[N° Indigo](#) 0 820 320 256

de 10 à 50 Mo [ok]

votre formule : de 10 à 50 Mo [ok]

de 10 à 50 Mo [ok]

Clicasso est à votre service depuis 7 ans avec des offres pour tous les budgets, peu de frais cachés, frais de mise en service offerts, support gratuit et illimité, ...

Fini le casse tête ! Gérez votre site avec toute la sécurité de DirectAdmin, 100% personnalisé, nous vous proposons un guide complet et ligne...



Pack.Perso
49 Ctc/an



PAOshop[®]
Votre communication à l'épreuve

PAOshop[®]

PAOshop[®]

PAOshop[®]

PAOshop[®]

PAOshop[®]

www.clicasso.fr

► N° Indigo 0 820 320 256

0,09 € TTC / MN

Listing 17b. Partitionnement par key multi-colonne

```
CREATE TABLE livre4 (
    isbn char(13) NOT NULL,
    titre char(255) NOT NULL DEFAULT '',
    auteurs tinyint unsigned NOT NULL,
) ENGINE=MyISAM
PARTITION BY KEY (isbn, titre) PARTITIONS 5 ;
```

prend une date en entrée et retourne un entier qui représente le nombre de jours depuis l'année 0. Comme vous pouvez le constater dans le listing 10, la fonction peut également être mise lors de la définition de chaque que la valeur `NULL` est considérée comme étant inférieure à tous les nombres entiers, en d'autres termes si le champ a des valeurs `NULL`, ces dernières iront dans la première partition.

Le partitionnement par LIST

La clause de partitionnement est `PARTITION BY LIST` (colonne). Il faut ensuite donner un nom à la partition : `PARTITION nom_partition` et définir la liste des valeurs pour chaque partition : `VALUES IN (valeur 1, valeur 2,...,valeurN)` (Listing 11).

L'algorithme `LIST` permet de spécifier la valeur `NULL` comme valeur de partitionnement. Par contre si vous essayez d'insérer un nombre qui n'est pas dans la liste des valeurs permises d'une des partitions, une erreur sera lancée par MySQL. Ce mécanisme renforce la cohérence des données dans le SGBDR (Listing 12).

Le partitionnement par HASH

La clause de partitionnement est `PARTITION BY HASH (colonne)`. Il faut ensuite donner le nombre de partitions souhaitées : `PARTITION nombre_partition` (Listing 13).

L'algorithme de partitionnement hash, idéal pour les valeurs séquentielles, comme les `AUTO_INCREMENT`, permet de distribuer de façon homogène les données dans les différentes partitions créées. La distribution des données sur les partitions est assez simple, c'est du modulo (le reste de la division). Si vous avez 3 partitions, `p0`, `p1`, `p2`, un enregistrement ira dans `p0`, le suivant dans `p1`, le suivant dans `p2`, le suivant dans `p0`, etc...

Pour être plus précis, le calcul fait par MySQL est le suivant :

```
IF (ISNULL  
    (valeur_partition), 0,  
ABS (valeur_partition))  
MOD nbr de partitions
```

Par exemple, avec 3 partitions :

- le nombre 1974 va dans la partition p_0 , car $1974 \leftrightarrow \text{NULL}$ et $1974 \bmod 3 = 0$
 - le nombre 1975 va dans la partition p_1 , car $1975 \leftrightarrow \text{NULL}$ et $1975 \bmod 3 = 1$
 - la valeur NULL va dans la partition p_0 .

L'algorithme hash, a une variante, `LINEAR HASH`, qui a pour effet de rendre plus rapides les tâches de maintenance sur les tables partitionnées (suppression, fusion, ajout...), mais en contrepartie, la distribution des données sur les différentes partitions est moins homogène (Listing 14).

Le partitionnement par KEY

La clause de partitionnement est `PARTITION BY KEY` (colonne 1, colonne 2,...), avec zéro ou plusieurs colonnes. Il faut ensuite donner le nombre de partitions souhaitées : `PARTITION nombre partition`. (Listing 15).

L'algorithme de partitionnement key est proche de celui du hash, il permet lui aussi de distribuer de façon homogène les données dans les différentes partitions créées. Il se différencie néanmoins par la possibilité donnée à l'utilisateur de ne pas choisir de colonne (Listing 16) ou alors d'en spécifier plusieurs, à condition qu'elles appartiennent toutes à la clé primaire (ou un index unique), ou alors que la table ne contient pas de clé primaire ou d'index uniques (Listing 17).

Il y a encore beaucoup de choses à dire sur les partitions comme comment gérer ses partitions, les fusionner, comment sous-partitionner, mettre ses partitions sur des disques différents...

Mais tout ceci sera pour un prochain numéro.

OLIVIER DASINI

Olivier DASINI est consultant et formateur officiel certifié MySQL. Militant pour la philosophie Open Source, il est responsable du pôle SGBDR chez Anaska Alter Way Group. Auteur de nombreux articles sur MySQL, il est également co-fondateur du MySQL User Group France et co-organisateur du MySQL User Group France.

Les nouveautés de HTML5

HTML5 est la 5ème version majeure de HTML. C'est le successeur de HTML 4.01 et de XHTML 1.0. Ce standard est encore en cours de définition mais certaines fonctionnalités sont déjà disponibles dans les navigateurs. Il intègre Web Forms 2.0, des éléments multimédias et permet une meilleure description du contenu.

Cet article explique :

- Les nouvelles fonctionnalités de HTML 5.
- Les différences entre HTML 4.01 et HTML 5.

Ce qu'il faut savoir :

- Vous devez connaître les bases du langage de balises HTML.

les éléments et attributs qui ne sont pas reconnus sont simplement ignorés. Le service de validation de pages HTML du W3C propose déjà la validation à titre expérimental de documents HTML5 (voir cadre *Sur Internet*).

Cet article présente une partie des nouveautés de HTML5. Vous allez découvrir de nouveaux éléments et attributs dédiés à une meilleure description de la structure de la page web, à l'extension des fonctionnalités des formulaires et à l'intégration de contenus multimédias (vidéos, sons, dessin en deux dimensions dans un canevas). Vous apprendrez également quelles seront les principales différences entre les versions 4 et 5 de HTML.

Niveau de difficulté



Le travail sur HTML5 (*HyperText Markup Language 5*) a débuté dès 2003, à l'initiative du groupe *Opera*. Lors d'un atelier du W3C (*W3C Workshop on Web Applications and Compound Documents*) en juin 2004, *Opera Software* et *Mozilla Foundation* ont présenté conjointement leur opinion sur le futur des applications web et des standards. Ils ont proposé un document de travail dédié à l'extension des fonctionnalités des formulaires de HTML 4. Cette proposition fut rejetée, car elle entraînait en conflit avec la vision du futur du web du W3C, qui travaillait sur la mise au point de XHTML 2.0 et XForms 1.0. Peu de temps après cet atelier du W3C, des personnes de chez *Apple*, *Mozilla* et *Opera* décidèrent de continuer leur travail sur l'extension des fonctionnalités de HTML et fondèrent un groupe de travail : le WHATWG (*Web Hypertext Application Technology Working Group*). En 2006, le W3C a décidé de collaborer avec le WHATWG. Il a créé un groupe (le W3C HTML WG) pour travailler conjointement avec le WHATWG sur la définition de la future norme HTML et XHTML. Depuis 2007, les deux groupes publient la même spécification HTML5, chacun sous sa licence. Des représentants des quatre navigateurs principaux (*Microsoft*, *Apple*, *Mozilla* et *Opera*) participent actuellement à ces groupes de travail.

Une recommandation du W3C est un document qui a valeur de standard. Il passe par plusieurs phases : *Working Draft* (brouillon de travail), *Last Call Working Draft* (dernier appel), *Candidate Recommendation* (candidat à la recommandation), *Proposed Recommendation* (recommandation proposée), *W3C Recommendation* (recommandation du W3C). Les navigateurs commencent l'implémentation des fonctionnalités décrites dans la recommandation bien avant qu'elle n'atteigne son statut final. Le premier *Working Draft* public de HTML5 est paru le 23 janvier 2008. La version actuelle publiée date du 12 février 2009 mais la version en cours (*Editor's Draft*) est modifiée très régulièrement sur le site du W3C et celui du WHATWG. Le W3C annonce la publication du *Last Call Working Draft* et de la *Candidate Recommendation* pour 2009. La recommandation finale est prévue pour 2010.

Bien que la spécification actuelle ne soit pas encore définitive et donc sujette à modifications, certaines sections ont un statut stable et peuvent donc être utilisées dès à présent. Les navigateurs ont commencé à implémenter des parties de HTML5 (*Internet Explorer*, *Firefox*, *Opera* et *Safari*). Les versions des navigateurs permettant de tester les fonctionnalités présentées dans cet article vous seront indiquées au fur et à mesure. Vous trouverez également sur le site de wikipédia une liste à jour des implémentations HTML5 pour les principaux navigateurs (cf. cadre *Sur Internet*). Les navigateurs qui ne sont pas compatibles sont capables de lire et afficher des pages HTML5,

Description du contenu

La nouvelle version de HTML permet une meilleure description des différentes parties d'un document Web, grâce à l'ajout de nouveaux éléments. Il sera possible de définir des en-têtes, des pieds de page, des sections de navigation, des encarts, ... Cela permettra entre autres une meilleure indexation du contenu pour les moteurs de recherche. Ces éléments définissant des parties du document pourront être utilisés dans les feuilles de style CSS. Les éléments *div* et *span* devraient donc être moins utilisés au profit de ces éléments HTML5 per-

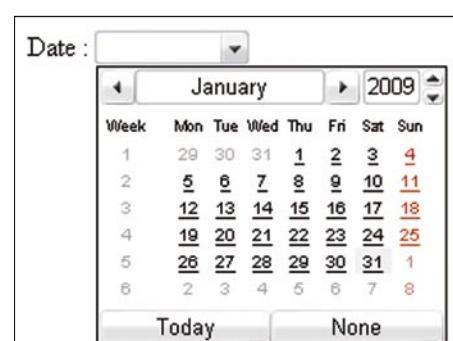


Figure 1. Champ input de type date en HTML5 (navigateur Opera, version 9.63)

mettant de mieux représenter la structure du document :

- L'élément `nav` marque une section de navigation. Les liens hypertextes placés dans cet élément correspondent à un menu de navigation ou à une table des matières (liens hypertextes pointant vers des ressources externes ou des sous-parties du document).
- L'élément `section` marque un groupe-ment sémantique de contenu correspon-dant, par exemple, à un chapitre. Une `section` contient un en-tête (un élément `h1` à `h6` ou le nouvel élément `header` défini ci-après), du contenu (paragraphes, ...) et éventuellement un pied de page (élé-ment `footer` défini ci-après).
- L'élément `header` marque l'en-tête du docu-ment ou le titre d'une section du document. Un `header` peut contenir un ou plusieurs éléments marquant des titres et sous-titres (`h1` à `h6`). Il peut également contenir du tex-te lié à un titre (numéro de version d'un docu-ment, date, historique des versions, ...).
- L'élément `footer` marque le pied de pa-ge du document ou d'une section du docu-ment. Le `footer` contient des éléments relati-ifs à la section dans laquelle il est con-tenu : lien à des pages en rapport avec le con-tenu, données de copyright, liste de liens (HTML5 préconise d'utiliser un élé-ment `footer` pour regrouper une liste de liens de bas de page, plutôt que d'utiliser un élément `nav`), ...
- L'élément `dialog` est utilisé pour indi-quer que le con-tenu est une con-versation. Les interventions des différents interlo-cuteurs sont marquées par des élé-ments `dt` (référance à l'auteur) et `dd` (phrases pro-noncées).
- L'élément `figure` permet de marquer une figure à laquelle un texte fait référ-ence (graphique, photo, vidéo, etc). Le titre de la figure est fixé avec l'élément `legend`.
- L'élément `aside` marque une section dont le con-tenu est lié au con-tenu principal de la page mais peut être lu sé-pa-rément, par exemple une note ou une re-marque.
- L'élément `article` marque une section de page indé-pen-dante du docu-ment. Cet-te section peut être lue de mani-ère indé-pen-dante du con-tenu (comme-mentaire utili-sateur, ar-ticle de magazine, ...).

Le Listing 1 donne un exemple de document HTML5 valide intégrant ces nouveaux élé-ments. Ceux-ci ne sont pas encore imple-mentés dans les navigateurs à la date de rédaction de cet article (février 2009).

Intégration de Web Forms 2.0

Les formulaires HTML4 ne permettent pas de typer les données ou de lier un champ à plu-

Listing 1. Document HTML5

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Mon site perso</title>
  </head>
  <body>
    <header>
      <h1>Site perso</h1>
    </header>
    <nav>
      <ul>
        <li><a href="accueil.html">Accueil</a><li>
        <li><a href="photos.html">Mes photos</a><li>
        <li><a href="blog.html">Mon blog</a><li>
      </ul>
    </nav>
    <article>
      <p>Le dernier film que j'ai vu ...</p>
    </article>
    <section>
      <h2>Bienvenue sur mon site</h2>
      <p>Ce que vous y trouverez</p>
      <section>
        <h3>Mes photos</h3>
        <p>Toutes les photos que j'ai prises...cliquez
           sur Mes photos dans le menu</p>
      </section>
      <section>
        <h3>Mon blog</h3>
        <p>Ma vie, mon train train quotidien...cliquez
           sur Mon blog dans le menu</p>
      </section>
    </section>
    <footer>
      <p>Copyright © 2009 Mon nom</p>
      <p>
        <a href="map.html">Plan du site</a> -
        <a href="contact.html">Contactez-moi</a>
      </p>
    </footer>
  </body>
</html>
```

sieurs formulaires. Dès 2004, *Opera* et *Mozilla* ont travaillé conjointement à l'extension des formulaires HTML4, informellement nommés *Web Forms* 1.0. Le WHATWG a pro-duit plusieurs versions d'un document de travail étendant les possibili-tés des formulai-rires (*Web Forms* 2.0 – *Working Draft*). Début janvier 2009, le WHATWG a indiqué que le document *Web Forms* 2.0 ne serait plus mis à jour car la spécification HTML5 contient une section sur les formulaires, basée sur *Web Forms* 2.0.

Une des nouveautés des formulaires HTML5 est le ty-page des données. De nouvelles valeurs ont été définies pour l'attribut `type` des champs `input` : dates (`date`, `datetime`, `month`, `week`,

`datetime-local`, `time`), `number`, `email`, `url`, `search`, `color`, `range`. Définir un type de con-tenu permettra d'indiquer le type attendu des données et de déléguer au navigateur la vérifi-cation de la validité des données, sans utiliser un langage de script. Des contrôles du type de données seront donc réalisés en natif par le na-vigateur avant envoi, par exemple un champ `number` ne pourra contenir qu'un nom-bre, un champ `date` devra suivre un format prédéfini, ... De nouveaux attributs seront disponibles en fonction du type des données. Par exemple, il sera possible de borner la valeur d'un nom-bre entier (`min`, `max`) et de définir les valeurs autorisées dans cet intervalle (`step`). Un élé-ment `input` de type `number`, dont les attributs

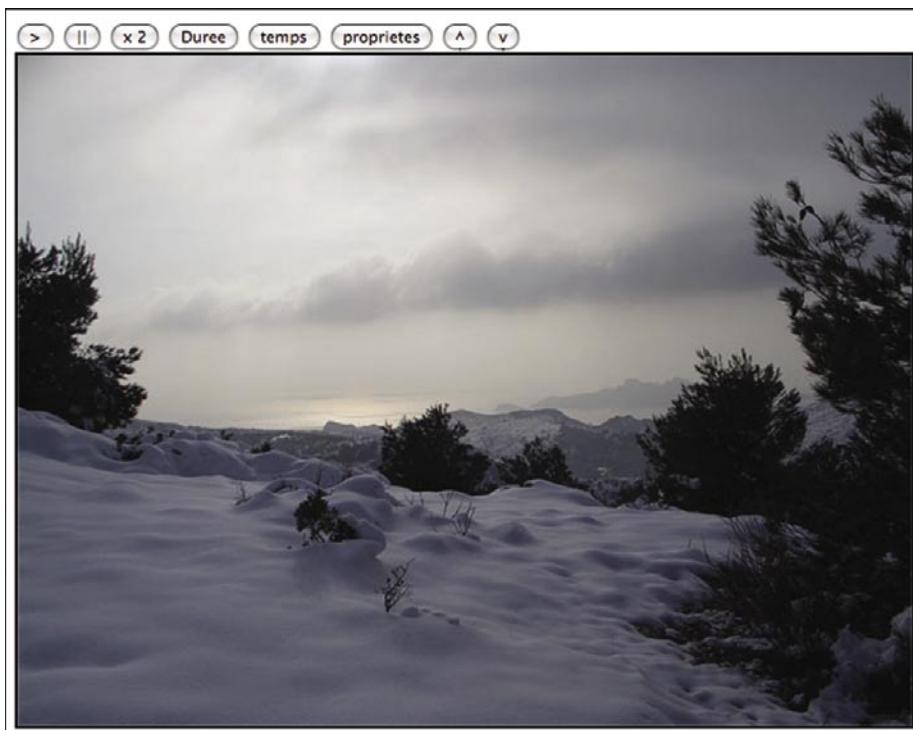


Figure 2. Élément `video` intégré dans une page HTML5, affichage du poster au chargement de la vidéo

`min`, `max` et `step` ont respectivement les valeurs 20, 50 et 10 acceptera les nombres 20, 30, 40 et 50.

La Figure 1 présente un exemple avec un champ du nouveau type date créé avec le code HTML suivant :

```
Date : <input type="date">
```

Depuis la version 9.0 le navigateur Opera fournit des fonctionnalités avancées pour les formulaires. Par exemple le code précédent affiche automatiquement un calendrier lorsqu'un internaute clique dans le champ date. Les nouveautés de HTML5 liées aux formulaires vous seront présentées en détail dans un prochain article.

Ajout d'éléments multimédias

La version 5 de HTML ajoute des éléments dédiés à la gestion de fichiers multimédias (audio, vidéo, animation,...) et une API pour les manipuler. Des formats de données multimédias seront supportés en natif par le navigateur. Cela évitera d'une part l'installation d'un plugin permettant de lire le fichier multimédia (Flash d'Adobe,...). Un panneau de contrôle sera proposé en natif par le navigateur. D'autre part, l'API rendra possible la personnalisation de l'exécution de la vidéo, ainsi que la définition et le contrôle de l'interface utilisateur. La lecture des ressources vidéo ou audio pourra ainsi s'effectuer dans la page par le biais d'un langage de script (pause, contrôle du volume, avance rapide, ...). Il est possible de définir le type de la ressource et le type de média des éléments multimédias vidéo et audio. Ceci permet de proposer plusieurs versions d'une

ressource, dans différents formats (attribut `type`). Le navigateur utilise les attributs pour déterminer quelle ressource il est capable de lire, avant d'effectuer un téléchargement. Ces nouveaux éléments multimédias acceptent également des contenus alternatifs destinés à informer un internaute que la ressource n'est pas disponible. Ces contenus sont destinés aux anciennes versions des navigateurs, ils ne sont pas prévus pour gérer l'accessibilité pour les personnes présentant des déficiences visuelles ou auditives. Dans ce cas, c'est une ressource particulière qui doit être créée, les navigateurs pour les personnes souffrant d'un handicap la sélectionneront en fonction du type de média indiqué (attribut `media`).

Elément vidéo

L'élément `video` a commencé à être implémenté par les navigateurs (*Opera 9.62, Firefox 3.1 beta, Safari 3.2*). Le premier *Working Draft* de HTML5 indiquait l'utilisation du format *Ogg Theora* pour la lecture de l'élément `video`, un format de compression vidéo libre et sans brevet. Cependant les différents groupes impliqués dans la spécification de HTML5 ne se sont pas entendus sur ce format, certains considérant que ce format pouvait potentiellement être breveté dans le futur (principalement les compagnies Nokia et Apple). Il a depuis été retiré de la spécification HTML5 et le format vidéo à implémenter reste une question en attente de réponse. Les navigateurs *Opera* (version expérimentale basée sur la 9.62) et *Firefox* (version 3.1 beta 2) implémentent l'élément `video` pour des ressources au format *ogg*. Le navigateur *Safari* l'intègre également (*Safari* version 3.2), il permet la lecture de fichiers au format *.mov* (format propriétaire d'Apple). Vous pouvez voir un exemple d'utilisation de l'élément `video` dans le Listing 2, dont le résultat est affiché dans la Figure 2.

L'attribut `src` indique l'URL de la ressource vidéo à intégrer dans la page web. Cet attribut est optionnel, il peut être remplacé par un ou plusieurs éléments `source` dans l'élément `video`, comme dans l'exemple du Listing 2. Cette seconde méthode permet d'indiquer plusieurs versions de la ressource, utilisant des formats de compression vidéo différents. Ainsi les différents navigateurs qui n'implémentent pas les mêmes formats vidéo sélectionneront l'élément `source` dont ils peuvent lire le format (attribut `type`) et ignoreront les autres éléments `source`. L'attribut `media` permet également de préciser le type de média pour lequel la ressource est prévue. Dans le Listing 2, le premier élé-

4.9.9 The `tr` element

Categories

None.

Contexts in which this element may be used:

As a child of a `thead` element.

As a child of a `tbody` element.

As a child of a `tfoot` element.

As a child of a `table` element, after any `caption`, `colgroup`, and `thead` elements, but only if there are no `tbody` elements that are children of the `table` element.

Content model:

Zero or more `td` or `th` elements

Element-specific attributes:

None.

DOM interface:

```
interface HTMLTableElement : HTMLElement {
  readonly attribute long rowIndex;
  readonly attribute long sectionRowIndex;
  readonly attribute HTMLCollection cells;
  HTMLElement insertCell([Optional] in long index);
  void deleteCell(in long index);
};
```

Figure 3. Extrait de la spécification HTML5, élément `tr`

Listing 2. Document HTML5 présentant un élément video.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Test video</title>
    <script type="text/javascript">
      function controlVideo(video_id, control, afficher_id){
        var obj_video = document.getElementById(video_id);
        var obj_affichage = document.getElementById(afficher_id);
        switch (control) {
          case 'play' :   obj_video.play();
                           break;
          case 'pause' :  obj_video.pause();
                           break;
          case 'fast' :   obj_video.playbackRate = 2.0;
                           break;
          case 'up' :     obj_video.volume += 0.1;
                           break;
          case 'down' :   obj_video.volume -= 0.1;
                           break;
          case 'duration' : obj_affichage.innerHTML = 'duree : \
                           '+Math.round(obj_video.duration)+' s';
                           break;
          case 'current' : obj_affichage.innerHTML = 'courant : \
                           '+obj_video.currentTime;
                           break;
          case 'properties' : var str = '';
                               for (var i in obj_video){
                                 str += i+"\n";
                               }
                               obj_affichage.innerHTML =
                               "<hr><pre>" + str + "</pre>";
                               break;
          default :      obj_affichage.innerHTML = 'non supporte';
        }
      }
    </script>
  </head>
  <body>
    <button onclick="controlVideo('ma_video', 'play', 'info')">&gt;</button>
    <button onclick="controlVideo('ma_video', 'pause', 'info');">||</button>
    <button onclick="controlVideo('ma_video', 'fast', 'info');">x 2</button>
    <button onclick="controlVideo('ma_video', 'duration', 'info');">Duree</button>
    <button onclick="controlVideo('ma_video', 'current', 'info');">temps</button>
    <button onclick="controlVideo('ma_video', 'properties', 'info');">
      proprietes</button>
    <button onclick="controlVideo('ma_video', 'up', 'info');"></button>
    <button onclick="controlVideo('ma_video', 'down', 'info');">v</button>
    <br>
    <video id='ma_video'
           width = 630
           height = 472
           style = "border:solid black 2px"
           poster="neige_marseille.jpg"
           autoplay
           controls
           loop
           playbackRate=1.0
           defaultPlaybackRate=1.0
           onplay="temps('ma_video', 'current')">
      <source scr='ma_video.ogg' type='video/ogg'>
      <source scr='ma_video.mov'>
    </video>
    <div id='info'></div>
  </body>
</html>

```

ment `source` indique que la ressource est un fichier au format .ogg. Cet élément sera retenu par le navigateur *Mozilla Firefox*, qui téléchargera le fichier ogg. *Safari* quant à lui ignorerà le premier élément `source` et retiendra le second élément `source` qui fournit un lien vers une ressource au format *Quicktime movie* (.mov), qu'il est capable de lire. Le navigateur *Opera* ne reconnaît pas encore les éléments `source` dans l'élément `video`. Il faut utiliser l'attribut `src` de l'élément `video` pour que ce navigateur lise le contenu d'un fichier vidéo (format .ogg).

De nombreux autres attributs sont disponibles pour l'élément vidéo :

- L'attribut `poster` permet d'afficher une image pendant le téléchargement de la vidéo. Si le navigateur n'arrive pas à lire le fichier vidéo alors cette image restera affichée à l'écran. Cet attribut prend comme valeur l'URL de l'image de remplacement. La Figure 2 montre l'affichage du poster dont l'URL est donnée dans le Listing 2 pendant le chargement de la vidéo.
- L'attribut `autoplay` est de type booléen, sa présence dans la balise `video` indique au navigateur que la lecture du fichier vidéo doit être démarrée automatiquement au chargement de la page.
- Les attributs `width` et `height` permettent de définir la taille de l'élément `video` sur la page HTML. Le fichier vidéo s'adaptera aux nouvelles dimensions données si ces attributs sont spécifiés dans la balise ouvrante.
- Les dimensions d'origine de la vidéo sont données par les attributs `videoWidth` et `videoHeight`.
- L'attribut `controls` (type booléen) indique au navigateur d'afficher le panneau de contrôle de la lecture vidéo. Ces contrôles dépendent du navigateur. A l'heure actuelle le navigateur *Opera* affiche des boutons *play*, *pause* et *stop*. Le navigateur *Mozilla Firefox* affiche un bouton *play*, un bouton *pause* et un bouton permettant de couper/remettre le son de la vidéo en lecture. Le navigateur *Safari* affiche ces mêmes boutons ainsi qu'une barre indiquant l'avancement de la lecture vidéo, qu'il est possible de déplacer. Il affiche de même deux boutons d'avance et de recul dans la lecture vidéo.
- Les vitesses de lecture de la vidéo par défaut et courante sont définies respectivement avec les attributs `defaultPlaybackRate` et `playbackRate`.
- L'attribut `volume` indique le volume de la vidéo, celui-ci peut être modifié au cours de la lecture.
- Les attributs `duration` et `currentTime` donnent respectivement la durée totale en

secondes de la vidéo et le temps écoulé depuis le lancement de la vidéo.

Certains attributs peuvent être modifiés au cours de la lecture de la vidéo, en utilisant du code JavaScript. Ceci permet d'implémenter ses propres contrôles vidéo. Le Listing 2 donne la possibilité de jouer sur la vitesse de lecture vidéo en modifiant la valeur de l'attribut `playBackRate`. Cet attribut doit être initialisé à la valeur 1.0 dans la balise `video`, tout comme l'attribut de vitesse de lecture par défaut `DefaultPlayBackRate`. Il est ensuite possible de modifier la valeur de l'attribut de vitesse de lecture `playBackRate` en JavaScript, pour accélérer ou ralentir la vitesse du film. Dans l'exemple du Listing 2, la vitesse de lecture est doublée lorsque l'utilisateur clique sur le bouton contenant la chaîne `x2`. Le clic sur le bouton provoque l'exécution de la fonction JavaScript `controlVideo`. Cette fonction JavaScript prend en argument l'identifiant unique de la vidéo, une chaîne indiquant le contrôle à appliquer à la vidéo et l'identifiant d'un bloc `div` dans lequel un texte peut être affiché. Les nœuds de l'arbre du document correspondants à la vidéo et au bloc `div` sont stockés respectivement dans les variables `obj_video` et `obj_affichage`. En fonction du contrôle souhaité, une action est effectuée : lancer la lecture (méthode pré-définie `play`), mettre en pause (méthode pré-définie `pause`), doubler la vitesse normale de défilement (propriété `playbackRate`), régler le volume (attribut `volume`), afficher la durée totale de la vidéo (attribut `duration`) et le temps courant (attribut `currentTime`). Le bouton propriétés, quant à lui, affiche l'ensemble des propriétés implémentées par le navigateur dans lequel la vidéo est lue. Bien entendu ces propriétés dépendent du navigateur lui-même et de sa version.

Elément audio

L'élément `audio` fonctionne de la même manière que l'élément `video`. L'URL de la ressource est fixée avec l'attribut `src`. La ressource peut être lancée automatiquement (`autoplay`), jouée en boucle (`loop`), un panneau de contrôles natif peut être affiché (`controls`) et de nombreux attributs et fonctions sont disponibles pour contrôler l'exécution de la ressource (`pause`, `volume`, `vitesse`, ...) et obtenir des informations (durée totale, volume).

Elément embed

L'élément `embed` permet d'ajouter un contenu dont la ressource n'est pas du HTML (application, contenu interactif), comme par exemple du contenu en *Flash*. Le chemin de la ressource est spécifié avec l'attribut `src`, son type avec l'attribut `type`, la hauteur et la largeur de l'élément dans la page avec les attributs `height` et `width`.

Elément canvas

L'élément `canvas` permet de créer une zone de dessin bitmap pour y représenter des graphiques ou des images générées à la volée. Il est déjà implémenté en natif dans plusieurs navigateurs, et est disponible depuis la version 1.5 de Mozilla Firefox, la 2.0.1 de Safari et la 9.0 du navigateur Opera. Le navigateur *Internet Explorer* ne l'a pas implémenté car il utilise la technologie *SilverLight* développée par Microsoft. Le plugin *SilverLight* permet d'ajouter et contrôler des éléments multimédias (vidéo, son, animation, dessin). La bibliothèque JavaScript `ExplorerCanvas`, distribuée selon les termes de la licence *Apache V2.0*, permet d'utiliser `canvas` dans *Internet Explorer*. Ce nouvel élément vous sera présenté en détail dans un prochain numéro de ce magazine.

Differences entre HTML4 et HTML5

La version actuelle du standard HTML est la 4.01 de décembre 1999 (*W3C Recommendation*). La réécriture de HTML 4.01 au format XML est décrite dans la Recommandation du W3C XHTML 1.0 de juin 2000, révisée en août 2002. HTML5 remplacera les spécifications HTML 4, et XHTML 1.0. C'est-à-dire que la spécification HTML5 est indépendante de la syntaxe utilisée pour décrire HTML4 et XHTML (voir section suivante). Elle peut se décliner en HTML ou XHTML.

Une spécification basée sur le DOM 2

Les langages de balises HTML 4.01 et XHTML 1.0 sont définis par des DTD (*Document Type Definition*). La DTD fixe la grammaire : liste des éléments composant le vocabulaire, attributs possibles pour chaque élément, modèle de contenu de l'élément, liste des entités. La DTD HTML 4.01 est définie en SGML (*Standard Generalized Markup Language*). L'extrait de la DTD HTML 4.01 ci-après définit l'élément `TR` (ligne de tableau). Celui-ci doit comporter une balise ouvrante (`<`), la balise fermante est optionnelle (`>`), il contient au moins un élément parmi `TH` et `TD` :

```
<!ELEMENT TR - O (TH|TD)+ -->.
```

XHTML 1.0 est une reformulation de la norme HTML 4.01 en XML. Un document XHTML doit être bien formé, c'est-à-dire qu'il doit suivre des règles XML comme, par exemple, fermer toute balise ouvrante. Le document doit également être valide, i.e. il doit déclarer la DTD XHTML utilisée et s'y conformer. L'élément `tr` est défini ainsi dans la DTD XHTML : `<!ELEMENT tr (th|td)+>`. La déclaration d'élément ne comporte plus d'information sur le caractère obligatoire de la balise fermante car c'est une condition obligatoire pour que le document soit bien formé.

Une différence majeure de HTML5 par rapport à HTML4 et XHTML est que la spécification est à présent basée sur le DOM (Niveau 2). Le document n'utilise plus de DTD pour décrire le langage. De plus, les interfaces proposées pour manipuler l'arbre du document (DOM), sont intégrées dans la spécification. Ces interfaces définissent les propriétés (attributs) et les méthodes disponibles pour chaque élément. La spécification HTML5 décrit le modèle de contenu : le contexte dans lequel l'élément peut être utilisé, les éléments qu'il peut inclure, ses attributs spécifiques et l'interface DOM qu'il utilise. Vous pouvez voir en Figure 3 un extrait de la spécification HTML5 de l'élément `tr`. Il n'y a plus de modèle `block` et `inline` comme en HTML 4. Les éléments ont un modèle de contenu (`Categories` dans la Figure 3) :

- `Metadata` : éléments qui donnent de l'information sur l'information (éléments qui apparaissent dans le `head`, exemple d'éléments de ce type : `title`, `meta`, `link`, `script`, `style`)
- `Flow` : éléments dans le `body` présentés en bloc ou en ligne, l'élément a au moins un descendant nœud texte qui n'est pas un espace entre éléments, ou au moins un descendant élément de type `Embedded` ;
- `Sectioning:sections` qui ont un `heading` ;
- `Heading:header`, `h1` à `h6` ;
- `Phrasing` : texte du document et éléments qui marquent le texte à l'intérieur d'un paragraphe (`abbr`, `span`, `strong`, `em`), le `phrasing content` est inclus dans le `flow content` ;
- `Embedded` : éléments qui chargent des ressources externes (`img`, `object`, `embed`, `video`, `audio`) ;
- `Interactive` : contenu destiné aux interactions avec l'utilisateur (éléments `a`, `input`, `select`, ...).

Eléments et attributs

Comme vous avez pu le voir dans cet article, HTML5 inclut de nouveaux éléments et attributs : éléments multimédias, amélioration de la description du contenu, nouvelles fonctionnalités dans les formulaires, ...

Des éléments HTML 4.01 ont été supprimés dans la version 5, entre autres : les cadres (`frame`, `frameset`, `noframes`), certains éléments de style (`basefont`, `big`, `center`, `font`, `s`, `strike`, `tt`, `u`), et les balises `acronym`, `applet`, `dir` et `isindex`. Certains attributs ont également disparu dans la nouvelle spécification HTML, comme par exemple des attributs liés à la présentation : `background`, `alink`, `link`, `text` et `vlink` dans l'élément `body`, ou encore l'attribut `border` dans l'élément `table` et l'élément `bgcolor` dans les tableaux (éléments `table`, `tr`, `td`, ...).

Sur Internet

- http://wiki.whatwg.org/wiki/Main_Page – Wiki du groupe WHATWG,
 - <http://www.w3.org/html/wg/html5/> – Proposition HTML5 en cours du W3C,
 - <http://www.whatwg.org/specs/web-apps/current-work/> – Proposition HTML5 en cours du groupe WHATWG, mise à jour plus régulière que sur le site du W3C,
 - <http://validator.w3.org/> – Validateur de pages HTML du W3C proposant la validation de pages HTML5,
 - [http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(HTML_5\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML_5)) – Implémentation de HTML5 dans les principaux navigateurs.

Enfin des éléments ont changé de signification. Par exemple les éléments `strong`, `hr`, `big`, `small`, `i`....

Conclusion

Vous avez vu dans cet article quelques unes des nouveautés du futur standard HTML. Celui-ci ajoute des fonctionnalités aux formulaires, en intégrant *Web Forms* 2.0, et propose de nouveaux éléments dédiés à la description de la structure d'une page web (en-tête, pied de page, section de navigation, ...) et à l'intégration de contenus multimédias. Les éléments multimédias en particulier permettront d'agir directement dans le navigateur sur du contenu qui jusque là nécessitait l'utilisation de *plugins*. La définition de ce nouveau standard n'est plus dépendante de SGML et XML, un seul document définit le futur HTML/XHTML. Il décrit les éléments, attributs et méthodes en se basant sur le DOM niveau 2.

Les premières versions de HTML codifiaient les pratiques déjà implémentées dans les nava-

teurs de l'époque. Ainsi la spécification HTML 3.2 du W3C ajoutait à HTML 2 des fonctionnalités déjà proposées par les navigateurs, telles que les tableaux et les applets. Des éléments utilisés par différents navigateurs étaient retenus et d'autres rejetés. Par exemple, dans la norme HTML 4.0, l'élément `embed`, utilisé par *Navigator* et *Internet explorer* pour intégrer des éléments multimédias n'a pas été retenu, la norme préconisant l'utilisation de l'élément `object`. La version 4 a mis l'accent sur la séparation du fond de la forme (feuilles de style), le multimédia et les formulaires. La future version de ce standard, le HTML5, est réalisée cette fois-ci *a priori* et non plus *a posteriori*. Les éditeurs des navigateurs les plus employés du marché collaborent à la définition de ce nouveau standard au sein des groupes de travail du WHATWG et du W3C. Les navigateurs peuvent ainsi aujourd'hui implémenter ces technologies en respectant ces standards. Cette nouvelle optique laisse espérer plus de compatibilité entre les navigateurs dans le futur.

Le HTML 5 est en projet depuis 2003 et bien qu'encore en cours de définition, certaines parties de la spécification sont suffisamment avancées aujourd'hui pour être implémentées dans de nombreux navigateurs. Le W3C a déjà intégré la version HTML5 à son validateur de code HTML. A la date de rédaction de cet article (février 2009) les principaux navigateurs implémentent tous l'élément `canvas` (ajout d'une bibliothèque JavaScript pour *Internet Explorer*). Beaucoup ont commencé à implémenter les éléments `video` et `audio`. Le navigateur *Opera* a implémenté de nouvelles fonctionnalités liées aux formulaires. Ce nouveau standard est donc bel et bien en train de se concrétiser et ouvre la voie à des applications web dont plus de fonctionnalités seront prises en charge nativement (*drag and drop*, contrôle des formulaires) et seront moins dépendantes du navigateur utilisé.

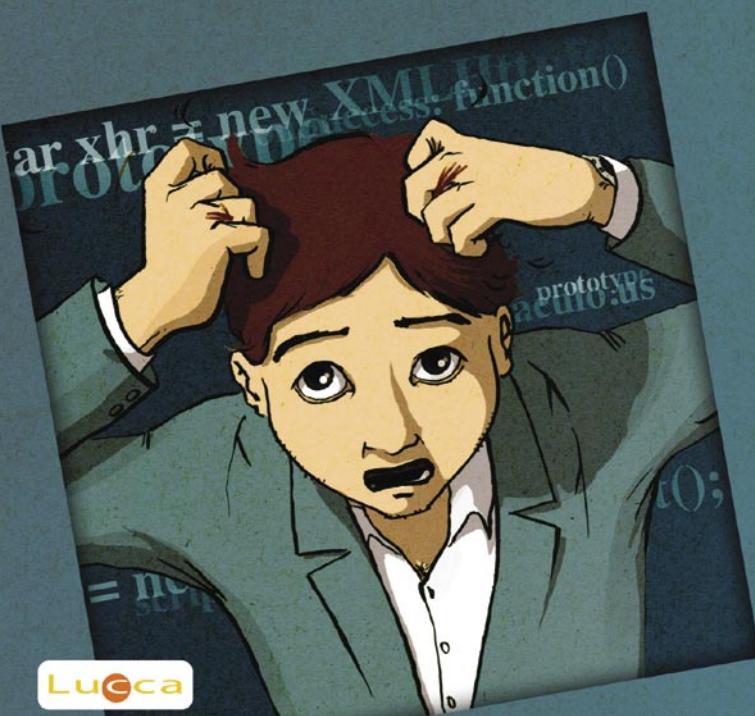
**CÉCILE ODERO,
MAGALI CONTENSIN**

Cécile Odero est spécialisée dans la conception et le développement d'applications Web en PHP. Elle travaille au CNRS comme ingénieur en développement et déploiement d'applications. Contact : cecile.oder@gmail.com.

Magali Contensin, auteur du livre Bases de données et Internet avec PHP et MySQL, est chef de projet en développement d'applications au CNRS, elle enseigne depuis neuf ans le développement d'applications Web à l'Université. Contact : <http://magali.co ntensin.online.fr>.

P U B L I C | T É

Mais non, AJAX n'est pas si difficile !



Lucca, éditeur d'applications multi-utilisateurs, organise des formations intra-entreprises.

Au cours de cette formation, d'une durée de 3 jours et animée par Nicolas Faugout, vous construirez de A à Z une application Ajax de type portail.

Cette formation est destinée aux développeurs d'applications web ayant déjà une bonne pratique du Javascript et des CSS.

Nicolas Faugout est un des auteurs de *Ajax, le guide complet* aux éditions Micro Applications.

ÉCONOMISEZ 30%

Savez-vous que vous pouvez vous offrir plus de joie de programmation et plus d'économies très facilement ?

Tout en profitant de notre offre d'abonnement, vous pouvez :

- avoir vos **6 numéros** pour **35 EUR** au lieu de **45 EUR** (soit **30 % d'économies**)
- profiter de la livraison du magazine à votre domicile
- obtenir un des cadeaux offerts par **PHP Solutions**



Voici votre choix des cadeaux* :

Exemplaire de l'édition spéciale de PHP Solutions

chaque numéro de hors-série de PHP Solutions présente une thématique particulière - à vous de le découvrir

Archives du magazine PHP Solutions 2007
vous y trouverez les six numéros de l'année 2007 en format pdf

Archives du magazine PHP Solutions 2006
vous y trouverez les six numéros de l'année 2006 en format pdf

ByteRun Protector for PHP (49 \$)
un encodeur professionnel qui vous permettra de distribuer vos projets en protégeant votre code

*Offre valable jusqu'à l'épuisement du stock

Pour profiter de notre offre, il suffit de remplir et nous faire parvenir le bulletin d'abonnement, par fax, par courrier ou par émail.

Bonne lecture!

bulletin d'abonnement

phpsolutions

Abonnez-vous
et obtenez
un cadeau !



Merci de remplir ce bon de commande et de nous le retourner par fax : **+31 (0) 36 5407252** ou par courrier :
Software-Wydawnictwo Sp. z o.o.,
Bokserska 1, 02-682 Varsovie, Pologne
Tél. **+31 (0) 36 5307118**
E-mail : **software@emdnl.nl**

Prénom/Nom

Entreprise

Adresse

.....

Code postal

Ville

Téléphone

Fax

Je souhaite recevoir l'abonnement à partir du numéro

.....

En cadeau je souhaite recevoir

.....

E-mail (indispensable pour envoyer la facture)

.....

PRIX D'ABONNEMENT À PHP SOLUTIONS : 35 €

Je règle par :

Carte bancaire n° CB

.....

code CVC/CVV

expire le date et signature obligatoires

type de carte (MasterCard/Visa/Diners Club/Polcard/ICB)

Virement bancaire :

Nom banque :

ABN AMRO

Account number EUR : 59.49.12.075

IBAN-code : NL14ABNA0594912075

BIC-code : ABNANL2A



Dans le prochain numéro de **phpsolutions**

Le périodique *phpsolutions* est publié par Software-Wydawnictwo Sp. z o.o.
Bokserska 1, 02-682 Varsovie, Pologne
Tél. +48 22 427 32 87, Fax. +48 22 244 24 59
www.phpsolmag.org

Président de Software-Wydawnictwo Sp. z o.o. : Paweł Marcinia

Imprimerie, photogravure :
101 Studio, Firma Tęgi
ul. Ekonomiczna 30/36,
93-426 Łódź
Imprimé en Pologne/Printed in Poland

Abonnement (France métropolitaine, DOM/TOM) :
1 an (soit 6 numéros) 35 €

Dépôt légal : à parution
ISSN : 1731-7037
Distribution : MLP
Parc d'activités de Chesnes, 55 bd de la Noirée
BP 59 F - 38291 SAINT-QUENTIN-FALLAVIER CEDEX
(c) 2008 Software-Wydawnictwo, tous les droits réservés

Rédacteur en chef : Łukasz Bartoszewicz

Préparation du CD : Andrzej Kuca
Maquette : Agnieszka Marchocka
Couverture : Agnieszka Marchocka

DTP : Graphics & Design Marcin Ziolkowski
www.gdstudio.pl
Composition : Andrzej Kuca

Correction : Clément Quinton

Bêta-testeurs : Fabrice Gyre, Brice Favre, Valérie Viel, Aymeric Lagier, Christophe Milhau, Alain Ribault, Stéphane Guedon, Eric Boulet, Mickael Puyfages, David Drapeau, Jean-Michel Delhalle, Isabelle Lupi, Antoine Beluze, Claude Brûlé, Timothée Neullas, Adrien Mogenet, Jean-François Montgaillard.

Les personnes intéressées par la coopération sont priées de nous contacter : editor@phpsolmag.org

Abonnement : software@emdn.nl
Fabrication : Marta Kurpiewska marta.kurpiewska@software.com.pl
Diffusion : Katarzyna Winiarz katarzyna.winiarz@software.com.pl
Publicité : publicite@software.com.pl

La rédaction fait tout son possible pour s'assurer que les logiciels sont à jour, pourtant elle décline toute responsabilité pour leur utilisation. Elle ne fournit pas de support technique lié à l'installation ou l'utilisation des logiciels enregistrés sur le CD-ROM. Tous les logos et marques déposés sont la propriété de leurs propriétaires respectifs.

Pour créer les diagrammes on a utilisé le programme  SmartDraw

Le CD-ROM joint au magazine a été testé avec AntiVirenKit de la société G Data Software Sp. z o.o

AVERTISSEMENT

Les techniques présentées dans les articles ne peuvent être utilisées qu'au sein des réseaux internes. La rédaction du magazine n'est pas responsable de l'utilisation incorrecte des techniques présentées. L'utilisation des techniques présentées peut provoquer la perte des données !

En vente dès juillet !

PROJETS

■ Le pathfinding pour jeu en PHP

Dans un article précédent sur le développement de jeu en PHP, la création de carte était décrite pas à pas. Ces cartes utilisées seules permettent au joueur de se représenter le monde dans lequel il fait évoluer son/ses personnage(s) et/ou ses terres. Le pathfinding est une des utilisations possibles de l'Intelligence Artificielle. Elle est très largement utilisée dans 2 principaux domaines : la robotique et les jeux vidéos. Dans cet article vous verrez les avantages et inconvénients des algorithmes les plus utilisés.

SÉCURITÉ

■ Les problèmes de spam dans les formulaires

En effet, si votre site n'a pas encore été victime d'un envoi massif de publicités douteuses pour des produits permettant d'améliorer vos performances sexuelles, sachez que vous n'êtes pas à l'abri, et que vous devez tout de même vous protéger des dommages pouvant arriver. À travers cet article vous verrez les méthodes de protection contre le spam.

PRATIQUE

■ Object/Relational Mapping avec Doctrine

La gestion de la base de données est un point épineux de la création d'une application. La librairie d'*Object/Relational Mapping* Doctrine vous permet de travailler d'un point de vue objet avec votre base de données relationnelle et ainsi de créer une application plus souple, cohérente et facile à maintenir. Après avoir lu cet article, vous saurez comment utiliser la librairie d'*Object/ Relational Mapping* Doctrine.

FICHE TECHNIQUE

■ Introduction à la veille technologique

Trop peu de développeurs PHP pratiquent la veille technologique. Cette activité consiste à suivre les évolutions et les tendances d'un domaine. Dans notre cas c'est le langage PHP qui nous intéresse. Il s'agit donc de surveiller l'activité de la communauté (forums, blogs, projets) du langage et son évolution (versions, fonctionnalités, extensions, etc).

Et de nombreux autres articles à ne pas manquer !

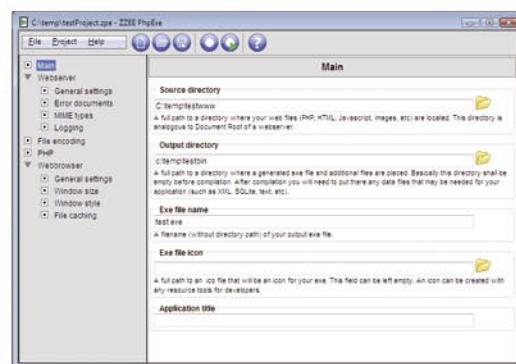
Conquer Windows with PHP

ZZEE PHPExe compiles PHP, HTML, Javascript and other web files into Windows GUI programs. You can rapidly develop Windows GUI applications by employing the familiar PHP web paradigm. Use the same code for online and Windows applications with little or no modification.



Benefits

- Applications compiled with ZZEE PHPExe look and behave like standard Windows GUI applications
- You can distribute your compiled applications royalty free
- Users don't see the contents of your web files or your web file tree, they can not change your scripts
- Integrated PHP encoder - no need to use third-party products for protection of your code
- UI based compiler - easily create and maintain your projects
- You can name compiled exe files as you like and you can assign your own icons to them
- Runtime size can be as low as 2.1Mb (depends on number of PHP extensions that your application uses and your code size)
- Advanced alternative to HTML compilers and HTAs (HTML applications)



Start using ZZEE PHPExe now and enter Windows market with your applications developed in PHP

zzee
<http://www.zzee.com>

Learn more and get your free trial copy here:



<http://zzee.com/phpexe/>

Tous les mois, 75 000 personnes visitent cette boutique.



7 solutions

à partir de

9,99€ HT/mois

soit 11,95€ TTC/mois*

TEST GRATUIT
PENDANT 30 JOURS !

QUELLE QUE SOIT VOTRE
ACTIVITÉ, CRÉEZ VOTRE
E-BOUTIQUE ET BOOSTEZ
VOS VENTES GRÂCE AUX
NOUVEAUX PACKS
E-COMMERCE D'AMEN.

L'offre e-commerce AMEN connaît, depuis son lancement en mars 2007, un succès sans précédent avec déjà plus de 6 000 boutiques installées.

Vous aussi, lancez-vous dans l'e-commerce sans aucune connaissance technique et bénéficiez de nombreuses fonctionnalités enrichies grâce à la nouvelle génération de boutiques en ligne.

- Une solution professionnelle et flexible
- Aucune connaissance technique requise
- Vente de produits et services en ligne avec module de réservation
- Solution multidevises et multilingue
- Systèmes de paiement en ligne intégrés
- Solution adaptée à tout type de catalogue, jusqu'à 30 000 produits
- Design 100% personnalisable et 300 modèles fournis
- Gestion des stocks et statistiques de fréquentation
- Optimisation de votre référencement
- Gestion de vos actions marketing et commerciales (newsletters, coupons, promotions...)

PARTENAIRE
INDUSTRIEL



Pour plus de renseignements : 0892 55 66 77 (0.34 €/mn) ou www.amen.fr

NOMS DE DOMAINE - EMAIL - HÉBERGEMENT - CRÉATION DE SITE - E-COMMERCE - RÉFÉRENCEMENT