



# phpsolutions

Nouvelles technologies et solutions pour les développeurs PHP

PHP Solutions N° 1/2009 (31) ISSN 1731-4593 Prix : 7,50 EUR CD offert France Metro : 7,50 EUR DOM : 8,80 EUR MAR : 80 MAD TOM/S : 990 XPF

## LE POUVOIR DU MULTIMÉDIA

# STREAMING VIDÉO

### PLUS

#### PLONGEZ DANS PHP5

DÉCOUVREZ LES NOUVELLES FONCTIONNALITÉS DE PHP5.3

GÉNÉREZ VOTRE BACK-OFFICE AVEC SYMFONY

XOAD ET VOS APPLICATIONS AJAX EN PHP

EXPORTEZ VOS CLASSES PHP VERS JAVASCRIPT

MANIPULEZ XML AVEC MYSQL 5.1

MYSQL S'ADAPTE À MERVEILLE

ATTENTION AUX ERREURS DE RÉFÉRENCIEMENT !

NE SOYEZ PAS PERDUS

### POUR LES DÉBUTANTS

#### OBJECT ORIENTED PROGRAMMING

ARCANES D'OOP  
DANS PHP5

### OUTILS

#### API ORANGE

ENRICHISSEZ VOTRE  
APPLICATION WEB

### SUR LE CD

#### UNIQUEMENT CHEZ NOUS

COURS VIDÉO

PHPMYADMIN ET SOLITEMANAGER

ENTIÈREMENT EN FRANÇAIS

#### VISTA BUTTONS

VERSION COMPLÈTE DE L'APPLICATION

D'UNE VALEUR DE 49.95 \$

ILLIMITÉE DANS LE TEMPS

L 14389 - 31 - F: 7,50 € - RD



# FAÏTES GRIMPER VOS VENTES

## avec nos solutions e-commerce et/ou facturation nomade

Arcanal vous propose des outils simples, uniques et évolutifs, bâtis sur les références du marché

**arcanal.**

Bâtissons ensemble vos systèmes d'informations

solution n°1

**arcanal** a assemblé pour vous **OSCommerce** et de nombreuses contributions: sécurité, ergonomie, référencement, paiements CB sécurisés...  
Pour créer **votre site marchand OSCommerce** et en assurer l'hébergement :

DÉCOUVREZ NOTRE OFFRE COMPLÈTE

pour 99€ ht / mois.

+ L'hébergement mutualisé  
**ORANGE Business Services** (Oléane)

+ La référence des logiciels  
marchands gratuits : **OSCommerce**

La fourniture et l'installation de la version **OSCommerce MS2** enrichie de diverses contributions.

Les démarches pour la création d'un hébergement mutualisé chez Orange ainsi que pour la création de votre nom de domaine.

Une journée d'assistance, formation et paramétrage (hors éventuels frais de déplacement).

La maintenance d'**OSCommerce** et la mise à disposition de nouveaux modules.

Le premier référencement gratuit chez **Google, Yahoo, Kelkoo et Le Guide.**

Une assistance téléphonique et un accès privilégié aux forums.



solution n°2

**arc@dia** vos livraisons, facturations

et encaissements itinérants

Avec **arcanal** vous choisissez un partenaire qui vous assure sans surprise et sans dérive de budget une solution forfaitaire prenant en compte la totalité des prestations nécessaires à la mise en œuvre de votre système de facturation nomade.

**www.arcanal.fr**

DÉCOUVREZ NOTRE OFFRE COMPLÈTE

pour 31€ ht / mois / vendeur

**www.oscommerce-arcanal.fr** votre solution de vente en ligne  
**arc@dia** vos livraisons, facturations et encaissements itinérants

Pour nous contacter, découvrir nos autres activités, TMA JDE, création et hébergement portail NPDS, formations bureautique:

**arcanal** rue du Champ de Courses Z.I. Monplaisir 38780 Pont-Evêque - Tél: 04 74 16 18 68 - Fax: 04 74 16 18 66 - E-mail : [arcadia@arcanal.fr](mailto:arcadia@arcanal.fr)

[www.arcanal.fr](http://www.arcanal.fr)

[www.oscommerce-arcanal.fr](http://www.oscommerce-arcanal.fr)

[www.arcanal.com](http://www.arcanal.com)

[www.arcanal.net](http://www.arcanal.net)

# PHP, plus qu'un divertissement !

**B**ienvenue dans ce nouveau numéro de PHP Solutions et merci de la confiance que vous accordez à notre magazine !

Nous l'avons vu en 2008, PHP et le modèle objet convergent de plus en plus. Aujourd'hui, PHP 5.3 offre les espaces de noms et la *Standard PHP Library* (SPL) – un ensemble de classes et d'interfaces pré-définies pour PHP – prend une importance telle qu'elle sera indispensable aux futures versions de PHP.

Cette évolution nous apporte des outils et des applications nouvelles, qui se distinguent par leur richesse et leur capacité à évoluer toujours plus importantes. Citons par exemple Zend Framework qui exploite énormément le modèle objet de PHP5, et Magento, une application e-Commerce de référence qui connaît une vitesse de croissance record, tant sur le plan technique que pour son adoption.

PHP fait partie de ces outils qui évoluent rapidement et s'adaptent facilement aux technologies du moment. Malgré ces changements rapides, il est toujours aussi divertissant de développer en PHP, tout comme il est divertissant aujourd'hui de découvrir et de re-découvrir la solution Open Source la plus populaire de tous les temps : Linux.

Que nous réserve PHP à l'avenir ? Nul ne peut le prévoir de manière précise. Il est cependant indéniable que l'adoption de PHP dans l'entreprise en fait une technologie de plus en plus stratégique. Si nous devons reprendre le parallèle entre PHP et Linux, il ne serait pas étonnant de voir dans le futur une implantation de PHP telle qu'elle concurrencerait sérieusement les solutions propriétaires. L'évolution de PHP prouve, autant que celle de Linux, que le modèle Open Source est viable. Alors qu'un développement propriétaire devient rapidement anarchique à partir de 30-50 participants, ce sont des centaines, voir des milliers de personnes géographiquement dispersées qui réussissent à collaborer efficacement sur PHP ou Linux.

Pour en revenir à l'évolution de PHP, la version 6 tarde un peu à se préciser au profit de la branche 5.3. En effet, de nombreuses évolutions initialement prévues pour PHP 6 ont été décalées dans cette branche. Au final, PHP 6 apportera essentiellement la compatibilité totale avec UTF-8.

La communauté évolue elle aussi. En France, l'AFUP est toujours active et une émission de télévision, PHPTV ([www.phptv.fr](http://www.phptv.fr)), voit le jour. Les utilisateurs de PHP chez nos confrères européens, notamment irlandais et belges, sont également de plus en plus actifs.

Guillaume Ponçon – auteur de Best practices PHP 5 et Zend Framework, bien développer en PHP aux éditions Eyrolles - gérant de la société OpenStates, partenaire de PHP Solutions – [www.openstates.com](http://www.openstates.com).



à  
ri  
o  
t  
i  
e

## TABLE DES MATIÈRES

### VARIA

#### 6 Actualités

Actualités du monde du développement

#### 8 Description du CD

Présentation du contenu du CD joint au magazine

#### 12 Interview de Davy Cremer

Interview de Davy Cremer, directeur de COGNIX SYSTEMS – Agence de Rennes

### OUTILS

#### 14 Développer plus efficacement

Adrien Mogenet

XDebug est une bibliothèque pour PHP, distribuée sous la forme d'extension, qui est capable de fournir bon nombre d'informations précieuses. Notez également qu'XDebug s'intègre parfaitement avec symfony. Si vous n'utilisez pas encore d'outils de *debugging*, cet article vous ouvre la voix vers un mode de développement plus efficace.

### PROJETS

#### 18 XOAD : PHP pour un environnement Ajax côté serveur

Laurent Capocchi

Ajax peut être vu à la fois comme un regroupement de technologies et comme une architecture. En effet, le terme Ajax sous entend la mise en œuvre d'un canal de communication asynchrone entre un navigateur et un serveur web, faisant intervenir du Javascript et du XML. Cet article présente la mise en place de l'environnement XOAD pour développer une application web permettant la gestion de tâches en mettant en avant la fonctionnalité du glisser/déposer.

### DOSSIER

#### 24 Le B.A.- BA de la vidéo sur Internet

Vincent Veauclin, Alexandre Rondepierre

Les deux différences principales entre *Streaming* et téléchargement concernent la gestion du cache, ou buffer, et la gestion de la bande passante ; le *P2P*, bien que différent, se rapprochant plus des techniques de téléchargement que du *Streaming*. Après avoir suivi cet article, vous découvrirez les différents types de diffusion : *Streaming*, *téléchargement*, *Progressive Download*, *peer-to-peer*.



### PRATIQUE

#### 32 Générer un document PDF avec PHP

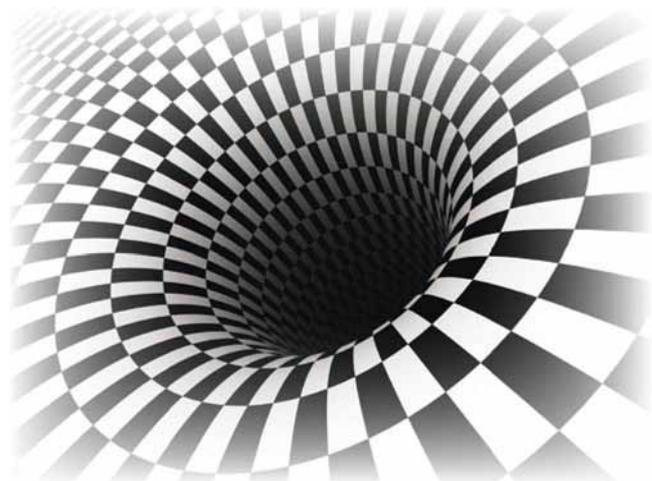
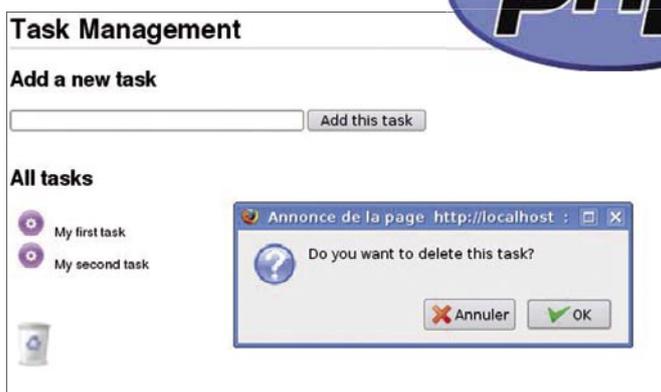
Martin Richard

FPDF est un projet qui propose un lot de fonctionnalités comme le choix du format des pages et des unités, les sauts automatiques de page, la justification, l'import d'images *GIF*, *JPG* et *PNG*, les couleurs et les liens hypertextes. Si la librairie *zlib* est installée, il est alors possible de compresser les fichiers. Le projet est également livré avec un script PHP permettant de convertir des fichiers de police *TrueType* et *Type1* pour les utiliser dans vos PDF.

#### 38 Les erreurs de référencement

Bertrand Andres

Après la lecture de cet article, vous vous apercevrez qu'en évitant les magouilles, et en faisant preuve de bon sens, on peut aboutir à un référencement correct. Cependant, il faut garder à l'esprit que votre site doit constamment évoluer, ainsi que votre référencement. Vous verrez comment éviter les erreurs classiques, et quelles sont les techniques pour l'améliorer.



## FICHE TECHNIQUE

### 42 Génération du BACK-OFFICE avec symfony

Azza Nafti, Adel Safi

Symfony est un framework qui apporte une vraie révolution dans le domaine de développement PHP. Cet article est consacré à l'exploration de quelques fonctionnalités de base de symfony. En plus l'auteur présente la procédure de génération du module back-office d'un site web. L'auteur montre également comment faire des tests pour contrôler les données saisies par l'utilisateur.

### 50 Utiliser XML avec MySQL 5.1

Olivier Dasini, Fabien Allard



MySQL s'adapte à merveille à la gestion de systèmes d'informations éventuellement lourds et complexes et XML a su devenir un langage universel permettant aux applications de communiquer et d'échanger des informations. Grâce à cet article vous allez découvrir les différentes fonctionnalités XML disponibles avec MySQL 5.1.

### 54 Closures et traits

Raphaël Rougeron

Tout comme les *closures*, les *traits* représentent donc une nouvelle fonctionnalité intéressante, qui permettrait aux développeurs PHP d'éviter le copier/coller et de coder d'une façon plus élégante. Encore une fois, cela ne risque pas de bouleverser votre quotidien, mais il est clair que tous les développeurs qui ont eu un jour à gérer un code source imposant seront satisfaits de disposer de ce genre d'outil.



## SÉCURITÉ

### 58 Sécurité avec PDO

Christophe Villeneuve

La sécurité est un point qu'il ne faut pas perdre de vue et cela mériterait de nombreuses lignes d'explications. Il ne faut pas utiliser seulement ce qui est présenté ici car ce n'est pas suffisant. Il faut toujours utiliser l'ensemble des fonctions que vous connaissez et que vous utilisez pour avoir un niveau de protection assez élevé. Aujourd'hui, pour changer un peu des autres exemples déjà utilisés dans les précédents numéros concernant le PDO, vous allez voir l'utilisation de requêtes testant les connexions avec un identifiant et un mot de passe.

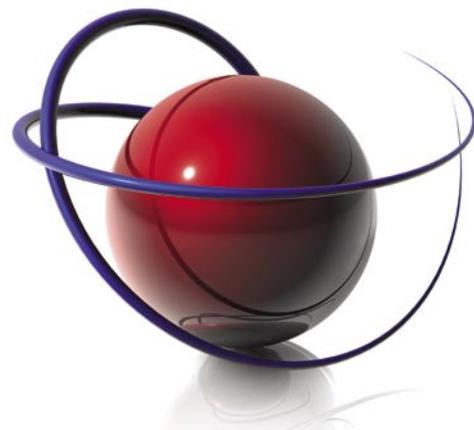


## POUR LES DÉBUTANTS

### 66 Introduction dans les arcanes d'OOP dans PHP5

Jerome Manchon

Alors que PHP6 est en ce moment même en pleine préparation, les statistiques montrent très clairement que PHP5 tend à remplacer PHP4 sur les serveurs web. Il est donc grand temps si vous ne le faites pas déjà de profiter de cette 5ème version, notamment par la mise en œuvre de toute la puissance apportée par l'OOP au sein de celle-ci, avant la sortie imminente de la prochaine version.



### 74 Communication avec authentification avec PHP/CURL

Cécile Otero, Magali Contensin

Dans le numéro précédent de *phpsolutions* vous avez utilisé l'extension *PHP/CURL* pour échanger des données avec un serveur web distant par les méthodes HTTP *GET* et *POST*. Les ressources demandées étaient en accès libre. Vous allez apprendre dans cet article à accéder à des ressources protégées par une authentification HTTP ou par un formulaire de *login* dans une navigation par session. Vous verrez comment échanger des *cookies* entre CURL et le serveur distant et également comment gérer les *cookies* de session.

**LEMUG**

LeMUG (*Le MySQL User Group*) est une nouvelle association, qui mérite quelques lignes. Le but de l'association est de stimuler la communauté MySQL, comme le fait AFUP (*Association Française des Utilisateurs PHP*) et d'accroître les échanges d'expérience pour tirer le meilleur parti de la technologie, et faire émerger les tendances lourdes pour les prochaines années. Un programme ambitieux, et à la taille de la communauté française.

<http://www.lemug.fr>

**Ovidentia 6.7.1**

Ovidentia est un outil de gestion de publication de contenu Open Source et une plateforme collaborative. Ovidentia vous permet d'intégrer simplement et facilement un environnement de publication sur internet ou un portail groupware (internet, extranet, intranet) pour une entreprise, organisation, communauté...

<http://www.ovidentia.org/>

**LimeSurvey 1.72**

LimeSurvey (anciennement PHPSurveyor) est un logiciel libre de sondage en ligne. Il permet aux utilisateurs sans connaissance en développement de publier un sondage et d'en collecter les réponses directement dans leur site internet.

<http://www.limesurvey.org/>

**PHP Excel 2007**

Ce projet est constitué d'un ensemble de classes PHP, qui vous permettent d'écrire et/ou de lire des fichiers Excel 2007. Ce projet est construit autour de Microsoft *OpenXML* standard et PHP. Il intègre de nombreuses fonctionnalités, telles que le paramétrage des méta-données des feuilles de calcul (auteur, titre, description...), plusieurs feuilles de travail, différentes polices et styles de polices, la gestion des bordures des cellules, l'ajout des images à votre feuille de calcul et bien plus encore !

<http://www.codeplex.com/PHPEXcel>

**Trouver les liens dans une page**

Pour extraire les liens d'une page, vous avez trois possibilités : Avec DOM (fonction `getElementsByTagName`), XPath ou les expressions rationnelles. Bien sûr les 2 premiers choix sont possibles au point de vue XML. Mais la dernière possibilité, qui est un peu plus barbare, mais peut réaliser cette extraction.

[http://www.reiersol.com/blog/1\\_php\\_in\\_action/archive/162\\_get\\_links\\_with\\_xpath.html](http://www.reiersol.com/blog/1_php_in_action/archive/162_get_links_with_xpath.html)

**Y! OS 1.0**

Y!OS est la nouvelle plateforme de Yahoo Open Strategy. Elle a pour but de regrouper une plateforme sociale, communautaire et une plateforme application. Un SDK en PHP sera disponible. Mais nous aurons l'occasion d'en reparler car à l'heure où nous écrivons ces lignes, Y!OS n'est pas encore disponible.

[http://developer.yahoo.net/blog/archives/2008/10/yos\\_10\\_launch.html](http://developer.yahoo.net/blog/archives/2008/10/yos_10_launch.html)

<http://developer.yahoo.com/yap/>

# Google Chrome

Google Chrome vous permet de naviguer sur Internet plus vite, plus facilement et en toute sécurité, à l'aide d'une interface très simple qui sait rester discrète.

Dans Google Chrome, accédez à l'intégralité du Web depuis un champ unique. Située dans la partie supérieure de la fenêtre standard de Google Chrome, la barre d'adresse simplifie la navigation sur Internet en faisant également office de champ de recherche. Vous pouvez également l'utiliser pour ajouter des sites Web à vos favoris et vous informer sur la sécurité d'un site Web.

Aux premiers abords, l'interface minimaliste de Chrome peut en surprendre plus d'un, mais après une prise en main facile, Chrome se révèle efficace. Le mot d'ordre pour Chrome est la *rapidité* : La rapidité d'installation, deux minutes suffisent pour le télécharger puis l'installer (avec une connexion haut débit), 30 secondes de plus

si vous souhaitez importer vos données de votre ancien navigateur (vos favoris par exemples). La rapidité des pages sur Chrome dépasse celle du nouveau Firefox 3 qui supplantait déjà amplement *Internet Explorer 7* de Microsoft.

La rapidité de recherche grâce à la barre d'adresse qui simplifie la navigation sur Internet en faisant également office de champ de recherche. Mais Google Chrome n'a pas que des avantages, il est en effet très gourmand en terme de mémoire pour ouvrir rapidement les sites web mis en parallèle dans les onglets. De plus vu son jeune âge, on peut s'attendre à des failles de sécurité.

Google Chrome n'est disponible uniquement sur *Windows XP / Vista* en version BETA. Les versions Linux et Mac sont toujours en cours de développement.

La bataille des navigateurs est ré-ouverte!

<http://www.google.com/chrome/index.html>



français

**Google Chrome (BÊTA) pour Windows**

Google Chrome est un navigateur qui associe une interface très simple à une technologie sophistiquée afin que vous puissiez surfer sur Internet plus vite, plus facilement et en toute sécurité.



Télécharger Google Chrome

Pour Windows Vista/XP SP2

**Zone unique de saisie**

Entrez du texte dans la barre d'adresses et obtenez des suggestions de pages Web et de recherches.

**Miniatures des sites que vous consultez le plus souvent**

Accédez en un clin d'œil à vos pages préférées à partir de tout nouvel onglet.

**Raccourcis vers vos applications**

Créez des raccourcis sur le Bureau pour lancer vos applications Web préférées.

[En savoir plus sur Google Chrome »](#)

## POSH 2.0

La technologie avance vite, mais l'internet va encore plus vite et se tenir informer demande de prendre du temps et comme tout le monde le sait, le temps, c'est de l'argent.

Posh va vous permettre de réduire ce temps que vous passez à rechercher les dernières actualités.

Posh est un outil réalisé par Portaneo, permettant avant tout de regrouper de nombreuses actualités à partir des flux RSS sur votre site comme iGoogle. Grâce à cela, vous pouvez aussi réaliser votre propre portail. Il va vous permettre de regrouper sur une seule page tout ce qui compose internet : vos informations, vos outils et vos produits préférés. Posh est bien-sûr réalisé en PHP avec de l'Ajax et une base de données MySQL. Il propose actuellement plus de 100 modules, par exemple :

- Déplacement, ajout, suppression et configuration des gadgets dans ses pages personnalisables,
- API pour réaliser ses propres widgets ou en importer,
- Gestion des pages personnalisables par onglets,
- Gestion des thèmes graphiques.

Mais la liste augmente régulièrement, Pour vous faire une idée des possibilités, vous pouvez visualiser les vidéos de démonstration fournies par son éditeur, mais aussi des sites comme ACTU F1 ou MyFoot qui ont été réalisés par cet outil.

<http://www.portaneo.com/site/index.php?lang=fr>

## RPG Illusion

Construire son propre univers ou son propre monde n'est pas toujours réalisable. RPG Illusion, vous fera changer d'avis, car vous allez pouvoir le réaliser le monde que vous rêvez. Bien sur des jeux en lignes, il en existe beaucoup, mais créer son propre jeu en ligne, c'est plus rigolo et beaucoup plus intéressant. RPG Illusion est totalement gratuit et par conséquent se trouve être en open source. Il est maintenu par une communauté qui pense avant tout de s'occuper de vos loisirs.

Les possibilités qui vous sont offertes sont :

- Ajout d'une map,
- Ajout d'un espace d'entraînement,
- Commentaires possible sur la news des villes,
- Ajout d'une banque,
- Envoi de mails aux inscrits via l'administration,
- Affichage news - résumé + complet (ajout d'images possible),



- Ajout des 150 images de monstres,
- Ajout d'infobulle dans le classement,
- Possibilité de créer une maison..

La communauté autour de ce projet est assez active car elle propose des mods à ajouter facilement pour améliorer votre jeu... Après avoir réalisé son univers, il ne vous reste plus qu'à ouvrir ses portes aux autres joueurs, et tout se déroule tranquillement. Vous pouvez vous rendre sur le site internet car plusieurs versions de démonstration existent pour vous donner les idées. Donc à découvrir sur le site <http://www.rpgillusion.fr/>

## MAGENTO

Magento est une plateforme e-commerce. Elle a été développée sur les bases du Framework Zend et donc en PHP, et d'après ce que j'ai pu voir dans le source, utilise PDO et est donc compatible avec toutes les bases de données. Magento est disponible sous la licence logiciel libre GNU General Public Licence version 3. Et en 2008, Magento a fêté sa première bougie. Mais il n'est pas seul car il existe une large communauté Française avec de nombreux sites références. En 2008, déjà 160 modules ont été créés et c'est dans la logique de voir les chiffres augmenter. PHP Solution vous parle de cette plateforme car la nouvelle version 1.1.6 vient de sortir. Mais surtout quand nous l'avons testée, nous pensions que c'était un outil pensé par des personnes qui connaissent la vente pour vous faciliter la

vente en général. De nombreux modules gratuits ont été développés par la communauté, qui sont :

- Éditeur WYSIWYG,
- Configuration des modes de livraisons (La poste par exemple),
- Un module compatible pour les utilisateurs *FIA-NET*,
- Systèmes de paiement en ligne,
- etc.

Mais Magento propose aussi des modules pour communiquer avec d'autres applications comme. Des blogs : *Wordpress, Lazymonks* :

- ERP : *OpenERP*

<http://www.magentocommerce.com/fr/>



### GEOIP PHP API

Il existe différentes méthodes d'interrogation des bases de données *GeoIP* en PHP et la plus facile reste l'utilisation d'une API de votre langage favoris. Mais *Mod\_geoip* propose une autre méthode avec PHP. Il est un compromis entre performance et facilité d'installation. Le module Apache est plus rapide (de référence), mais le pure module PHP est beaucoup plus facile à installer.

<http://www.maxmind.com/app/php>

### GreenSQL

La sécurité n'est pas à prendre à la légère. *GreenSQL*, qui est une base de données Open Source conçue pour être utilisée comme un proxy MySQL, permet de jouer un rôle de *Firewall* (pare feu) : au lieu de se connecter directement à la base de données, le serveur Web se branche sur *GreenSQL* pour détecter le code malicieux.

<http://www.greensql.net/>

### Dwoo 1.0

Dwoo est un moteur de template Smarty compatible avec les modèles. Réalisée en PHP5, cette version apporte de nouvelles fonctionnalités et de la syntaxe de sucre. Si vous voulez en savoir plus je vous conseille d'aller jeter un coup d'œil sur l'application.

<http://dwoo.org>

### PHP 5.3 alpha1 released

L'équipe de développement PHP est fière d'annoncer la première sortie alpha de la mise à jour de la version mineure de PHP 5.3. Les fichiers binaires pour Windows seront dès le départ disponibles avec alpha2. On attend de la nouvelle version PHP 5.3 l'amélioration de la stabilité et l'optimisation des performances ainsi que l'ajout de nouvelles syntaxes de langue et d'extensions. Plusieurs nouvelles fonctions ont déjà été documentées dans la documentation officielle...

### Dotclear 2.0.2 automatisé

Dotclear est un logiciel libre de publication web simple d'emploi permettant à tout un chacun de publier sur le web et ce, quel que soit son niveau de connaissances techniques. Cette nouvelle version permet entre autre de gérer le multi blogs, le multilingue, les photos et vidéos, et l'installation est dorénavant automatisée !

<http://www.dotclear.net/>

### Protégez votre site web contre les robots

La librairie *Artichow* vous permet de protéger efficacement et facilement vos pages contre les requêtes automatisées de robots. Cette fonctionnalité peut être utilisée pour filtrer l'inscription à un site communautaire par exemple, ou lutter contre le  *flood*.

<http://www.artichow.org>



## Cours vidéo : initiation à PHPMyAdmin et à SQLiteManager

Nous vous présentons un cours vidéo réalisé par Aymeric Lagier. Dans cette vidéo vous allez découvrir 2 gestionnaires de bases de données : PHPMyAdmin et SQLiteManager. L'un pour MySQL et l'autre comme son nom l'indique pour SQLite.

Les bases de données sont devenues incontournables dans le développement d'applications sur Internet. Vous pouvez y stocker des gigaoctets d'informations très facilement. Cependant tout le monde n'a pas les connaissances suffisantes pour administrer une base de données uniquement avec le langage SQL. C'est pourquoi il existe des gestionnaires de bases de données permettant d'utiliser une base de données grâce à une interface web intuitive ne demandant peu voire pas de connaissances en SQL. Ce système permet également une gestion beaucoup plus rapide de la base de données pour les développeurs web.

Vous connaissez tous PHPMyAdmin, le fameux gestionnaire de bases de données MySQL installé sur tous les hébergements mutualisés que vous proposent des sociétés comme OVH, 1&1, etc...

Vous apprendrez comment réaliser les opérations de base comme SELECT, INSERT, DELETE, UPDATE, CREATE TABLE et tout ceci sans aucune connaissance du langage SQL, simplement en remplissant des formulaires. Pour les personnes un peu plus initiées, la vidéo traite de l'exportation/importation de bases de données sans passer par la ligne de commande. Enfin nous passerons un peu de temps sur les petites fonctionnalités assez pratiques que propose PHPMyAdmin et qui sont souvent mal connues des utilisateurs standards.

Malheureusement, nous n'apprenons pas à installer PHPMyAdmin sur un serveur car l'installation est un peu complexe et la traiter en détails pour une bonne compréhension serait trop long. Pour ce qui est de SQLiteManager, il est assez peu connu par le grand public car utilisant SQLite et non MySQL. SQLite a la particularité par rapport à MySQL de ne pas avoir une architecture client/serveur mais d'être intégré directement aux programmes en utilisant des fichiers de bases de données. Dans la vidéo nous apprendrons à faire une installation basique de SQLiteManager. Attention l'installation que vous allez voir

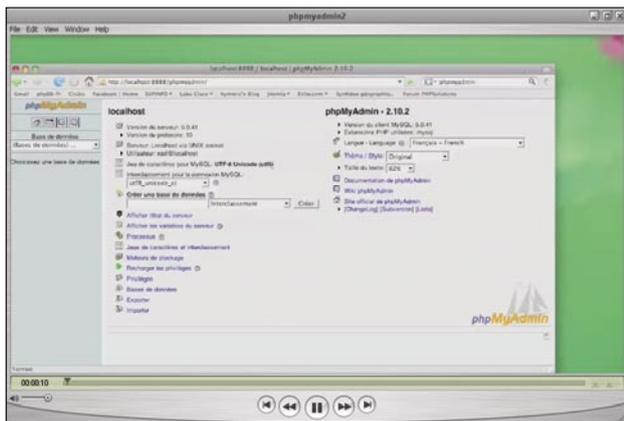


Figure 1. Initiation à PHPMyAdmin et à SQLiteManager

n'est pas sécurisée, tout le monde va pouvoir accéder à vos bases de données sans aucun login ni mot de passe. Je souhaitais juste vous montrer comment il est simple d'installer un SQLiteManager de base. Nous reprendrons le même plan que pour PHPMyAdmin, c'est à dire les commandes de bases (INSERT, SELECT, UPDATE, etc...). L'interface de SQLiteManager étant beaucoup plus simple que celle de PHPMyAdmin, il y a par conséquent moins d'options que dans cette dernière.

## Vista Buttons (49.95 \$)

Les Vista Buttons vous aident à créer facilement des boutons de défilement HTML et des menus déroulants pour Vista/XP/Windows 98/Mac. Avec une interface claire et compréhensible offerte par Vista Web Buttons, il vous faut simplement 4 étapes élémentaires pour créer vos boutons Web et vos menus HTML et si vous utilisez un exemple pré-conçu comme base, vous aurez ce résultat encore plus rapidement.

Les Vista Buttons fournissent + de 500 boutons Web, 100 sous-menus, 50 exemples déjà prêts, de telle façon que vous pourrez créer des menus et des boutons HTML avec un look web vraiment sympa, avec peu ou aucune connaissance en conception. Les boutons Vista, XP, Windows 98, Mac, Aqua, circulaire, brillant, métallique, style 3D, menus tabulaires, menus déroulants, seront décoiffants.

Vous n'avez pas besoin de connaître ni HTML, ni Javascript, ni CSS, ni aucun des autres langages de programmation pour réaliser des boutons WEB allers/retours multi-états, des cross-navigateurs professionnels, des menus DHTML sympas de moteurs de recherche. Tout ce que vous avez à faire se limite à quelques clics et à ajuster les boutons comme vous souhaitez qu'ils apparaissent. Les Vista Web Buttons vont générer toutes les images, code HTML, Javascripts, styles CSS nécessaires automatiquement.

Chaque bouton ou paramètre de menu peut être facilement configurable dans les Vista Buttons pour correspondre à la conception de votre site web ou à vos besoins. Créer vos propres menus HTML, aussi simples et aussi complexes que vous voulez.

Nous vous proposons la version complète de l'application d'une valeur de 49.95 \$.

Pour plus d'informations sur le produit, nous vous invitons à visiter le site : <http://vista-buttons.com>

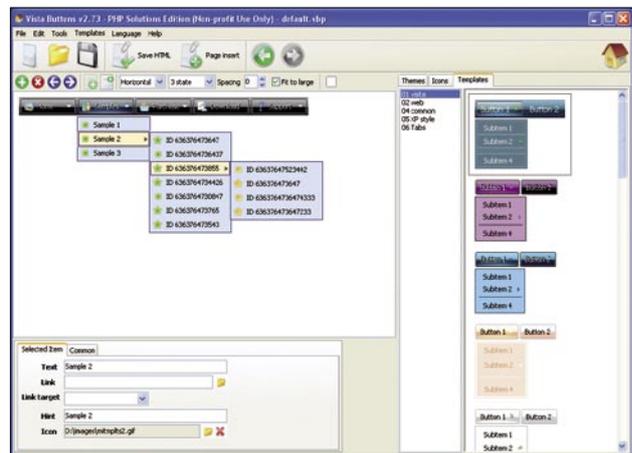
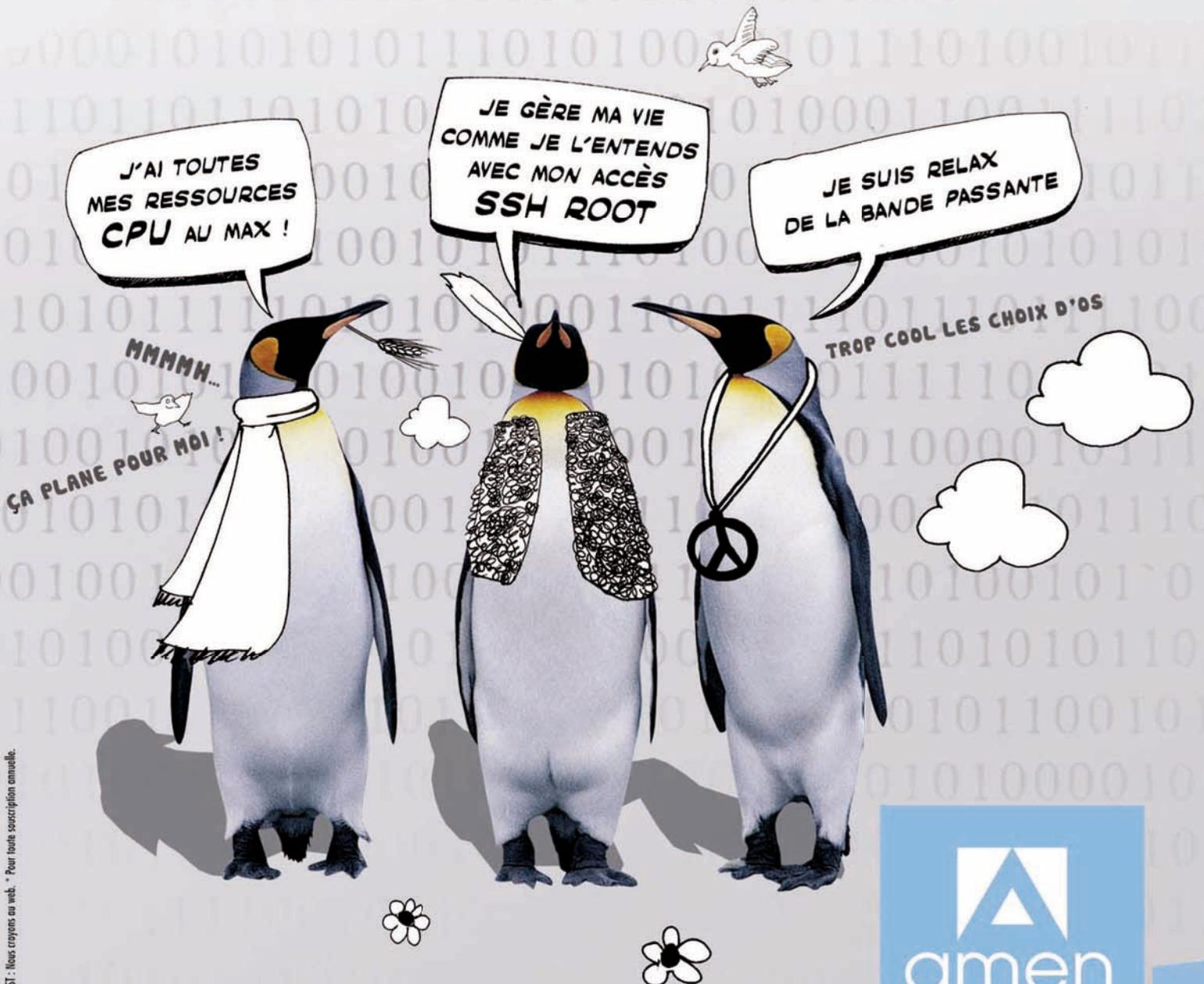


Figure 2. Vista Buttons

# NOUVEAU VDS+ d'AMEN : le bonheur est dans le serveur !



À PARTIR DE  
**5€<sup>HT</sup>** /MOIS  
soit 5,98 € TTC/MOIS\*

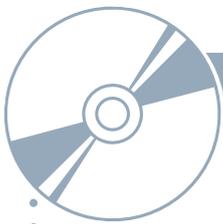
**SERVEURS PRIVÉS AMEN :  
BÉNÉFICIEZ DE RESSOURCES  
GARANTIES QUI VOUS SONT  
PROPRES (PROCESSEUR,  
MÉMOIRE, DISQUE DUR...)  
TOUT EN PROFITANT D'UNE  
PLATEFORME INFOGÉRÉE  
24H/24 - 7J/7.**

- Hébergement multi-sites/multi-domaines
- Interface d'administration : Plesk 8.6
- Systèmes d'exploitation : Fedora Core 8, Suse 10.3, Debian 4.0, Ubuntu 8.04 ou CentOS 5
- Part CPU minimum : de 1 à 6
- Mémoire garantie : de 256 Mo à 1 Go
- Espace disque : de 5 Go à 30 Go
- Bases de données : illimitées
- 1 adresse IP fixe
- Accès Root



Pour plus de renseignements : 0892 55 66 77 (0.34 €/mn) ou [www.amen.fr](http://www.amen.fr)  
NOMS DE DOMAINE - EMAIL - HÉBERGEMENT - CRÉATION DE SITE - E-COMMERCE - RÉFÉRENCIEMENT

REFLEXIONFAITE - Conditions Générales de Vente sur le site [www.amen.fr](http://www.amen.fr) AMEN RCS PARIS : 8 431 527 797. IN WEB WE TRUST : Nous croyons au web. \* Pour toute souscription annuelle.



### Matériaux supplémentaires

Nous avons mis à votre disposition de nombreuses applications Open Source. En supplément des articles du numéro vous trouverez : CMS Made Simple, Drupal 6.6, eZ Publish 4.0.1, Joomla! 1.5.7, Typo3 4.0.1, Xoops 2.0.18, MySQL 6.0.7, symfony 1.1.4, XAMPP 1.6.8, SQLiteManager 1.2, PHPMyAdmin 3.0.1, Xdebug 2.0.3, Xoad 0.6.0.0.

### CMS Made Simple

CMS est un système de gestion de contenu (en anglais, CMS, pour *Content Management system*).

Un CMS est un logiciel web qui vous permettra de créer un site internet dynamique en toute simplicité. CMS Made Simple, écrit en PHP / MySQL, est un logiciel CMS Open Source distribué sous licence GNU/GPL (gratuit), extensible, OO (*Orienté Objet*), simple, efficace et facile à utiliser. CMS Made Simple est conçu pour développer des sites de toutes tailles, personnels ou communautaires, des portails d'entreprise, associatifs, corporatifs ou des sites intranet. Vous pourrez mettre en ligne du contenu et mettre à disposition de vos visiteurs des services (forum, boutique en ligne, galerie photos...).

Pour plus d'informations, nous vous invitons à visiter le site : <http://www.cmsmadesimple.fr/>



Figure 3. CMS Made Simple

### Joomla!

Joomla! est un système de gestion de contenu pour l'Internet (Content Management System) qui se démarque des Nukes, Xoops, eXoops, Zope... par sa modularité et de Typo3 par sa simplicité d'utilisation et d'administration. Joomla! a été développé pour tous. Sous licence GNU/GPL, il est facile à installer, à administrer et il est fiable. Joomla! n'exige pas que l'utilisateur ou l'administrateur du système maîtrise le langage HTML pour permettre son utilisation.

Il permet une grande flexibilité pour l'affichage de votre contenu : si vous utilisez Joomla! pour un blog, des nouvelles ou le site Web d'une société, vous constaterez qu'un ou plusieurs styles de contenu

permettent de présenter vos informations. Vous pouvez aussi changer le style de contenu dynamiquement selon vos souhaits.

Côté visiteur, le contenu est affiché au travers du gabarit (ou *template*) constitué habituellement d'un seul fichier (*index.php*) en plus de la classique feuille de style et des fichiers images. Les fichiers d'origine permettant l'affichage habituel peuvent également être modifiés pour une mise en conformité accrue avec les standards web en s'affranchissant totalement des tables dans la mise en page. Les emplacements ainsi définis dans le template (*main, left, right, top, bottom* ou des emplacements personnalisés *user 1, user 2...*) permettent de paramétrer et de personnaliser totalement l'aspect de votre site simplement en plaçant vos publications et blocs d'affichage depuis l'interface d'administration (une zone de nouvelle flash, une bannière publicitaire, des nouvelles, des articles, un login de connexion, un module de recherche, etc.).

Avec Joomla! il est extrêmement facile d'ajouter et de présenter le contenu de différentes manières. Tout le contenu est placé dans le corps principal (*mainbody*) de votre template (*gabarit*). Il y a trois types principaux de dispositions disponibles dans Joomla! Tous peuvent être personnalisés par différents paramètres lors de la création de contenu ou de liens de menu.

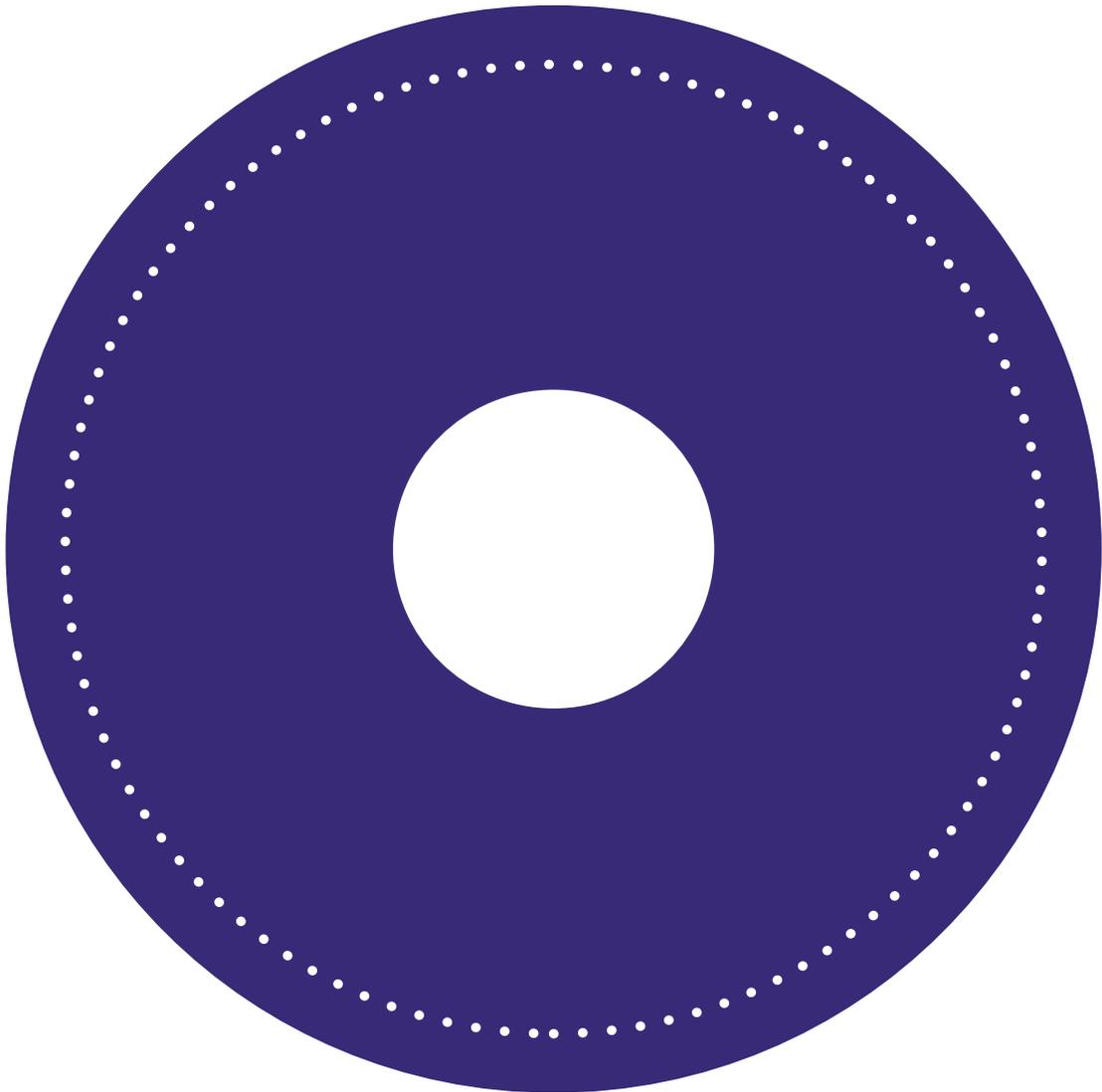
Pour plus d'informations, nous vous invitons à visiter le site : <http://www.joomla.fr/>



Figure 4. Joomla!

*Note* : nous présentons nos excuses à Monsieur Henri Thérét car nous avons oublié d'informer nos lecteurs qu'il a réalisé des tutoriels vidéo qui ont été publiés dans le numéro 4/2008. Ces tutoriels portaient sur la gestion du routage avec Zend Framework. Henri Thérét est concepteur et développeur web, il a travaillé en *freelance* et au sein de plusieurs sociétés de création multimedia. Aujourd'hui, il a intégré l'équipe de production de *Stonfield InWorld*. Son activité consiste d'une part à rechercher et étudier des solutions techniques, d'autre part à développer des applications internet classiques, rich-medias ou destinées à interagir avec des univers virtuels.

**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](mailto:cd@phpsolmag.org)**



# Interview de Davy Cremer

## Davy Cremer

directeur de COGNIX SYSTEMS  
Agence de Rennes

La perspective de PHP5, en particulier au niveau du développement objet, était rassurante sur l'évolution à moyen terme du langage et j'avoue avec le recul, ne pas regretter le choix de PHP

### PHP Solutions : Pourquoi avez-vous choisi de vous orienter vers les technologies PHP et AJAX ?

**Davy Cremer :** Lorsqu'en 2001, mon futur associé Fabrice Sourdonnier et moi-même nous sommes rencontrés, nous avons la vision commune que l'avenir des applicatifs passerait par les interfaces Web. C'est de cette première intuition qu'est né Cognix Systems. A l'époque nous disposions déjà d'une expertise forte en PHP et de bonnes connaissances en génie logiciel et savions qu'il était possible de réaliser des applicatifs importants sur ce langage en faisant preuve de rigueur. De plus, la perspective de PHP5, en particulier au niveau du développement objet, était rassurante sur l'évolution à moyen terme du langage et j'avoue avec le recul, ne pas regretter le choix de PHP. Aujourd'hui, la maturité du langage PHP nous permet de travailler sur de vrais applicatifs critiques, tout en étant compétitif sur le marché grâce au gain de temps permis par PHP.

L'orientation Stratégique de Cognix Systems est de continuer nos investissements sur ces langages de manière à conserver un pôle d'expertise fort sur PHP et AJAX. Nous misons beaucoup sur la formation des équipes et sur le développement de nos propres outils. Parallèlement, nous avons entamé une démarche de certification et déjà certains de nos développeurs ont acquis la „PHP5 Zend Certified Engineer“ afin de garantir à nos clients un bon niveau d'expertise.

### PS : Qu'en est-il d'AJAX, apporte-t-il des gains d'usage reconnus par vos clients ou est-ce plutôt un effet de mode ?

**DC :** L'arrivée d'AJAX a été pour nous une vraie révélation, il faut dire que l'on a toujours considéré l'ergonomie comme l'élément essentiel qui conditionne la réussite d'un projet. En effet, notre philosophie a toujours été de placer l'utilisateur au centre de nos développements, pour cela nous devons être capables de lui simplifier la compréhension de nos interfaces et de lui permettre d'effectuer rapidement ses actions. Sur ces deux points AJAX a apporté de vraies solutions, avec des interfaces Web aux comportements proches

des logiciels lourds, nous permettons à l'utilisateur de retrouver ses marques. De plus, l'AJAX permet réellement des gains de productivité pour l'utilisateur grâce à la réactivité des interfaces, nous avons estimé que par rapport aux interfaces HTML classiques, des interfaces AJAX permettent de réduire de moitié le temps nécessaire à l'utilisateur pour réaliser ses tâches.

### PS : Quelles sont les difficultés majeures rencontrées dans le développement de sites massivement AJAX ?

**DC :** Si au tout début de l'AJAX, les principaux problèmes provenaient des difficultés de débogage et manque d'uniformité du comportement des navigateurs dans l'interprétation du Javascript, l'éco système logiciel autour d'AJAX est maintenant mûre. Chez Cognix Systems, nous utilisons la bibliothèque JQuery qui apporte un vrai niveau d'abstraction par rapport aux navigateurs et surtout permet de vraiment simplifier le code. Pour le débogage de nos applications, nous utilisons le plugin Firebug de FireFox qui permet de remonter les bugs de façon pertinente.

### PS : Utilisez-vous systématiquement l'AJAX dans vos développements ?

**DC :** Non, tout d'abord parce que l'AJAX n'est pas approprié sur tout type de projet, pour un site Internet institutionnel par exemple, on préfère rester sur une ergonomie Web classique en XHTML/CSS pour des raisons de référencement et pour coller aux habitudes des Internauts. Idem pour les portails Web ou nous limitons l'AJAX aux seules fonctions apportant un vrai gain à l'utilisateur. En revanche, nous utilisons de plus en plus l'AJAX dans nos interfaces BackOffice et pour le développement de solutions logicielles (ERP, CRM, etc.). Le principal frein à l'AJAX est généralement le coût des interfaces qui reste supérieur à celui des interfaces HTML plus classique. Notre vision est d'industrialiser la mise en place de ces technologies pour diffuser largement les gains apportés par ces nouvelles IHM. En ce début 2009, nous lançons un nouveaux framework de travail qui

intègre très fortement la technologie AJAX au travers de quelques briques à très fortes valeurs ajoutées dont nous pouvons ainsi mutualiser les coûts.

**PS : Pourriez vous nous en dire plus sur votre nouveau framework ?**

**DC :** Bien sûr ! Notre nouveau framework était une attente forte de la part de nos développeurs et nous y avons consacré beaucoup d'énergie. Pour une société comme la nôtre, utiliser un framework représente le véritable ciment de nos sites, portails et applicatifs. Il s'agit à la fois d'un outil qui améliore la productivité, définit un niveau de qualité et qui dessine des processus autour desquels se rassemble l'ensemble de notre équipe, du chef de projet au développeur en passant par le support. Un framework constitue l'âme et l'avantage concurrentiel d'une société comme la nôtre.

Le travail sur le nouveau framework va nous permettre de franchir une nouvelle étape de notre développement, à la fois en y intégrant les dernières technologies mais aussi en capitalisant sur l'expérience acquise depuis plusieurs années. Il sera basé sur le Zend Framework qui a des fondamentaux solides et dont le niveau de maturité est bon. Nous faisons ce choix également car nous avons la conviction que ce framework pourra devenir un futur „standard“. Il est également basé sur le moteur de template Smarty et sur JQuery pour l'AJAX.

**PS : Quels sont les outils que vous utilisez pour assurer à vos clients la qualité du code généré ?**

**DC :** Au delà des seuls outils, la qualité de nos prestations est liée à la qualité de l'équipe et à son organisation. Nous avons une véritable démarche projet, qui passe par une étude précise des besoins de nos clients,

la rédaction d'un dossier de spécifications techniques et fonctionnelles et le maquetage de nos interfaces. Cette démarche projet permet de garantir à nos clients que nos livraisons seront conformes à leurs attentes.

Au niveau du développement, nous avons fortement standardisé la création de nos logiciels, nous disposons d'outil de génération automatique de code, et nous essayons de fortement réutiliser nos composants. Pour garantir la qualité du code, chaque développeur se doit de suivre une fiche de procédure et en fin de chaque projet, un contrôleur qualité valide le respect des règles de qualité. Pour finir, nous disposons d'une équipe de test qui valide point par point nos applications. L'ensemble de cette méthodologie nous permet des livraisons de qualité en limitant les allées-retours.

Une fois les projets en production, notre équipe support veille également à apporter une réponse rapide à nos clients en cas de problème et nous avons mis en place une plateforme de monitoring chargée de surveiller les problèmes classiques pouvant intervenir sur nos applications (fatal error, erreur mysql, etc.). Quand un problème est avéré, il est remonté afin d'être corrigé à sa source.

**PS : Quels sont aujourd'hui les produits et services que propose COGNIX SYSTEMS ?**

**DC :** Cognix Systems est une société qui a fortement évolué ces dernières années et dont les produits et services s'étoffent avec le temps. Nous avons débuté comme Web Agency et avons développé une vraie expertise dans la création de site Internet évolué, puis nous avons naturellement évolué dans le domaine du développement spécifique (Intranet, Extranet, logiciel Web). Parallèlement, nous avons développé plusieurs logiciels dont une solution de gestion de

contenu WebGazelle, une solution de transfert de fichier WebGazelle Data Transfert, une solution de stockage et diffusion de video et autre média WebGazelle Media Center, une solution de gestion de CV en ligne et une solution de diffusion d'information ciblée Webazelle News Center. L'ensemble de ces produits est décrit sur notre site Internet.

**PS : Concernant votre CMS WebGazelle, pouvez-vous raconter à nos lecteurs ce qui le différencie de Joomla ou de SPIP qui contiennent une documentation en ligne très complète ?**

**DC :** La différence fondamentale entre WebGazelle et des solutions comme SPIP et Joomla est sans nul doute l'ergonomie, WebGazelle se veut à la porté de tous, de l'assistance de direction au webmaster en passant par le responsable de publication.

L'autre raison pour laquelle nous avons investi sur notre propre outil est qu'il est important de maîtriser parfaitement notre outil, tant dans ses fonctionnalités que dans ses évolutions. Cela permet de mieux répondre à des besoins spécifiques et de mieux gérer notre parc de plus de 300 sites.

**PS : Pouvez-vous dire à nos lecteurs quels sont vos projets pour l'avenir ?**

**DC :** Pour 2009, nous allons continuer nos efforts outils de notre nouveaux framework et avons en développement plusieurs projets d'outils pour l'entreprise que nous lancerons courant 2009. L'ensemble de ses outils tirera partie de nos nouvelles interfaces AJAX.

**PS : Merci beaucoup pour le temps que vous avez bien voulu nous accorder.**

**DC :** Merci à vous de nous avoir permis de nous présenter à vos lecteurs et félicitations pour votre excellent magazine.



Cognix Systems, SSII et Web Agency depuis 2002

Expertise autour des technologies Web sur PHP

Développeurs PHP, venez nous rejoindre !

Intégrez 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



# Développer

## plus efficacement

En tant que développeur PHP, vous avez sûrement du avoir affaire à des scripts regorgeant de fonctions, de lignes de codes entremêlées, et surtout bon nombre de bugs tous plus mystérieux les uns que les autres.

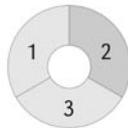
### Ce que cet article explique :

- Présentation de la bibliothèque XDebug.
- Comment déboguer avec XDebug sous Eclipse.
- Sensibilisation au débogging et profiling.

### Ce qu'il faut savoir :

- Connaître la notion de bibliothèques pour PHP.
- Connaître les bases de PHP.
- Avoir été confronté à des bugs au sein d'un projet.

### Niveau de difficulté



**D**u coup, vous passez votre temps à déboguer grâce aux fidèles *echo* et autres *print\_r*. Si vous êtes dans cette situation et que vous n'utilisez pas encore d'outils de *debugging*, cet article vous ouvre la voix vers un mode de développement un peu plus efficace.

### Présentation d'XDebug

XDebug est une bibliothèque pour PHP, distribuée sous la forme d'extension, qui est capable de fournir bon nombre d'informations précieuses parmi lesquelles :

- Les différentes traces (des appels sur la pile, au sein d'une fonction...),
- Les allocations mémoire,
- Protection contre les récursivités infinies,
- *Profiling* de code (voir plus loin).

Notez également qu'XDebug s'intègre parfaitement avec symfony (via la *toolbar de debug*).

### L'utilité d'une telle bibliothèque

Deux jours. Deux jours que vous venez de finir le développement d'un nouveau module de gestion de panier en ligne, et vous n'arrivez toujours pas à comprendre l'origine de ce message d'erreur si insignifiant. Il vous indique une ligne, certes, mais celle-ci vous semble totalement anodine. C'est là qu'XDebug intervient.

L'utilisation d'une bibliothèque tierce comme celle-ci permet – entre autre - de rendre les messages d'erreur beaucoup plus explicites, et détaillés. L'affichage complet des appels de fonctions effectués vous permettra de découvrir que le problème vient peut être du cœur même de votre application, un problème non soupçonné jusque là ! Alors que bon nombre de développeurs tentent de résoudre leur problème en insérant eux-même des instructions telles que *debug\_backtrace*, *echo* ou *print\_r* dans leur code, une utilisation optimale d'une bibliothèque comme XDebug, couplée à un débogueur intégré à votre IDE, vous fera gagner de très, très précieuses minutes, heures ou jours.

Cet article s'adresse aux développeurs de tous niveaux, non seulement parce que cela fait partie des habitudes à prendre lorsqu'on débute, mais de surcroît car opter pour l'utilisation d'XDebug peut rendre bien des services dans un projet complexe. Je m'en sers très souvent dans mes projets symfony, et je m'imaginerais très mal développer sans dès lors que j'ai eu l'occasion d'essayer.

### XDebug... et les autres ?

Faisons si vous le voulez bien un tour d'horizon des solutions existantes... Il n'y en a pas pléthore ! Nous pouvons citer, parmi les plus connues, *DBG*, *Pear::Benchmark*, *Pecl::APD*... Or il se trouve que la plupart de ces solutions ne proposent pas autant de fonctionnalités qu'XDebug. Bien sûr chacune a son petit avantage à proposer, mais notre choix nous permet de profiter de toutes les fonctionnalités citées au dessus via une seule et unique extension, plutôt efficace et fort bien documentée.

### Installation d'XDebug

Commencez par vous rendre sur le site officiel du projet, <http://www.XDebug.org>. Vous trouverez directement sur la page d'accueil différents liens. Choisissez celui qui convient à votre système d'exploitation et à votre version de PHP. Notez qu'une installation par PEAR est également possible. Il vous suffit de taper, dans un terminal :

```
pecl install XDebug
```

L'installation par PEAR sous Unix nécessite la présence du paquet *php5-dev* (ou *php-dev*, ou encore *php5-devel*...). Une fois XDebug téléchargé, il vous reste à configurer correctement PHP pour prendre en compte cette nouvelle extension. Ouvrez votre fichier de configuration *php.ini*, et ajoutez-y les lignes suivantes :

```
; Remplacez par le chemin vers le
    fichier .dll
zend_extension_ts="C:\wamp\bin\php\
    php5.2.6\ext\php_xdebug.dll"
```

```
XDebug.remote_enable=1
XDebug.remote_handler=dbgp
XDebug.remote_mode=req
XDebug.idekey=default
XDebug.idekey=ECLIPSE_DBGP
XDebug.profiler_enable=1
XDebug.remote_port=9000
XDebug.show_local_vars=1
```

Sous Linux, remplacez la première ligne par

```
zend_extension="/usr/lib/php5/
    apache2/XDebug.so"
```

### Tester l'installation de la bibliothèque

Afin de contrôler la bonne installation de la bibliothèque, je vous invite à écrire un script très basique, exploitant une fonctionnalité d'XDebug :

```
<?php
    function fix_string($a) {
```

```

echo "Called @ " . XDebug_call_file()
. " : " . XDebug_call_line() .
" from " . XDebug_call_
function();
}
$ret = fix_string(array('Tekover'));
?>

```

Si l'installation s'est correctement déroulée, vous devriez obtenir un message du genre *Called @ /home/httpd/html/test/XDebug\_caller.php:12 from {main}*.

Dans la suite de cet article, nous ne ferons plus appel aux fonctions fournies par la bibliothèque par nous-même, mais rien ne vous empêche de le faire au sein de vos scripts. La documentation en ligne est très bien détaillée.

## Intégration d'XDebug avec Eclipse PDT

Désolé pour tous les utilisateurs de *Netbeans* ou d'un autre IDE, je ne vais traiter ici que de l'exemple d'Eclipse PDT. La manipulation est néanmoins réalisable - et plus ou moins similaire - au sein des autres IDE (autres distributions d'Eclipse incluses). Vous pouvez retrouver la liste des éditeurs supportés sur le site officiel d'XDebug, à cette page : <http://www.XDebug.org/docs/remote>.

Ouvrez les préférences d'Eclipse, et rendez-vous dans la section *PHP... Debug*. Sélectionnez le débogueur XDebug, comme sur la Figure 1. Le port 9000 est normalement configuré par défaut, mais vous pouvez toujours vérifier par vous-même.

## Utilisation du débogueur

Maintenant que tout est - normalement - en place, nous pouvons passer à un cas pratique d'utilisation d'XDebug. Pour prendre en main ce nouvel outil, créez-vous un script basique, du type :

```

<?php
function add_two($nb) {
    return $nb + 2;
}
function foo($nb) {
    return add_two($nb);
}
$a = 42;
$b = foo($a);
?>

```

Dans Eclipse, faites un clic droit sur le script, et choisissez *Debug as... PHP Web Page* (Figure 2). Eclipse peut vous demander, la première fois, d'entrer l'URL de la page ou du serveur (Figure 3). Montrez-vous sympathique et obéissez aux ordres de votre IDE. Votre script doit alors s'ouvrir au sein de votre navigateur, et doit sembler attendre quelque chose. Si Eclipse ne l'a pas fait de lui-même, passez dans la vue *PHP Debug* (menu *Window... Open Perspective... Debug*). Vous devriez avoir une

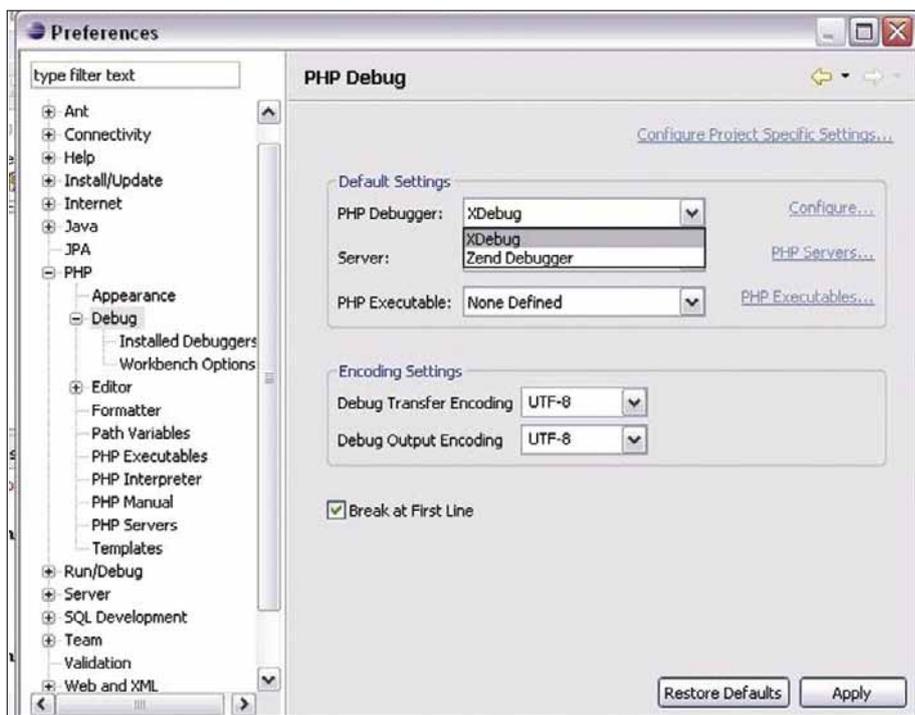


Figure 1. Panneau de préférences

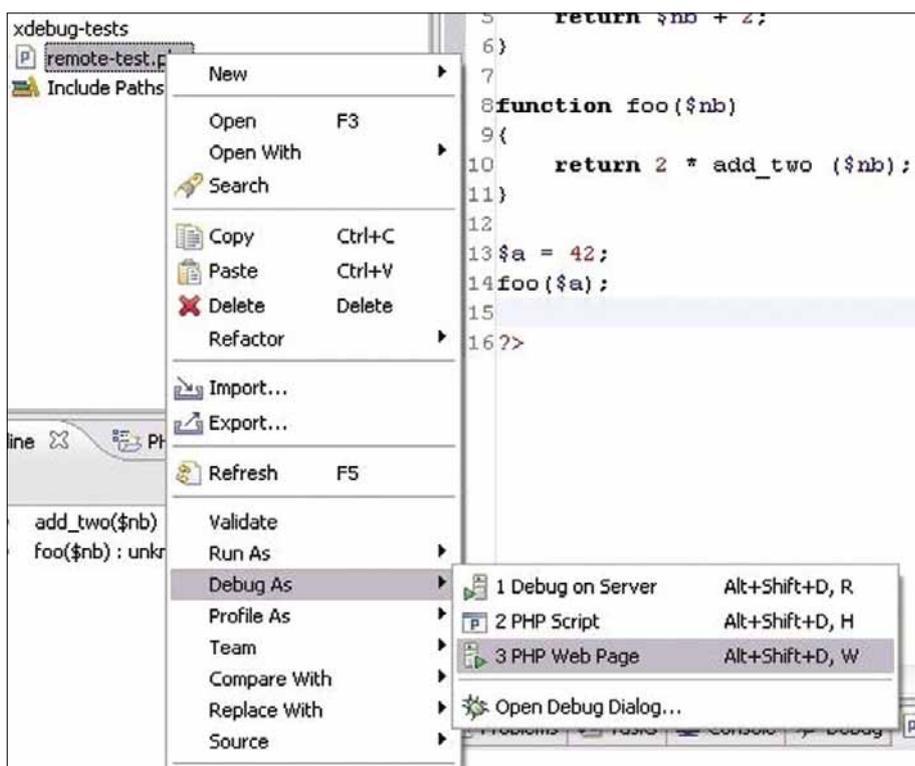


Figure 2. Lancement d'une session de débogage



Figure 3. Spécification de l'URL à utiliser

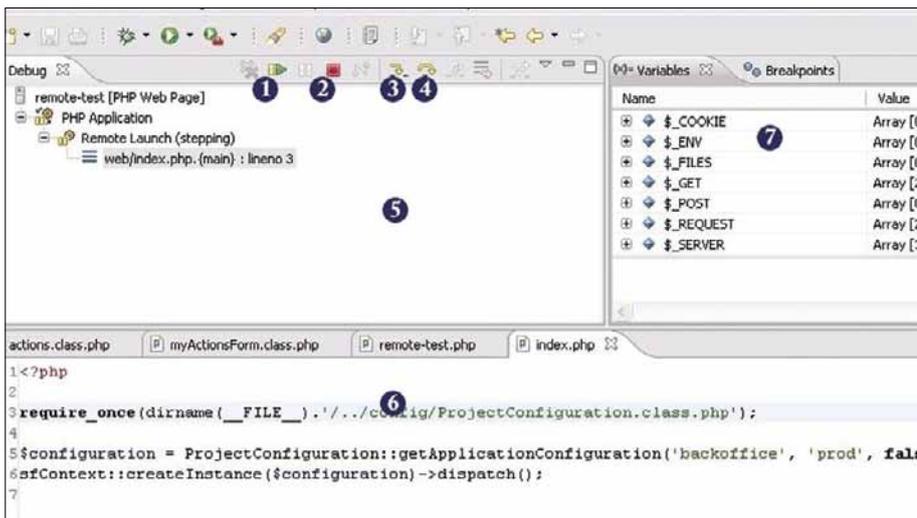


Figure 4. Interface du débogueur Eclipse

fenêtre comme sur la Figure 4. Voici un descriptif du comportement de chacun des boutons :

- 1: Lancer le script jusqu'au prochain point d'arrêt,
- 2: Arrêter le script,
- 3: Avancer à la ligne suivante,
- 4: Passer la boucle/fonction suivante,
- 5: Trace actuelle (pile d'appel),
- 6: Votre code,
- 7: Variables.

Il y a deux autres fonctionnalités et concepts de base que vous devriez connaître pour une utilisation efficace du débogueur : premièrement, la notion de *breakpoint* (point d'arrêt). Vous pouvez en effet indiquer à votre débogueur d'exécuter normalement votre script, mais de s'arrêter à une ligne précise. Habituellement, il suffit de cliquer (ou double cliquer) à gauche du numéro de ligne pour poser un point d'arrêt sur cette ligne. Dans notre exemple, essayez de lui dire de s'arrêter au début de la fonction `foo`.

La deuxième chose à connaître est la notion de *watcher*. Ces *watchers* vous permettent de surveiller le contenu d'une variable. Dans notre exemple précédent, sélectionnez la variable `$a`, faites clic-droit et sélectionnez *watcher*. À chaque évaluation de la variable `$a`, vous pouvez suivre sa valeur dans le cadre dédié.

Maintenant que vous connaissez le comportement et le fonctionnement général du débogueur, entraînez-vous sur notre exemple. En cliquant sur le bouton 3, essayez d'observer

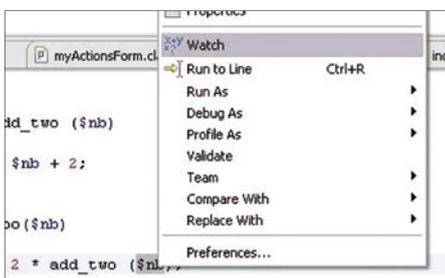


Figure 5. Positionnement d'un Watcher

le comportement. Mettez des *watchers* où bon vous semble. L'exemple ici est très simple, mais cette méthode est parfaitement exploitable dans le cadre d'un développement au sein d'une application complexe (avec un framework par exemple). Lors d'une telle situation, notez que vous avez juste à ouvrir une page via Eclipse (clic droit sur `index.php` par exemple, puis *Debug As... PHP Web Page*). Vous pouvez ensuite naviguer au sein de votre application (tenter de vous identifier exemple) et suivre le parcours de l'interpréteur PHP via le débogueur. Par défaut, ce dernier s'arrête au début de l'exécution de chaque page. Si cela vous ennuie, et c'est fort probable, vous pouvez désactiver ce comportement en décochant la case *Break at first line* dans le panel de *configuration* (voir Figure 1).

### Profiling de code

Une des fonctionnalités intéressante d'XDebug est sa capacité de fournir des fichiers de *logs* avec l'intégralité des actions effectuées à l'affichage d'une page, avec le temps requis pour chacune de ces actions. Pour activer cette fonctionnalité, rajoutez les lignes suivantes dans votre fichier de configuration `php.ini` (ici, les fichiers seront automatiquement placés dans le répertoire `c:\wamp\log`).

```
XDebug.profiler_output_dir="C:\wamp\log\"
XDebug.profiler_output_name=
    "%t.cachegrind.out"
XDebug.profiler_enable_trigger=1
```

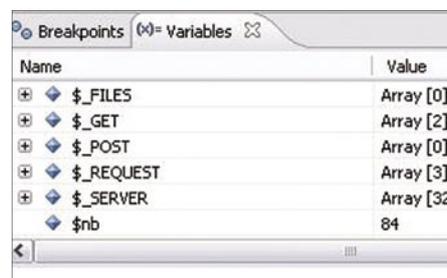


Figure 6. État en direct des variables

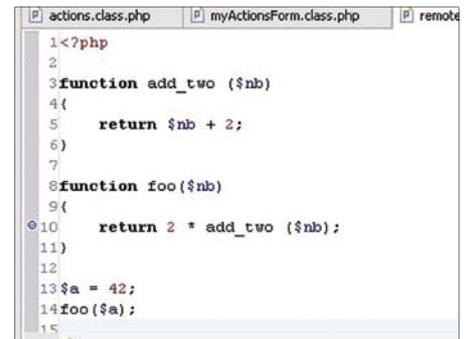


Figure 7. Breakpoint posé à la ligne 10

N'oubliez pas de redémarrer le serveur pour que les modifications soient prises en compte. Lancez l'affichage d'une page, et regardez le dossier `c:\wamp\log`. Un fichier y a été créé. Vous pouvez ouvrir ces fichiers avec *WinCacheGrind* (Windows) ou *KCacheGrind* (sous l'environnement KDE), par exemple.

Conseil : pensez à bien désactiver cette option si votre serveur doit servir dans un environnement de production ! L'écriture de ces fichiers prend un temps (et un espace disque) non négligeable.

### Pourquoi du profiling de code ?

La lecture de ces fichiers vous donnera de très précieuses informations sur les différents appels effectués, et vous laissera sûrement penser qu'une partie de votre code mériterait d'être revue et optimisée, sous peine de coûter énormément de temps en cas de forte charge. Vous avez maintenant tout le loisir de décortiquer les temps d'exécution de chaque appel de fonction (les fonctions natives de PHP sont préfixées par `php:`), les différentes inclusions de fichiers pas forcément toujours judicieuses, ou encore la consommation mémoire de votre script.

Une telle approche peut se révéler utile tant pour contrôler le bon fonctionnement de son application, que pour découvrir quelle fonction est particulièrement lente, au point de faire perdre 2 secondes à chaque génération.

### Conclusion

J'ai pris le temps d'exposer ces différentes méthodes de travail car trop souvent on assiste à des scènes de debuggages laborieuses pour lesquelles l'utilisation d'outils comme ceux-ci s'avèrerait déterminante et primordiale. Il ne me reste plus qu'à vous souhaiter bon courage dans l'utilisation de votre débogueur. J'espère que vous adopterez ces pratiques et qu'elles vous feront gagner des heures bien précieuses. N'hésitez pas à me contacter personnellement pour toute remarque, question ou commentaires !

### ADRIEN MOGENET

Actuellement étudiant à l'EPITA, Adrien Mogenet est passionné des technologies Open Source et par toutes les problématiques d'optimisation du développement d'un projet.

Contact : [adrien@frenchcomp.net](mailto:adrien@frenchcomp.net)

# Arrondissez vos fins de mois avec Progonline

## Qu'est-ce que Progonline ?

Le site [www.progonline.com](http://www.progonline.com) est la première plateforme Internet qui facilite la réalisation de projets informatiques à distance. Plus de 15 000 développeurs et 5 000 clients travaillent ensemble, provenant de plus de 20 pays francophones (France, Suisse, Belgique, Québec, Luxembourg, etc.). Le site constitue une source incessante de clients qualifiés qui cherchent à confier leurs projets informatiques à des spécialistes informatiques. La collaboration se déroule intégralement à distance, en télétravail.

## A qui s'adresse Progonline ?

La vocation de *Progonline* est de permettre à tout développeur qui souhaite réaliser des projets informatiques de trouver rapidement et facilement des clients francophones dans le monde entier, et de travailler avec eux dans un cadre sécurisé, directement sur Internet. Plus précisément, *Progonline* s'adresse aux catégories suivantes d'utilisateurs :

- les développeurs particuliers qui souhaitent arrondir leurs fins de mois en réalisant des projets de leurs choix (développement web, progiciels, applications métier, référencement, etc.),
- des sociétés de développement informatique, référencement, web, etc. qui cherchent des projets et des clients dans leur secteur d'activité.

## Quel est le fonctionnement de Progonline ?

Le fonctionnement de *Progonline* est très simple :

- vous vous inscrivez gratuitement,

- vous recevez les projets disponibles par mail, quotidiennement,
- vous réalisez les projets de votre choix directement par Internet, sans aucun déplacement,
- vous êtes payé par virement bancaire dès que le projet est finalisé.

Progonline se rémunère en prélevant une commission de 15% sur chaque projet réalisé avec succès par un développeur membre du réseau.

Des abonnements spéciaux optionnels sont également proposés aux utilisateurs intéressés, pour disposer d'informations supplémentaires concernant les clients et leurs projets, par exemple.

## Quelle est l'utilité de Progonline ?

L'utilité de *Progonline* peut être énoncée en 2 points-clés :

- vous aider à *trouver des clients* dans le monde entier. Progonline vous évite de passer un temps important en démarches commerciales sans succès, pour vous permettre de vous consacrer à ce que vous savez faire de mieux : le développement informatique.
- sécuriser votre travail, et *vous garantir un paiement sans retard*. Dans le cadre du travail à distance, entre clients et développeurs qui ne se connaissent pas, cette garantie est primordiale. *Progonline* fonctionne comme un tiers de confiance, qui vous garantit votre rémunération.

## Est-ce légal de travailler sur Progonline ?

Pour travailler sur *Progonline*, tout prestataire doit avoir une existence juridique, et avoir le droit de facturer. Plus précisément, si vous êtes un particulier, plusieurs possibilités s'offrent à vous :

- vous inscrire auprès d'une *société de portage salarial*. *Progonline* a établi un partenariat avec Webportage ([www.webportage.com](http://www.webportage.com)), qui est une référence dans le domaine de télétravail et des webmasters. Cette solution est la plus commode et la plus rapide.
- *créer votre micro-entreprise*. *Progonline* a établi des partenariats avec des plateformes qui vous permettent de créer votre entreprise directement en ligne.

Attention à voir avec votre employeur et avec votre contrat de travail (pour ceux qui exercent une activité professionnelle principale salariée) que vous ne tombiez pas sous l'incidence de clauses qui vous interdisent d'avoir des activités professionnelles annexes. La société *Progonline* vient de recevoir le label *JEI (Jeune Entreprise Innovante)* décerné par la *Chambre de Commerce de Paris*, ce qui confirme l'originalité et l'utilité de son concept.

## PROGONLINE

SAS PEL.COM au capital de 101 101euros – 481 012  
839 RCSNANTERRE. 20 bis rue Louis Philippe, 92200  
Neuilly-sur-Seine.  
[contact@progonline.com](mailto:contact@progonline.com)



## Vous avez un projet informatique ?

Travaillez directement sur Internet avec des prestataires du monde entier à des tarifs défiant toute concurrence

**infographie programmation webmastering ...**

**les meilleurs prestataires sont chez  
[www.progonline.com](http://www.progonline.com)**

# XOAD

## PHP pour un environnement Ajax côté serveur

Vous allez découvrir comment XOAD permet d'exporter des classes PHP vers du code Javascript implémenté côté client. Cette approche est résolument orientée objet et de plus totalement transparente pour le développeur d'applications web.

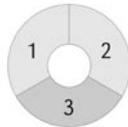
### Cet article explique :

- Comment utiliser l'environnement XOAD pour développer des applications Ajax en PHP avec une approche orientée objet côté serveur.
- Comment implémenter la fonctionnalité du glisser/déposer en Ajax.

### Ce qu'il faut savoir :

- Avoir de bonnes bases en Javascript, PHP et SQL.
- Connaître les concepts de la programmation orientée objet.
- Être familier avec la programmation web.

### Niveau de difficulté



Ajax (*Asynchronous Javascript and XML*) peut être vu à la fois comme un regroupement de technologies et comme une architecture. En effet, le terme Ajax sous-entend la mise en œuvre d'un canal de communication asynchrone entre un navigateur et un serveur web, faisant intervenir du Javascript et du XML (*Extensible Markup Language*). La naissance d'Ajax est liée au mode d'évolution des applications web. À l'origine, l'architecture d'une application web tournait autour de la définition d'une page qui correspondait à un événement de l'application (validation d'une commande, connexion sur un espace privé, etc...). Puis, chaque événement avait pour conséquence le renvoi d'une page complète vers un navigateur. Ce type d'architecture réduisait les applications web au partage de données et n'exploitait pas réellement l'interactivité.

Ajax constitue un ensemble de technologies permettant une connexion asynchrone entre le client et le serveur d'une application web. Les composants côté client peuvent alors effectuer des requêtes sur le serveur afin de ne mettre à jour que certaines parties de la page sans recharger totalement celle-ci. De plus, Ajax fournit un niveau d'interaction optimal dans le sens où l'application web est toujours disponible pendant l'envoi des requêtes et le

traitement côté serveur. Pour aller un peu plus loin, Ajax utilise l'objet Javascript XMLHttpRequest (*XHR*) pour établir les connexions asynchrones client/serveur. Un développeur web devra donc instancier, parfois plusieurs fois, cet objet si il veut bénéficier de la technologie Ajax dans son application. Cependant, il existe des environnements basés sur des bibliothèques agissant à différents niveaux de l'application qui permettent de simplifier l'utilisation de l'objet XHR.

Les bibliothèques d'accès distant (*remoting toolkit*) de bas niveau font partie. Elles ont pour but de masquer l'instanciation de l'objet XHR mais également de fournir une solution équivalente si Javascript n'est pas supporté par le navigateur (par l'utilisation d'*iframe* par exemple). Parmi les environnements offrant l'accès distant, certains fonctionnent sans être contraints par la technologie employée côté serveur (comme par exemple Dojo, JSON-RPC, Prototype, OpenRico ou Script.aculo.us) alors que d'autres permettent d'établir des passerelles entre client et serveur. En effet, l'environnement DWR (*Direct Web Remoting*) est une solution basée sur l'association du client Javascript avec des écouteurs sur le serveur spécialement implémentés pour la plate-forme Java. De la même manière, JSON-RPC permet d'associer des services avec plusieurs langages côté serveur. Il existe également des environnements Javascript orientés objet qui permettent au client d'accéder directement aux objets qui se trouvent sur le serveur. DWR,

JSON-RPC, SAJAX ou XOAD font partie de ces environnements que l'on qualifie d'ORB (*Object Request Broker*). Le développement d'applications web basées sur l'utilisation de ce type de technologie est particulièrement adapté aux architectures trois tiers. En effet, une équipe de développement peut concentrer ses efforts sur le tiers logique (*serveur*) tout en générant simultanément des résultats pour les deux tiers physiques (le serveur et le client).

Dans cet article, nous avons choisi de vous présenter la mise en pratique de l'environnement XOAD (*XMLHTTP Object-oriented Application Development*). Cette bibliothèque (anciennement appelée NAJAX) a été créée en novembre 2005 et permet grâce à un accès distant de type ORB de connecter du Javascript côté client avec des fonctionnalités PHP côté serveur. Pourquoi avoir choisi XOAD ? Nous pensons que cet environnement a fait ses preuves et est encore ce que l'on peut faire de mieux dans le développement d'application web faisant intervenir une intégration d'AJAX avec un environnement serveur basé sur PHP.

La suite de cet article se décompose comme suit. Dans la section *L'environnement XOAD*, nous présentons le fonctionnement des accès distants de type ORB avec l'utilisation de Javascript et PHP au travers de XOAD. Puis dans la section suivante intitulée *Mise en pratique : le drag and drop*, nous mettons en application l'environnement XOAD en présentant l'implémentation du *drag and drop* dans une application web simple.

### L'environnement XOAD

Dans cette section nous présentons en détails l'environnement XOAD. Vous verrez quels sont les modifications et les ajouts nécessaires au niveau du code pour permettre l'exportation des objets du côté serveur vers le côté client.

## Le mariage d'Ajax et de PHP

PHP est un langage utilisé par de nombreux développeurs du fait qu'il soit gratuit et surtout Open Source. Ce dernier critère lui confère une communauté d'utilisateurs et de développeurs très active mettant à disposition un nombre de plus en plus croissant de bibliothèques d'extensions. Le succès d'Ajax n'a pas échappé à la règle puisque XOAD est le résultat du mariage de cette technologie avec PHP. XOAD est un environnement basé sur un ORB et par conséquent permet la réutilisation sur le client du code côté serveur présenté comme un même modèle objet sur les deux tiers. En fait, dans XOAD la couche Javascript est liée au code serveur, en l'occurrence PHP. De plus, cet environnement est orienté objet et il propose l'exportation des méthodes mais également des objets instanciés en PHP directement dans le code Javascript avec une transparence totale.

## Accès distant de type ORB

Un ORB (*Object Request Broker*) est un ensemble de fonctions qui implémentent un *bus logiciel* par lequel des objets envoient et reçoivent des requêtes et des réponses, de manière transparente et portable. La mise en place d'un ORB confère au développeur d'applications web la possibilité d'écrire du code côté client qui accède directement au modèle de domaine côté serveur. Comme il est montré sur la Figure 4, le développeur a la possibilité d'utiliser des objets côté client qui invoquent à distance des méthodes appartenant à d'autres objets distribués sur un serveur.

## JSON

JSON (*JavaScript Object Notation*) est un format ouvert décrivant la représentation d'objets Javascript sous la forme d'un texte facile à créer et à analyser. Autrement dit, JSON est une alternative à XML pour envoyer des données au navigateur codées sous la forme d'objets JSON qui peuvent être facilement convertis en objet Javascript. Avec JSON, du fait que les objets soient typés, on évite de donner une couche supplémentaire au XML pour spécifier le type des données manipulées. De plus, les données traitées n'ont pas besoin d'être analysées et cela évite également d'écrire du code pour manipuler le DOM (*Document Object Model*) du XML.

La stratégie d'intégration d'Ajax avec le langage PHP proposée par XOAD procure les avantages suivants :

- l'utilisation de JSON : au lieu d'utiliser XML, XOAD exploite JSON afin de faciliter la manipulation des données dans l'application,
- la vision côté serveur : XOAD donne la possibilité aux développeurs d'avoir constamment un œil sur les événements côté

serveur afin d'optimiser les performances globales de l'application,

- la facilité d'implémentation : XOAD permet de développer des applications Ajax sans être confronté aux problèmes liés à l'interaction entre Javascript, XML et HTML,
- la mise en cache : XOAD évite les dysfonctionnements d'*Internet Explorer* (IE) parfois provoqués par la mise en cache d'une trop grosse quantité d'informations.

Voyons à présent de quelle manière s'effectue la mise en place de XOAD.

## Du procédural à l'orienté objet

XOAD est orienté objet et il permet d'exporter des objets et des classes. Pour ce faire, il faut ajouter des métadonnées aux classes côté serveur afin de spécifier les méthodes à exporter. Le Listing 1 présente une approche consistant à définir les méthodes de la classe à exporter en utilisant la fonction `XOAD_CLIENT::mapMethods()`. Ici, la méthode `my_xoad_method` de la classe `MyXoadClass` est exportée. Enfin, il faut définir la liste des méthodes en accès public par la fonction `XOAD_CLIENT::publicMethods()`.

Du côté client, il est nécessaire d'ajouter le code permettant de définir les classes auxquelles

on veut pouvoir accéder à distance. En considérant que les sources de l'environnement XOAD sont stockées dans un sous-répertoire `xoad`.

Le Listing 2 présente le code à rajouter côté client afin d'enregistrer les classes accessibles à distance (ici `MyXoadClass` implémentée dans le fichier `my_xoad_class.php`). Cette procédure s'effectue grâce à la méthode `XOAD_Server::allowClasses()` implémentée dans le fichier `./xoad/xoad.php`. Le test conditionnel dans `XOAD_Server::runServer()` permet de gérer la requête XHR. Il renvoie faux si la requête n'est pas un rappel au XHR. Sinon cette procédure invoque la méthode serveur selon les paramètres de la requête et renvoie les résultats. Enfin, il faut ajouter le code Javascript d'assistance XOAD grâce à la méthode `XOAD_Utilities::header()`.

Pour enregistrer une classe et y accéder via Javascript il faut utiliser une autre fonction d'assistance `XOAD_Client::register()`. Le Listing 3 présente le code à insérer dans le fichier principal dès le début de la balise `<head>` pour enregistrer la classe `MyXoadClass`.

Vous savez à présent comment utiliser XOAD afin d'exporter des classes et des méthodes directement exploitables dans le code côté client. Nous allons à présent aborder une mise en pratique de XOAD avec l'implémentation du *drag and drop*.

**Listing 1.** Code serveur pour exporter la méthode `my_xoad_method` de la classe `MyXoadClass`

```
Class MyXoadClass {
...

    public function xoadGetMeta () {
        XOAD_CLIENT::mapMethods($this, array('my_xoad_method'));
        XOAD_CLIENT::publicMethods($this, array('my_xoad_method'));
    }
...
}
```

**Listing 2.** Code client pour l'initialisation de l'environnement XOAD

```
<?php
    require_once("../xoad/xoad.php");
    require_once("my_xoad_class.php");
    XOAD_Server::allowClasses('MyXoadClass');
    if (XOAD_Server::runServer()) {
        exit;
    }
    XOAD_Utilities::header('../xoad');
?>
```

**Listing 3.** Code côté client d'enregistrement de la classe `MyXoadClass`

```
<head>
    <script type="text/javascript">
        var obj = <? = XOAD_Client::register(new MyXoadClass()) ?>;
    </script>
...
</head>
```

**Listing 4.** Code SQL pour générer la table `tasks`

```
CREATE TABLE tasks (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  order INT UNSIGNED NOT NULL default '0',
  description VARCHAR(100) NOT NULL default '',
  PRIMARY KEY(id)
);
```

**Listing 5.** Code XHTML du fichier principal `index.php`

```
<?php
  require_once("taskslist.class.php");
  require_once("../xoad/xoad.php");
  XOAD_Server::allowClasses('TasksList');
  if(XOAD_Server::runServer()){
    exit;
  }
  XOAD_Uilities::header('../xoad');
?>

<!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">

<head>
  <script type="text/javascript">
    var obj = <? XOAD_Client::register(new TasksList()) ?>;
  </script>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>AJAX Drag and Drop Sortable List</title>
  <link href="drag-and-drop.css" rel="stylesheet" type="text/css" />
  <script type="text/javascript" language="JavaScript" src="../xoad/js/xoad_
    optimized.js"></script>
  <script src="../aculo/prototype.js" type="text/javascript"></script>
  <script src="../aculo/scriptaculous.js" type="text/javascript"></script>
  <script src="drag-and-drop.js" type="text/javascript"></script>
</head>

<body onload="startup()">
  <h1>Task Management</h1>
  <h2>Add a new task</h2>

  <div>
    <input type="text" id="txtNewTask" name="txtNewTask"
      size="30" maxlength="100" onkeydown="handleKey(event)"/>
    <input type="button" name="submit" value="Add this task"
      onclick="process('txtNewTask', 'addNewTask')"/>
  </div>
  <br />
  <h2>All tasks</h2>
  <ul id="tasksList" class="sortableList" onmouseup="process('tasksList',
    'updateList')">
  </ul>
  <br /><br />
  <div id="trash">

  <!--DROP HERE TO DELETE -->
  <br /><br />
  </div>
</body>
</html>
```

## Installation

L'installation de l'environnement XOAD est très simple. Il suffit de télécharger l'archive à l'adresse <http://sourceforge.net/projects/xoad/> puis de décompresser celle-ci à la racine de votre application web. Le répertoire contient un ensemble de sous-répertoires contenant les fichiers PHP qui définissent la configuration, les classes et les extensions de XOAD. Le lecteur pourra trouver une documentation complète du contenu de ces répertoires ainsi que des exemples d'utilisation dans les répertoires *docs/source* et *docs/tutorials*. Il est à noter que le fichier *xoad.php* qui se trouve à la racine du répertoire xoad est le plus important. C'est ce fichier qui fait référence aux autres fichiers de configuration et que vous devrez par la suite insérer dans votre code côté client.

## Mise en pratique: le drag and drop

Le but de cette section est de présenter un exemple d'application mettant en œuvre l'environnement XOAD : la gestion d'une liste de tâches par le glisser/déposer (*drag and drop*). Nous exploiterons la technologie Ajax avec la bibliothèque d'accès distant *Script.aculo.us* et XOAD afin d'implémenter une fonctionnalité que l'on peut retrouver dans les applications de gestion de tâches comme <http://www.tadalist.com>, <http://aimbo.com> ou <http://www.taskfreak.com/>. Dans ce type d'applications, l'utilisateur a la possibilité d'entrer une tâche au travers d'un champ d'acquisition et peut valider son entrée soit en appuyant sur la touche « Entrée » soit en cliquant sur un bouton *Add this task*. La Figure 1 montre le résultat visuel que nous allons obtenir après avoir implémenté notre application.

Les tâches sont insérées dans une liste de puces ordonnée et peuvent être supprimées par un *glisser/déposer* dans une poubelle. La confirmation de la suppression se fait par l'intermédiaire d'une fenêtre de dialogue (Figure 2).

Le cahier des charges de l'application peut être décrit à l'aide des points suivants :

- Pour ajouter une tâche,
  - l'utilisateur entre une tâche,
  - lorsque l'utilisateur clique sur la touche *Entrée* ou sur le bouton *Add this task*, la donnée est envoyée au serveur par une requête HTTP asynchrone. Le script côté serveur ajoute la nouvelle tâche dans une base de données et retourne la liste mise à jour qui est injectée dans le code HTML, par l'intermédiaire du Javascript,
- Pour réarranger la liste des tâches,
  - chaque tâche est une puce dans une liste d'élément (`<li>`). L'utilisateur



Figure 1. Capture d'écran de l'application

commence le glisser/déposer et une requête HTTP est envoyée au serveur. Cette requête consiste à supprimer l'élément et à mettre à jour la liste,

- côté client, la liste est réarrangée uniquement lorsque le serveur a terminé son processus,
- Pour faire un glisser/déposer dans la poubelle,
  - l'utilisateur fait un glisser/déposer d'un item dans la zone de drop,
  - une requête HTTP est envoyée au serveur qui effectue le processus de suppression dans la base de données et l'élément XHTML est instantanément détruit.

Les tâches sont stockées dans une base de données MySQL. Cette base nommée *ajax* est composée d'une table *tasks* avec 3 champs : l'identifiant, le numéro et la description de la tâche. Pour les besoins de l'exemple, la base de données et la table peuvent être implémentées avec *PhpMyAdmin* (Figure 3) ou générées par SQL avec le code montré sur le Listing 4.

Avant de passer à l'implémentation des fichiers, il faut installer les librairies que l'on devra utiliser :

- Télécharger les librairies `script.aculo.us` (<http://script.aculo.us/downloads>) et `XOAD` (<http://sourceforge.net/projects/xoad>),
- Décompresser les archives dans votre répertoire web et renommer les répertoires des librairies simplement en *aculo* et *xoad*.

La librairie `script.aculo.us` nous servira entre autre pour implémenter la liste ordonnée grâce à la fonction `Sortable`.

Comme vous pouvez le voir sur la Figure 1 et 2, nous utilisons des figures pour représenter la poubelle ainsi que les puces permettant d'identifier les tâches. Nous laissons le choix au lecteur de personnaliser ces images et nous ne donnerons pas le fichier de style au format `css` dans cet article.

Nous allons à présent commenter les fichiers qui composent notre application. Nous verrons dans un premier temps le fichier principal *index.php* puis nous détaillerons le contenu des fichiers *taskslist.class.php* et *drag-and-drop.js*.

### Le fichier `index.php`

Le fichier *index.php* est le cœur de notre application car il constitue l'interface de gestion des tâches. Cette gestion se fait par l'intermédiaire de fonctions Javascript qui utilisent les méthodes de la classe `TasksList`. Cette procédure est évidemment assurée par `XOAD` qui va permettre l'exportation de deux méthodes appartenant à cette classe.

Comme vous pouvez le voir sur le Listing 5, le fichier principal débute avec un script PHP qui invoque les instructions permettant



Figure 2. Capture d'écran de la fenêtre de dialogue confirmant la suppression d'une tâche

de charger et d'initialiser notre environnement `XOAD`. Le fichier *taskslist.class.php* est demandé et la classe `TasksList` implémentée dans ce fichier est enregistrée. Le code HTML débute avec le chargement des fichiers Javascript nécessaires à l'application : `prototype.js`, `scriptaculous.js`, `drag-and-drop.js`. Le fichier

### Listing 6. Code Javascript du fichier `drag-and-drop.js`

```
function startup() {
    Sortable.create("tasksList", {tag:"li"});
    Droppables.add("trash", {
        onDrop: function(element) {
            var deleteTask = confirm("Voulez vous supprimer cette tâche?");
            if (deleteTask) {
                Element.hide(element);
                process(element.id, "delTask");
            }
        }
    });
    obj.BuildTasksList("not_used", function(result){postUpdateProcess(result)});
}

function postUpdateProcess(data) {
    if (data.indexOf("ERRNO") >= 0 || data.indexOf("error") >= 0)
        alert(data);
    document.getElementById("tasksList").innerHTML = data;
    Sortable.create("tasksList");
    document.getElementById("txtNewTask").value = "";
    document.getElementById("txtNewTask").focus();
}

function process(content, action){
    var tab= new Array(content,action);
    if (action == "delTask")
        obj.Process(tab,function(result){postUpdateProcess(result)});
    else if ((action == "updateList") || (action == "addNewTask")) {
        tab[0]=document.getElementById("txtNewTask").value;
        obj.Process(tab,function(result){postUpdateProcess(result)});
    }
}

function handleKey(e) {
    e = (!e) ? window.event : e;
    code = (e.charCode) ? e.charCode :
        ((e.keyCode) ? e.keyCode :
        (e.which) ? e.which : 0));
    if (e.type == "keydown") {
        if(code == 13) {
            process("txtNewTask", "addNewTask");
        }
    }
}
```

*xoad\_optimized.js* est utilisé pour optimiser les fonctionnalités de XOAD. Le fichier *drag-and-drop.css* qui contient la définition des conteneurs *div* utilisés plus bas dans le code est assez simple et n'est pas présenté dans cet article.

Lorsque le fichier est interprété par le navigateur, la fonction Javascript `startup()` est invoquée afin d'initialiser la fonctionnalité du *drag and drop* et d'afficher la liste éventuelle des tâches répertoriées au cours de la session précédente. Ensuite, le champ de texte est introduit avec la redéfinition du gestionnaire d'événements afin d'invoquer l'ajout d'une tâche lorsque l'utilisateur clique sur la touche *Entrée*. Ce champ est suivi d'un bouton qui implémente, par la fonction `process()`, la même fonctionnalité lorsqu'il est cliqué. Enfin, la liste est implémentée et une redéfinition du gestionnaire d'événements sur la souris est spécifiée pour réordonner la liste si l'utilisateur essaie de la modifier sans supprimer une tâche. Lorsqu'une tâche est ajoutée la fonction `process()` mo-

Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/> id	int(10)		UNSIGNED	Non		auto_increment	    
<input type="checkbox"/> order	int(10)		UNSIGNED	Non	0		    
<input type="checkbox"/> description	varchar(100)	latin1_swedish_ci		Non	"		    

Tout cocher / Tout décocher Pour la sélection:      

Figure 3. Capture d'écran de la création de la base de données *ajax* dans *PhpMyAdmin*

difie (par l'intermédiaire d'une autre fonction que nous verrons plus tard) le DOM afin d'ajouter les balises `<li>` supplémentaires. Vous remarquerez la présence du conteneur *div* avec l'identifiant *trash* qui n'est autre que notre poubelle.

Vous pouvez noter que le fichier principal est simple et n'invoque que des fonctions Javascript.

### Le fichier *drag-and-drop.js*

Le fichier *drag-and-drop.js* contient toutes les fonctions Javascript permettant la manipulation de la liste des tâches mais également la gestion des événements utilisateurs de notre application comme l'appui sur la touche *Entrée*.

Comme vous pouvez le voir sur le Listing 6, la fonction `startup()` permet de créer la liste énumérée grâce à la fonction `sortable()` disponible dans la librairie *script.aculo.us*.

De plus, nous voyons pour la première fois l'invocation d'une méthode exportée de la classe `TaskList: BuildTasksList()`. Cette méthode PHP est appelée de manière transparente dans le code Javascript. Elle permet de questionner la base de données *ajax* en lui demandant de donner la liste des tâches déjà présentes dans la table *tasks*. Si la requête est positive, la liste s'affiche à l'écran. Vous remarquerez que la méthode est appelée à partir de l'instance `obj` définie dans l'entête du fichier *index.php*.

#### Listing 7. Code PHP du fichier *tasklist.class.php*

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'guest');
define('DB_PASS', 'guest');
define('DB_NAME', 'ajax');
define('DB_TABLE', 'tasks');
class TaskList {
    public function xoadGetMeta(){
        XOAD_Client::mapMethods($this, array('Process',
            'BuildTasksList'));
        XOAD_Client::publicMethods($this, array('Process',
            'BuildTasksList'));
    }
    public function getConnection() {
        $connection = mysql_connect(DB_HOST, DB_USER, DB_PASS);
        mysql_select_db(DB_NAME, $connection);
        return $connection;
    } public function closeConnection($connection) {
        mysql_close($connection);
    } public function BuildTasksList($data){
        $myList = '';
        $connection = $this->getConnection();
        $sqlQuery = 'SELECT * FROM `tasks` ORDER BY `order` ASC';
        mysql_query($sqlQuery, $connection);
        $sqlResult = mysql_query($sqlQuery);
        while ($row = mysql_fetch_assoc($sqlResult)) {
            $myList .= '<li id="' . htmlentities($row['id']) .
                '>' . htmlentities($row['description']) .
                '</li>';
        }
        mysql_free_result($sqlResult);
        $this->closeConnection($connection);
        return $myList;
    } public function Process($tab) {
        $content=$tab[0];
        $action=$tab[1];
        $connection = $this->getConnection();
        switch($action) {
            case 'updateList':
                $new_order = explode('_', $content);
                for ($i=0; $i < count($new_order); $i++) {
                    $sqlQuery = 'UPDATE `tasks` SET order="' .
                        $i . " WHERE id=" . $new_order[$i] . "'";
                    $result = mysql_query($sqlQuery, $connection);
                }
                $updatedList = $this->BuildTasksList("");
                return $updatedList;
            break;
            case 'addNewTask':
                $sqlQuery = 'SELECT (MAX(`order`) + 1) . `AS` `order`
                    FROM `tasks`';
                $result = mysql_query($sqlQuery, $connection);
                $row = mysql_fetch_assoc($result);
                $order = $row['order'];
                if (!$order) $order = 1;
                $sqlQuery = 'INSERT INTO `tasks` (`order`,
                    `description`) VALUES (" . $order . ", "' .
                    $content . '")';
                $result = mysql_query($sqlQuery, $connection);
                $updatedList = $this->BuildTasksList("");
                return $updatedList;
            break;
            case 'delTask':
                $sqlQuery = 'DELETE FROM `tasks` WHERE id="' . $content
                    . "'";
                $result = mysql_query($sqlQuery, $connection);
                $updatedList = $this->BuildTasksList("");
                return $updatedList;
            break;
        }
        mysql_free_result($result);
        $this->closeConnection($connection);
    }
}
?>
```



# Le B.A.

## BA de la Vidéo sur Internet

Tandis que la consommation de vidéo explose sur Internet en liaison avec l'essor foudroyant des sites, de la vidéo à la demande (VoD) et de la Catch-Up TV, un petit rappel s'impose sur les différentes technologies qui pullulent derrière ce marché.

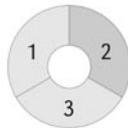
### Cet article explique :

- Les différentes technologies de diffusion vidéo sur le Web.
- La bataille des principaux Players du marché.

### Ce qu'il faut savoir :

- Des connaissances en architecture client / serveur.
- Les notions de base des protocoles TCP/IP.

### Niveau de difficulté



Nous commençons par vous présenter les différents types de diffusion : Streaming, téléchargement, Progressive Download, peer-to-peer. Il existe trois grandes familles de protocoles pour diffuser une vidéo sur Internet : le téléchargement, qui regroupe plusieurs techniques et protocoles comme le FTP ou *FTP over SSH*, mais nous ne parlerons ici que du téléchargement progressif (ou *Progressive Download* en anglais) popularisé par YouTube ou Dailymotion, le *Streaming* et le *Peer-to-Peer* (P2P).

Les deux différences principales entre Streaming et téléchargement concernent la gestion du cache, ou buffer, et la gestion de la bande passante ; le P2P, bien que différent, se rapprochant plus des techniques de téléchargement que du Streaming.

Pour délivrer un fichier vidéo en *Progressive Download*, il suffit d'un serveur web qui, via le protocole http, va délivrer le fichier vidéo comme n'importe quel autre objet, en utilisant au maximum la bande passante dont il dispose, sans se limiter. C'est un avantage si la différence entre la bande passante disponible à un instant  $T$  et la qualité d'encodage de la vidéo est grande.

En effet, dans le cas du téléchargement progressif, le fichier vidéo est téléchargé sur le disque dur et le lecteur vidéo lit la vidéo depuis le disque. Si la bande passante est suffisante, le fi-

chier se retrouve plus vite sur le disque dur que le lecteur met de temps à le lire, le protocole TCP, une des sous-couches réseaux sur lequel est basé HTTP, se chargeant d'assurer l'intégrité du transfert.

Dans le cas contraire, si la bande passante est insuffisante, la lecture progresse alors plus vite que le téléchargement et celui-ci sera interrompu. Pour éviter ce phénomène de coupure au beau milieu d'une vidéo, les lecteurs en *Progressive Download* ne débutent la lecture du fichier que lorsqu'ils sont quasi sûrs qu'elle ne sera pas interrompue avant la fin, ce qui ralentit d'autant le lancement de la vidéo.

L'autre moyen pour éviter ces interruptions est de faire du Streaming. Contrairement au téléchargement, un simple serveur web ne suffit pas dans ce cas. Il faut un serveur spécialisé dans la distribution de fichiers son ou vidéo via un protocole de Streaming. Le protocole de base du Streaming est le RTSP, lui-même basé sur le RTP. Une des particularités principales du protocole RTSP est qu'il sait s'adapter à la bande passante dont il dispose.

Lorsqu'un lecteur de Streaming demande une vidéo à un serveur, il se connecte à deux ports distincts, l'un qui reçoit les données de la vidéo et l'autre qui gère le buffering en liaison avec le serveur en fonction de la bande passante disponible, ainsi que le déplacement au sein de la vidéo (fonction d'avance ou de recul dans une séquence). Dans l'exemple de fichiers multi-bitrates, c'est-à-dire contenant plusieurs qualités d'encodage dans le même fichier, le lecteur demande au serveur de lui envoyer la qualité correspondante à la valeur actuelle de sa bande passante.

Le second avantage du Streaming sur le *Progressive Download* tient à la gestion du Buffer, une zone de mémoire tampon, en mémoire vive, jamais sur le disque dur sauf exception très particulière. Cette particularité est importante dans l'impression de sécurité qu'elle procure aux ayants-droits. En effet, même si aujourd'hui des logiciels de gestion de droits (DRM) apportent une sécurisation du contenu encore plus grande, celle, implicite, du Streaming est déjà garante de sécurité auprès de bon nombre d'acteurs du marché.

Néanmoins, si la technologie Streaming peut sembler séduisante en théorie, dans la pratique la discussion entre lecteur et serveur est souvent problématique, et les choix pris par le lecteur rarement excellents. Il est généralement préférable d'encoder les fichiers de telle manière que le lecteur n'ait pas à faire ces choix (nous en reparlerons dans la section dédiée à l'encodage des fichiers).

Par ailleurs, si l'adaptation à la bande passante disponible est un atout pour le protocole RTSP lorsque celle-ci est disponible en quantité, c'est un inconvénient dans le cas contraire. En effet, si la bande passante n'est pas suffisamment régulière, la discussion entre le lecteur vidéo et le serveur de Streaming ne se passe pas bien.

Enfin, si l'on prend en compte les bandes passantes disponibles chez les utilisateurs, l'avantage du Streaming par rapport au *Progressive Download* dans la gestion de la bande passante, se réduit considérablement sur certains contenus.

Notons que nous n'avons parlé ici que de transfert de fichiers, soit en téléchargement soit en Streaming, sans aborder le cas du transfert en quasi temps réel d'un évènement, en direct ou en direct simulé.

De la même manière, nous ne ferons qu'évoquer la distribution *peer-to-peer*, qui s'apparente à du téléchargement, à la différence

qu'au lieu d'avoir une connexion d'un serveur vers un client, on a une multitude de serveurs dont chacun diffuse des paquets du fichier, dans un ordre dispersé. Le P2P est une tech-

nologie qui permet de diffuser rapidement, sans coût excessif et à un grand nombre de personnes, un fichier volumineux ou non. Et si l'on rajoute une variable permettant

d'ordonner l'arrivée des morceaux d'une vidéo, alors on peut aussi parler de téléchargement progressif (cf la technologie 1-Click). Cette technologie aura l'avenir devant elle le

### Listing 1. L'exemple type d'un flux YouTube

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns='http://www.w3.org/2005/Atom' xmlns:openSearch='http://a9.com/-/spec/opensearchrss/1.0/' xmlns:gml='http://www.opengis.net/gml' xmlns:georss='http://www.georss.org/georss' xmlns:media='http://search.yahoo.com/mrss/' xmlns:batch='http://schemas.google.com/gdata/batch' xmlns:yt='http://gdata.youtube.com/schemas/2007' xmlns:gdt='http://schemas.google.com/g/2005'>
<id>http://gdata.youtube.com/feeds/api/videos</id>
<updated>2008-10-27T18:45:20.000Z</updated>
<category scheme='http://schemas.google.com/g/2005#kind' term='http://gdata.youtube.com/schemas/2007#video' />
<title type='text'>YouTube Videos matching query: skateboarding dog</title>
<logo>http://www.youtube.com/img/pic_youtubelogo_123x63.gif</logo>
<link rel='alternate' type='text/html' href='http://www.youtube.com' />
<link rel='http://schemas.google.com/g/2005#feed' type='application/atom+xml' href='http://gdata.youtube.com/feeds/api/videos' />
<link rel='http://schemas.google.com/g/2005#batch' type='application/atom+xml' href='http://gdata.youtube.com/feeds/api/videos/batch' />
<link rel='self' type='application/atom+xml' href='http://gdata.youtube.com/feeds/api/videos?start-index=21&max-results=1&vq=skateboarding+dog' />
<link rel='previous' type='application/atom+xml' href='http://gdata.youtube.com/feeds/api/videos?start-index=20&max-results=1&vq=skateboarding+dog' />
<link rel='next' type='application/atom+xml' href='http://gdata.youtube.com/feeds/api/videos?start-index=22&max-results=1&vq=skateboarding+dog' />
<author>
<name>YouTube</name>
<uri>http://www.youtube.com</uri>
</author>
<generator version='2.0' uri='http://gdata.youtube.com/'>YouTube data API</generator>
<openSearch:totalResults>5381</openSearch:totalResults>
<openSearch:startIndex>21</openSearch:startIndex>
<openSearch:itemsPerPage>1</openSearch:itemsPerPage>
<entry>
<id>http://gdata.youtube.com/feeds/api/videos/qzyjvXSL3es</id>
<published>2008-02-08T19:02:26.000Z</published>
<updated>2008-10-27T18:45:20.000Z</updated>
<category scheme='http://gdata.youtube.com/schemas/2007/keywords.cat' term='Tillman' />
<title type='text'>Skateboard dog Tillman "The Greatest American Dog" Venice Beach California</title>
<content type='text'>An amazing bulldog Tillman skateboarding "The Greatest American Dog!!! Filmed by Name On Rice.com</content>
<link rel='alternate' type='text/html' href='http://www.youtube.com/watch?v=qzyjvXSL3es' />
<link rel='http://gdata.youtube.com/schemas/2007#video.responses' type='application/atom+xml' href='http://gdata.youtube.com/feeds/api/videos/qzyjvXSL3es/responses' />
<link rel='http://gdata.youtube.com/schemas/2007#video.related' type='application/atom+xml' href='http://gdata.youtube.com/feeds/api/videos/qzyjvXSL3es/related' />
<link rel='http://gdata.youtube.com/schemas/2007#mobile' type='text/html' href='http://m.youtube.com/details?v=qzyjvXSL3es' />
<link rel='self' type='application/atom+xml' href='http://gdata.youtube.com/feeds/api/videos/qzyjvXSL3es' />
<author>
<name>NameOnRice</name>
<uri>http://gdata.youtube.com/feeds/api/users/nameonrice</uri>
</author>
<media:group>
<media:title type='plain'>Skateboard dog Tillman "The Greatest American Dog" Venice Beach California</media:title>
<media:description type='plain'>An amazing bulldog Tillman skateboarding "The Greatest American Dog!!! Filmed by Name On Rice.com</media:description>
<media:keywords>american, angeles, beach, bulldog, california, dog, greatest, los, nameonriceCom, robinson, skateboard, skateboarding, the, Tillman, venice, vivianne</media:keywords><yt:duration seconds='195' />
<media:category label='Animaux' scheme='http://gdata.youtube.com/schemas/2007/categories.cat'>Animals</media:category>
<media:content url='http://www.youtube.com/v/qzyjvXSL3es&f=gdata_videos' type='application/x-shockwave-flash' medium='video' isDefault='true' expression='full' duration='195' yt:format='5' />
<media:content url='rtsp://rtsp2.youtube.com/CigLEny73wIaHwnr3Yt0vaM8qxMYDSANFEgUGxNZGF0YV92aWRlb3MM/0/0/0/video.3gp' type='video/3gpp' medium='video' expression='full' duration='195' yt:format='1' />
<media:content url='rtsp://rtsp2.youtube.com/CigLEny73wIaHwnr3Yt0vaM8qxMYESARFEgUGxNZGF0YV92aWRlb3MM/0/0/0/video.3gp' type='video/3gpp' medium='video' expression='full' duration='195' yt:format='6' />
<media:thumbnail url='http://i.ytimg.com/vi/qzyjvXSL3es/2.jpg' height='97' width='130' time='00:01:37.500' />
<media:thumbnail url='http://i.ytimg.com/vi/qzyjvXSL3es/1.jpg' height='97' width='130' time='00:00:48.750' />
<media:thumbnail url='http://i.ytimg.com/vi/qzyjvXSL3es/3.jpg' height='97' width='130' time='00:02:26.250' />
<media:thumbnail url='http://i.ytimg.com/vi/qzyjvXSL3es/0.jpg' height='240' width='320' time='00:01:37.500' />
<media:player url='http://www.youtube.com/watch?v=qzyjvXSL3es' />
</media:group><yt:statistics viewCount='1702' favoriteCount='9' />
<gd:rating min='1' max='5' numRaters='2' average='5.00' />
<gd:comments>
<gd:feedLink href='http://gdata.youtube.com/feeds/api/videos/qzyjvXSL3es/comments' countHint='3' />
</gd:comments>
</entry>
</feed>
```

jour ou elle cessera d'être associée au phénomène du piratage.

## L'encodage des vidéos

Maintenant nous allons vous présenter quatre questions essentielles s'il s'agit de l'encodage des vidéos.

### Formats et codecs

Un fichier multimédia est composé d'un ensemble d'éléments. Il y a tout d'abord un conteneur qu'on appelle format, qui est l'enveloppe du fichier, qui peut être simple et ne contenir que la vidéo et une piste son (stéréo ou non), ou beaucoup plus compliquée, avec par exemple différentes qualités d'un flux vidéo, plusieurs pistes son en différentes langues, des images, du texte pour les sous-titres, etc.

Chacune de ces pistes est ensuite encodée d'une certaine manière : c'est ce qu'on appelle les codecs. Certains codecs sont liés et se confondent même avec un format, comme c'est souvent le cas des formats propriétaires (*wmv*, *rm* ou *flv*). D'autres distinguent véritablement conteneur et codecs, comme Apple avec le format *mov* ou le consortium *MPEG* avec le format *mp4*.

Les formats et *codecs* vidéo ou audio sont une véritable jungle dans l'Internet multimédia. Nous serons volontairement partiaux

pour n'aborder que les formats et codecs représentant une part significative de la vidéo et du son circulant sur Internet. Nous passerons donc rapidement sur les formats sous licences OpenSource (*Ogg Vorbis* par exemple) qui, bien qu'étant de très bons projets, n'ont pas été adoptés par les grands producteurs de contenus et sont donc peu utilisés dans les sites vidéos ou musicaux.

De plus, ces formats ne sont supportés ni par les lecteurs *embedded* classiques que l'on inclut dans les pages web, ni par les serveurs de streaming *standards* que sont Windows Media Server de Microsoft, Darwin Streaming Server d'Apple (Quicktime) ou encore Helix (Real Networks).

En fait, en citant ces trois logiciels qui sont spécialisés dans la diffusion de flux streamés, on fait le tour de la très grande majorité des formats diffusés sur Internet. Les formats associés au téléchargement progressif ont, eux, fait un retour sur le devant de la scène depuis deux ans avec l'avènement des sites de partage vidéo comme YouTube ou Dailymotion.

### Les codecs Son

La qualité du son est très importante, voire primordiale suivant la nature de la production. Les codecs actuels liés aux formats *wmv* et *h264*, sont le *wma9.1* et le *AAC*. Ils sont

tous deux de qualité à peu près deux fois supérieure au célèbre *mp3*. Ainsi, si l'on souhaite une qualité équivalente à du 128 kbps en *mp3*, on peut utiliser du 64 kbps en *wma9.1* ou en *AAC*. Mais si l'on veut mettre l'accent sur la musique, un encodage à 128 kbps permettra à l'utilisateur d'entendre un son identique à celui de sa chaîne hi-fi. Tout cela se fait néanmoins au détriment de la vidéo, bien entendu. Le tuyau ne bouge pas, et si l'on veut augmenter la qualité du son, il faut savoir baisser la qualité de l'image.

### La bataille des Players : Windows Media Vs Flash

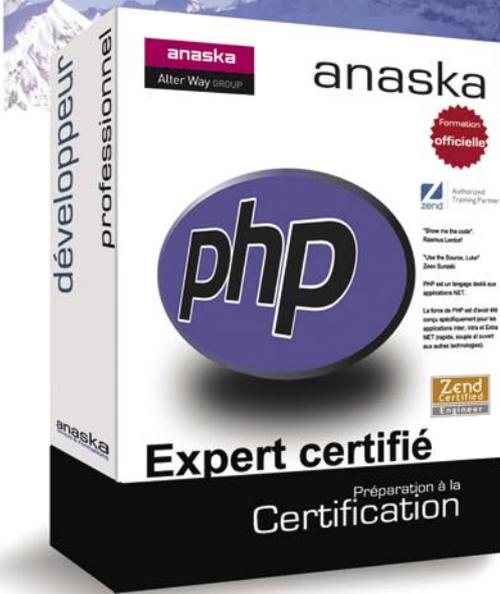
En Europe, deux players se partagent aujourd'hui le marché de la diffusion vidéo sur le web : le lecteur Windows Media, capable de lire les formats *wmv9*, 8 et 7 plus quelques autres comme le *mp3*, et le lecteur *Flash* capable, lui, de lire les formats historiques liés à *Flash* comme le *Sorenson*, les *On2 VP5* et *VP6*, mais aussi plus récemment, et ce qui a profondément modifié la donne, le format *mp4* et l'un de ces codecs associés, le désormais célèbre *h264*.

Il y a quelques années, le format de *Real Networks* (RN) avait la vie belle. Acteur incontournable du monde du streaming, RN avait participé à la création du RTSP et fait partie des

**Listing 2.** La librairie agrégation en PHP

```
require_once 'Zend/Loader.php';
Zend_Loader::loadClass('Zend_Gdata_YouTube');
// on récupère la variable en POST ou en GET pour la recherche
$queryType = isset($_POST['queryType']) ? $_POST['queryType'] :
    null;
if ($queryType === null) {
    /* affichage par défaut de l'interface */
    include 'interface.html';
} else if ($queryType == 'show_video') {
    /* affichage d'une video */
    if (array_key_exists('videoId', $_POST)) {
        $videoId = $_POST['videoId'];
        echoVideoPlayer($videoId);
    } else if (array_key_exists('videoId', $_GET)) {
        $videoId = $_GET['videoId'];
        echoVideoPlayer($videoId);
    } else {
        echo 'No videoId found.';
        exit;
    }
} else {
    /* affichage d'une liste de vidéos */
    $searchTerm = $_POST['searchTerm'];
    $startIndex = $_POST['startIndex'];
    $maxResults = $_POST['maxResults'];
    $yt = new Zend_Gdata_YouTube();
    $query = $yt->newVideoQuery();
    $query->setQuery($searchTerm);
    $query->setStartIndex($startIndex);
    $query->setMaxResults($maxResults);
    /* regarde le standard voulu (plus vues, plus récentes, top,
    etc ... ) */
    switch ($queryType) {
        case 'most_viewed':
            $query->setFeedType('most viewed');
            $query->setTime('this_week');
            $feed = $yt->getVideoFeed($query);
            break;
        case 'most_recent':
            $query->setFeedType('most recent');
            $feed = $yt->getVideoFeed($query);
            break;
        case 'recently_featured':
            $query->setFeedType('recently featured');
            $feed = $yt->getVideoFeed($query);
            break;
        case 'top_rated':
            $query->setFeedType('top rated');
            $query->setTime('this_week');
            $feed = $yt->getVideoFeed($query);
            break;
        case 'all':
            $feed = $yt->getVideoFeed($query);
            break;
        default:
            echo 'ERROR - unknown queryType - "' . $queryType . "'";
            break;
    }
    echoVideoList($feed);
}
```

anaska



# Formation PHP Expert Certifié

*“ Montez en puissance sur PHP et préparez vous à passer la certification PHP ”*

## AU PROGRAMME DES 5 JOURS

- \* Certification PHP et rappels
- \* La programmation orientée objet PHP 5
- \* Les bases de données et PDO
- \* XML et les services Web
- \* Flux et réseau
- \* Sécurité
- \* Optimisation des performances
- \* Debug et audit de code



Authorized  
Training Partner



### Prochaines sessions

Paris 2008

23 Juin, 7 et 21 Juillet  
8 Septembre, 6 et 17 Octobre  
24 Novembre, 1 et 15 Décembre

Lyon 2008

23 Juin, 7 Juillet  
8 Septembre, 6 Octobre  
24 Novembre, 15 Décembre

Tarif

1999 € HT

**anaska**

Alter Way GROUP

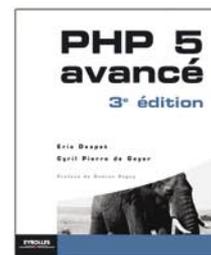
## Nos parcours de formation PHP

PHP est devenu le langage de développement web le plus utilisé par les entreprises. Nos formations vous accompagnent de la prise en main à des niveaux plus experts.

### Parcours développeur



### Les références



Informations : 01 45 28 09 82  
<http://www.anaska.com>

acteurs ayant mis au point le standard MPEG4. Le codec *Real G2* n'était autre qu'une très légère variation du *h263*, qui allait être réutilisé comme standard pour la vidéo sur mobiles. Les codecs suivants, les *Real 9* et *10*, sont d'excellents codecs, mais faute d'une politique agressive sur ce marché, *RN* s'est laissé distancer par son principal concurrent : Microsoft.

Microsoft (MS) a commencé sa carrière dans le streaming avec un codec médiocre nommé *wmv7*. La troisième mouture de leur codec, le *wmv9*, allait franchir le fossé qui séparait MS de ses concurrents. Une révision du codec audio accompagnant le codec vidéo, baptisé *9.1 advanced*, a été nécessaire pour améliorer sa qualité.

Pour créer des vidéos en *wmv9*, il suffit de posséder un ordinateur fonctionnant sous Windows et de télécharger, sur le site de MS, le

*Windows Media Encoder* (WME) qui est gratuit et simple d'utilisation.

Le paysage vidéo Internet a été bouleversé par l'arrivée des sites comme Dailymotion et YouTube. Ces sites ont déclenché un véritable raz de marée sur Internet : on pouvait d'un seul coup diffuser facilement des vidéos, visibles par tous, et gratuitement. Effet collatéral de ces sites, le format *flv*, c'est à dire le format vidéo utilisé par Macromedia, racheté depuis par Adobe, a submergé le marché.

Pour publier des vidéos sur Internet, les sites comme YouTube et Dailymotion ont eu l'excellente idée de transcoder automatiquement les fichiers qu'on déposait chez eux vers un format *Flash* standard, pouvant être vu sur toutes les machines ou presque. Pour cela, ils ont utilisé pour une majeure partie un programme nommé *ffmpeg*. Ce programme repose sur deux

fameuses bibliothèques, *libavformat* et *libavcodec*, celles utilisées par le célèbre projet VLC, véritables couteaux suisses du développement vidéo et son sur Internet. Grâce à ces bibliothèques, on a pu monter à peu de frais, et en batterie, des moteurs qui transcendent les dizaines de milliers de vidéos que reçoivent des plates-formes comme YouTube.

Mais il y a trois ans, les formats/codecs supportés par Flash étaient le *On2 VP5*, le *On2 VP6* qui venait de sortir, et le *Sorenson*. De nettement moins bonne qualité que le *VP6* qui n'était pas encore intégré dans la partie encodage des bibliothèques *libavformat* et *libavcodec*, le *On2 VP5* avait l'avantage d'être libre, et les plates-formes ont donc pu au début transcoder vers du *VP5* sans payer de droits d'utilisation. Et lorsqu'on est un YouTube et que l'on reçoit plusieurs dizaines de milliers de vidéos par

**Listing 3.** Le code *Mash-Up Video PHP*

```
<html>
  <head>
    <title>API Video en PHP</title>
    <link href="video_browser.css" type="text/css"
      rel="stylesheet"/>
    <script src="video_browser.js" type="text/javascript">
    </script>
  </head>
  <body>
    <div id="main">
      <div id="titleBar">
        <div id="titleText"><h1>YouTube data API Video Browser
          in PHP</h1></div>
        <div id="searchBox" style="display: none;">
          <form id="searchForm" onsubmit="ytvbp.listVideos(this.
            queryType.value, this.searchTerm.value, 1);
            return false;">
            <select name="queryType" onchange="ytvbp.queryType
              Changed(this.value, this.form.searchTerm);">
              <option value="all" selected="true">All Videos
              </option>
              <option value="top_rated">Top Rated Videos
              </option>
              <option value="most_viewed">Most Viewed Videos
              </option>
              <option value="recently_featured">Recently
                Featured Videos</option>
            </select>
            <input name="searchTerm" type="text" value="puppy">
            <input type="submit" value="Search">
          </form>
        </div>
        <br />
      </div>
      <br clear="all" />
      <div id="mainSearchBox">
        <h2>Search YouTube:</h2>
        <form id="mainSearchForm" onsubmit="ytvbp.listVideos(
          this.queryType.value, this.searchTerm.value,
          1); document.forms.searchForm.searchTerm.value=
          this.searchTerm.value; ytvbp.hideMainSearch();"
          </form>
      </div>
      <div id="searchResults">
        <div id="searchResultsListColumn">
          <div id="searchResultsVideoList"></div>
          <div id="searchResultsNavigation">
            <form id="navigationForm">
              <input type="button" id="previousPageButton"
                onclick="ytvbp.listVideos(ytvbp.previous
                  QueryType, ytvbp.previousSearchTerm,
                  ytvbp.previousPage);" value="Back"
                style="display: none;"></input>
              <input type="button" id="nextPageButton"
                onclick="ytvbp.listVideos(ytvbp.
                  previousQueryType, ytvbp.previousSearchTerm,
                  ytvbp.nextPage);" value="Next" style=
                  "display: none;"></input>
            </form>
          </div>
        </div>
        <div id="searchResultsVideoColumn">
          <div id="videoPlayer"></div>
        </div></div>
    </body>
  </html>
```

jour, le coût de logiciels propriétaires est très important. Pouvoir s'en passer est une économie substantielle qui mérite qu'on fasse de la R&D dessus.

Arrive alors le codec *h264*! Adobe a eu la très bonne idée voilà un peu plus d'un an, d'annoncer que son Player intégrerait le codec *h264*, lié au format *mp4*. Le fait qu'Adobe ait choisi d'intégrer ce codec présent sur 98% des postes des utilisateurs à son Player, a été un signal fort au monde de la vidéo sur Internet. C'est la libéralisation de la création de contenus vidéo qui arrivait sur Internet.

Le format *mp4* contient à la fois du son, de la vidéo, du réseau, du contrôle, etc. et même une définition de caractères texte, et peut s'adapter à tout type de support de diffusion, du Broadcast au mobile, du *SD* à la *HD*, en passant par Internet, la TNT ou le Blu-Ray.

Ne restait plus que l'Internet où le consensus n'était pas atteint, et ne l'est toujours pas. La bataille sera intense entre le géant de Redmond et le reste du monde qui va avoir tendance à adopter le format *MPEG-4* et son dernier codec associé, le *h264*.

### Silverlight, la réponse de Microsoft au MPEG-4

Silverlight est au format *wmv* ce que Flash est au format *flv*. Sauf que la vague du format *flv* a déferlé bien après le monopole de Flash sur les contenus dynamiques. A contrario,

l'immense base de vidéos encodées au format *wmv* sur le web pourrait faciliter l'installation de Silverlight.

Un choix va en effet s'offrir aux sites web : soit l'éditeur dispose de nombreux contenus au format *wmv*, et cela lui coûtera cher de transcoder l'ensemble de ses vidéos dans un nouveau format, soit il a peu de vidéos, ou encodées dans un format supporté par *Flash*, et cela lui coûtera cher de faire développer un site dans une nouvelle technologie.

On voit bien dans ces deux schémas que l'un est plus facilement adoptable que l'autre. Si Adobe n'avait pas fait un effort important sur la vidéo et n'avait pas adopté le format *MPEG-4*, mettant derrière lui l'ensemble de la profession vidéo autre que Microsoft et certains de ses alliés, Adobe n'aurait pu résister au rouleau compresseur MS. Le rapport de force est à présent équilibré, et l'avenir nous dira s'il y a de la place pour deux formats.

Avec l'arrivée tardive de Silverlight, les deux concurrents redoublent d'annonces pour s'attirer la communauté des développeurs. Leur format est de plus en plus orienté développeur : Flex pour Adobe et Silverlight (*XAML // javascript*) pour Microsoft.

Ces technologies sont de conceptions assez communes : les deux reposent sur une description par balisage et communiquent avec l'extérieur via javascript. Elles proposent par ailleurs des outils pour faciliter le développement d'ap-

plication sur leur produit. Microsoft propose sa gamme Expression ainsi qu'un Toolkit accessible ici : <http://www.codeplex.com/Silverlight>, sinon vous avez le B.A. BA à cette adresse : <http://dotnet.developpez.com/faq/silverlight/>.

Les langages compatibles avec Silverlight :

Version 1.0 : Javascript uniquement.

Version 2.0 : Tout langage de la CLR, donc tout langage compilable en MSIL, ainsi que 4 langages dynamiques : IronPython, IronRuby, JScript, VBScript.

Microsoft fait développer pour les linuxiens et les gens du libre un plug-in Moonlight pour toucher cette communauté qui est généralement réfractaire à tout ce qui vient de la firme de Redmond.

De plus, pour promouvoir le développement de Silverlight, Microsoft apporte son soutien, notamment financier, à Soyatec afin que cette équipe de développeurs intègre Silverlight à Eclipse, sous le nom de code du projet Eclipse4SL. Cette extension sera diffusée sous la licence Eclipse Public License Version 1.0 sur SourceForge. Adobe publie de son côté un framework open source pour Flex.

Les premiers développements sur ces différentes technologies utilisaient PHP, pour générer les flux xml ou faire le lien avec la base de données. Avec ces ouvertures et l'utilisation de ces technologies des langages de balisage, il devient possible de générer et manipuler plus avant ces objets qui rendront les sites web plus riches encore.

Liens :

- silverlight: <http://silverlight.net/default.aspx>,
- le xaml <http://www.xaml.fr/silverlight.html>,
- flex framework: <http://www.adobe.com/products/flex/flexdownloads/index.html>.

### Comment réaliser de bonnes vidéos pour Internet ?

Il faut tout d'abord définir à qui l'on s'adresse : une petite image de bonne qualité satisfera le plus grand nombre, une vidéo avec une belle image nécessitera des compromis sur l'audience possible, et si l'accent est mis sur la musique, la qualité de la vidéo en sera affectée.

La taille de la vidéo dépend de l'environnement dans lequel on veut que l'utilisateur la voit. Plus le site diffuseur est spécialisé dans la vidéo, plus celle-ci aura une place importante. Une taille de 320x200 (PAL), de 320x240 (NTSC) ou de 320x180 (16/9ème) est suffisante pour une vidéo illustrant un article dans un Blog généraliste. Pour un site spécialisé dans la vidéo, on préconisera dans ce cas l'utilisation d'un 512x384 pour du 3/4, ou 512x288 pour du 16/9, ou 680x382 si l'on dispose de beaucoup de place. Enfin, si la vidéo est le produit d'un site, on peut alors fournir une qualité DVD. Dans ce cas, il faudra peut-être passer à la dimension supérieure et encoder la vidéo en qualité PAL ou NTSC, voire en HD. Attention, dans ces tailles

Listing 4. La classe Mash-Up Yahoo!

```
require_once('yahooapi.class.php');
class Exemple extends YahooAPI {
    // votre constructeur qui prend les paramètres ID et type votre sortie
    // par défaut.
    public function __construct($ID) {
        $this->setParam('output', 'php');
        $this->setService('http://local.yahooapis.com/'. 'LocalSearchService/V3/
            localSearch');
        $this->setAppID($ID);
    }
    public function Search($query,$in_location) {
        $this->setParam('query', $query);
        $this->setParam('location',$in_location);
        return $this->doAPICall();
    }
    public function Results() {
        $return = $this->getResults();
        $return = $return['ResultSet']['Result'];
        return $return;
    }
}
??
<?Php
require_once('Exemple.class.php');
$query = new Exemple(Votre_ID);
$query->Search('PHP','mashup');
$output = $query->Results();
??
```

d'images, la bande passante nécessaire à une bonne visualisation est importante, et le choix du téléchargement progressif peut s'avérer judicieux.

Se pose alors la question de la bande passante nécessaire dans chacun des cas. On remarquera que les tailles d'images sont grosso modo des multiples entiers de la première. En effet, le 512x384 est à peu près deux fois plus gros, en nombre de pixels, que le 320x240, le PAL environ deux fois plus gros que le 512x384, etc. Donc, pour garder une qualité constante entre ces formats, il suffit de doubler la bande passante nécessaire à chacun : si l'on encode une vidéo de taille 320x240 en 350 kbps, ce qui donne une qualité assez bonne, on aura alors une qualité à peu près équivalente en encodant une vidéo de 512x384 en 700 kbps, et ainsi de suite.

Ensuite, c'est surtout une question d'image. Suivant la nature de la vidéo, on aura intérêt à augmenter tel ou tel paramètre. Une vidéo normale a par exemple une fréquence de 25 images par seconde, et 30 pour le NTSC américain. Or, en adaptant la fréquence d'images à celle de la vidéo, on peut gagner 30% à 40% de bande passante. Pour avoir la même qualité d'image pour une fréquence d'images de 25 i/s et 12 i/s, on peut passer d'une bande passante de 350 kbps à une bande passante comprise entre 250 kbps et 280 kbps. Bien sûr, en appliquant les coefficients précédemment cités, on retrouvera les qualités pour les autres tailles d'image.

Néanmoins si l'on divise la fréquence d'images par deux, on ne divise pas pour autant la bande passante par deux. Cela est dû à la nature des compressions des codecs vidéo. Le principe de la compression vidéo reste le même entre les différents codecs. Pour compresser une vidéo, vous prenez une image toutes les 2, 5 ou n secondes. Ces images sont appelées des *images clefs*. Elles servent à fournir l'information entière d'une image composant la vidéo. Tout le travail des codecs va être de compresser toutes les images intermédiaires en ne retenant que les différences entre les images clefs.

Ensuite, la compression, sa qualité et son efficacité, dépendront des algorithmes qui la constituent. Bien sûr, les images clés elles-mêmes sont aussi compressées, tout comme le sont les *jpeg* par exemple.

Ainsi, lorsqu'on réduit la fréquence d'images d'une séquence, on réduit le nombre d'images mais les différences entre images étant plus grandes, on doit donc utiliser un peu plus de bande passante pour encoder ces différences, ce qui ne diminue pas la bande passante nécessaire autant qu'on pourrait le croire.

Un dernier paramètre entre en jeu dans la qualité de diffusion d'une vidéo sur Internet : le fait de pouvoir encoder sa vidéo, non pas en une seule fois, mais en plusieurs fois, le plus courant étant deux fois d'où son surnom de *double passe*.

Lors d'un encodage en double passe, le logiciel écrit des statistiques sur la vidéo : il transforme la vidéo en données réutilisables. Lors de la deuxième passe, le logiciel se sert des données qu'il a précédemment écrites pour améliorer ses algorithmes de compression, et ainsi nettement améliorer la qualité de l'image. Seul défaut de cette pratique, la double passe revient à allonger le temps de transcodage de la vidéo, et sur des codecs comme le *h264*, cela peut rapidement devenir très sensible. Au final, le choix de la fréquence des images clés est important et est fonction du mode de distribution de la vidéo, de la complexité de l'image de la vidéo, et de la qualité du rendu. Ce compromis entre qualité et bande passante est le résumé de l'histoire de la vidéo sur Internet.

### Le Mashup vidéo

Diffuser des vidéos sur un site Internet peut rapidement devenir coûteux, aussi bien en espace de stockage vidéo qu'en bande passante, consommée ou limitée par un hébergeur. Ainsi, faire supporter les coûts par l'un des grands acteurs de la vidéo sur le Web peut être une astucieuse solution. Les principaux portails vidéo ouvrent aujourd'hui leur moteur, ce qui permet de réaliser des Mashups vidéo. Ainsi Yahoo! Pipes [<http://pipes.yahoo.com/pipes/>], Popfly [<http://www.popfly.com>] pour MSN, Mashup Editor [<http://code.google.com/gme/>] pour Google, qui est une interface avec une syntaxe simplifiée, ou encore GDATA, une API développée sous différents langages, toujours par Google.

Le concept d'une Mashup vidéo est assez simple : les principaux agrégateurs de contenus fournissent une url vers laquelle on peut interroger leur moteur, en mettant en fonction de chacun des paramètres spécifiques pour définir les critères de recherche.

Ceux-ci vous répondent via des flux *xml* compatibles avec les normes rss 2.0. Il ne reste plus alors qu'à transformer ce flux *xml* et l'afficher sur un front...

Pour interroger l'agrégateur de contenu, il faut former une url qui se composera ainsi en fonction de la recherche :

- Recherche d'une vidéo à partir de son ID : <http://gdata.youtube.com/feeds/api/videos/videoID>, avec videoID qui correspond à l'id de la vidéo. Recherche de vidéos par mots-clefs :
  - début de l'url : <http://gdata.youtube.com/feeds/api/videos/>

composée avec les paramètres suivants pour définir la requête :

- `alt` : définit le format du fichier retour, peut prendre les valeurs suivantes: atom, rss, json and json-in-script, par défaut le format est l'atom,

- `author` : définit l'auteur des uploads vidéo,
- `max-results` : le nombre de résultat maximal,
- `start-index` : définit l'index du premier élément renvoyé, par défaut le premier élément est 1,
- `vq` : spécifie les mots clés recherchés, les caractères spéciaux doivent être protégés,
- `time` : date d'upload, s'applique uniquement pour les recherches `top_rated`, `top_favorites`, `most_viewed`, `most_popular`, `most_discussed`, `most_linked` et `most_responded`, peut prendre les valeurs `today` (1 jour), `this_week` (7 jours), `this_month` (1 mois) et `all_time` qui est sa valeur par défaut,
- `sortorder` : définit l'ordre d'apparition des commentaires, `ascending` ou `descending`,
- `restriction` : restriction par rapport à une IP ou un pays (ex : DE pour l'Allemagne),
- `orderby` : paramètre permettant de trier les résultats, peut prendre les paramètres suivants : `relevance`, `published`, `viewCount` et `rating`. On peut également préciser le code de l'encodage attendu (par exemple, si le résultat est attendu en mandarin, le code est zh-Hans),
- `restriction` : restriction par rapport à une IP ou par pays (ex : DE),
- `lr` : paramètre de restriction de la recherche à un langage (ex : zh-Hans),
- `location` : paramètre de restriction géographique,

### Glossaire

- HTTP : *HyperText Transfert Protocol*,
- TCP : *Transport Control Protocol*,
- UDP : *User Datagram Protocol*,
- RTP : *Real Time transport Protocol*,
- RTSP : *Real Time Streaming Protocol*,
- RTCP : *Real Time Control Protocol*,
- Unicast : connexion point à point,
- Multicast : connexion point à multiple points,
- bande passante : quantité de données pouvant circuler sur un lien réseau exprimée, en bits par secondes,
- kbps : kilobits par seconde (1 kilobits = 1024 bits = 128 octets ou bytes), c'est une unité qui représente la bande passante disponible sur un lien réseau mais aussi la *qualité* d'encodage d'une vidéo ou d'une musique, un mp3 à 128kbps par exemple. Il ne faut pas non plus confondre les kbps qui mesure la bande passante d'un lien avec les ko/s qui mesure les vitesses de transfert d'un disque dur. En général on utilise les bits par secondes dans le domaine du réseau, et les octets par secondes dans le domaine du stockage,
- `buffer` : zone de mémoire tampon,
- `codec` : *co(der)-dec(oder)*, algorithmes de compression permettant d'encoder puis de décoder différents types de fichiers, souvent audio et/ou vidéo.

- `format` : les vidéos attendues doivent être à un format spécifique,
- 1 RTSP pour le Streaming sur mobile H.263 video (dimension: plus de 176x144) et audio AMR,
- 5 pour une diffusion embed avec un player SWF,
- 6 RTSP Streaming pour le mobile, format MPEG-4 SP video (dimension: plus de 176x144) et audio AAC,
- `client` : pour spécifier votre id client,
- `category` : recherche des vidéos par rapport à un mot clé ou une catégorie.

ce qui peut donner par exemple :

- <http://gdata.youtube.com/feeds/api/videos?vq=skateboarding+dog&start-index=21&max-results=1>.

Tous les paramètres sont disponibles à l'adresse : [http://code.google.com/apis/youtube/1.0/reference.html#youtube\\_data\\_api\\_tag\\_entry](http://code.google.com/apis/youtube/1.0/reference.html#youtube_data_api_tag_entry). Le code retour standard est le suivant (Listing 1). Après avoir défini la manière d'interroger le moteur d'agrégation pour rechercher des vidéos, et le format des flux en retour, il ne reste plus qu'à créer une fonction qui va parser le flux *xml* réponse pour l'afficher sur le site, ou qu'à utiliser l'API fournie par Google, nommée Gdata. Regardons plus avant cette API *googlesque* qui est développée en PHP (avec une version au moins supérieure à la 5.1.4), et qui est un composant du framework Zend. Au travers de cette API, on peut interroger tous les outils de Google, Calendar, Blogger, Earth ou YouTube. Intéressons-nous plus précisément à ce dernier. Une fois l'installation du framework Zend effectué, il faut bien faire attention à définir dans `index.php` le chemin qui mène à la librairie Zend :

```
define("ZEND_FRAMEWORK_DIR", "/home/local/
Zend/ZendFramework/library");
```

Pour ceux qui ont accès à leur fichier de configuration PHP (*php.ini*), il est possible de le définir directement dans `include_path`. On peut désormais utiliser sans problème le module Gdata du framework Zend. Voici l'exemple standard. C'est un peu le *Hello World* local, les sources sont là `ZendGdata-1.6.2/demos/Zend/Gdata/YouTubeVideoBrowser/` pour ceux qui ont déjà dézippé `ZendGdata-1.6.2` dans le fichier `index.php` (Listing 2). Dans un fichier `interface.html` (Listing 3). Retrouvez le fichier javascript qui rend dynamique la page avec l'affichage des vidéos (`video_browser.js`), et `css` pour la mise en forme (`video_browser.css`) ici : `ZendGdata-1.6.2\demos\Zend\Gdata\YouTubeVideoBrowser`.

Avec ce petit exemple, on peut rechercher et afficher des vidéos de YouTube sur un site Web. Par ailleurs, il est également

## Sur Internet

- [WP-MP4] – [http://fr.wikipedia.org/wiki/Mpeg\\_4](http://fr.wikipedia.org/wiki/Mpeg_4),
- [DC-FFM] – <http://ffmpeg.mplayerhq.hu/>,
- [DC-MPL] – <http://www.mplayerhq.hu/design7/news.html>,
- [DC-264] – <http://rob.opendot.cl/index.php/useful-stuff/ffmpeg-x264-encoding-guide/>,
- [DC-PLA] – [http://jroller.com/page/abstractScope?entry=flash\\_mp3\\_imagesideshow\\_media\\_player](http://jroller.com/page/abstractScope?entry=flash_mp3_imagesideshow_media_player),
- [DC-SUP] – <http://www.erightsoft.com/SUPER.html>,
- [DC-VIS] – <http://techspanion.com/>
- Zend framework – <http://framework.zend.com/download/gdata/>,
- Class Zend\_Gdata\_YouTube – [http://framework.zend.com/apidoc/core/Zend\\_Gdata/Zend\\_data\\_YouTube.html](http://framework.zend.com/apidoc/core/Zend_Gdata/Zend_data_YouTube.html),
- Pour en savoir plus : [http://code.google.com/apis/youtube/1.0/developers\\_guide\\_php.html](http://code.google.com/apis/youtube/1.0/developers_guide_php.html),
- Protocole API de Google – [http://code.google.com/apis/youtube/1.0/developers\\_guide\\_protocol.html](http://code.google.com/apis/youtube/1.0/developers_guide_protocol.html), [http://code.google.com/apis/youtube/1.0/reference.html#youtube\\_data\\_api\\_tag\\_entry](http://code.google.com/apis/youtube/1.0/reference.html#youtube_data_api_tag_entry),
- Yahoo! Developpeur – critère de recherche : <http://developer.yahoo.com/search/web/V1/webSearch.html>,
- Yahoo! Developpeur – code API : <http://www.sitepoint.com/examples/yahooapi/YahooAPIarchive.zip>,
- Tutoriels Yahoo! API : <http://developer.yahoo.com/php/>.

possible d'afficher uniquement des vidéos sur un compte ou des Playlists. Pour faire de même avec Yahoo!, voici l'url à interroger : <http://api.search.yahoo.com/WebSearchService/V1/webSearch>. avec les paramètres suivants :

- `appid` (obligatoire) : id de votre application... merci Yahoo!,
- `query` (obligatoire) : vos mots-clés de recherche encodés en utf-8,
- `region` (défaut us) : pour rechercher sur un moteur de Yahoo! spécifique (par exemple : `region=uk` va interroger le moteur `uk.search.yahoo.com`),
- `type` : all retourne tous les résultats. Any retourne les résultats matchant avec au moins un critère phrase retourne les résultats contenant la phrase,
- `results` (integer: défaut 10) : nombre de résultats attendus allant de 10 à 100,
- `start` (défaut 1) : définit l'index du premier élément, ne peut pas dépasser 1000,
- `format` : spécifie le format de la recherche, any (défaut), `html`, `msword`, `pdf`, `ppt`, `rss`, `txt`, `xls`,
- `adult_ok` : (par défaut tout public) pour des recherches spécifiques aux personnes majeures, mettre à 1,
- `output` : format de sortie du flux *xml* (défaut), `json`, `php` (nous opterons pour ce dernier pour l'exemple qui sérialise les données).

Exemple :

```
$output = file_get_contents('
http://api.search.yahoo.com/WebSearch
Service/V1/webSearch?appid=Yahoodemo
&query=SitePoint&results=2&output=p
hp');
$output = unserialize($output);
print_r($output, TRUE);
```

Une url pour connaître tous les critères de recherche, ainsi que les tags contenus dans la réponse, est également disponible à l'adresse suivante : <http://developer.yahoo.com/search/web/V1/webSearch.html>.

Il est par ailleurs tout à fait possible d'utiliser l'API conçue par Yahoo! (le code est à télécharger à l'adresse <http://www.sitepoint.com/examples/yahooapi/YahooAPIarchive.zip>). On peut ainsi faire des recherches avec l'API comme suit :

```
$api = new YahooAPI();
$api->setAppID('app_id');
$api->setService('http://
api.search.yahoo.com
/WebSearchService/V1/webSearch');
$api->setParam('output', 'php');
$api->setParam('query', 'SitePoint');
$api->setParam('results', '2');
$output = $api->doAPICall();
```

ou créer une classe qui interroge l'API :

- fichier `Exemple.class.php` (Listing 4).

Ceci fait, il ne reste plus qu'à créer un Mashup qui interroge tous les portails ensemble. Quelques liens pour en savoir plus : voir l'encadré sur Internet.

## ALEXANDRE RONDEPIERRE, VINCENT VEAUCLIN, CYRIL LALLOUM

Vincent Veauclin et Alexandre Rondepierre, respectivement responsable éditorial et responsable développement logiciel chez ipercast, travaillent dans le monde de l'Internet depuis 10 ans. Quant à Cyril Lalloum, il est à l'origine du réseau d'ipercast. Vous pouvez les contacter à cette adresse : [info@ipercast.net](mailto:info@ipercast.net)

# Générer un document PDF avec PHP

L'usage du format PDF est généralement judicieux pour permettre à vos visiteurs d'obtenir une version hors-ligne de votre site internet.

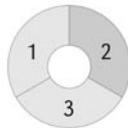
## Cet article explique :

- Les différentes solutions permettant de produire un document PDF avec PHP.
- Le fonctionnement de la classe FPDF dans un cas concret.

## Ce qu'il faut savoir :

- Avoir une idée des usages d'un fichier PDF.
- Connaître les bases du modèle objet de PHP4 et PHP5.

## Niveau de difficulté



Le format PDF est un format développé et supporté par la société *Adobe System*, sa utilisation est devenue particulièrement courante depuis plusieurs années et le taux de pénétration des logiciels permettant de les lire est particulièrement élevé. Sa version 1.7 est devenue un standard ISO depuis décembre 2008, et sa norme a été publiée le premier juillet. Le format PDF pour *Portable Document Format* est particulièrement utile dans la mesure où il conserve à l'identique sa mise en forme quels que soient le logiciel utilisé et sa destination. Il est par ailleurs tout à fait adapté pour un document qui pourra être imprimé. Le format PDF respecte la mise en page, les éléments graphiques (les images et les couleurs, par exemple) mais aussi la police de caractère, même si celle-ci n'est pas installée sur le système du lecteur. Il faut tout de même noter que même si ce format est ouvert et sa documentation disponible gratuitement, il reste un format propriétaire. Par ailleurs, créer un document PDF en travaillant directement avec sa syntaxe se révèle être une tentative périlleuse et globalement une perte de temps face aux fonctionnalités proposées par les outils disponibles avec PHP.

## Une variété d'outils pour PHP

Il existe plusieurs librairies et classes ayant pour but de manipuler des fichiers PDF pour

PHP, ayant une approche du format variable et exploitant des fonctionnalités différentes. Étant des extensions natives de PHP, les librairies sont globalement plus performantes, mais elles nécessitent une installation manuelle et donc la possibilité d'administrer le serveur hôte. La plupart des classes ont moins de pré-requis, si ce n'est la version de PHP. Si la portabilité est un critère de taille, ces classes deviennent de fait les seules solutions abordables.

Voici un échantillon des principales solutions disponibles.

- La librairie `PDFlib`, peut-être installée comme module complémentaire à PHP, elle est utilisable gratuitement selon la documentation PHP, mais un usage commercial nécessite l'acquisition d'une licence. C'est une librairie très complète qui permet de créer des documents originaux ou de travailler sur des documents existants. C'est également la librairie la mieux mise à jour. Vous pouvez vous renseigner sur le site de l'éditeur : <http://www.pdflib.com>,
- La bibliothèque `eZPDF`, composée de deux classes et quelques fichiers complémentaires permettant de gérer les polices. Elle est éditée par R&OS, qui permet son exploitation gratuite sans limite d'utilisation. La première classe `PDF` est utilisée pour générer les documents, tandis que la classe `eZPDF` offre quelques fonctionnalités complémentaires, comme la numérotation automatique des pages. Vous

pouvez télécharger `eZPDF` sur le site <http://www.ros.co.nz/pdf/>,

- FPDF est une bibliothèque composée d'une seule classe du nom du projet. Des fichiers complémentaires permettant de gérer les polices sont également nécessaires. Ce projet est un *freeware* que vous pouvez utiliser, modifier et redistribuer librement. Notre exemple utilisera FPDF, notre choix sera détaillé dans une prochaine section de cet article. Nous vous invitons à télécharger FPDF sur le site internet <http://www.fpdf.org> pour suivre cet article.

La dernière solution consisterait à générer le document PDF directement à partir de sa syntaxe, ce qui se révélerait rapidement complexe à mettre en œuvre, comme nous l'avons dit précédemment. Cette dernière alternative offre malgré tout un avantage majeur : une économie de ressources qui pourrait s'avérer non négligeable pour certains projets.

## Le projet FPDF

Le projet FPDF est un projet conçu en PHP4. Il n'avait pas été mis à jour depuis plusieurs années, mais le développeur distribue depuis le 3 août une nouvelle version (1.6). Le projet ne traite pas les dernières améliorations du format PDF, mais reste très efficace pour un usage standard. La classe `FPDF` ne nécessite par ailleurs aucun module particulier et peut être utilisée sur une grande majorité de serveurs PHP. Elle reste l'une des classes les plus utilisées, qui bénéficie d'une documentation conséquente et accessible. Notre choix s'est porté sur cette librairie pour ces raisons, ainsi que sa simplicité d'utilisation.

FPDF propose un lot de fonctionnalités comme le choix du format des pages et des unités, les sauts automatiques de page, la justification, l'import d'images *GIF*, *JPG*

et PNG, les couleurs et les liens hypertextes. Si la librairie *zlib* est installée, il est alors possible de compresser les fichiers. Le projet est également livré avec un script PHP permettant de convertir des fichiers de police TrueType et Type1 pour les utiliser dans vos PDF.

L'Unicode n'est pas supporté par cette classe avant la version 1.6, mais il est possible de produire des documents avec des jeux de caractères comme le cyrillique, le balte et le grec, ou encore le coréen, le chinois et le japonais. Le site internet du projet donne plus de détails sur ces fonctionnalités.

L'installation de FPDF est très sommaire : copiez la classe et le sous répertoire *font* dans le répertoire de vos sources, si vous souhaitez localiser le dossier contenant les polices ailleurs sur votre serveur, il suffira de déclarer une constante ainsi : `define('FPDF_FONTPATH', 'localisation/du/dossier');`. Vous pouvez placer cette ligne au début du fichier *fpdf.php*.

## Concevoir une facture

Pour avoir un bon aperçu des possibilités qu'offre la bibliothèque FPDF, nous allons produire une facture. La manière la plus simple est de générer le document à la volée. Il s'avère cependant plus intéressant d'utiliser le paradigme objet pour éditer un modèle de document réutilisable, et ainsi encapsuler les instructions propres à FPDF.

Nous allons donc créer une classe PHP appelée *FacturePDF*, héritant de la classe *FPDF*. Elle contiendra les propriétés suivantes :

- *Devise* : une chaîne de caractères contenant la monnaie utilisée,
- *Destinataire* : un tableau contenant l'adresse du destinataire de la facture,
- *Articles* : un tableau contenant l'intitulé, la quantité et le prix des articles,
- *Taxe* : un flottant contenant le taux des taxes à appliquer,
- *Total* : un flottant contenant le montant total de la facture.

Les méthodes de notre classe seront les suivantes :

- Le constructeur `__construct()` : est public, il servira à initialiser les principales propriétés de notre objet et à appeler le constructeur parent (de la classe *FPDF*).
- `AjouterArticle()` : est une méthode publique qui servira, comme son nom l'indique, à ajouter un article à notre facture,
- `Generer()` : est une méthode publique qui éditera le document PDF,
- `Header()` : est une méthode publique, elle existe déjà dans la classe *FPDF* mais elle est vide. Surchargée, elle sera appelée

à chaque début de nouvelle page pour constituer l'en-tête,

- `Footer()` : est l'équivalent de `Header()` pour le pied de page, c'est également une méthode publique,
- `Separeateur()` : cette méthode permettra d'afficher une ligne horizontale servant de séparateur,
- `LigneDestinataire()` : affichera l'adresse du destinataire de la facture,
- `EnteteArticles()` : affichera l'en-tête du tableau de la facture,
- `LigneArticle()` : affichera la ligne d'un article (sa référence, son prix, sa quantité),
- `LignesTotal()` : affichera le sous-total, les taxes et le total.

Les méthodes `Separeateur()`, `igneDestinataire()`, `EnteteArticles()`, `LigneArticle()` et `ignesTotal()` sont également protégées, elles ne sont appelées que par la méthode `Generer()` et servent à éditer les différents éléments du document. Nous allons en pre-

mier préparer notre classe *FacturePDF* héritant de *FPDF*, il convient donc d'importer cette dernière avant tout. Le Listing 1 présente le premier prototype de la classe. Le constructeur de *FacturePDF* recevra les premiers paramètres renseignant `$Destinataire`, `$Taxe` et `$Devise`. Il appellera ensuite le constructeur du parent *FPDF* avant de définir les en-têtes du document.

Le constructeur de *FPDF* admet trois paramètres optionnels. Le premier concerne l'orientation du document, il prend les valeurs `P` (correspondant à l'orientation portrait) ou `L` (paysage, ou *Landscape*). Le second permet de définir l'unité de travail, par défaut le millimètre *mm*, les autres valeurs possibles étant le centimètre *cm*, le point *pt* ou le pouce *in*. Le dernier paramètre est le format de page par défaut. Vous pouvez travailler avec un format standard comme *A3*, *A4*, *A5*, *Letter* ou *Legal* ou votre propre format défini par un tableau (largeur, hauteur) dans l'unité de travail actuel.

Listing 1. Prototype de la classe *FacturePDF*

```
<?php
require('fpdf.php');
class FacturePDF extends FPDF {
    private $Devise = '';
    private $Destinataire = array();
    private $Articles = array();
    private $Taxe = 0.0;
    private $Total = 0.0;
    // Méthodes de la classe
}
?>
```

Listing 2. Constructeur de la classe *FacturePDF*

```
public function __construct($ref, $destinataire, $adresse, $taxe, $devise) {
    # On enregistre les paramètres
    $this->Devise = $devise;
    $this->Taxe = $taxe;
    $this->Destinataire = array('ref' => $ref, 'nom' => $destinataire, 'adresse'
        => $adresse);
    # Création du document
    parent::FPDF('P', 'mm', 'A4');
    $this->SetAuthor('Service de facturation');
    $this->SetCreator('Generated By FPDF 1.6/PHP'.phpversion());
    $this->SetTitle('Facture n°'.$this->Destinataire['ref']);
    $this->SetSubject('Facture de '.$this->Destinataire['nom']);
    $this->SetKeywords('facture, ma société, mes produits');
    $this->SetDisplayMode('real', 'continuous');
}

public function AjouterArticle($ref, $designation, $quantite, $prix_unitaire) {
    $this->Articles[] = array(
        'ref' => $ref,
        'designation' => $designation,
        'quantite' => $quantite,
        'prix_unitaire' => $prix_unitaire,
        'total' => $quantite*$prix_unitaire);
}
```

La classe `FPDF` est composée de méthodes permettant de gérer les en-têtes du document, comme l'auteur `SetAuthor()`, le créateur `SetCreator()`, le titre du document `SetTitle()`, son sujet `SetSubject()` et les mots clés associés au document `SetKeywords()`. Elles ne sont pas nécessaires mais permettent d'ajouter quelques informations complémentaires lues par les moteurs de recherche, par exemple. On appellera ces fonctions depuis le constructeur de notre classe. Chacune de ces méthodes admet deux paramètres, le premier étant la chaîne de caractères correspondant à l'information, le second, optionnel, permet de préciser si la chaîne est encodée en UTF-8. Par défaut, ce paramètre vaut `false`. La méthode `SetDisplayMode()` permet de définir la manière dont la page sera affichée par le lecteur lors de son ouverture : on définit le zoom par le premier paramètre et la disposition des pages grâce au second. La valeur du zoom peut être un entier correspondant au facteur de zoom, ou une chaîne de caractères valant `fullpage` pour afficher la page entière à l'écran, `fullwidth` pour que la largeur de la page corresponde à la largeur de l'écran, `real` correspond à un zoom de 100%. La disposition des pages est choisie par les valeurs

`single` (affichage d'une seule page à la fois), `continuous` (affichage continu d'une page à l'autre) ou `two` (affichage de deux pages sur deux colonnes). Pour les deux paramètres, `default` laisse le choix à la charge du logiciel. Le constructeur de la classe `FacturePDF`, accompagné de la méthode `AjouterArticle()` est présenté au Listing 2.

Nous allons maintenant étudier la méthode `Generer`. Nous commencerons par définir la marge du document à l'aide de la méthode `SetMargins()` qui prendra pour paramètres la largeur de la marge de gauche, la hauteur de la marge supérieure, et éventuellement la largeur de la marge de droite si celle-ci doit différer de celle de gauche.

Nous allons ensuite choisir la police utilisée dans notre document. Elle peut être modifiée à tout moment, sachant que la dernière police déclarée sera utilisée tant qu'elle n'a pas été redéfinie. On utilise alors la méthode `SetFont()` en lui indiquant la famille de la police, son style et sa taille indiquée en points comme dans la plupart des logiciels de traitement de texte.

Ces deux derniers paramètres sont optionnels. Pour définir le style des caractères, vous devrez utiliser les valeurs `B` pour *gras*, `I` pour *italique* et `U` pour *souligné*, il est également

possible de faire une combinaison de ces valeurs, `IB` produira du texte gras et italique. Il est possible d'utiliser toutes les polices de type TrueType ou Type1 dans la mesure où vous la transformez en un format exploitable par FPDF et l'appellez avec la méthode `AddFont()`. Cette procédure est détaillée dans le manuel aux adresses <http://www.fpdf.org/fr/tutorial/tuto7.htm> et <http://www.fpdf.org/fr/doc/addfont.htm>.

Pour gagner du temps, nous appelons la méthode `SetAutoPageBreak()` avec la valeur `true`, qui aura pour effet de déclencher un saut de page automatique dès que le pointeur arrivera en bas de page. Par défaut, le saut de page sera réalisé lorsque le pointeur atteindra la marge inférieure, fixée par défaut à deux centimètres.

Comme nous envisageons d'ajouter un pied de page cette marge sera redéfinie grâce au code `$this->SetAutoPageBreak(true, 30);` plaçant ainsi le retour automatique à la ligne. Par ailleurs, le retour n'est effectif qu'à la condition que la fonction `AcceptPageBreak()` retourne `true`.

Par défaut, elle retournera la valeur définie par `SetAutoPageBreak()`. Vous pouvez surcharger cette méthode dans une classe pour créer un document sur plusieurs colonnes par exemple. La documentation officielle donne plus de détails sur la page de la fonction `AcceptPageBreak()` à l'adresse <http://www.fpdf.org/fr/doc/acceptpagebreak.htm>.

Nous allons enfin créer la première page avec la méthode `AddPage()`. A l'appel de cette méthode, une nouvelle page est créée, les méthodes `Header()` et `Footer()` sont exécutées et le pointeur est placé au début de cette nouvelle page. FPDF travaille avec un pointeur, que l'on peut voir comme la position de la pointe du crayon qui dessine le document. Il existe deux méthodes appelées `GetX()` et `GetY()` qui retournent la position actuelle du pointeur. `X` étant l'abscisse (distance relative au bord gauche) du pointeur et `Y` son ordonnée (distance relative au bord supérieur).

Notre méthode `Generer()` peut maintenant afficher les différents éléments de page, en appelant successivement les méthodes `LigneDestinataire()`, `EnteteArticles()`, `LigneArticle()` et `LignesTotal()` que nous détaillerons ensuite. La dernière méthode appelée, `Close()`, termine le document et le rend prêt à être transmis. Cet appel n'est pas nécessaire, mais nous assure que le document ne pourra plus être modifié. Le code de `Generer()` est présenté au Listing 3.

D'autres méthodes permettent de travailler sur la mise en page du document. `Cell()` permet de créer une cellule comparable à un bloc en HTML. On lui attribue une largeur, une hauteur, un contenu et une éventuelle bordure. Il est également possible de renseigner d'autres paramètres comme la position du

### Listing 3. La méthode `Generer()`

```
public function Generer() {
    $this->SetMargins(15, 15);
    $this->SetFont('Arial', '', 10);
    $this->SetAutoPageBreak(true, 30);
    $this->AddPage();
    $this->LigneDestinataire();
    $this->EnteteArticles();
    for($i = 0; $i < sizeof
        ($this->Articles), $i++) {
        $this->LigneArticle($i);
        $this->Total += $this->
            Articles[$i]['total'];
    }
    $taxe = ($this->Taxe/100) *
        $this->Total;
    $this->LignesTotal($taxe);
    $this->Close();
}
```

### Listing 4. Les méthodes `Separateur()` et `LigneDestinataire()`

```
protected function Separateur() {
    $this->Ln(20);
}

protected function LigneDestinataire() {
    $this->SetXY(-100, 70);
    $this->MultiCell(85, 4, $this->Destinataire['nom']."\n".$this->Destinataire[
        'adresse']);
    $this->Separateur();
    $this->Cell(0, 5, 'Référence de la facture : '.$this->Destinataire['ref'], 1, 1);
}
```

# Misez sur vos compétences

On vous demande de mettre en place des applications extrêmement riches en termes de fonctionnalités et d'ergonomie. On vous demande aussi de maîtriser strictement vos coûts. Dans cette perspective, la complexité et la variété des technologies induisent un nouveau risque : celui des **compétences**.

**SEWATECH**, organisme de formation et de conseil, spécialiste des technologies objet, propose plusieurs formules. Pour vous conseiller dans vos choix au

démarrage de vos projets, nous mettons en place des **séminaires** de tours d'horizon ou de l'**assistance** en architecture, outillage et organisation. Pour accélérer la montée en compétences de vos équipes, nous proposons des **formations** et ateliers pratiques, sur programme standard ou sur mesure, ainsi que des **accompagnements** techniques et méthodologiques. Pour résoudre vos problèmes et reprendre le bon chemin, ces solutions peuvent être associés à un **audit** de projet ou une revue d'architecture.

## Langage Java

Initiation java avec Eclipse	4 jours
Approfondissement java	3 jours
Développement XML en java	2 jours
Services Web en Java	3 jours
Tests unitaires pour Java EE	2 jours
Techniques de refactoring avec Eclipse	1 jours

## Java EE et frameworks

Développement Web : JSP et servlet	3 jours
Framework MVC – JSF	3 jours
Framework MVC – Struts	3 jours
Web 2.0 avec AJAX et javascript	2 jours
EJB 2, les bases	3 jours
EJB 3, les bases	3 jours
Mapping O/R avec EJB 3 et JPA	3 jours
Initiation à Spring 2.0 / 2.5	3 jours
Framework Spring – compléments	2 jours
Mapping O/R avec Hibernate	3 jours
Développement avec les frameworks Struts, Spring et Hibernate	5 jours
Développement avec les frameworks JSF, Spring et Hibernate	5 jours

## Autres langage

Langage PHP 5	3 jours
/ntégration PHP 5	3 jours
Initiation à .NET et C#	4 jours
Langage C	4 jours
Langage ANSI/ISO C++	4 jours
Langage Python	4 jours
Langage PERL 5	3 jours
Approfondissement PERL 5	2 jours
Initiation à Ruby on Rails	3 jours

## Conception Objet

Programmation orientée objet	2 jours
Principes et conception Objet	2 jours
Architectures Objet	1 jours
Conception et Design Patterns	2 jours
Bonnes Pratiques Java EE	3 jours
Design Patterns pour Java EE	3 jours

## Système Linux

Utilisation du système Linux	5 jours
Administration du système Linux	5 jours
Sécurité sous Linux	3 jours

## Tours d'horizon (séminaires)

Java, Java EE : tour d'horizon	1 jours
Sécurité des applications Java EE	1 jours
Java et l'Open Source	1 jours
Serveurs d'applications Java	1 jours
.NET : tour d'horizon	1 jours
Comparaison Java - .Net	1 jours
Architectures Web	1 jours
Architectures SOA	2 jours
Outils pour le SOA	2 jours

## Modélisation et méthodologie

UML 2, analyse et conception	4 jours
UML pour chefs de projets	2 jours
UML pour développeurs	2 jours
Gestion de projet Objet	2 jours
Initiation au Unified Process	1 jours
Méthodes agiles : XP et SCRUM	1 jours
Mise en oeuvre de l'eXtreme Programing	2 jours

## Serveurs d'applications Java

Administration JBoss 4	3 jours
Administration Apache Tomcat	3 jours

pointeur après la cellule ou l'alignement du texte à l'intérieur.

La méthode `MultiCell()` est identique, sauf dans sa gestion des retours à la ligne. En effet, `Cell()` affichera le texte sur une ligne, même après avoir atteint les limites de la cellule, tandis que `MultiCell()` ajoutera une nouvelle cellule identique en dessous pour simuler un retour en fin de ligne et prendra en charge le caractère `\n` (retour chariot).

Vous pouvez maintenant déplacer le pointeur grâce à la méthode `Ln()`, qui génère un saut de ligne. Sans paramètre, le pointeur est ramené au bord gauche du document en dessous de la dernière cellule créée, sinon, vous pouvez renseigner la hauteur du saut de ligne.

Les méthodes `SetX()` et `SetY()` et `SetXY()` permettent de modifier la position du pointeur plus précisément, en court-circuitant le flot normal. En définissant des valeurs négatives, la position sera alors relative au bord inférieur pour `SetY()` et droit pour `SetX()`. La présence ou non d'une marge n'aura pas d'influence sur ces deux méthodes.

La méthode `LigneDestinataire()` imprimera une cellule avec `MultiCell()`, placée de manière fixe à 7 centimètres du bord supérieur et 10 centimètres du bord droit, pour pouvoir placer l'adresse dans une enveloppe standard. La largeur de la cellule correspond à la largeur de page restante, marge comprise, soit 85 millimètres. La hauteur sera de 4 millimètres (ce qui correspond à un peu plus de 10 points). Le code de la méthode est présenté au Listing 4. Nous afficherons par la même occasion la référence de la facture, après un retour à la ligne de 2 centimètres. L'appel à la fonction `Cell()` admet ici deux paramètres optionnels après le texte. Le paramètre bordure, valant 1. Il est possible de définir quelles bordures afficher avec les valeurs `L` (*left*, gauche), `R` (*right*, droite), `T` (*top*, haut) ou `B` (*bottom*, bas) ou une combinaison des quatre. Le deuxième paramètre concerne la gestion du pointeur après la cellule. La valeur 0 place le pointeur en haut à droite de la cellule, 1 provoquera un retour à la ligne identique à celui de la méthode `Ln()`, enfin, la valeur 2 placera le pointeur au niveau du bord inférieur gauche de la cellule, pour faciliter les retours à la ligne si la coordonnée X du pointeur n'est pas celle de la marge de gauche.

Pour concevoir notre tableau d'articles, nous allons devoir placer des cellules les unes à côté des autres. Nous allons définir des largeurs de colonnes idéales. Il est également possible de définir la largeur relativement au contenu des cellules selon vos propres critères, car la méthode `GetStringWidth()` retourne la largeur d'une chaîne de caractères passée en paramètre dans la police courante.

Nous avons déjà étudié la plupart des instructions qui composent les méthodes `EnteteArticles()`, `LigneArticle()` et `LignesTotal()`.

Nous changeons la police une première fois pour mettre en valeur les en-têtes de colonnes, cette instruction sera à nouveau appelée pour revenir à la police de base de notre document une fois que l'en-tête a été générée.

Nous allons utiliser deux nouvelles instructions ; `SetLineWidth()` permet de définir l'épaisseur des traits à l'aide de son seul paramètre : l'épaisseur de la ligne définie dans l'unité de travail. Par défaut, la valeur est 0,2 millimètres. `SetFillColor()` est une méthode qui sert à définir la couleur de remplissage des cellules.

Elle peut prendre un paramètre correspondant à la couleur en niveau de gris de 0 à 255 ou trois paramètres correspondant respectivement aux niveaux des couleurs rouge, vert et bleu. Par commodité et en lien avec les codes couleurs utilisés en HTML, vous pouvez utiliser la représentation hexadécimale des nombres (en ajoutant le préfixe `0x`).

Nous avons utilisé deux paramètres de la méthode `Cell()` que nous n'avions pas étudié plus tôt : après le paramètre définissant le comportement du pointeur après la cellule, nous avons précisé le paramètre précisant

**Listing 5.** Les méthodes `EnteteArticles()`, `LigneArticle()` et `LignesTotal()`

```
protected function EnteteArticles() {
    $this->Separateur();
    $this->SetFont('Arial', 'I', 10);
    $this->SetLineWidth(0.4);
    $this->SetFillColor(0xDE);
    $this->Cell(30, 5, 'Référence', 'B', 0, 'C', true);
    $this->Cell(60, 5, 'Désignation', 'B', 0, 'L', true);
    $this->Cell(30, 5, 'Prix unitaire', 'B', 0, 'C', true);
    $this->Cell(30, 5, 'Quantité', 'B', 0, 'C', true);
    $this->Cell(30, 5, 'Total', 'B', 0, 'C', true);
    $this->Ln();
    $this->SetFont('Arial', '', 10);
    $this->SetLineWidth(0.2);
    $this->SetFillColor(0xFF);
}

protected function LigneArticle($i) {
    if($i%2 == 0)
        $this->SetFillColor(0xFE);
    else
        $this->SetFillColor(0xFF);
    $this->Cell(30, 5, $this->Articles[$i]['ref'], 0, 0, 'C', true);
    $this->Cell(60, 5, $this->Articles[$i]['designation'], 0, 0, 'L', true);
    $this->Cell(30, 5, $this->Articles[$i]['prix_unitaire'].' '.$this->Deviser, 0, 0, 'C', true);
    $this->Cell(30, 5, $this->Articles[$i]['quantite'], 0, 0, 'C', true);
    $this->Cell(30, 5, $this->Articles[$i]['total'].' '.$this->Deviser, 0, 0, 'C', true);
    $this->Ln();
}

protected function LignesTotal($taxe) {
    $this->Cell(30, 5, 'T', 0, 'C', true);
    $this->Cell(120, 5, 'Total HT', 'T', 0, 'L', true);
    $this->Cell(30, 5, $this->Total.' '.$this->Deviser, 'T', 0, 'C', true);
    $this->Ln();
    $this->Cell(30, 5, ' ', 0, 0, 'C', true);
    $this->Cell(120, 5, 'Taxes TVA '.$this->Taxes.' %', 0, 0, 'L', true);
    $this->Cell(30, 5, $taxe.' '.$this->Deviser, 0, 0, 'C', true);
    $this->Ln();
    $this->Cell(30, 5, 'T', 0, 'C', true);
    $this->Cell(120, 5, 'Total TTC', 'T', 0, 'L', true);
    $this->SetFont('Arial', 'B', 10);
    $this->Cell(30, 5, ($this->Total+$taxe).' '.$this->Deviser, 'T', 0, 'C', true);
    $this->SetFont('Arial', '', 10);
    $this->Ln();
}
```

**Listing 6. Les méthodes Header() et Footer()**

```
public function Footer() {
    $this->SetY(-30);
    $this->Cell(0, 15, 'Ma société - www.masociete.com', 0, 0, 'C',
        false, 'http://www.masociete.com');
}

public function Header() {
    $this->Image('/monlogo.png', $this->GetX(), $this->
        GetY(), 0, 0, 'PNG', 'http://www.masociete.com');
}
```

**Sur Internet**

- <http://www.fpdf.org/> – Site officiel du projet FPDF, il contient plusieurs tutoriaux pratiques et une documentation bien fournie, le tout en français,
- <http://www.abracadabrapdf.net/> – Site de ressources francophones sur le format PDF et la suite logicielle Acrobat développée par Adobe System.

l'alignement du texte. Il prend l'une des trois valeurs L (*left*, aligné à droite), C (*center*, centré) ou R (*right*, aligné à gauche). Le dernier paramètre sert à ajouter (*true*) ou non (*false*) un fond coloré à la cellule, si celui-ci a été défini plus tôt grâce à `SetFillColor()`. Les méthodes `EnteteArticles()`, `LigneArticle()` et `LignesTotal()` sont présentées au Listing 5.

Nous allons à présent gérer le pied de page, et découvrir le dernier paramètre de la fonction `Cell()`, qui permet d'ajouter un lien hypertexte. Le prototype des fonctions `Cell()` et `MultiCell()` complet est alors : `Cell($largeur, $hauteur, $texte, $bordure, $pointeur, $alignement, $remplissage, $lien)`. Il est possible de définir des liens internes ou externes. Le paramètre `$lien` peut être une chaîne de caractères contenant une URL ou un descripteur

de lien interne, retourné par la méthode `AddLink()`. La méthode `SetLink()` servira à placer l'ancre du lien interne dans la page. Reportez vous à la documentation de FPDF pour plus de détails.

Nous pouvons finir par travailler sur l'entête qui devra accueillir un logo de notre société, donc une image. FPDF permet d'importer des images aux formats *JPG*, *PNG* et *GIF* sous certaines conditions, notamment la présence de l'extension *GD* dans certains cas ; n'hésitez pas à consulter la documentation en cas de problème. L'insertion d'une image se fait à l'aide de la méthode `Image()`. Le premier paramètre correspond au lien ou à l'URL de l'image, il est en effet possible d'importer une image qui n'est pas disponible sur le serveur local, dans les conditions imposées par la configuration de PHP.

Les autres paramètres sont optionnels. Les deuxièmes et troisièmes paramètres précisent la position de l'image, par défaut, la position courante du pointeur est utilisée. Il est ensuite possible de modifier la largeur et la hauteur de l'image dans l'unité du document grâce aux deux paramètres suivants. Si ces paramètres valent 0, alors les dimensions sont calculées automatiquement. Le sixième paramètre vous permet d'indiquer le type d'image à importer, il est recommandé de l'utiliser, sinon, FPDF se base sur l'extension du fichier. Le dernier paramètre permet enfin d'ajouter un lien, de la même manière que pour les cellules. Le Listing 6 présente les méthodes `Header()` et `Footer()`.

Il existe d'autres méthodes pratiques permettant d'imprimer du texte en ligne (ou *mode flot*), vous pourrez alors utiliser différentes polices sur une même ligne, sans travailler avec les cellules.

Le projet FPDF possède par ailleurs plusieurs extensions développées par des contributeurs, telles que UFPDF ou FPDFI qui permettent respectivement de travailler avec un jeu de caractères Unicode et travailler sur la base d'un document existant.

Le forum du projet FPDF est régulièrement consulté et s'avère être une aide efficace pour découvrir les subtilités de cette bibliothèque, n'hésitez pas à consulter cette mine d'informations utiles.

**MARTIN RICHARD**

*Jeune développeur autodidacte, l'auteur a déjà une expérience solide dans la conception de sites et d'applications pour internet. Étudiant en informatique et développeur web indépendant, il publie depuis plusieurs années des cours de conception web sur son site internet [www.martiusweb.net](http://www.martiusweb.net).*

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 !**

Intégrez notre équipe **jeune et dynamique** présente sur Rennes pour le développement d'applications Internet/intranet/extranet.

[recrutement@cognix-systems.com](mailto:recrutement@cognix-systems.com)
**02 99 27 75 92**

Plus d'informations sur  
[www.cognix-systems.com](http://www.cognix-systems.com)



# Les erreurs de référencement

Les années ayant vu le Web s'ouvrir et s'étirer tel une fleur au soleil, le nombre de sites Internet présents sur celui-ci à cru de façon exponentielle.

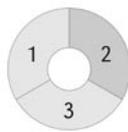
## Cet article explique...

- Cet article explique les erreurs classiques à ne pas faire pour le référencement de votre site.

## Ce qu'il faut savoir...

- Vous devez au moins connaître l'(X)HTML (pour un développement correct), et les bases du PHP. Une connaissance basique de Zend Framework et son fonctionnement en MVC vous sera utile.

## Niveau de difficulté



Dans un premier temps, vous allez voir, les abus passés, ayant conduit les moteurs de recherche à être plus sévères dans leurs critères, puis vous verrez les erreurs courantes à ne plus faire d'aujourd'hui, que ce soit syntaxiquement, ou dans la façon de penser les choses, et, conjointement à cette partie, nous verrons ce que le Zend Framework peut nous apporter grâce à sa mise en œuvre MVC.

## Les abus sur les moteurs de recherche

Les années passant, les moteurs de recherche apparaissent, de plus en plus nombreux. Et hélas, des abus apparaissent. Dans cette partie, nous allons voir les premiers abus effectués, puis des abus plus récents, ayant contraint les moteurs de recherche à être plus stricts dans leur analyse des sites Internet.

### Abus passés

Les premiers grands moteurs de recherche utilisaient les *Méta-tags*, ces petites balises placées dans la partie `HEAD` de votre document (X)HTML. Une contenant les mots-clés pouvant être indexés par les moteurs de recherche, et une autre contenant la description de votre site de-

vant être affichée sur le moteur de recherche. Les balises meta ont beaucoup d'autres utilisations, mais elles sortent du cadre de cet article.

Les premiers abus, très simples à réaliser, consistaient à mettre n'importe quoi dans les balises meta contenant les mots clés, afin d'apparaître plus facilement. On pouvait ainsi se faire référencer en utilisant des mots clés pour apparaître là où l'on ne devait pas théoriquement. Cas d'exemple : Un site parle d'un sujet *A*, mais ses meta-tags de référencement contiennent des mots parlant du sujet *A*, mais aussi d'un sujet *B*, qui n'a rien à voir avec *A*. Ainsi, en faisant des recherches sur le sujet *A*, le site pouvait apparaître, mais aussi si l'on faisait des recherches sur le sujet *B*, bien que le site n'en parle pas !

Pour contrer cela, les moteurs de recherche se sont davantage focalisés sur le contenu des pages.

### Abus récents

Google, le plus célèbre moteur de recherche actuel, faisant même parfois partie de polémiques. Google s'est développé et a imposé un drôle de climat. À cette époque, beaucoup d'entreprises ne pensaient qu'à faire des portails de services, comme *multimania* ou *caramail* par exemple. Peu de gens pensaient ou voulaient développer de gros projets de moteurs de recherche.

Google s'est développé dans cette optique, en proposant un moteur de recherche qui se voulait le plus rapide à charger, surtout au niveau de sa page d'accueil, mais aussi en étant un moteur de recherche qui ne se laisserait pas dépasser par les abus de certains webmasters peu scrupuleux.

Pourtant, est apparu l'abus du *Google bombing*. L'algorithme de référencement de Google fonctionnait en partie sur les liens. Hors, dans un lien, nous avons deux choses qui se distinguent (hors CSS et Javascript) :

- L'URL, c'est-à-dire la destination,
- Le texte de votre lien.

Ainsi, si vous faisiez un lien vers un autre site, le moteur de Google comptait cela comme un vote vers un autre site, et utilisait le texte comme étant le sujet du vote. Un exemple de *Google bombing* qui eut lieu en France, fut vers 2002 la quantité de liens pointant vers le site de l'Elysée, avec comme texte magouilleur intouchable. Si vous tapiez magouilleur intouchable dans le moteur de recherche Google, vous obteniez ainsi le site de l'Elysée comme premier résultat de la première page. Un autre plus connu, fut l'apparition du site de la maison blanche (toujours vers 2002) lorsque vous tapiez miserable failure (échec épouvantable). Tout cela a aujourd'hui été corrigé, et, de plus, les algorithmes de classements de Google ont été fortement durcis, et continuent de l'être, car l'algorithme de recherche est en développement continu.

## Les erreurs très lourdes pour votre site

Je vais commencer avec deux erreurs, parfois connues, parfois hélas conseillées, à tort. Ne faites jamais cela, votre site se retrouverait banni de l'indexation Google, et ce, définitivement. Vous n'auriez plus qu'à acheter un autre nom de domaine.

### L'erreur ultime

#### à ne (plus) jamais faire : le cloacking

Abus moins connu que ceux précédemment cités, mais ayant forcé Google à durcir ses al-

algorithmes. Le principe est de cacher à Google le vrai contenu de votre site, pour lui mettre à lui (et uniquement lui) un autre contenu. Imaginez, vous faites votre site, avec votre contenu. Mais, vous mettez en place une routine pour faire apparaître une autre page lorsqu'un bot de référencement de Google passe sur votre site. L'algorithme est ici très simple : l'adresse IP des bots de référencement de Google est trouvable partout (même en utilisant Google). Il vous suffirait ici de vérifier l'adresse IP de qui consulte cette page, la comparer avec celle de Google, et le cas échéant, rediriger vers une autre page. Le Listing 1 illustre ce cas d'exemple.

### Bonne et mauvaise pratique de l'échange de liens

La popularité de votre site est un élément important pour Google. En effet, si vous faites beaucoup de liens vers d'autres sites, et que ces sites font des liens vers votre site, Google analysera votre site comme important, donc vous serez mieux classé(e). N'hésitez pas à faire ces liens sur plusieurs pages. Un des abus (et donc aujourd'hui une erreur à ne plus faire) qui a longtemps été fait, fut de faire des pages avec des milliers et des milliers de liens. Ceci était parfois couplé avec le *cloacking*.

### Les problèmes d'erreur de syntaxe

Pour se faire correctement comprendre sur le web, c'est comme pour se faire comprendre dans la vie, il faut parler un langage correct, intelligible de tous. Pour votre site Internet, c'est pareil, votre code doit éviter les erreurs pour être compris par les moteurs de recherche.

### Le non-respect des normes du web

On ne le dira jamais assez : le respect strict des normes du W3C est un élément on ne peut plus important pour votre site, sur ces deux niveaux :

- Pour que le navigateur comprenne comment votre site doit être affiché,
- Pour que votre référencement soit le meilleur possible.

En effet, un document correct et respectueux dans sa syntaxe (on ne parle pas ici du contenu) sera toujours mieux référencé. En effet, il sera toujours plus simple pour un moteur de recherche quel qu'il soit de différencier ainsi un titre d'un paragraphe, et comprendre quelle est votre structure de page. Comprenez par là qu'il est toujours meilleur de respecter les normes du W3C, plutôt que d'utiliser des segments propriétaires, comme `EMBED`, préférez plutôt la balise `OBJECT`. Et oubliez la balise `MARQUEE`.

### Oublis de balises et d'attributs de balise

Accessibilité : un autre mot très important. Une erreur souvent faite par certains développeurs est de négliger certains attributs de

balises pourtant recommandés par le W3C. En effet, lorsque vous insérez une image dans votre site, n'oubliez pas l'attribut `alt`, qui vous per-

met d'afficher un texte, si votre image ne peut se charger, notamment sur certains navigateurs pour personnes dont la visibilité est réduite.

#### Listing 1. Exemple de cloacking

```
<?php // LE PARFAIT EXEMPLE A NE PAS SUIVRE:
// Ici, nous déclarons un tableau avec les IPs de Google :
$array_ips = array('66.249.64.28','66.249.64.37');
/* On va parcourir le tableau, pour vérifier si l'adresse
   IP du client est celle d'un bot Google.*/
if(in_array($_SERVER['REMOTE_ADDR'], $array_ips) === TRUE) {
    // C'est un bot Google, on redirige.
    header('Status: 301 Moved Permanently');
    header('Location: pour_google.php');
    exit(); } // Si on est ici, c'est qu'on fait face au visiteur, on continue ...
```

#### Listing 2. Un exemple de sitemap

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.google.com/schemas/sitemap/0.84"><url>
  <loc>http://localhost/~bertrand/articles/erreurs_referencement/cloacking.php
  </loc>
  <lastmod>2008-05-12</lastmod>
  <changefreq>monthly</changefreq>
  <priority>0.8</priority></url>
  <url><loc>http://localhost/~bertrand/articles/erreurs_referencement/cloacking_
    error.php</loc>
  <lastmod>2008-05-19</lastmod>
  <changefreq>monthly</changefreq>
  <priority>0.5</priority></url>
</urlset>
```

#### Listing 3. Vérification des adresses entrées dans la sitemap

```
<?php // Ouverture du sitemap:
$map = simplexml_load_file('sitemap.xml');
// On parcourt tous les noeuds "url":
foreach($map->url as $node) {
    /* $node est un objet, ayant pour attribut tous les noeuds contenus dans le noeud
       url. Ici, on va utiliser la fonction fopen() pour vérifier que l'url est correcte.
       J'ai placé un @ pour éviter les messages d'erreurs quand le fichier n'existe pas */
    if(@fopen($node->loc, 'r') !== FALSE) {
        // Le fichier existe, on le signale.
        echo "L'url ' . $node->loc . ' existe<br />";
    } else { // Le fichier n'existe pas, on le met en rouge, c'est plus voyant.
        echo "<span style='color:#FF0000;'>Erreur: l'url ' . $node->loc . ' n'existe
           pas.</span><br />";
    }
}
```

#### Listing 4. Bootstrap de la galerie photos

```
<?php
// On indique qu'on veut afficher les erreurs :
error_reporting(E_ALL|E_STRICT);
ini_set('display_errors', 'On');
// On met le répertoire du framework dans l'include_path
ini_set('include_path', ini_get('include_path') . PATH_SEPARATOR . '../library');
// On utilise la classe Zend_Loader pour éviter l'autoload :
include_once('Zend/Loader.php');
Zend_Loader::registerAutoload(); // Chargement du contrôleur :
$controller = Zend_Controller_Front::getInstance(); // Affichage des erreurs :
$controller->throwExceptions(true); // On renseigne le répertoire des contrôleurs :
$controller->setControllerDirectory('../application/controllers'); // Affichage :
$controller->dispatch();
```

**Listing 5. Contrôleur de la galerie photos**

```
<?php
class Galerie_photosController extends Zend_Controller_Action {
    // Galerie photo pour Strasbourg :
    public function strasbourgAction() { // Création de la galerie
        $galerie = new galerie(); // On indique sur quelle ville on travaille :
        $galerie->ville = 'Strasbourg';
        /* Autres traitements possibles pour votre template [...] */
    }
}
```

Cela permet, en plus de ne pas négliger une partie de vos visiteurs, de stipuler au moteur de recherche à quoi correspond une image, ce qui va encore vous aider dans votre référencement.

Par contre, ne faites pas l'erreur de mettre une trop longue description dans le `alt` pour espérer y gagner. Cela jouerait contre vous, puisque les moteurs de recherche délaisseraient cette partie. En bref, ne mettez que quelques mots dans cet attribut. C'est largement suffisant pour aider au maximum votre indexation.

**Google sitemap incohérent**

Google a mis en place un outil formidable pour l'aider dans votre référencement : le *sitemap*. Il s'agit d'un fichier XML contenant la structure de votre site. Du moins, vous saisissez (en respectant la norme XML et la syntaxe Google) quelles sont les pages les plus importantes à vos yeux et qui doivent être surveillées. L'intérêt est double :

- Vous indiquez à Google quelles sont les pages les plus intéressantes de votre site,
- En plus, dans votre fichier XML, vous pouvez lui indiquer en direct quelle est la date de modification de chaque fichier, en plus d'une priorité sur les fichiers à surveiller et à afficher.

Deux erreurs sont à éviter absolument :

- Indiquer dans le *sitemap* des fichiers obsolètes où n'étant plus sur le serveur,
- Ne pas mettre le *sitemap* à jour en fonction des modifications de vos fichiers.

Si vous faites l'une de ces erreurs, votre *sitemap* devient alors inutile... Voici un cas d'exemple pratique. Les URLs sont celles de mon serveur local. Voici tout d'abord un *sitemap*, dans le Listing 2. Pour la syntaxe du fichier, je vous renvoie à la documentation Google, cet article n'est pas là pour vous expliquer comment développer un *sitemap*. Vous remarquerez que j'y ai entré deux URLs :

- [http://localhost/~bertrand/articles/erreurs\\_referencement/cloacking.php](http://localhost/~bertrand/articles/erreurs_referencement/cloacking.php),
- [http://localhost/~bertrand/articles/erreurs\\_referencement/cloacking\\_error.php](http://localhost/~bertrand/articles/erreurs_referencement/cloacking_error.php).

La première est bonne, la seconde n'existe pas. Que faire face à ça ? On va utiliser PHP ! Ce

langage va nous dire si les URLs sont correctes ! Tout est dans le Listing 3.

**Cacher des éléments : il y a la façon de le faire !**

Respect des normes classées strictes du W3C oblige, le CSS *inline* est à proscrire. Ainsi, évitez les balises du genre `div style="display:none;"`, la plupart des moteurs de recherche estiment que vous voulez leur cacher des choses, et pourraient l'assimiler à un ersatz de *cloacking*. Pour faire les choses bien, faites votre balise comme par exemple ci-dessous :

```
<div class="noDisplay"> Lorem ipsum dolor
sit amet</div>
```

Puis appliquez la règle CSS du :

```
noDisplay {display:none;}
```

**Les problèmes de contenu**

Voilà, vos fondations sont solides, votre code est propre, respectueux des standards du W3C, et vos URLs sont lisibles proprement. Il nous reste à aborder le dernier point, que vous ne devez surtout pas négliger : entrant en compte pour les moteurs de recherches, mais aussi vos visiteurs : le contenu.

**L'oubli de mises à jour**

Une autre erreur qui à la longue sera presque fatale. Celle-ci ne vous fait pas *blacklister* (loin de là), mais si votre site n'est pas mis à jour régulièrement, votre positionnement va dégringoler. Votre site est visité par Google tous les X temps. Si votre site est régulièrement mis à jour, Google va le visiter plus souvent, et ainsi, toutes vos mises à jour seront prises en compte et classées plus rapidement. En revanche, si vous mettez à jour votre site une fois tous les 36 mois, Google visitera votre site moins souvent. Cela aura pour effet :

- Que vos mises à jour seront plus longues à être prises en compte par Google,
- Google privilégiera d'autres sites, qui seront mieux classés que le votre.

En bref, il vaut mieux avoir un site très dynamique. Bien sûr, il n'est pas nécessaire de modifier toutes les pages tous les jours (ça n'aurait pas de sens). Simplement, mettez des actualités sur vos

tre page d'accueil. Et faites-en de nouvelles assez régulièrement. Cela aura pour effet de montrer à Google que votre site change régulièrement, mais à vos visiteurs aussi !

**Les URLs incompréhensibles**

Un aspect simple. Vos URLs comptent dans votre référencement en tant que mots clés. Attention, les variables ne comptent pas. Ainsi, une URL du type : `index.php?page=mot1,mot2,mot3,...,motn`.

Ne servira à rien, les moteurs de recherche ne s'intéressent pas aux variables. Sur un cas réel, l'URL `page.php?p=gal&ville=48` ne vous apportera rien ! Mais une URL *propre*, comme par exemple `http://www.monsite.com/galerie_photos/Strasbourg` est davantage lisible, et utile pour un moteur de recherche. Pour l'installation et la mise en place du Zend Framework, je vous renvoie au numéro 2/2008 de PHP Solutions. Le Listing 4 présente le code de votre *bootstrap*. Le code du *bootstrap* est classique, et suit le schéma du *screen-cast*. Vous ne devriez pas avoir de problème. Si on suit notre exemple, il nous faut un contrôleur pour la galerie photo, et au moins une méthode, celle de l'action correspondant à Strasbourg. Attention, cet article ne couvre pas la création d'une galerie photo avec le Zend Framework, le code que vous lirez dans la méthode est une supposition de code que vous pourriez écrire, à titre de démo, et ayant pour but la compréhension. Le Listing 5 vous présentera le contrôleur de la galerie photos.

**Rester figé**

Une autre erreur, mais qui n'influe que très peu, voire pas du tout sur votre référencement, est de rester figé sur votre *look*. Les *look* de sites web sont comme les modes : ça va un temps, tout le monde (en partie) trouve ça joli, mais ça ne dure jamais. Pensez à vos visiteurs (c'est quand même à la base pour eux que vous faites un site), et pensez de temps en temps à améliorer le *look* de votre site, ce qui a toujours pour effet de lui redonner une nouvelle jeunesse, mais aussi de plaire à votre visiteur (sauf en cas de mauvais goût de votre part). Et du neuf dans votre *look* pourrait inciter vos visiteurs à faire des liens vers votre site depuis leur blog.

**Conclusion**

En conclusion, que retenir ? Le référencement ne s'invente pas, il s'agit d'une science, certes, mais si vous avez bien lu cet article, vous vous apercevrez qu'en évitant les magouilles, et en faisant preuve de bon sens, on peut aboutir à un référencement correct. Cependant, il faut garder à l'esprit que votre site doit constamment évoluer, tout comme votre référencement.

**BERTRAND ANDRES**

*Programmeur évoluant en entreprise à titre professionnel depuis plus d'un et demi. Parlant anglais et français, joignable sur msn à l'adresse : ethtezah@hotmail.fr*

# Améliorer son référencement : campagne de netlinking dans les annuaires

Le référencement d'un site dans les moteurs de recherche résulte principalement de 2 leviers : l'optimisation du contenu thématique des pages du site ET la mise en place d'une stratégie de netlinking. En d'autres mots, faire connaître votre site sur des sites proches de votre thématique. L'objectif d'une stratégie de netlinking est de placer un lien sur un site partenaires vers une page de site (communément appelé BackLink).

Nous ne traiterons ici pas de la question de l'optimisation du contenu et de la structuration des pages Internet. Mais du deuxième levier : la mise en place d'une stratégie de Netlinking et principalement de la soumission de votre site dans les annuaires.

Les annuaires regroupent des sites par thématique. Chaque webmaster peut soumettre son site afin qu'il soit intégré et présenté dans une catégorie de l'annuaire.

Par exemple, le site Ta-voiture.com spécialisé dans l'univers de l'automobile dispose d'un annuaire qui rassemble par ville l'ensemble des professionnels de l'auto (garage, casse, auto école...).

Le fait d'inscrire son site dans un annuaire procure, à votre site, un double avantage. Cela permet :

D'améliorer son référencement (classement dans les moteurs) en bénéficiant d'un back link de qualité (lien dit en dur) dans l'annuaire. Pour cela, il faut choisir des annuaires qui proposent des liens en dur c'est à dire pris en compte par les moteurs comme Google,

D'améliorer son trafic en bénéficiant de la visibilité déportée de l'annuaire et ainsi capter une part du trafic ciblé de l'annuaire. (Un internaute sélectionnera d'autant mieux votre site dans la liste de l'annuaire, que votre titre et votre descriptif correspondra à ses attentes).

La mise en place d'une campagne de soumission dans les annuaires est une tâche souvent longue et fastidieuse puisse qu'elle nécessite de trouver les annuaires qui traitent de votre thématique et de les sélectionner en fonction de critères de qualités (liens en dur, lien de retour...).

Yooda, l'éditeur de solutions professionnels pour le référencement, propose son logiciel de gestion de soumission manuelle assistée dans les annuaires : Yooda Submit.

Étudié spécialement pour les webmasters et les référenceurs, il permet d'assister le webmaster dans toutes les étapes d'une campagne de soumission dans les annuaires :

- 1) Gestions des informations à soumettre,
- 2) Recherche automatisée d'annuaires sur une thématique,
- 3) Prè-remplissage du formulaire de soumission des annuaires,
- 4) Gestion des réponses d'acceptation des annuaires,
- 5) Graphiques de suivi des inscriptions (tableau de bord).

## Yooda Submit se déroule en 4 étapes :

### La première étape : renseigner les informations qui seront utilisées pour pré-remplir les formulaires de soumission

Cette étape consiste à renseigner les informations demandées par un annuaire lors de la soumission de votre site. Vous aller devoir indiquer, dans le logiciel Yooda Submit, les informations générales : Nom, Prénom, E-mail, Identifiant, mot de passe et url de retour. Ces informations permettront à terme d'informer le webmaster de l'annuaire : Qui vous êtes, Où vous joindre, créer votre espace privé de gestion dans l'annuaire et indiquer l'URL de votre site afin de vérifier la présence d'un lien retour éventuel. Ensuite vous allez devoir renseigner les champs Titres et Descriptifs de votre site afin de décrire et de présenter votre site dans les annuaires.

Yooda Submit, gère l'enregistrement de plusieurs titres et descriptifs afin que vous puissiez présenter différemment votre site selon la thématique et la spécificité de l'annuaire. Ce qui vous permettra de décrire et de présenter votre site différemment en fonction de la catégorie et de l'annuaire choisis. Le fait d'avoir un descriptif approprié à la thématique permettra d'accroître votre taux d'acceptation dans les annuaires et mais également votre propre référencement dans les moteurs.

### La deuxième étape : recherche d'annuaires par Yooda Submit

Yooda Submit vous liste automatiquement les annuaires qui correspondent à votre site. Vous renseignez à l'aide de mots clés la thématique d'annuaire recherché, et Submit vous fournit une liste d'annuaires qualifiée.

Lors de cette étape, vous avez la possibilité de choisir le mode Expert qui va vous permettre de cibler des annuaires selon votre stratégie de netlinking. Vous aurez la possibilité de cibler des annuaires avec un Page Rank Minimum, le type de lien désiré (liens en dur, nofollow, Java, etc...), les annuaires qui demande un lien de retour (obligatoire ou pas) et la contribution financière demandée par l'annuaire (des annuaires peuvent vous demander un micro ou macro paiement ou aucun paiement).

### La troisième étape : la soumission assistée

Vous accédez directement à la page du formulaire de soumission de l'annuaire. Yooda Submit pré-rempli automatiquement le formulaire de soumission avec les informations renseignées à l'étape 1. Il ne vous restera plus qu'à remplir le captcha (code image) et valider votre soumission dans l'annuaire. Après cette étape, il vous suffit de retourner à la liste d'annuaire et de choisir un autre annuaire. Chaque annuaire soumis sera supprimé de la liste des annuaires proposés et transféré dans l'espace de Gestion.

### La quatrième étape : gestion des réponses des annuaires

À la réception du mail de réponse du Webmaster vous indiquant l'acceptation de votre site ou de son refus, vous renseignez l'état de chacun des annuaires (accepté, refusé). Au fil de vos inscriptions, vous allez bénéficier de tableaux de bords d'activité de votre campagne de soumission.

Yooda Submit va vous permettre d'effectuer en toute simplicité votre campagne de soumission. Et ainsi soumettre votre site dans un annuaire en moins de 2 minutes. Il est toutefois conseillé de ne pas soumettre votre site dans plus de 20 annuaires par jours. Pour tester le logiciel, Yooda.com met à disposition sur son site, une campagne d'essai qui vous permettra de soumettre gratuitement votre site dans une 20aine d'annuaires.

Vous pouvez télécharger gratuitement le logiciel sur le site [www.yooda.com](http://www.yooda.com).



Figure 1. Yooda

# Génération

## du BACK-OFFICE avec symfony

Vous découvrirez les différentes étapes de la réalisation d'un site web dont le but est de regrouper un grand nombre d'examens. Ce site peut être considéré comme une sorte de dépôt dans lequel l'étudiant va chercher les examens qui l'intéressent.

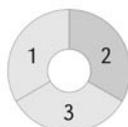
### Cet article explique:

- Dans cet article on présente la procédure de génération du module back-office d'un site web, l'accent sera particulièrement mis sur les fonctions de vérifications et de validations qu'offre symfony.

### Ce qu'il faut savoir :

- Cet article s'adresse à un public débutant mais ayant une première expérience avec symfony.

### Niveau de difficulté



La réussite d'un projet dépend de plusieurs facteurs, entre autres, les choix techniques et le langage de développement. Heureusement, avec symfony (qui est un framework open source développé en PHP5) vous pouvez réaliser une solution gratuite, Open Source et évolutive. Les limites du développement classique avec le langage PHP ont été dépassées avec symfony; cette dernière est entièrement basée sur la séparation entre le code et le design. On commence par présenter les différentes fonctionnalités attendues du site ainsi que la conception qui en résulte. On passe ensuite à la description de la génération du module *back office* du site qui constitue l'interface d'administration du site.

### Description et conception

Dans cette partie, on va d'abord décrire le site, puis on va présenter : en premier lieu la méthode Merise, en second lieu l'outil *DB Designer* et enfin le modèle conceptuel de données.

### Description du site

Comme on l'a déjà mentionné, ce site s'adresse essentiellement à une population étudiante et dont la finalité consiste à offrir une interface Web pour la recherche et le té-

lchargement des examens, et ce, à travers une interface à accès publique. Une autre interface, reversée aux utilisateurs inscrits va servir à charger les examens sur le serveur. Pour faciliter la tâche à l'utilisateur, celui-ci devra uniquement spécifier le chemin du document à charger et les mots clés. Il peut, en option, spécifier l'organisme dans lequel l'examen a été posé, qui lui même appartient à une université.

### Présentation de la méthode Merise

MERISE est une démarche à 3 dimensions, qui découpe l'organisation en domaines. Elle modélise les données et les traitements du système d'informations d'une manière séparée. C'est le fruit d'un ensemble de travaux de recherches dans le domaine de l'informatique de gestion et de recherches en systémi-

que appliquée aux organisations. Elle a été créée en 1979 et dont l'objectif est de concevoir et définir une méthode d'intérêt national qui intègre l'approche par les données et qui propose une démarche garantissant la rigueur de la méthode et sa facilité d'application sur le terrain. Elle est particulièrement utilisée dans les pays francophones.

### L'outil DB Designer

Plusieurs outils de conception de génération automatique de base de données existent. Le choix s'est arrêté sur l'outil *DB designer 4* qui est un logiciel open source permettant de concevoir des bases de données à partir d'un MCD (*modèle conceptuel de données*). Il possède également des fonctions permettant de créer et de gérer la base de données sous MySQL. Pour de plus amples informations sur les fonctionnalités de cet outil, veuillez consulter l'adresse : <http://www.framasoft.net/article4695.html>.

### Modèle Conceptuel de Données

La description ci-dessus vous a conduit au *Modèle Conceptuel* suivant qui a été réalisé avec *DB designer 4* (voir Figure 1).

#### Listing 1. Exemple du fichier Schema.yml

```
propel:
  weblog_membre:
    _attributes: { phpName: Membre }
  id:
    identifiant: varchar(20)
    motdepasse: varchar(20)
    nom: varchar(20)
    prenom: varchar(20)
    DateNaissance: date
    LieuNaissance: varchar(20)
    Adresse: varchar(20)
    CodePostal: integer
    Region: varchar(20)
```

## Génération du BACK-OFFICE

Symfony vous donne les moyens de réaliser rapidement et efficacement la partie *back office* du site qui va permettre l'administration des différents modules du site web, et c'est à partir de quelques lignes de commandes et sans éditer beaucoup de code.

On détaillera la réalisation des modules suivants:

- Inscription (configuration et validation),
- Authentification,
- L'ajout des examens, des matières et des établissements,
- La navigation entre les différentes pages.

En utilisant l'invite de commande, vous commencez par la création de votre application pour le *back-office* via la commande :

```
symfony init-app backend
```

Vous allez procéder à la génération de la base de données à l'aide du fichier de configuration *schema.yml*.

*Schema.yml* : pour la construction de la base, chaque entité présente dans le MCD est une table définie dans le fichier de configuration *schema.yml* (voir Listing 1).

Après avoir enregistré ce fichier, vous allez taper la commande suivante :

```
Symfony propel-build-model
```

Qui permet de gérer la classe du modèle. Ensuite :

```
Symfony propel-build-sql
```

Qui gère le SQL du schéma. Et enfin :

```
Symfony propel-insert-sql
```

Qui permet l'insertion du SQL dans la base. Et maintenant vous pouvez créer vos modules. Il vous faudra ensuite créer pour chaque classe Propel le module d'administration correspondant.

Dans cet exemple, vous avez besoin seulement de 4 classes Propel ce qui exige la création de 4 modules d'administration :

```
symfony propel-init-admin backend membre
Membre
symfony propel-init-admin backend examen
Examen
Symfony propel-init-admin backend matiere
Matiere
Symfony propel-init-admin backend
etablissement Etablissement
```

Ces instructions vous permettent de créer 4 modules avec une configuration par défaut dont tous les champs sont pris en compte.

Listing 1. Exemple du fichier *Schema.yml* - suite

```
Telephone: integer
Email: varchar(20)
categorie: varchar(20)
photo: varchar(50)
sexe: varchar(20)
etablissement_id:
  dateinscrit: date
weblog_etudiant: _attributes: { phpName: Etudiant }
id:
  niveau: varchar(20)
membre_id:
  weblog_enseignant: _attributes: { phpName: Enseignant }
id:
  niveau: varchar(20)
  titre: varchar(20)
  annee: varchar(20)
  etablissement: varchar(20)
membre_id:
  weblog_examen: _attributes: { phpName: Examen }
id:
  date: date
  nature: varchar(20)
  mot: varchar(20)
  chemin: varchar(50)
  membre_id:
  matiere_id:
weblog_matiere: _attributes: { phpName: Matiere }
id:
  Nom: varchar(20)
weblog_section:
  _attributes: { phpName: Section }
id:
  Nom: varchar(20)
weblog_etablissement: _attributes: { phpName: Etablissement }
id:
  Nom: varchar(60)
  agregation: varchar(60)
  univ: integer
weblog_universite: _attributes: { phpName: Universite }
id:
  Nom: varchar(20)
  agregation: varchar(60)
weblog_appartient: _attributes: { phpName: Appartient }
  etablissement_id:
  section_id:
  matiere_id:
weblog_amie: _attributes: { phpName: Amie }
id:
  source: integer
  destination: integer
  etat : integer
weblog_message: _attributes: { phpName: Message}
id:
  source: integer
  destination: integer
  date: varchar(50)
  sujet: varchar(200)
  texte: longvarchar
  lu: boolean
  effacer: integer
```

**Inscription (configuration)**

Passez maintenant à la configuration. Vous allez modifier la description de vos données dans le fichier `apps/backend/modules/membre/config/generator.yml` (voir Listing 2). Ainsi, vous pouvez titrer la page : `Title : Liste des Membres`. Définir une mise en page : `Layout : tabular`.

Définir aussi les champs que vous voulez afficher :

```
display: [=fullName, _photo, Region,
         categorie, etablissemnt_id,
         dateinscrit]
```

Préciser le nombre maximum d'enregistrements par pages :

```
max_per_page: 5
```

Réaliser un filtre dont le but est de faciliter

la recherche des données que vous voulez récupérer :

```
filters: [nom, prenom, Region, _categorie,
         dateinscrit, etablissemnt_id]
```

Dans le fichier de configuration `schema.yml` vous n'avez pas le droit d'écrire des champs contenant des espaces tel que *mot de passe*, on est donc obligé d'écrire comme cela *motdepasse*, et ce n'est pas exactement ce que vous voulez. Vous pouvez alors les renommer :

```
fields:
  motdepasse: { name: Mot de passe }
  Codepostal: { name: Code Postal }
  dateinscrit: { name: Date d'inscription
               }
```

Déposer des photos :

```
photo:
  type: admin_input_file_tag
```

en donnant un message d'aide qui sera affiché au-dessous de la zone du texte de la photo :

```
help: SVP charger votre photo
```

vous pouvez aussi l'afficher et donner la possibilité à l'utilisateur de la supprimer :

```
upload_dir: photos
params: include_link=photos
        include_remove=true
```

Pour récupérer la photo même dans la page des listes des membres et non pas le nom, vous devrez mettre le signe `_` devant le champ photo lors de sa définition dans les champs que vous voulez afficher, et en créant un nouveau fichier nommé `_photo.php` sous `apps/backend/modules/membre/templates` contenant le code de la récupération de la photo :

```
<?php echo image_tag('/'.sfConfig::get('
sf_upload_dir_name').'/photos/'
.$membre->getPhoto()) ?>
```

De même si vous voulez rendre le mot de passe crypté, sous `apps/backend/modules/membre/templates` en créant le fichier `_motdepasse.php` portant le code suivant :

```
<?php echo input_password_tag() ?>
```

Les champs sexe et catégorie sont de type liste déroulante dont le contenu est statique. Vous définissez leur contenu en créant les fichiers suivants `_sexe.php` (Listing 3) et `_categorie.php` (Listing 4) se trouvant sous

**Listing 2. Exemple du fichier generator.yml**

```
generator:
  class: sfPropelAdminGenerator
  param:
    model_class: Membre
    theme: default
    fields:
      identifiant: { name: Identifiant }
      motdepasse: { name: Mot de passe }
      nom: { name: Nom }
      prenom: { name: Prenom }
      DateNaissance: { name: Date de Naissances }
      LieuNaissance: { name: Lieu de Naissance }
      Adresse: { name: Adresse }
      Codepostal: { name: Code Postal }
      Region: { name: Region }
      Telephone: { name: Telephone }
      Email: { name: e-mail }
      categorie: { name: Categorie }
      sexe: { name: Sexe }
      etablissemnt_id: { name: Etablissement }
      dateinscrit: { name: Date d'inscription }
    photo:
      help: Please upload a photo
      type: admin_input_file_tag
      upload_dir: photos
      params: include_link=photos include_remove=true
    list:
      title: Liste des Membres
      layout: tabular
      display: [=fullName, _photo, Region, categorie, etablissemnt_id, dateinscrit]
      max_per_page: 5
      filters: [nom, prenom, Region, _categorie, dateinscrit, etablissemnt_id]
    edit:
      title: Inscription
      display: [identifiant, motdepasse, nom, prenom, DateNaissance, LieuNaissance,
              Adresse, Codepostal, Region, Telephone, Email, _categorie, _sexe,
              photo, etablissemnt_id, dateinscrit]
```

**Listing 3. Exemple du fichier \_sexe.php**

```
<?php $sexe = array( 'hom' => 'Homme', 'fem' => 'Femme');
echo select_tag('sex', options_for_select($sexe, 'hom')); ?>
```

**Listing 4. Exemple du fichier \_categorie.php**

```
<?php $categorie = array( 'etud' => 'Etudiant', 'ens' => 'Enseignant');
echo select_tag('type', options_for_select($categorie, 'etud')); ?>
```

apps/backend/modules/membre/templates.  
 Pour récupérer le nom et le prénom ensemble dans le même champ, il faut surcharger la fonction `__toString()` dans le fichier `lib/model/Membre.php` qui appelle à son tour la fonction `getFullName()` comme valeur de retour et cette dernière fonction permet de récupérer le nom et le prénom séparés par un espace.

Et voilà le code :

```
public function __toString() {
    return $this->getFullName();
} public function getFullName() {
    return $this->getNom(). ' '
        . $this->getPrenom();
}
```

En mettant le signe = devant un champ, un lien amenant à la page d'édition de cet enregistrement sera automatiquement créé.

```
display: [=fullName]
```

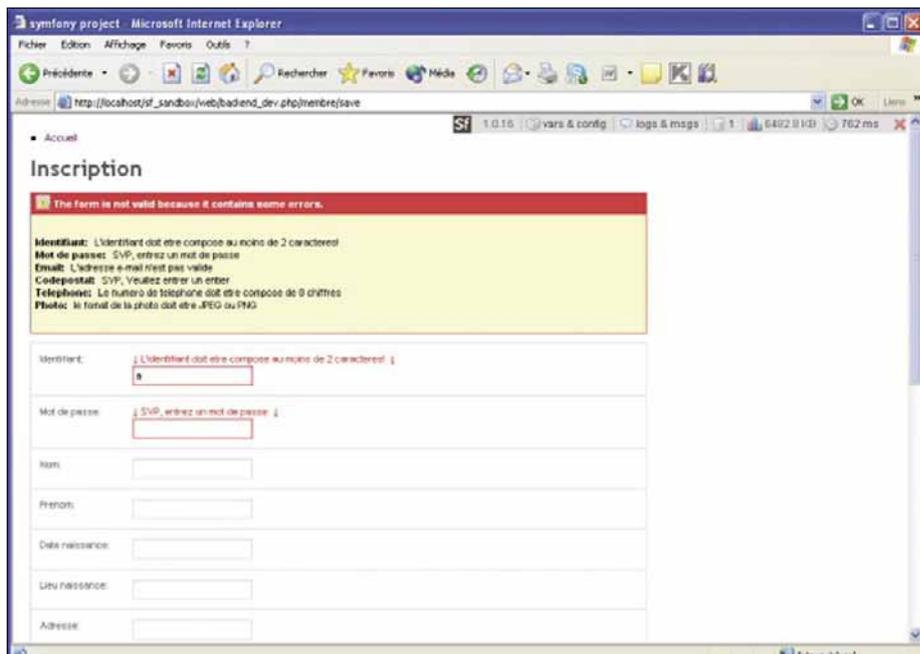


Figure 2. Vérification des coordonnées pour le déroulement d'une bonne inscription

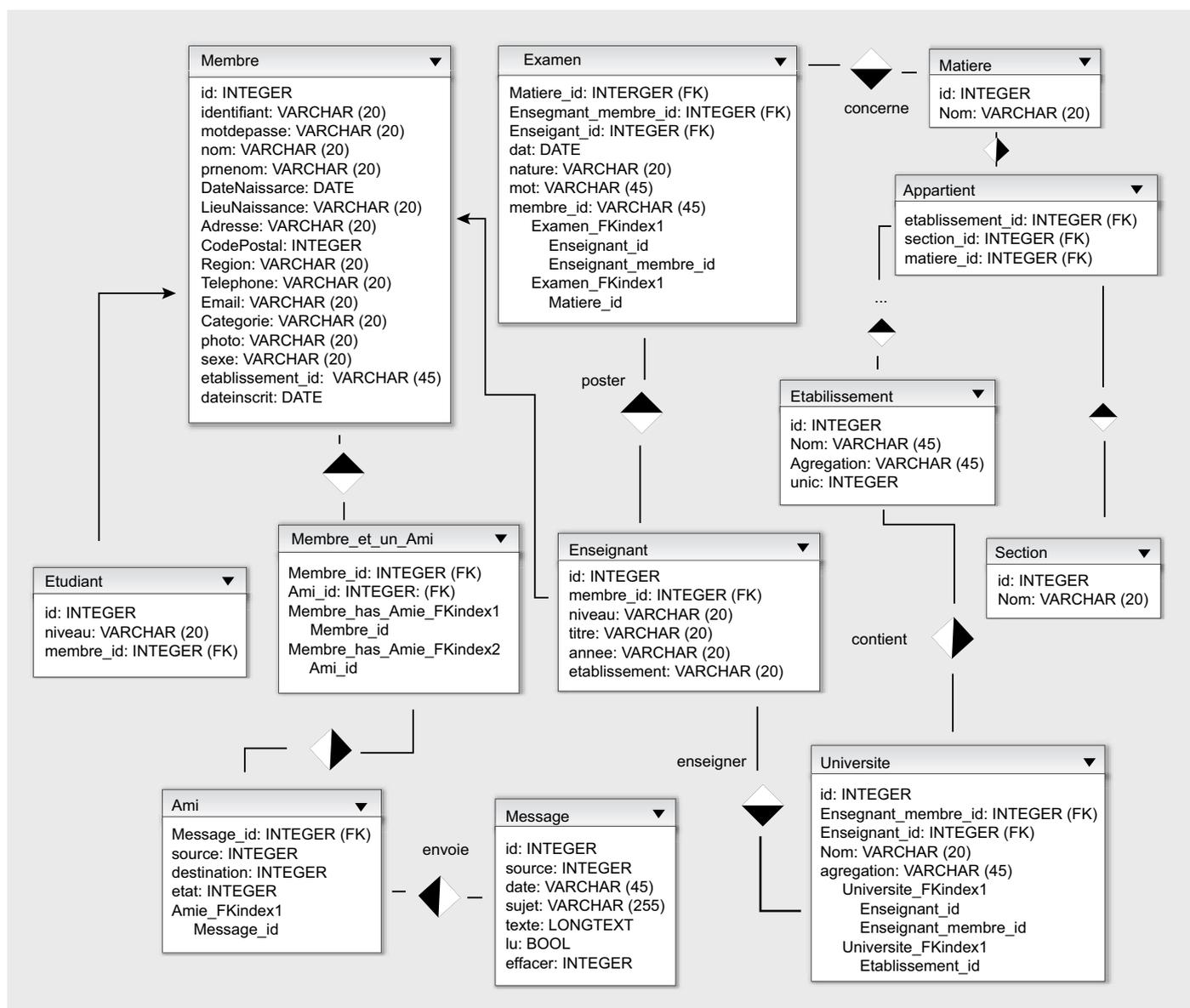


Figure 1. Modèle Conceptuel de Données

**Inscription (validation)**

Vous allez ajouter un fichier dont le nom est celui de l'action à faire. Pour l'action de l'édition, vous créez le fichier *edit.yml*, qui sera placé sous le dossier *validate* du module membre (voir Listing 5). Dans ce fichier on a fait plusieurs vérifications telles que les champs obligatoires. Dans cet exemple les champs identifiant et mot de passe sont obligatoires :

Le champ email n'est pas obligatoire mais il doit comprendre une adresse email valide, ce qui se fait grâce au validateur standard *emailValidator* intégré dans symfony.

Vous vérifiez aussi la longueur d'un champ, en indiquant une valeur limite à ne pas dépasser (min ou max ou les deux ensemble) à l'aide du validateur *sfStringValidator* :

```
param:
  min: 1000
  max: 4000

max_size: 2000000
mime_types:
  - 'image/jpeg'
  - 'image/png'
```

et en affichant un message d'erreur pouvant aider l'utilisateur à corriger son éventuelle erreur :

```
min_error: L'identifiant doit être composé
           au moins de 2 caractères!
email_error: L'adresse e-mail n'est
            pas valide
```

Le validateur *sfNumberValidator* vous permet de vérifier si la valeur entrée est bien une valeur entière ou non tel que le code postal ou bien le numéro du téléphone.

En outre, la vérification de la taille maximale de la photo et des formats autorisés (*jpeg*, *png*, *jpeg*, *x-png*) se font avec les paramètres suivants :

```
max_size: 2000000
mime_types:
  - 'image/jpeg'
  - 'image/png'
```

```
- 'image/jpeg'
- 'image/x-png'
```

Afin de s'inscrire, on accède à la page d'inscription via le lien qui se trouve dans la page d'accueil ou bien via l'adresse : [http://localhost/sf\\_sandbox/web/backend\\_dev.php/membre/create](http://localhost/sf_sandbox/web/backend_dev.php/membre/create).

Lors du remplissage du formulaire par des fausses coordonnées, des messages d'erreurs seront affichés ce que vous montre la Figure 2.

**Authentification**

Pour l'instant, notre application n'est pas sécurisée. On veut restreindre l'accès à la page de chargement d'examen aux personnes inscrites au site.

Vous allez commencer par créer un fichier nommé *security.yml* sous *apps/backend/modules/examen/config* contenant le code suivant :

**Listing 5. Exemple du fichier *edit.yml***

```
methods:
  post:
    - "membre{identifiant}"
    - "membre{motdepasse}"
    - "membre{email}"
    - "membre{codepostal}"
    - "membre{telephone}"
    - "membre{photo}"
names:
  membre{identifiant}:
    required: true
    required_msg: SVP, entrez un identifiant
    validators: idValidator
  membre{motdepasse}:
    required: true
    required_msg: SVP, entrez un mot de passe
    validators: psValidator
  membre{email}:
    required: false
    validators: emailValidator
  membre{codepostal}:
    required: false
    validators: numValidator
  membre{telephone}:
    required: false
    validators: telValidator
  membre{photo}:
    required: false
    file: true
    validators: photoValidator
idValidator:
  class: sfStringValidator
  param: min: 2
  min_error: L'identifiant doit être composé au moins de 2
             caractères!
psValidator:
  class: sfStringValidator
  param: min: 6
  min_error: Le mot de passe doit être composé au moins de
             6 caractères!
emailValidator:
  class: sfEmailValidator
  param:
    email_error: L'adresse e-mail n'est pas valide
numValidator:
  class: sfNumberValidator
  param:
    nan_error: SVP, Veuillez entrer un entier
    min: 1000
    min_error: La valeur du code postal doit être supérieur
              ou égal à 1000
    max: 4000
    max_error: La valeur du code postal doit être inférieur
              ou égal à 4000
telValidator:
  class: sfNumberValidator
  param:
    nan_error: SVP, Veuillez entrer un entier
    min: 11111111
    min_error: Le numéro de téléphone doit être composé de 8
              chiffres
    max: 99999999
    max_error: Le numéro de téléphone doit être composé de 8
              chiffres
photoValidator:
  class: sfFileValidator
  param:
    max_size: 2000000
    max_size_error: Max size is 2Mo
    mime_types:
      - 'image/jpeg'
      - 'image/png'
      - 'image/pjpeg'
      - 'image/x-png'
    mime_types_error: le format de la photo doit être JPEG ou PNG
```

```
all:
  is_secure: on
```

L'étape suivante est de rendre le module `security` comme un module par défaut, pour le faire, vous ouvrez le fichier `settings.yml` sous `apps/backend/config` et vous modifiez cette partie du code :

```
#all:
# .actions:
# login_module: security
# login_action: index
```

Ensuite, vous créez deux modules l'un est nommé `security` dont le `template` contient les champs identifiant et mot de passe pour la connexion, l'autre est nommé `verife` qui servira à vérifier l'authentification.

```
symfony init-module backend security
symfony init-module backend verife
```

D'abord vous supprimez le code défini par défaut dans la fonction `executeIndex` contenue dans le fichier `actions.php` du module `security`.

Éditez le fichier `apps/backend/modules/security/templates/indexSuccess.php` pour créer l'interface d'authentification (Listing 6). Éditez le fichier `apps/backend/modules/verife/actions/actions.php` pour ajouter l'action d'authentification en définissant cette fonction (Listing 7).

Ici, vous avez deux tests. Le premier consiste à diriger les inscrits (étudiant ou enseignant) directement à la page de l'ajout des examens après leur connexion. Le deuxième concerne l'administrateur qui possède déjà un identifiant et un mot de passe (administrateur et gestion, dans notre exemple), qui sera dirigé à la page de l'espace administration et qui contient un menu représentant les différentes actions disponibles (gestion des matières, des établissements, des examens et des membres).

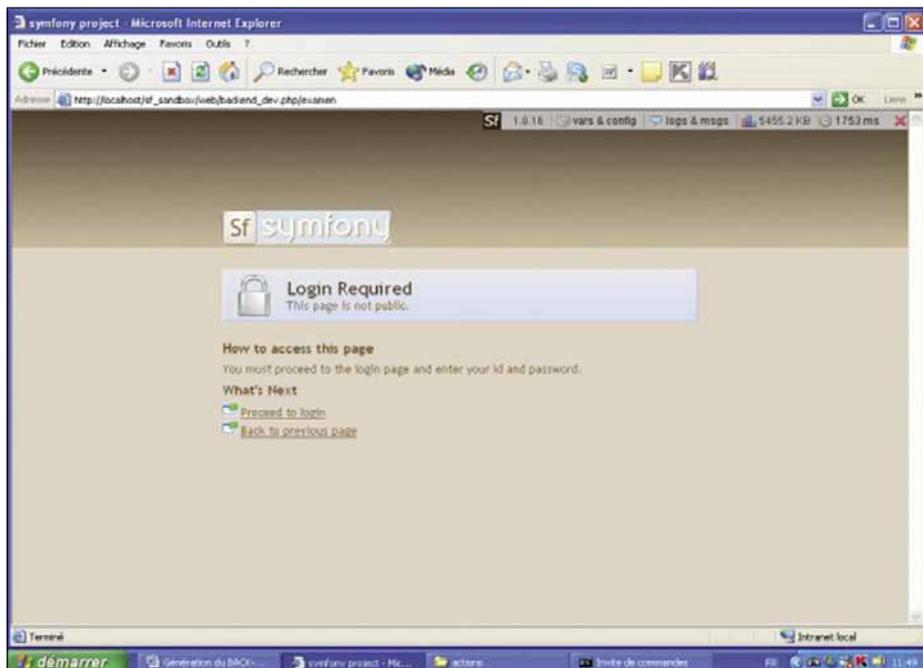


Figure 3. Page sécurisée

**Listing 6. Exemple du fichier `indexSuccess.php`**

```
<div id="title">
  <h1><center><?php echo link_to('Soyez le Bienvenue', '@homepage') ?></center></h1>
</div>
<h4>
  Veuillez vous connecter si vous posséder un compte sinon vous pouvez vous
  inscrire:<?php echo link_to('Inscription', 'membre/create') ?>
</h4>
<?php echo form_tag('verife') ?>
  <label for="identifiant">Identifiant:</label>
  <?php echo input_tag('identifiant', $sf_params->get('identifiant')) ?>
  <label for="motdepasse">Mot de passe:</label>
  <?php echo input_password_tag('motdepasse') ?>
  <?php echo submit_tag('Connecter', 'class=default') ?>
</form>
```



# 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)

(1) TTC, sans frais cachés.

**Autres services :**

- :: Création de site web,
- :: Référencement,
- :: Location de serveurs dédiés



**www.clicasso.fr**

**N° Indigo 0 820 320 256**

0,09 € TTC / MN

## L'ajout des examens, des matières et des établissements

La génération du module examen, matière et établissement se fait exactement comme le module inscription par contre l'accès à ces pages est sécurisé. Un accès non authentifié via l'adresse : `http://localhost/sf_sandbox/web/backend_dev.php/examen` nous permet de voir le résultat (Figure 3).

## La navigation entre les différentes pages

On peut faire des liens globaux qui seront présents sur toutes les pages, et on peut aussi faire

des liens partiels qui se trouvent juste dans une seule page. Dans cet exemple, on a fait le lien *Accueil* comme un lien global et c'est en ajoutant le code suivant dans le fichier *layout.php* qui se trouve sous *apps/backend/templates* :

```
<li><?php echo link_to('Accueil', 'http://localhost/sf_sandbox/web/backend_dev.php/security') ?></li>
```

On a aussi défini plusieurs autres liens qui se trouvent dans la page de l'espace administration et qui se présentent juste sur cette page car ils sont définis sous le *template*

du module menu, et c'est en ajoutant le code suivant dans le fichier *indexSuccess.php* se trouvant sous le dossier *apps/backend/modules/menu/templates* (Listing 8).

## Connaissances supplémentaires

Pour rendre la page d'inscription un peu plus claire, on peut placer les champs obligatoires ensemble et les champs non obligatoires ensemble. Dans ce site les champs obligatoires sont l'identifiant et le mot de passe, tout le reste est non obligatoire.

Donc on peut éditer le fichier de configuration *generator.yml* qui se trouve sous *apps/backend/modules/membre/config/*, en modifiant ce code :

```
edit:
  display: [identifiant, motdepasse,
            nom, prenom, DateNaissance,
            LieuNaissance, Adresse, Codepostal,
            Region, Telephone, Email, _categorie,
            _sexe, photo, etablisement_id,
            dateinscrit]
```

par ceci :

```
edit: display:
  Champs Obligatoires : [identifiant,
                        motdepasse]

  Champs Non Obligatoires : [nom, prenom,
                             DateNaissance, LieuNaissance,
                             Adresse, Codepostal, Region,
                             Telephone, Email, _categorie, _sexe,
                             photo, etablisement_id,
                             dateinscrit]
```

Vous avez affiché la liste des membres dans un tableau, si vous voulez que l'affichage soit juste sous forme de lignes et pas de colonnes, vous allez donc modifier le code suivant sous *apps/backend/modules/membre/config/generator.yml* :

```
list :
  layout : tabular
```

par ceci :

```
list :
  layout : stacked
```

Si vous voulez renommer le bouton *create* dans la page *liste des examens* par *ajouter un examen*, c'est sous *apps/backend/modules/examen/config/generator.yml* qu'on ajoute ce code :

```
actions:
  _create: {name: ajouter un examen}
```

Si vous voulez que l'action dans la page d'ajout des examens soit juste la sauvegarde, vous pouvez ajouter ce code dans le même fichier cité précédemment :

### Listing 7. Définition de la méthode *executeIndex*

```
public function executeIndex() {
  $test='0';
  $mesg='';
  $identifiant=$this->getRequestParameter('identifiant');
  $motdepasse=$this->getRequestParameter('motdepasse');
  $this->verifie=MembrePeer::doSelect(new Criteria());
  foreach($this->verifie as $m) {
    if( ($m->getIdentifiant()==$identifiant) && ($m->getMotdepasse()==$motdepasse) ) {
      $this->test='1';
      break;
    }
  }
  if( $this->test=='1' ) {
    $_SESSION['identifiant']=$identifiant;
    $_SESSION['motdepasse']=$motdepasse;
    $this->getUser()->setAuthenticated(true);
    return $this->redirect('examen/create');
  } else {
    ?>
    <body onLoad="javascript:history.back()">
    </body>

    <?php }
    if( $this->getRequestParameter('identifiant') == 'administrateur' && $this->
      getRequestParameter('motdepasse') == 'gestion' ) {
      $this->getUser()->setAuthenticated(true);
      return $this->redirect('menu');
    } else {
      $this->getRequest()->setError('identifiant', 'incorrect entry');
      return $this->forward('security', 'index');
    }
  }
}
```

### Listing 8. Exemple du fichier *indexSuccess.php*

```
<li>
  <?php echo link_to('Gestion des matieres?', 'matiere/list') ?>
</li>
<li>
  <?php echo link_to('Gestion des etablisements?', 'etablisement/ list') ?>
</li>
<li>
  <?php echo link_to('Gestion des membres?', 'membre/list') ?>
</li>
<li>
  <?php echo link_to('Gestion des examens?', 'examen/list') ?>
</li>
```

```
edit:
  actions:
    _save: ~
```

Pour faciliter la tâche d'édition ou de suppression d'un membre ou d'un examen pour l'administrateur, vous pouvez ajouter ces deux actions dans la page des listes, ainsi l'administrateur visitera la page des listes et il aura le choix soit d'éditer un enregistrement pour voir toutes les informations qui lui appartiennent, soit le supprimer.

Pour les examens, vous ajoutez le code suivant dans le même fichier que précédemment :

```
list:
  object_actions:
    _edit: ~
    _delete: ~
```

L'administrateur est le responsable de toutes les gestions, il a le droit d'ajouter un membre, un examen, etc. Pour faciliter les tâches, on peut regrouper toutes les actions (ajout des examens, ajout des matières, ajout des établissements, ajout des membres) dans la page *liste des membres*. Par exemple, pour l'ajout des examens, c'est dans le fichier `apps/backend/modules/membre/config/generator.yml` qu'on ajoute le code suivant :

## Sur Internet

- <http://www.symfony-projet.org/> – Le site officiel de symfony,
- <http://c-maneu.developpez.com/tutorial/web/php/symfony/intro/> – Le site du *developpez.com*,
- [http://www.symfony-project.org/book/1\\_1/](http://www.symfony-project.org/book/1_1/) – Le livre *The Definitive Guide to symfony* écrit par Fabien Potencier et François Zaninotto.

```
list:
  object_actions:
    ajouterExam: {name: ajouter un
                  examen, action: Ajout}
```

Et dans le fichier `apps/backend/modules/membre/actions/actions.class.php` qu'on ajoute la fonction suivante :

```
public function executeAjout() {
    $this->redirect('examen/create');}
```

Ou bien si vous voulez que l'action soit la visite de la liste des examens, vous modifiez donc le mot `create` dans cette dernière fonction par `list`.

## Conclusion

Symfony est un framework qui apporte une vraie révolution dans le domaine de développement PHP.

On a consacré cet article à l'exploration de quelques fonctionnalités de base de symfony, ainsi, on a généré le *back-office* du site en éditant un minimum de code. On a aussi montré comment faire des tests pour contrôler les données saisies par l'utilisateur.

On a aussi développé la phase de l'authentification pour rendre l'accès sécurisé. Et à la fin, on a ajouté quelques connaissances qui servent à rendre le site plus organisé et plus clair.

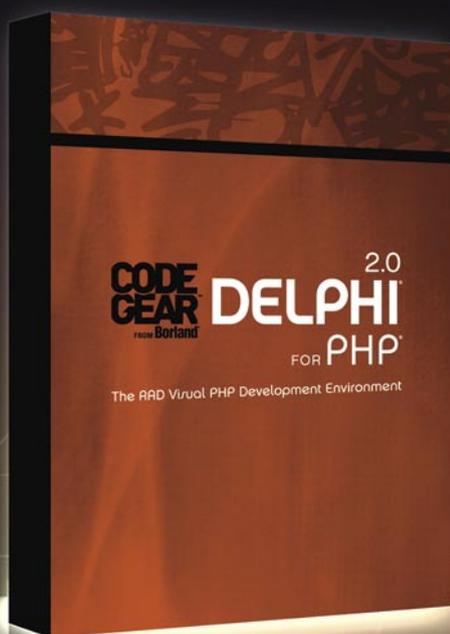
## AZZA NAFTI, ADEL SAFI

*Azza Nafti : étudiante en informatique à l'ESSTT (École Supérieure des Sciences et Techniques de Tunis). Le développement web constitue pour elle une passion. Contact : azza\_esstt@yahoo.fr*

*Adel Safi : Ingénieur chercheur en informatique à l'ESSTT. La programmation et l'administration sont ces domaines préférés. Contact : adel.safi@imag.fr*

P U B L I C I T É

# DELPHI FOR PHP 2.0



l'environnement de  
développement RAD pour PHP

[WWW.CODEGEARFRANCE.COM](http://WWW.CODEGEARFRANCE.COM)

[codegear@open-way.com](mailto:codegear@open-way.com)  
01 49 97 51 28

# Utiliser XML avec MySQL 5.1

MySQL avec son modèle relationnel et XML avec son approche hiérarchique ont en commun la vocation d'organiser et de structurer des ensembles de données.

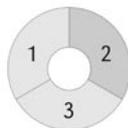
## Cet article explique :

- Les différentes fonctionnalités XML disponibles avec MySQL 5.1.

## Ce qu'il faut savoir :

- Il vous sera utile d'avoir des connaissances techniques sur MySQL, sur XML et Xpath.

## Niveau de difficulté



En regardant de plus près, ces technologies peuvent offrir des solutions complémentaires :

- MySQL s'adapte à merveille à la gestion de systèmes d'informations éventuellement lourds et complexes,
- XML a su devenir un langage universel permettant aux applications de communiquer et d'échanger des informations.

Leur complémentarité apparaît alors évidente : XML rend possible l'exploitation et le partage des informations contenues dans MySQL par des applications distantes.

## Générer du XML avec les programmes clients mysql et mysqldump

La version de MySQL utilisée pour cet article est la 5.1.22. Générer du XML avec MySQL est très simple. La première possibilité est d'utiliser le client texte mysql avec l'option *xml* ou *X*. Un fois connecté au serveur MySQL, les résultats des commandes sont sortis sous format XML comme on peut le voir sur le Listing 2.

Le Listing 3 nous montre que le résultat d'une requête peut être redirigé vers un fichier. On peut par exemple créer le fichier

*ville\_fra.xml* avec le contenu de la table *city* du schéma *world* ou plus généralement avec le résultat d'une requête. L'autre façon d'exporter du SQL dans un fichier au format XML est d'utiliser le client texte qui permet de faire des sauvegardes (*dump*) *mysqldump*. Il devient alors possible d'obtenir un *backup*, structures et données, au format XML du schéma *world*, comme le montre le Listing 4.

## Insérer du XML dans MySQL

Un fichier XML, n'est finalement qu'un fichier contenant du texte et insérer du texte dans un SGBDR, est loin d'être la chose la plus difficile. Une simple requête *INSERT* fait l'affaire. Cela nous donne quelque chose comme ceci :

```
INSERT INTO ma_table (champ_texte)
VALUES ('<?xml version="1.0"?>...')
```

Cela dit, si le flux XML est important, cette façon de faire n'est guère pratique. En fait, la méthode la plus simple est d'utiliser la fonction MySQL : *LOAD \_ FILE()*.

Cette fonction prend en argument un fichier et renvoie son contenu sous la forme d'une chaîne de caractères.

À noter que vous devez avoir le droit *FILE* pour exécuter cette commande, et que le fichier doit se trouver sur la machine hôte (en *local*). Voir Listing 1.

Arrêtons nous quelques instants sur la commande *LOAD XML* qui est apparue avec la version 6 de MySQL (MySQL 6.0.3). Cette commande permet d'importer simplement le contenu d'un fichier XML dans une table. Elle a une syntaxe assez proche de la commande *LOAD DATA* que vous connaissez peut-être, et qui elle aussi facilite le travail lorsque le besoin d'importer des données d'un fichier texte se fait sentir. Le Listing 5 nous montre un exemple d'utilisation de *LOAD XML*.

## Interroger un flux XML avec Xpath

L'emploi de XML ne présenterait pas beaucoup d'intérêt sans la possibilité de pouvoir interroger sa structure pour accéder à une information particulière : le langage *XPath* répond à cette attente et est (en partie du moins) aujourd'hui intégré à MySQL. Mais avant de voir comment utiliser *XPath* à travers MySQL, il est nécessaire de se pencher sur la syntaxe de ses expressions, dont la construction la plus importante est sans conteste le chemin de *localisation* (*location path*).

### Listing 1. Insérer un fichier XML dans MySQL

```
mysql> CREATE DATABASE IF NOT EXISTS xml;
mysql> USE xml;
mysql> CREATE TABLE rss (
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    type CHAR(30) NOT NULL,
    flux_rss TEXT )
ENGINE=MyISAM;
mysql> INSERT INTO xml.rss VALUES (NULL, 'cinema', LOAD_FILE('./cinema.xml'));
```

### Le chemin de localisation

De manière similaire à la navigation au sein d'un système de fichiers, un chemin de localisation se compose d'une succession d'étapes, séparées les unes des autres par un /. La description d'un chemin peut se faire de façon absolue ou relative, auquel cas un nœud contexte est utilisé afin de déterminer à partir de quel endroit doit s'effectuer la recherche.

Chaque étape d'un chemin de localisation peut se diviser en trois parties (axe de recherche, test du nœud et ensemble de prédicats), réunies sous la forme suivante :

```
Axis::Node-Test[Predicate1][Predicate2]...
```

### L'axe de recherche (Axis)

L'axe de recherche permet de spécifier dans quel *sens* orienter la sélection des nœuds que l'on veut récupérer par rapport au nœud courant. Ainsi, il est possible de choisir de chercher dans l'ensemble des nœuds des descendants, plutôt que de se limiter (par défaut) aux nœuds directement enfants.

Il existe une dizaine d'axes de recherche, dont les plus usités sont :

- *child*, qui limite à l'exploration des nœuds éléments directement enfants du nœud contexte,
- *descendant*, qui englobe, en plus des nœuds enfants, tous les éléments descendants,
- *descendant-or-self*, identique à descendant avec en plus le nœud contexte inclus,
- *ancestor*, qui contient tous les nœuds ancêtres,
- *ancestor-or-self*, identique à ancestor avec en plus le nœud contexte,
- *following-sibling*, qui fait référence aux éléments frères suivants,
- *preceding-sibling*, qui fait référence aux éléments frères précédents,
- *attribute*, qui se cantonne à la recherche dans les nœuds attributs.

### Le test du nœud (Node-Test)

Généralement, le test repose sur le nom du nœud que l'on veut retrouver. Cependant, *XPath* ne se limite pas à la recherche de simples éléments, mais à tous types de nœuds. Il est donc ainsi possible de préciser le type de nœud que l'on veut atteindre grâce aux fonctions suivantes :

- *node()*, qui valide tous types de nœuds,
- *text()*, pour les nœuds texte,
- *comment()*, pour les commentaires,
- *processing-instruction()*, pour les instructions de traitements.

### Les prédicats (Predicates)

Les prédicats, délimités par un [ et un ], permettent de filtrer plus finement l'ensemble des

nœuds retrouvés par le chemin de localisation, en évaluant l'expression qu'ils contiennent et en ne retenant que les nœuds qui la valident. Plusieurs prédicats peuvent se suivre, filtrant ainsi un peu plus précisément l'ensemble des nœuds à garder.

Les expressions mises en place dans les prédicats peuvent, en plus des quelques fonctions existantes et des opérateurs habituels, exploiter des ensembles de nœuds (*node-sets*) récupérés grâce à... des chemins de localisation ! Ainsi, il est possible d'adapter la validité d'un

#### Listing 2. Générer du XML en mode console

```
shell> mysql -u daz -p --xml
mysql> SHOW SCHEMAS ; /*Liste de tous les schémas (databases)*/
<?xml version="1.0"?>
<resultset statement="show schemas" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <row>
    <field name="Database">information_schema</field>
  </row>
  <row>
    <field name="Database">world</field>
  </row>
  <row>
    <field name="Database">xml</field>
  </row>
</resultset>
mysql> USE xml /*connexion au schéma qui se nomme xml */
Database changed
mysql> SHOW TABLES ; /*Liste de toutes les tables du schéma xml*/
<?xml version="1.0"?>
<resultset statement="SHOW TABLES" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <row>
    <field name="Tables_in_xml">rss</field>
  </row>
</resultset>
```

#### Listing 3. Création d'un fichier XML avec le client texte mysql

```
shell>mysql -X -e" SELECT * FROM world.city WHERE countrycode = 'fra';"
--default-character-set=utf8 > c:/city_fra.xml
shell> more city_fra.xml
<?xml version="1.0"?>
<resultset statement="SELECT * FROM world.city WHERE countrycode = 'fra'" xmlns:xsi
="http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <field name="ID">2974</field>
    <field name="Name">Paris</field>
    <field name="CountryCode">FRA</field>
    <field name="District">Île-de-France</field>
    <field name="Population">2125246</field>
  </row>
  <row>
    <field name="ID">2975</field>
    <field name="Name">Marseille</field>
    <field name="CountryCode">FRA</field>
    <field name="District">Provence-Alpes-Côte</field>
    <field name="Population">798430</field>
  </row>
  <row>
    <field name="ID">2976</field>
    <field name="Name">Lyon</field>
    ...
  </row>
</resultset>
```

**Listing 4. Générer une sauvegarde au format XML**

```
shell> mysqldump -u daz -p --xml --default-character-set=utf8 world > sauvegarde.xml
<?xml version="1.0"?>
<mysqldump xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <database name="world">
    <table_structure name="city">
      <field Field="ID" Type="int(11)" Null="NO" Key="PRI" Extra="auto_increment" />
      ...
    </table_structure>
    <table_data name="city">
      <row> <field name="ID">1</field> ... </field> </row>
    </table_data>
  </database>
</mysqldump>
```

**Listing 5. Utilisation de la commande LOAD XML de MySQL 6**

```
mysql6> USE xml;
mysql6> CREATE TABLE anaska (
  id_consultant TINYINT UNSIGNED NOT NULL PRIMARY KEY,
  nom CHAR(40),
  prenom CHAR(40)
)ENGINE=MyISAM;
mysql6> LOAD XML LOCAL INFILE 'anaska.xml' INTO TABLE anaska
ROWS IDENTIFIED BY '<consultant>';
mysql6> SELECT * FROM anaska;
+-----+-----+
| id_consultant | nom      | prenom      |
+-----+-----+
| 1 | Dasini | Olivier     |
| 2 | Allard | Fabien      |
| 3 | Pauli  | Julien     |
| 4 | Cyril  | Pierre de Geyer |
| 5 | Romain | Bourdon    |
+-----+-----+
```

**Listing 6. Extraire des données d'une variable contenant du XML**

```
/*Contenu du fichier XML */
shell> more anaska.xml
<anaska>
  Les formateurs de l'équipe sont :
  <formateur domaines="MySQL PHP"> <nom>Dasini</nom> <prenom>Olivier</prenom>
  </formateur>
  <formateur domaines="PHP XML"> <nom>Allard</nom>
  <prenom>Fabien</prenom>
  </formateur>
  <formateur domaines="Linux MySQL">
  <nom>Dumont</nom>
  <prenom>Pierre</prenom>
  </formateur>
  et sont passionnés par l'open source.
</anaska>
/* Initialisation de la variable @xml avec le contenu du fichier anaska.xml */
mysql> SET @xml = LOAD_FILE( './anaska.xml' );
/* Récupération de la valeur de l'élément racine */
mysql> SELECT extractValue( @xml, '/anaska' )\G
***** 1. row *****
extractValue( @xml, '/anaska' ):
  Les formateurs de l'équipe sont :
  et sont passionnés par l'open source.
```

chemin de localisation en fonction même de tout élément de l'arborescence du flux XML. Par exemple, si l'on considère l'expression *XPath* suivante :

```
// article[@rubrique = /site/menu/
rubrique[2]]
```

on récupère tous les articles qui possèdent un attribut rubrique ayant la même valeur que la deuxième rubrique du menu du site. Cet autre exemple :

```
// article[not( @rubrique = preceding-
// sibling::article/@rubrique)]
```

permet de ne sélectionner que les articles qui n'ont pas leur attribut rubrique égal à l'attribut rubrique d'un article précédent. En d'autres termes, on ne récupère ici que le premier article de chaque rubrique.

**Quelques raccourcis**

Afin de faciliter la lecture et l'écriture de requêtes *XPath*, il existe bien heureusement un ensemble de raccourcis pratiques. Le Tableau 1 présente une liste des ces raccourcis.

**Les fonctionnalités XPath de MySQL 5**

MySQL 5 offre la possibilité de pouvoir exécuter des requêtes *XPath* sur un flux XML, grâce à la fonction `extractValue()`. Le résultat récupéré est la concaténation de la valeur textuelle de chacun des nœuds trouvés (chaque valeur étant dissociée des autres par un espace blanc). Le Listing 6 nous montre comment récupérer la valeur de l'élément racine d'un flux XML.

Comme vous pouvez le constater dans le Listing 6, les éléments enfants de l'élément racine sont supprimés de la sélection : seul le contenu propre des éléments recherchés est donc intégré au résultat. Autre exemple, on veut sélectionner le deuxième formateur de notre flux XML, ainsi que les formateurs qui ont pour domaine de compétence MySQL (Listing 7).

Comme vous avez pu le constater, *XPath* est également un langage d'interrogation. Examinons d'un peu plus près sa syntaxe.

Une expression *XPath* représente ce qui est appelé un *chemin de localisation* (très similaire au chemin permettant d'identifier un fichier) dans lequel chaque nœud est séparé des autres avec un /. Généralement, la recherche s'effectue

**Tableau 1. Raccourcis XPath**

self::node()	.
parent::node()	..
attribute::name	@name
//descendant-or-self::node()/	//
element[position() = X]	element[X]

en fonction de deux types de nœuds distincts :

- les nœuds éléments, renseignés par leur nom,
- les attributs, indiqués par un @ accolé à leur nom.

Ainsi l'expression `formateur/@domaines` réfère l'attribut appelé `domaines` des éléments nommés *formateur*. Pour pouvoir filtrer plus finement les éléments, *XPath* dispose des *prédi-*

*cats*. Chacun de ces prédicats est spécifié dans une paire de crochets [*expression*]. Une expression peut être soit une simple évaluation, soit utiliser une (ou plusieurs) fonction(s).

Par exemple `formateur[@domaines="Linux MySQL"][contains(nom, "Dumont")]/prenom` retrouve tous les prénoms des formateurs qui ont pour domaines *Linux MySQL* et qui possèdent un nom valant *Dumont*.

Les chemins de localisation utilisés peuvent être indiqués de façon absolue ou de

façon relative. Dans ce deuxième cas, la recherche s'effectue en fonction du nœud contexte.

Dans l'expression précédente, l'attribut `domaines` testé est celui de l'élément `formateur`.

Enfin, il est possible de se déplacer dans l'arborescence (un `.` représente le nœud courant, et `..` le nœud parent).

Ainsi `formateur/prenom[../@domaines="Linux MySQL"][contains(.. / nom, "Dumont")]` est équivalente à l'expression précédente.

Jusqu'ici, la récupération des informations ne s'est opérée qu'en lisant un flux XML stocké dans une variable. Cependant, il est également possible de récupérer des informations XML stockées dans la base : il suffit pour cela de remplacer la variable par le nom du champs dans lequel est enregistré ce flux. Le Listing 8 permet de trouver le type et les URL des flux RSS qui contiennent le mot *France*.

MySQL 5 implémente une autre commande pour pouvoir traiter des flux XML, la fonction `updateXML()`. Elle permet de modifier la sortie d'un document XML, sans modifier ce dernier. Une remarque, cette fonction ne modifie que le premier nœud trouvé qui correspond à l'expression, ce qui limite son intérêt. De plus, la chaîne de remplacement est figée, il n'est donc pas possible de faire une transformation (Listing 9).

## Conclusion

Voilà un petit panorama de l'utilisation des fonctionnalités XML de MySQL. Comme nous l'avons vu, générer le résultat d'une requête au format XML reste très simple avec le client texte `mysql`, idem pour la génération d'une sauvegarde (`mysqldump`).

De plus, bien que n'étant pas une base de données XML, on peut aisément y stocker des flux XML, et les parcourir ou les modifier grâce aux fonctions `extractValue` et `updatexml`, et une bonne dose d'*XPath*. Le support de ce dernier, par MySQL, reste encore assez basique, mais ce manque sera comblé progressivement dans les prochaines versions.

## OLIVIER DASINI, FABIEN ALLARD

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 Francophone : LeMUG.fr. Fabien Allard est formateur et développeur certifié PHP, Ajax et Linux chez Anaska Alter Way Group. Il est également spécialiste des technologies XML.

### Listing 7. Extraire le deuxième formateur et les formateurs MySQL

```
mysql> SELECT extractValue(@xml, '/anaska/formateur[2]/nom') AS Deuxieme,
      extractValue(@xml, '/anaska/formateur[contains(@domaines, "MySQL")]/nom')
      AS MySQL;
+-----+-----+
| Deuxieme | MySQL |
+-----+-----+
| Allard   | Dasini |
|          | Dumont |
+-----+-----+
```

### Listing 8. Extraire les données d'un flux XML stocké en base

```
SELECT type, extractValue(flux_rss, "/rss/channel/item/link[../title[contains(
., 'France')]])") AS url FROM rss GROUP BY url HAVING url NOT LIKE ''\G
***** 1. row *****

type: international
url: http://www.lemonde.fr/asia-pacifique/article/2008/08/21/en-france-le-debat-s-
intensifie-sur-le-renfort-des-troupes-en-afghanistan-decide-par-m-sarkozy_
1086214_3216.html?xtor=RSS-3210
***** 2. row *****

type: technologie
url: http://www.lemonde.fr/aujourd-hui/article/2008/08/07/en-france-des-
adversaires-des-jo-s-expriment-en-video-sur-la-toile_1081170_3238.html?xtor=
RSS-651865
```

### Listing 9. Remplacement de données XML avec updatexml

```
SELECT updatexml(load_file('c:/anaska.xml'), '/anaska/formateur[2]/prenom',
      '<prenom>aka XML fan</prenom>') AS result \G
***** 1. row *****

result:
<anaska>
  Les formateurs de l'équipe sont :
  <formateur domaines="MySQL PHP">
    <nom>Dasini</nom>
    <prenom>Olivier</prenom>
  </formateur>
  <formateur domaines="PHP XML">
    <nom>Allard</nom>
    <prenom>Aka XML fan</prenom>
  </formateur>
  <formateur domaines="Linux MySQL">
    <nom>Dumont</nom>
    <prenom>Pierre</prenom>
  </formateur>
  et sont passionnés par l'open source.
</anaska>
/*Le fichier XML original n'est pas modifié*/
```

# Closures

## et traits

À la suite de l'article sur les nouveautés de PHP 5.3, nous allons découvrir 2 nouvelles fonctionnalités qui devraient prochainement voir le jour dans PHP.

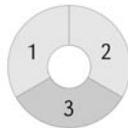
### Cet article explique :

- Les fonctions lambda et les closures.
- Les traits.

### Ce qu'il faut savoir :

- Bonnes notions de la programmation orientée objet.
- Connaissance générale des fonctionnalités de PHP.

### Niveau de difficulté



La première nouveauté que nous allons examiner est d'ores et déjà intégrée à PHP 5.3 : il s'agit des *closures* et des fonctions `lambda` (ou fonctions anonymes). Fin 2007, un *patch* avait déjà été proposé pour les fonctions `lambda`, mais certaines personnes trouvant dommage de ne pas supporter également les *closures*, le *patch* avait été rejeté. À l'initiative de Christian Seiler et Dimitri Stogov, une RFC a été écrite afin de décrire une méthode d'implémentation complète, et le *patch* a été modifié en fonction.

Les *closures* et fonctions `lambda` sont issues des langages de programmation fonctionnelle : le concept a été développé dans les années 60 avant d'être implémenté pour la première fois dans le langage *Scheme*. Ces constructions sont très utilisées dans tous les langages où les fonctions sont à l'honneur, c'est-à-dire les langages où les fonctions peuvent être passées en argument, retournées par un appel de fonction, placées dans des variables, etc... Nous pouvons citer *Lisp*, *Scheme*, *ML*, *Haskell*, *Smalltalk*, et plus proche de nos préoccupations, *Javascript*. Dans ce type de langage, le concept d'objet est souvent implémenté à l'aide des *closures*.

Examinons maintenant la différence entre les fonctions `lambda` et les *closures*. Une fonction `lambda` est tout simplement une fonction

jetable, définie dans un certain contexte mais qui n'est pas utilisée par ailleurs. Pour en comprendre l'intérêt, il suffit de penser aux fonctions de PHP prenant une fonction de *callback* en argument : `preg_replace_callback()`, `array_map()`, `array_reduce()`, `array_filter()`, `array_walk()`, `usort()`, etc... Afin de les utiliser, vous devez recourir à l'un des 3 stratagèmes suivants :

- Définir la fonction de *callback*... quelque part ! Ce qui ne favorise pas la lisibilité du code si la fonction est définie 300 lignes plus loin que son contexte d'utilisation,
- Définir la fonction de *callback* juste avant son utilisation, et si nécessaire, à l'intérieur d'une autre fonction. Cela implique de devoir vérifier que la fonction n'est pas déjà définie, ce qui diminue là encore la lisibilité du code,
- Utiliser `create_function()` pour définir la fonction en temps réel. La lisibilité est ici minimale (pas de coloration syntaxique), et la compilation se faisant également en temps réel, les performances risquent d'être fortement impactées si cette astuce est trop souvent utilisée.

Afin d'illustrer ces 3 stratégies, vous trouverez dans le Listing 1 l'implémentation d'une fonction `array_cube()` dont le but est, comme son nom l'indique, d'élever au cube tous les éléments d'un tableau. La dernière implémentation utilise les fonctions `lambda` : vous noterez qu'elle est légèrement plus lisible et indéniablement plus propre.

Les fonctions `lambda` sont utiles, mais présentent un sérieux inconvénient : elle n'ont pas accès au *scope*, c'est-à-dire à l'environnement lexical dans lequel elles sont définies. Le code suivant, par exemple, ne fonctionne pas comme on aurait pu l'espérer :

```
function additionneur($nombre) {
    return function($valeur) {
        return $valeur + $nombre;
    }
}
$additionneur = additionneur(10);
echo $additionneur(1); // retourne 1,
pas 11 !!!
```

La fonction `lambda` n'a pas accès au *scope* de la fonction `additionneur()`, et n'a donc pas accès à la valeur de `$nombre`. Pour avoir accès à une partie de cet environnement lexical, nous devons utiliser une *closure*. Notez d'ailleurs qu'ici l'implémentation de PHP diffère fortement des implémentations de la plupart des langages fonctionnels. En effet, dans ces langages, la création d'une *closure* est automatique dès lors que la fonction anonyme fait référence à des variables locales ou des arguments de la fonction (ou méthode) dans laquelle elle est définie. Dans le cas de PHP, nous devons spécifier explicitement les variables que nous voulons importer du *scope* parent vers le *scope* de la *closure*, à l'aide du mot réservé `use`. La fonction `additionneur()` s'écrit donc ainsi :

```
function additionneur($nombre) {
    return function($valeur) use ($nombre) {
        return $valeur + $nombre;
    }
}
$additionneur = additionneur(10);
echo $additionneur(1); // retourne 11
```

En résumé, les *closures* sont donc des blocs de code possédant des liens vers l'environnement

lexical duquel elles proviennent. La syntaxe des *closures* et des fonctions `lambda` est la suivante :

```
function & (parameters) use (lexical vars)
{ block }
```

Le `&` est bien sûr optionnel et indique que la fonction doit retourner une référence. Le `use(...)` est également optionnel et spécifie les variables à importer dans la *closure*. Si l'on affecte une *closure* à une variable :

```
$lambda = function() { echo 'Hello world';
};
```

Cette variable `$lambda` contient une ressource que l'on peut appeler de différentes façons :

```
$lambda();
call_user_func($lambda);
call_user_func_array($lambda, array());
```

Par défaut, les variables importées à l'aide de `use(...)` sont copiées, mais il est possible de les importer sous forme de références en ajoutant un `&` devant le nom de la variable. Une fonctionnalité intéressante lorsque la *closure* est définie à l'intérieur d'un objet est que la *closure* a un accès complet aux propriétés et méthodes de l'objet via `$this`, même si elles sont protégées ou privées. Le Listing 2 vous montre un exemple d'utilisation.

Maintenant que nous avons vu la théorie, il est temps de nous intéresser aux applications pratiques. Dans les autres langages, les *closures* sont souvent utilisées par les créateurs de librairie afin de permettre aux utilisateurs de personnaliser le comportement de certaines fonctions en leur fournissant une *closure* en argument. Il est donc possible que certaines librairies ou extensions PHP adoptent peu à peu les *closures*. Un exemple qui me vient à l'esprit est PHP-GTK : pouvoir utiliser des *closures* comme fonctions de *callback* serait plutôt élégant, comme vous le montre l'exemple 1 du Listing 3. Les *closures* (en fait les fonctions `lambda`) sont également très pratiques dès lors qu'il s'agit d'opérer un tri dans une collection. Imaginez une classe Entreprise encapsulant une collection d'employés. Nous voulons disposer de méthodes `managers()` et `developers()` qui retournent une sous-collection d'employés. Cela implique une certaine redondance dans le code de ces méthodes, puisque l'on doit utiliser une *foreach* sur les employés dans chacune d'entre elles (cf exemple 2 du Listing 3). Avec les *closures*, nous pouvons créer une classe parente Collection avec une méthode `select()` qui boucle sur le contenu de la collection, et le code de notre classe Entreprise devient moins redondant et plus lisible.

Enfin, les *closures* sont utiles dans un contexte transactionnel, c'est-à-dire dès lors que vous ne devez pas oublier de faire quelque chose à la fin d'une méthode. Cela peut être le traitement

d'un fichier, où il faut ouvrir le fichier, le traiter, et ne pas oublier de le fermer, ou bien une transaction de base de données, où il ne faut pas oublier le `commit` ou le *rollback* (Listing 4).

Bien sûr, il n'y a rien de très révolutionnaire là-dedans, il ne s'agit là que d'un idiome pratique. De manière générale, les *closures* ne seront de toute façon pas à l'origine d'une révolution

de PHP. Il s'agit juste d'une fonctionnalité intéressante, un outil de plus qui dans certains cas bien précis peut vous permettre d'écrire un code plus élégant et plus facile à maintenir.

## Traits

Intéressons-nous maintenant à une nouveauté qui n'a pas encore atterri dans le code source de

### Listing 1. Possibilités d'implémentation des fonctions de callback

```
// Implémentation 1
function cube($n) {
    return ($n * $n * $n);
} function array_cube($array) {
    return array_map('cube', $array);
}

// Implémentation 2
function array_cube($array) {
    if (!function_exists('cube')) {
        function cube($n) {
            return ($n * $n * $n);
        }
    }
    return array_map('cube', $array);
}

// Implémentation 3
function array_cube($array) {
    $cube = create_function('$n', 'return ($n * $n * $n);');
    return array_map($cube, $array);
}

// Implémentation utilisant une fonction lambda
function array_cube($array) {
    $lambda = function($n) {
        return ($n * $n * $n);
    };
    return array_map($lambda, $array);
}

// Utilisation
$a = array(1, 2, 3, 4, 5);
$b = array_cube($a); // $b = array(1, 8, 27, 64, 125);
```

### Listing 2. Exemple de closure définie au sein d'un objet

```
class Employe {
    private $nom;
    private $salaire;
    public function __construct($nom, $salaire) {
        $this->nom = $nom;
        $this->salaire = $salaire;
    } public function setSalaire($salaire) {
        $this->salaire = $salaire;
    } public function paidLess() {
        return function($autre_salaire) {
            return $this->salaire > $autre_salaire;
        };
    }
}

$emp = new Employe('John', 2000);
$paidLess = $emp->paidLess();
$bool = $paidLess(1500); // true
$emp->setSalaire(1200);
$bool = $paidLess(1500); // false
```

## Listing 3. Exemples d'utilisation des closures

```
// Exemple 1 : utilisation d'une closure comme callback dans PHP-GTK
$win = new GtkWindow();
$but = new GtkButton('Quitter');
$win->add($but);
$win->show_all();
$but->connect_simple('clicked', function() { Gtk::main_quit(); });
Gtk::main();

// Exemple 2 : utilisation des closures comme critères de tri d'une collection
// Sans closures
class Entreprise { private $employees;
    public function __construct($employees) { $this->employees = $employees;
    } public function managers() { $managers = array();
        foreach ($this->employees as $e) { if ($e->isManager()) $managers[] = $e;
        }
        return $managers;
    } public function developers() {
        $developers = array();
        foreach ($this->employees as $e) { if ($e->isDeveloper()) $developers[] = $e;
        }
        return $developers;
    }
}

// Avec closures
class Collection {
    private $objects;
    public function __construct($objects) { $this->objects = $objects;
    } public function select($closure) { $result = array();
        foreach ($this->objects as $o) { if ($closure($o) === true) $result[] = $o;
        }
        return $objects;
    }
}
class Entreprise extends Collection {
    public function managers() { return $this->select(
        function($employee) { return $employee->isManager(); });
    } public function developers() {
        return $this->select(
            function($employee) { return $employee->isDeveloper(); }
        );
    }
}
```

## Listing 4. Utilisations possibles des closures

```
class File {
    static function open($filename, $mode, $closure) {
        $file = fopen($filename, $mode);
        $closure($file);
        fclose($file);
    }
}

File::open('test.txt', 'r',
    function($file) {
        ...
    }
);

Db::transaction(
    function($db) {
        ...
    }
);
```

PHP, mais qui devrait faire son apparition dans une éventuelle version 5.4 : les *traits*. Il s'agit d'un système destiné à faciliter la réutilisation de code dans les langages à héritage simple comme PHP. En effet, la réutilisation du code est un des buts principaux de la programmation orientée objet, but qu'elle s'efforce d'atteindre à l'aide de l'héritage. Mais dans le cas de l'héritage simple, la seule solution pour réutiliser du code est de déplacer les méthodes correspondantes plus haut dans la hiérarchie des classes (les classes *parentes*), ce qui rend le code plus difficile à comprendre et donc à maintenir. Afin de contourner ce problème, C++ emploie l'héritage multiple, tandis que *Ruby* propose les *mixins*, mais ces mécanismes sont complexes et peuvent engendrer des comportements surprenants. Les *traits* sont eux conçus pour encapsuler un ensemble de méthodes utilisées librement dans plusieurs classes indépendantes. Ils ont été intégrés ou sont près de l'être dans de nombreux langages dont *Perl 6*, *Scala*, *Java* et *C#*.

Afin de vous aider à mieux comprendre l'intérêt des *traits*, je vais devoir parler de *Ruby On Rails* (non, ne partez pas tout de suite ;). Comme vous le savez sans doute, Rails propose une implémentation du motif *ActiveRecord* particulièrement élégante, et surtout extensible via des *plugins acts as*. Ces plugins, qui sont en fait des *mixins*, proposent des comportements additionnels aux classes héritant de *ActiveRecord*. Par exemple, *acts\_as\_list* permet de gérer les enregistrements d'une table comme une liste ordonnée. Il suffit d'écrire *acts\_as\_list* dans la définition d'une classe, et cette classe disposera de méthodes supplémentaires permettant de modifier l'ordre des items, comme *move\_higher* ou *move\_to\_bottom*:

```
class TodoItem < ActiveRecord::Base
    acts_as_list
end

todo_item.move_to_bottom
todo_item.move_higher
```

On trouve ainsi un grand nombre de plugins, comme *acts\_as\_tree* ou *acts\_as\_nested\_set* pour gérer un arbre d'enregistrements, *acts\_as\_taggable*, *acts\_as\_attachment*, *acts\_as\_versioned*, etc... Pour faire ce genre de choses avec PHP, la seule solution serait d'avoir toute une hiérarchie de classes héritant d'une classe *ActiveRecord*, comme *ActsAsListActiveRecord*, *ActsAsTreeActiveRecord* et de faire hériter sa classe métier de l'une d'entre elles. Le problème est, outre la lourdeur de cette solution, que cela ne permet pas d'intégrer deux comportements à une même classe. L'avantage de la solution de Rails est qu'une classe peut à la fois déclarer *acts\_as\_list* et *acts\_as\_tree*, et ainsi avoir des enregistrements organisés

en arbre dont les branches sont ordonnées ! Les *traits* nous permettraient d'implémenter une solution très proche de celle de Rails.

Voyons maintenant comment fonctionneraient les *traits* en PHP. Vous trouverez dans le Listing 5 des exemples de code pour toutes les

fonctionnalités décrites. Le premier exemple vous montre la syntaxe de base. Comme vous pouvez le constater, l'utilisation du mot réservé `use` fait que les méthodes définies dans le *trait* sont injectées dans la classe. Notez que tout se fait pendant la compilation du code PHP : les méthodes des *traits* sont directement ajoutées dans les classes, comme un copier/coller automatique. Il n'y donc pas de notion de *trait* en *runtime*.

Certaines règles de précedence existent au niveau des *traits* : si une classe définit une méthode existant déjà dans le *trait* qu'elle utilise, la méthode de la classe surclassera celle du *trait*. Par contre, si la méthode n'est pas définie dans la classe déclarant le *trait*, mais dans une classe parente, la méthode du *trait* prévaut. Étant donné qu'il est évidemment possible d'utiliser plusieurs *traits* dans une même classe comme le montre l'exemple 2, il fallait également trouver un moyen de gérer les conflits de méthodes entre *traits*. Plutôt que d'implémenter un algorithme de résolution de conflits complexe et dont les résultats pourraient potentiellement dérouter les utilisateurs, les développeurs ont choisi de laisser le choix à l'utilisateur. Concrètement, en cas de conflit entre deux méthodes de deux *traits*, PHP générera une erreur signalant le conflit et les méthodes en cause. L'utilisateur disposera alors de la syntaxe présentée dans l'exemple 3 pour régler le conflit. `B::helloWorld` instead `A::helloWorld` spécifie à PHP qu'il doit employer la méthode du *trait* `B`, tandis que `B::helloWorld as hello` permet de créer un alias `hello()` de la méthode `helloWorld()`.

Étant donné que les *traits* ne possèdent pas de propriétés, il est nécessaire de disposer d'un moyen de spécifier certains *besoins* du *trait* afin qu'il puisse interagir avec les propriétés et l'état de l'objet dans lequel il est utilisé. Pour cette raison, il est possible de définir des méthodes abstraites dans les *traits*, que les classes doivent implémenter afin que le *trait* puisse être utilisé. L'exemple 4 vous montre comment procéder.

Tout comme les *closures*, les *traits* représentent donc une nouvelle fonctionnalité intéressante, qui permettraient aux développeurs PHP d'éviter le copier/coller et de coder d'une façon plus élégante. Encore une fois, cela ne risque pas de bouleverser votre quotidien, mais il est clair que tous les développeurs qui ont eu un jour à gérer un code source imposant seront satisfaits de disposer de ce genre d'outil.

### Listing 5. Syntaxe et utilisation des Traits

```
// Exemple 1 : syntaxe et utilisation de base
trait HelloWorld {
    public function helloWorld() {
        echo 'Hello world !';
    }
}
class Foo { use HelloWorld; }
class Bar { use HelloWorld; }

$foo = new Foo();
$foo->helloWorld();
$bar = new Bar();
$bar->helloWorld();

// Exemple 2 : traits multiples
trait Hello {
    public function hello() {
        echo 'Hello';
    }
}
trait World {
    public function world() {
        echo ' world !';
    }
}
class Foo {
    use Hello, World;
}

$foo = new Foo();
$foo->hello();
$foo->world();

// Exemple 3 : résolution de conflits
trait A {
    public function helloWorld() {
        echo 'Hello world !';
    }
}
trait B {
    public function helloWorld() {
        echo 'Hello world :)';
    }
}
class Foo {
    use A, B {
        B::helloWorld instead A::helloWorld,
        B::helloWorld as hello
    }
}

// Exemple 4 : expression des "prérequis" d'un trait
trait Hello {
    public function say() {
        echo 'Hello '.$this->getName();
    }
    abstract public function getName();
}
class User {
    use Hello;
    private $name;
    public function __construct($name) {
        $this->name = $name;
    }
    public function getName() {
        return $this->name;
    }
}
```

### RAPHAËL ROUGERON

Raphaël Rougeron travaille au sein du groupe Linagora (<http://www.linagora.com>) sur des projets web de développement et d'intégration de briques logicielles libres. Il est également le créateur du framework Stato (<http://www.stato-framework.org>), publié sous licence MIT.

Pour contacter l'auteur : [goldoraf@gmail.com](mailto:goldoraf@gmail.com)

Son blog : <http://www.rafael-rougeron.com>

# Sécurité

## avec PDO

La sécurité, vous en voyez partout dans tous les sens, mais ne vous y trompez pas, le point le plus important concerne les valeurs entrantes et le format PDO est là et propose une fonction pour vous faciliter la vie entre votre page et la communication vers votre base de données.

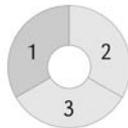
### Cet article explique :

- Protéger des valeurs entrantes.
- Utiliser les fonctions.
- Utiliser les classes.

### Ce qu'il faut savoir :

- Avoir lu le numéro précédent sur PDO.
- Avoir quelques connaissances sur la sécurité en général.

### Niveau de difficulté



Lorsque vous manipulez des chaînes de caractères, vous pouvez rencontrer des chaînes avec des caractères qui risquent de gêner l'affichage. Ces chaînes de caractères proviennent de données arrivant de formulaires ou d'entrée de valeurs.

Les gênes concernent entre autres l'utilisation des guillemets ou des apostrophes et provoquent des perturbations dans l'affichage et la connexion avec votre base de données.

Pour remédier à ce phénomène, PDO propose la fonction `PDO->QUOTE` qui permet de sécuriser votre chaîne de caractères.

La fonction `PDO->quote` reçoit en arguments :

```
PDO->quote (
    string, // chaîne à échapper
    paramètre // fournit un
    // type de données
)
```

Le premier critère correspond à la valeur à protéger.

Le deuxième critère correspond aux options :

- `PDO::PARAM_STR` : Chaîne de caractères,
- `PDO::PARAM_INT` : Integer de SQL,
- `PDO::PARAM_NULL` : NULL de SQL,

- `PDO::PARAM_BOOL` : Booléen,
- `PDO::PARAM_LOB` : Objet large de SQL.

Ce paramètre n'est pas obligatoire pour l'utilisation de la fonction. Si vous ne mettez rien la valeur utilisée par défaut sera `PDO::PARAM_STR`.

Pour illustrer cette fonction, vous allez voir 3 exemples montrant les possibilités de cette fonction. Le terme employé sera chaîne échappée ou chaîne non échappée. Avant de commencer, vous devez établir une connexion au format *PDO* :

```
<?php
$cnx = new PDO('
    mysql:host=localhost; // serveur
    port=3306; // port
    dbname=phpsolution', // nom de la
    // base de données
    'root', // login
    ''); // mot de passe
?>
```

Vous allez à partir d'une chaîne de caractères simples, ajouter des quotes pour en délimiter son utilisation.

```
<?php
$chaîne = "Bonjour";
echo "Chaîne non échappée : $chaîne<br>";
echo "Chaîne échappée : " .
    $cnx->quote($chaîne) . "<br>";
?>
```

Pour obtenir le résultat suivant :

```
Chaîne non échappée : Bonjour
Chaîne échappée : 'Bonjour'
```

Le résultat montre que des quotes ont été rajoutées si votre variable n'en possède pas. Comme ceci, notre chaîne est délimitée et sécurisée.

Mais si notre chaîne possède déjà des apostrophes, elle peut être dangereuse et contenir des attaques touchant la sécurité.

```
<?php
$chaîne = "L'illustration d'une
    chaîne";
echo "Chaîne non échappée : $chaîne<br>";
echo "Chaîne échappée : " .
    $cnx->quote($chaîne) . "<br>";
?>
```

Pour obtenir le résultat suivant :

```
Chaîne non échappée : L'illustration d'une
    chaîne
Chaîne échappée : 'L\'illustration d'une
    chaîne'
```

Bien sur, si vous envoyez des informations entrantes avec une chaîne très complexe, ce qui se trouve être très possible, vous pouvez avoir un exemple comme ceci :

```
<?php
$chaîne = "L'illustration d'une chaîne
    \"complexe\" pour vérifier";
echo "Chaîne non échappée : $chaîne
    <br>";
echo "Chaîne échappée : " . $cnx->
    quote($chaîne) .
    "<br>";
?>
```

Pour obtenir le résultat suivant :

```
Chaîne non échappée : L'illustration d'une
    chaîne "complexe" pour vérifier
Chaîne échappée : 'L\'illustration d'une
    chaîne \"complexe\" pour vérifier'
```

Cette fonction est à prendre avec des pincettes. Elle permet de rajouter des quotes dans une chaîne de caractères si celle-ci n'en possède pas, surtout si vous communiquez avec votre base de données.

Si vous utilisez cette fonction directement pour vérifier ou contrôler des valeurs entrantes, comme des données venant d'un formulaire, il vaut mieux ne pas l'utiliser seule car vous pouvez être confronté à des attaques XSS ou sous la forme d'injection SQL.

Par contre, si vous avez effectué tous les tests nécessaires au niveau de la sécurité en amont, vous pouvez imposer l'utilisation des quotes comme ceci cette fonction sécurise vos paramètres qui ont été employés dans votre requête SQL.

## Exemple du jour

Aujourd'hui, pour changer un peu des autres exemples déjà utilisés dans les précédents numéros concernant le PDO, vous allez voir l'utilisation de requêtes testant les connexions avec un identifiant et un mot de passe.

Avant de commencer, vous devez créer une base de données (ici ce sera phpsolution) et vous allez trouver les tables et données dans un petit source SQL vous permettant ainsi de pouvoir continuer l'article (Listing 1 – Données du jour en SQL). Dans notre exemple, l'article ne parlera pas de codification du mot de passe, ni de l'utilisation des techniques pour encoder / décoder des caractères ou de la fonction MD5.

## Injection SQL

Ce point a été traité dans les anciens numéros de PHP solution, il s'agit plutôt d'un petit rappel.

Une injection SQL correspond à l'utilisation d'un code malicieux dans votre code sans que vous lui en donniez les droits.

Cette injection est possible par les failles XSS (*Cross Site Scripting*).

Si vous ne protégez pas vos requêtes SQL, voici le genre d'attaque auquel vous allez être confronté.

Dans l'exemple (Listing 2. Exemple d'injection SQL), vous allez envoyer des informations sans connaître le code d'identification ni le mot de passe, pour obtenir le script suivant.

Pour obtenir le résultat suivant :

```
SELECT id FROM user WHERE identifiant =
    '' OR 1 AND motpasse =
    '' OR 1
Ligne Trouvée
```

Le résultat montre que sans connaître le mot de passe et son identifiant, l'enregistrement a été trouvé et donc toute manipulation peut provoquer des actions dont vous n'avez pas le contrôle.

Il s'agit d'un exemple parmi tant d'autres auquel vous pouvez être confronté, car vous pouvez utiliser les critères identifiant et mot de passe dans les exemples qui vont suivre et vous obtiendrez un résultat différent.

## Sécuriser vos requêtes

Quelques lignes au dessus, vous avez vu la sécurisation d'une chaîne et vous avez vu aussi un aspect de trou de sécurité qui peut être résolu par différentes manières.

Mais ce qui vous intéresse, concerne l'utilisation de cette fonction `PDO::QUOTE` avec l'exécution d'une requête SQL PDO.

Il existe 2 possibilités pour l'utilisation de cette fonction :

- Les requêtes sans préparation,
- Les requêtes préparées.

### Les requêtes sans préparation

Les requêtes sans préparation utilisent la fonction `QUERY`. Cette requête se présente sous la forme suivante :

- Connexion,
- Exécution d'une requête SQL,
- Retourne un jeu de résultats.

### Listing 1. Données du jour en SQL

```
CREATE TABLE 'blocnote' (
    'id' int(11) NOT NULL auto_increment,
    'iduser' tinyint(5) NOT NULL,
    'date' date NOT NULL,
    'remarque' varchar(255) NOT NULL,
    PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

INSERT INTO 'blocnote' ('id', 'iduser', 'date', 'remarque') VALUES
(1, 1, '2008-09-01', 'Ouverture compte'),
(2, 1, '2008-09-02', 'Mise A jour dossier'),
(3, 2, '2008-09-15', 'Ouverture compte'),
(4, 2, '2008-09-15', 'Mise A jour dossier');

CREATE TABLE 'user' (
    'id' tinyint(5) NOT NULL auto_increment,
    'identifiant' varchar(20) NOT NULL,
    'motpasse' varchar(20) NOT NULL,
    PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

INSERT INTO 'user' ('id', 'identifiant', 'motpasse') VALUES
(1, 'admin', 'admin'),
(2, 'user', 'user');
```

### Listing 2. Exemple d'injection SQL

```
<?php
try {
    $cnx = new PDO('mysql:host=localhost;port=3306;dbname=phpsolution', 'root', '');
    $cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $identifiant="'' OR 1";
    $motpasse="'' OR 1";
    $sql = "SELECT id FROM user WHERE identifiant = $identifiant AND motpasse =
        $motpasse";
    echo $sql."<br>";
    $result = $cnx->query($sql);

    if ($result->rowCount(>0)
        echo "Ligne Trouvée";
    else
        echo "Aucune Données";
    $cnx = null;
} catch(PDOException $e) {
    echo $e->getMessage();
}
```

**Listing 3. Les requêtes sans préparation**

```
<?php
try {
    $cnx = new PDO('
mysql:host=localhost; // serveur
port=3306; // port
dbname=phpsolution', // nom de la base de données
'root', // login
''); // mot de passe
$cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$identifiant="admin";
$motpasse="admin";
$identifiant=$cnx->quote($identifiant, PDO::PARAM_STR);
$motpasse=$cnx->quote($motpasse, PDO::PARAM_STR);
$sql = "SELECT id FROM user WHERE identifiant = $identifiant AND motpasse =
    $motpasse";
$result = $cnx->query($sql);

if ($result->rowCount()>0)
    echo "Ligne Trouvée";
else
    echo "Aucune Données";
$cnx = null;
} catch(PDOException $e) {
    echo $e->getMessage();
}
?>
```

**Listing 4. Les requêtes préparées**

```
<?php
$login="admin";
$motpasse="admin";

try {
    $cnx = new PDO('
mysql:host=localhost; // serveur
port=3306; // port
dbname=phpsolution', // nom de la base de données
'root', // login
''); // mot de passe
$cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$identifiant="admin";
$motpasse="admin";
$identifiant=$cnx->quote($identifiant, PDO::PARAM_STR);
$motpasse=$cnx->quote($motpasse, PDO::PARAM_STR);
$sql = "SELECT id FROM user WHERE identifiant = $login AND motpasse = $motpasse";
$qid=$cnx->prepare($sql);
$qid->execute();

if ($qid->rowCount()>0)
    echo "Ligne Trouvée";
else
    echo "Aucune Données";
$cnx = null;
} catch(PDOException $e) {
    echo $e->getMessage();
}
?>
```

Lorsque vous mettez cette technique en application, vous obtenez le source se trouvant dans le Tableau 1 dans la première colonne (simple saisie).

Le résultat que vous obtenez correspond à celui-ci :

```
SQLSTATE[42S22]: Column not found: 1054
    Unknown column 'admin' in
    'where clause'
```

Ce message que vous voyez, signale qu'il n'a pas trouvé d'enregistrement et par conséquent n'a pas pu compter le nombre de lignes.

Le problème apparaît car il manque les quotes dans les valeurs entrantes, c'est à dire \$login et \$motpasse.

Pour remédier à ce phénomène, il suffit de rajouter des quotes à notre exemple, c'est à dire mettre sous cette forme : \$login et \$motpasse. Comme cela le nouveau code source se trouve dans le Tableau 1 pour la deuxième colonne (*Saisie Indispensable*).

Pour obtenir le résultat suivant :  
Connexion réussie

Bien sur, si vous ne fournissez pas les bonnes valeurs, vous recevrez le résultat suivant :

Connexion échouée

Cependant l'ajout des quotes à la main ne corrige pas tous les problèmes d'attaques pouvant exister. Un autre type d'attaque peut se présenter à vous sous la forme suivante :

```
$identifiant="" OR ";
$motpasse="" <> '";
```

et donne :

```
$sql= "SELECT id FROM user
    WHERE identifiant = '' OR
    ' AND motpasse = ' <> ' '";
```

Le résultat obtenu :

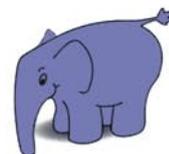
```
connexion réussie
Ligne Trouvée
```

Par conséquent, même avec les ' on peut avoir des failles de sécurité. Pour pallier à ce problème, il est possible d'imposer l'utilisation des quotes avec la fonction PDO::QUOTE(). Comme ceci vous obtenez une sécurité supplémentaire pour votre requête SQL.

Cette sécurité se trouve illustrée par le Listing 3 (Listing 3 – Les requêtes sans préparation).

Le résultat montre que la ligne est réussie si vous mettez les bons identifiants. Vous pouvez toutefois mettre les deux exemples d'attaques et vous verrez le résultat vous montrant

# VISITEZ NOTRE SITE INTERNET



[WWW.PHPOLMAG.ORG/FR](http://WWW.PHPOLMAG.ORG/FR)

## *Vous y trouverez*

les articles les plus  
intéressants à télécharger

listings, outils indispensables

forum

actualités, informations sur  
les prochains numéros

qu'aucun enregistrement n'est trouvé. L'exemple (Listing 3) vous a montré que vous pouvez éviter certains problèmes juste avec certaines fonctions qui sont proposées en standard.

**Les requêtes préparées**

Cette partie peut donner l'impression de doublon avec la partie précédente, mais étant donné que vous pouvez utiliser différentes méthodes d'interrogation, il est nécessaire d'en montrer toutes les possibilités.

Les requêtes préparées sont des requêtes possédant une très bonne approche au niveau de la sécurité et surtout une utilisation plus claire pour l'exécution d'une requête SQL. Une requête préparée s'utilise sous la forme suivante :

- Connexion,
- Préparation d'une requête SQL,
- Exécution de la requête préparée,
- Retourne un jeu de résultats.

Lorsque vous mettez cette technique en application, vous obtenez le code source se situant dans le Tableau 2 de la colonne 1 (simple saisie).

Le résultat obtenu sera :

```
SQLSTATE[42S22]: Column not found: 1054
Unknown column 'admin' in
'where clause'
```

Nous obtenons le même résultat qu'avec la partie précédente car nous venons d'interroger de la même façon la fonction SQL. Le message que vous voyez, signale qu'il n'a pas trouvé d'enregistrement et par conséquent n'a pas pu compter le nombre de lignes. Le problème apparaît parce qu'il manque les quotes dans les valeurs entrantes, c'est à dire \$login et \$motpasse.

Pour remédier à ce phénomène, il suffit de rajouter des quotes à notre exemple, et plus exactement il suffit de mettre sous la forme suivante \$login et \$motpasse. Le nouveau code source se trouve dans le Tableau 2 pour la deuxième colonne (saisie indispensable).

Pour obtenir le résultat suivant :  
Connexion réussie

Bien sûr, si vous ne fournissez pas les bonnes valeurs, vous recevrez le résultat suivant :

Connexion échouée

Vous pouvez comme précédemment imposer l'utilisation des quotes et vous obtenez un niveau de sécurité nettement meilleur. Vous pouvez voir le Listing 4. Les requêtes préparées

Pour obtenir le résultat suivant :

Ligne Trouvée

Vous pouvez aussi avec le Listing 4, utiliser les différentes attaques possibles qui ont été décrites plus haut pour voir que le résultat obtenu sera différent.

Concernant le choix que vous allez retenir entre les requêtes préparées et celles qui ne le seront pas, il faut penser au format de la base de données, car certains formats n'acceptent

**Tableau 1.** Requêtes sans préparation

	Simple saisie	Saisie indispensable
Connexion	<?php try { \$cnx = new PDO('mysql:host=localhost;port=3306;dbname=phpsolution','root',''); \$cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); echo "connexion réussie "; }	
Critères de filtres	\$identifiant="admin"; \$motpasse="admin";	
Requête SQL	\$sql = "SELECT id FROM user WHERE identifiant = \$identifiant AND motpasse = \$motpasse";	\$sql = "SELECT id FROM user WHERE identifiant = '\$identifiant' AND motpasse = '\$motpasse'";
Exécution	\$result = \$cnx->query(\$sql);	
Affichages	if (\$result->rowCount()>0) echo "Ligne Trouvée"; else echo "Aucune Données";	
Déconnexion	\$cnx = null; }	
Récupération d'erreurs	catch(PDOException \$e) { echo \$e->getMessage(); }	
Résultat affiché	connexion réussie SQLSTATE[42S22]: Column not found: 1054 Unknown column 'admin' in 'where clause'	connexion réussie Ligne Trouvée

**Tableau 2.** Les requêtes préparées

	Simple saisie	Saisie indispensable
Connexion	<?php try { \$cnx = new PDO('mysql:host=localhost;port=3306;dbname=phpsolution','root',''); \$cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); echo "connexion réussie "; }	
Critères de filtres	\$identifiant="admin"; \$motpasse="admin";	
Requête SQL	\$sql = "SELECT id FROM user WHERE identifiant = \$identifiant AND motpasse = \$motpasse";	\$sql = "SELECT id FROM user WHERE identifiant = '\$identifiant' AND motpasse = '\$motpasse'";
Préparation	\$qid=\$cnx->prepare(\$sql);	
Exécution	\$qid->execute();	
Affichages	if (\$qid->rowCount()>0) echo "Ligne Trouvée"; else echo "Aucune Données";	
Déconnexion	\$cnx = null; }	
Récupération d'erreurs	catch(PDOException \$e) { echo \$e->getMessage(); }	
Résultat affiché	connexion réussie SQLSTATE[42S22]: Column not found: 1054 Unknown column 'admin' in 'where clause'	connexion réussie Ligne Trouvée

# Rejoignez le Club .PRO

Pour plus de renseignement : [editor@phpsolmag.org](mailto:editor@phpsolmag.org)



## Stonfield Inworld

Stonfield Inworld propose aux entreprises des solutions globale d'intégration d'Internet et des Univers Virtuels dans leur stratégie de développement. Au-delà de ses services, la société consacre 30% de ses ressources à des travaux de R&D sur le e-Commerce et le e-Learning dans les Mondes Virtuels.



## Anaska Formation

Anaska est le spécialiste des formations sur les technologies OpenSource. En partenariat avec MySQL AB, Mandriva, Zend et d'autres acteurs de la communauté, Anaska vous propose un catalogue de plus de 50 formations dédiés aux technologies du Libre.

<http://www.anaska.com>



## COGNIX Systems

Conseil, conception et développement d'applications évoluées pour les systèmes d'informations Internet/intranet/extranet. Alliant les compétences d'une SSII et d'une Web Agency, Cognix Systems conçoit des applicatifs et portails web à l'ergonomie travaillée et des sites Internet à forte valeur ajoutée.

<http://www.cognix-systems.com>



## WEB82

Création et hébergements de sites web pour particuliers, associations, entreprises, e-commerce. Développement entièrement aux normes W3C ([www.w3.org](http://www.w3.org)) de sites web de qualité, au graphisme soigné et employant les dernières technologies du web (PHP5, MySQL5, Ajax, XHTML, CSS2).

<http://www.web82.net>



## Core-Techs

Expert des solutions de gestion et de communication d'entreprise en Open Source, Core-Techs conçoit, intègre, déploie et maintient des systèmes de Gestion de Contenu Web, de Gestion Documentaire, de Gestion de la Relation Client (CRM), d'e-commerce et de travail collaboratif.

<http://www.core-techs.fr>



## POP FACTORY

PoP Factory, SSII spécialisée Web. Développement de solutions applicatives spécifiques ; offre de solutions packagées : catalogue numérique, e-commerce, livre/magazine numérique, envoi SMS. Nous accompagnons nos clients tout au long de leur projet : audit, conseil, développement, suivi et gestion.

<http://www.popfactory.com> / [info@popfactory.fr](mailto:info@popfactory.fr)



## Blue Note Systems

Spécialistes en CRM Open Source, nous proposons une offre complète de prestations sur la solution SugarCRM. Notre valeur ajoutée réside dans une expertise réactive et une expérience des problématiques de la GRC. Nous vous aidons à tirer le meilleur parti de votre solution CRM.

<http://www.bluenote-systems.com>

Club .PRO

Listing 5. Classe PDO

```
<?php
class classPDO {
    public function connexion() {
        $cnx = new PDO(
            'mysql:host=localhost;
            port=3306;
            dbname=phpsolution', 'root', '');
        $cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        return $cnx;
    } public function affiche() {
        return $this->id." - ".$this->motpasse;
    }
}
?>
```

Listing 6. Liste en double connexion

```
<?php
try {
    $cnx = new PDO('mysql:host=localhost;port=3306;dbname=phpsolution', 'root', '');
    $cnx->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "SELECT id,identifiant,motpasse FROM user";
    $qid=$cnx->prepare($sql);
    $qid->execute();
    $result = $qid->fetchAll();
    foreach($result as $row) {
        echo "<h2>".$row['id'].'-'. $row['identifiant'].'-'. $row['motpasse']. "</h2>";
        echo autre_cnx($row['id']);
    }
    $cnx = null;
} catch(PDOException $e) {
    echo $e->getMessage();
}
?>
```

Listing 7. Deuxième connexion dans une fonction

```
<?php
function autre_cnx($iduser) {
    try {
        $cnx2 = new PDO('mysql:host=localhost;port=3306;dbname=phpsolution', 'root',
            '');
        $cnx2->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $iduser=$cnx->quote($iduser, PDO::PARAM_STR);
        $sql2 = "SELECT * FROM blocnote WHERE iduser = $iduser ";
        $qid2=$cnx2->prepare($sql2);
        $qid2->execute();
        $result2 = $qid2->fetchAll();
        echo "<ul>";

        foreach($result2 as $row2) {
            echo $row2['date'].'-'. $row2['remarque']. "<br>";
        }
        echo "</ul>";
        $cnx2 = null;
    } catch(PDOException $e) {
        echo $e->getMessage();
    }
}
?>
```

pas la fonction `QUERY` (par exemple : utilisation du format `ODBC`). Par contre, les requêtes préparées sont utilisables par toutes les bases de données, c'est pourquoi il est préférable de choisir cette méthode.

## Éviter les répétitions

Lorsque vous décidez d'imposer l'utilisation des quotes, vous arrivez à penser qu'il faut répéter la fonction `PDO->quote` de nombreuses fois si vous avez de nombreux critères à insérer dans votre fonction SQL.

Pour éviter ce genre de manipulation assez longue, vous pouvez utiliser la fonction `Function`, qui va se présenter de la façon suivante :

```
<?php
    $login=pdo_quote($login);
    $motpasse=pdo_quote($motpasse);
    function pdo_quote($variable) {
        GLOBAL $cnx;
        return $cnx->quote($variable, PDO::
            PARAM_STR);
    }
?>
```

L'exemple qui vous est montré peut être utilisé à tout moment et pour l'ensemble des fonctions. C'est à vous maintenant de réaliser vos routines pour éviter de taper des lignes de programmation.

## La classe PDO

PDO n'échappe pas à la possibilité qui est offerte d'utiliser le format PDO avec une classe. Pour utiliser cette classe, la fonction à connaître et qui est mise à votre disposition est : `PDOStatement::setFetchMode()`.

Pour réaliser cette opération, vous devez créer une classe représentée par le Listing 5 – classe PDO.

Dans cette classe, deux fonctions ont été réalisées :

- Connexion,
- Affiche le résultat.

La fonction *connexion* permet d'établir une connexion avec votre base de données. Les champs se remplissent comme cela avait été expliqué dans les articles précédents. La fonction *affiche* permet d'afficher le numéro de la ligne avec le champ id et le mot de passe de la ligne concernée.

Lorsque votre classe est réalisée, vous pouvez commencer à réaliser les lignes de programmation pour afficher le résultat voulu.

Vous devez déclarer la classe que nous pouvons appeler `classPDO` :

```
<?php
    $classPDO = new classPDO;
?>
```

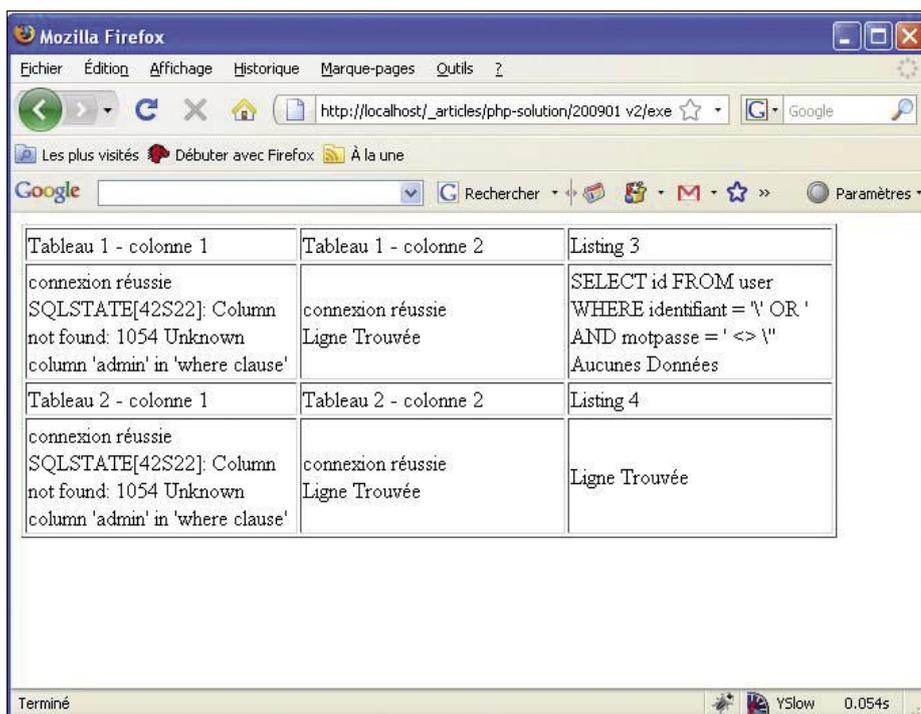


Figure 1. Comparaison des résultats obtenus

Il est nécessaire de faire appel à la fonction `connexion` qui a été déclarée dans la fonction `classPDO` :

```
<?php
try {
    $cnx = classPDO::connexion();
?>
```

Il est maintenant nécessaire de préparer et d'exécuter notre requête SQL :

```
<?php
$sql = "SELECT * FROM user";
$sqlid=$cnx->prepare($sql);
$sqlid->execute();
?>
```

Vous devez définir le mode de récupération avec la fonction `setFetchMode` qui a été préparée dans la classe avec la fonction `affiche` pour ensuite visualiser le résultat :

```
<?php
$sqlid->setFetchMode(PDO::FETCH_INT0, new
    classPDO);
foreach($sqlid as $classPDO) {
    echo $classPDO->affiche()."<br>";
}
?>
```

Il est obligatoire de fermer la connexion lorsque la connexion a réussi. Pour clore la connexion vous devez détruire l'objet. Ceci sera réalisé en signifiant `NULL` à la connexion. Si vous ne le faites pas, le langage PHP fermera automatiquement la connexion lorsque le script arrive à la fin de la page :

```
<?php
    $cnx = null;
} catch(PDOException $e) {
    echo $e->getMessage();
}
?>
```

Bien sûr le récupérateur d'erreur est présent dans notre exemple.

Le résultat que vous obtenez sera :

```
1 - admin
2 - user
```

## Cas particuliers

Comme dans la majorité des langages ou n'importe quelles méthodes que vous pouvez utiliser, il existe toujours des cas particuliers que vous pouvez rencontrer surtout pour les données provenant de formulaire.

Lorsque vous réalisez un site internet avec la présence d'un ou plusieurs formulaires, vous pouvez avoir besoin d'afficher différents critères qui ne peuvent pas être gérés dans la même requête SQL. Vous pouvez être confronté à ce genre de cas pour :

- Des listes déroulantes,
- Des cases à cocher,
- Afficher des listes de résultats,
- etc.

Ces options peuvent être alimentées avec d'autres tables de votre base de données. L'exemple retenu concernera l'affichage d'une liste par rapport aux identifiants. Pour afficher cette liste, vous utilisez une liste classique représentée sous la forme du

Listing 6. Liste en double connexion. Lors de l'affichage du résultat contenant la liste des comptes utilisateurs, vous pouvez appeler une autre table de votre base de données pour afficher par exemple une situation du compte. Cela se présentera sous la forme du Listing 7 (deuxième connexion dans une fonction).

Cette fonction se compose comme l'affichage d'une page classique, car vous devez aussi respecter un certain ordre :

- Connexion,
- Préparation de la requête SQL,
- Exécution,
- Affichage,
- Déconnexion.

Vous obtenez le résultat suivant :

- 1-admin-admin :

2008-09-01-Ouverture compte

2008-09-02-Mise A jour dossier

- 2-user-user :

2008-09-15-Ouverture compte

2008-09-15-Mise A jour dossier

Ici l'exemple peut-être utilisé dans une même requête, mais l'article montre qu'avec la norme PDO, on peut gérer plusieurs connexions en même temps et par conséquent être confronté aux différentes attaques dans tous les niveaux.

## Conclusion

Dans l'article de ce numéro, la sécurité est un point qu'il ne faut pas perdre de vue et cela mériterait de nombreuses lignes d'explications. Je vais le redire, il ne faut pas utiliser seulement ce qui est présenté ici car ce n'est pas suffisant. Il faut toujours utiliser l'ensemble des fonctions que vous connaissez et que vous utilisez pour avoir un niveau de protection assez élevé.

Bien sûr, les possibilités avec PDO sont nombreuses, surtout au niveau de la sécurité car il s'agit d'une couche supplémentaire. Mais en résumé, vous devez penser qu'il est possible de tout faire.

## CHRISTOPHE VILLENEUVE

Auteur du livre *PHP & MySQL-MySQLi-PDO, Construisez votre application, livre français aux Editions ENI. Il 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...*

Contactez l'auteur : <http://www.hello-design.fr>

# Introduction

## dans les arcanes d'OOP dans PHP5

Depuis sa 5ème version, PHP rejoint les rangs des autres Java, C++ et Python pour ne citer qu'eux en supportant lui aussi le modèle objet et donc l'OOP (Object-Oriented Programming), quelle réjouissante nouvelle pour les dresseurs d'éléphants bleus dont vous faites sûrement partie !

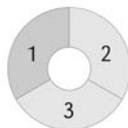
### Cet article explique :

- Les différents concepts objets exploitables dans les scripts PHP5.
- L'implémentation de ces concepts et quelques-unes de leurs utilisations possibles.

### Ce qu'il faut savoir :

- Les bases de la programmation PHP.

### Niveau de difficulté



Alors que PHP6 est en ce moment même en pleine préparation, les statistiques du site *Nexen* de juillet 2008 montrent très clairement que PHP5 tend à remplacer PHP4 sur les serveurs web. Il est donc grand temps si vous ne le faites pas déjà de profiter de cette 5ème version, notamment par la mise en œuvre de toute la puissance apportée par l'OOP au sein de celle-ci, avant la sortie imminente de la prochaine.

Le modèle objet a probablement été le sujet du plus grand nombre d'améliorations lors du passage de PHP4 à PHP5, conférant ainsi à ce langage toute la puissance offerte par les concepts de la programmation orientée objet (*aka* OOP pour *Object-Oriented Programming*), et mettant fin aux éventuels complexes qu'aurait pu avoir le PHP face à Java.

Sachez avant toute chose que le modèle objet va radicalement changer votre approche de la programmation, et qu'une fois adopté, vous aurez du mal à vous en séparer. En effet, celui-ci présente de nombreux atouts ! Réutilisabilité et indépendance des portions de code permettent une programmation modulaire, ainsi qu'une programmation bien moins floue la plupart du temps : si on peut le qualifier ainsi, le modèle objet une fois bien appliqué rendra votre code bien plus *humain*.

D'un point de vue théorique, un objet peut représenter absolument tout : une voiture, un utilisateur, un bâtiment, un elfe etc, d'un point de vue plus pragmatique, un objet est la composition de variables appelées propriétés ou attributs et d'actions qui lui sont associées sous forme de fonctions (appelées méthodes); dans vos applications PHP, un objet pourra représenter par exemple un membre, les résultats d'une requête SQL ou un *template*.

Pour créer des objets, vous aurez besoin de définir les modèles à partir desquels ils seront instanciés, on appelle ces modèles *classes*: ce sont en quelque sorte les moules de vos objets, et c'est grâce à eux que vous définirez les fonctionnalités de vos objets. Leur déclaration se fait grâce au mot-clé *class*, et l'opération d'instanciation (opération qui consiste à créer une instance à partir d'une classe) à l'aide du mot-clé *new*. Une instance est une variable ayant le comportement de l'objet que vous avez défini dans la classe dont il est issu: cette variable comprend des attributs et des méthodes (l'accès à ces membres est abordé un peu plus loin). Vous pouvez créer à partir d'une classe autant d'instances que vous voulez, et chaque instance est totalement indépendante de la classe dont elle est issue ainsi que des autres instances issues de la même classe. Un exemple valant mieux que mille explications, le processus de création d'un objet à partir d'une classe est présenté dans le Listing 1.

Les déclarations des méthodes et propriétés se font au sein de la classe, respectivement

comme celles des fonctions et des variables, vous pouvez néanmoins leur spécifier des visibilitées : `public`, `protected`, ou `private` dont vous retrouverez les descriptions dans le tableau 2 (si vous ne la précisez pas, elle sera par défaut de type `public`) les visibilitées vous permettent de définir les domaines d'accès des attributs de vos objets, il est tout à fait possible qu'un attribut soit accessible directement dans le script, ce qui peut être pratique par exemple pour l'application de paramètres à une instance, ou que cet attribut soit seulement accessible par des méthodes : le Listing 2 vous donne un exemple de déclaration d'attribut et méthode avec leurs visibilitées. Remarque: Vous pouvez déclarer les méthodes et propriétés d'un objet n'importe où au sein de la classe, mais pour plus de clarté, il est conseillé de déclarer en premier lieu les propriétés puis les méthodes.

Pour accéder aux membres d'une instance, vous devrez toujours passer par la variable qui la représente, que ce soit dans le script en général ou dans l'une de ses méthodes. Il vient alors un problème : lors de la déclaration des méthodes, on ne connaît pas encore le nom de l'instance qui va en être issue, nom dont nous avons besoin si nous voulons accéder aux membres de l'instance par ses propres méthodes. Pour résoudre ce problème il existe une variable spéciale, que vous pouvez utiliser dans le corps de vos classes, faisant référence à l'instance appelée: la variable `$this`.

Dans vos méthodes, vous aurez parfois besoin d'accéder à l'instance à laquelle elles appartiennent, pour cela vous devrez utiliser une variable référence sur l'instance courante : la variable `$this`; pour ce qui est de l'appel des méthodes et propriétés d'une instance, vous avez à votre disposition l'opérateur `->` (ces deux appels sont explicités dans le Listing 3). En ce qui concerne les constantes : il vous est possible de définir des constantes dans le

corps de la classe à l'aide du mot-clé *const*, vous pourrez alors y accéder en dehors de la classe en utilisant le nom de la classe suivi de l'opérateur `::` et enfin le nom de la constante. En dehors des constantes utilisateur (celles que vous déclarez et définissez vous-même), il existe des constantes que l'on qualifie de magiques, elles sont définies par PHP lui-même et vous fourniront des renseignements sur votre script, vous les retrouverez dans le Tableau 1.

*Remarque* : vous pouvez définir un *typage* pour vos méthodes : vous pouvez préciser qu'un seul type d'objet ne peut être accepté comme paramètre d'une méthode, ce qui conduira à une erreur si un autre type est utilisé, pour cela vous n'avez qu'à précéder le nom de la variable par le type d'objet (à savoir le nom de la classe d'où vient l'objet) dans le prototype de votre méthode comme montré dans le Listing 4.

### L'héritage

Vous aurez remarqué qu'il existe quelques différences entre les différents mots-clés définissant la visibilité des attributs. La subtile différence entre les mots-clé *private* et *protected* tient dans le fait que le membre défini avec le premier mot-clé ne sera accessible que dans la classe dans laquelle il est défini.

Mais qu'est-ce que l'héritage ? C'est l'opération qui consiste à étendre une classe par une autre : la déclaration d'une classe héritant d'une autre se fait en ajoutant le mot-clé *extends* suivi du nom de la classe que vous souhaitez étendre au niveau de la définition de la classe héritée (voir Listing 5).

La classe fille (ou classe dérivée), à savoir celle qui hérite, assimile toutes les propriétés et méthodes de la classe mère, vous pourrez utiliser dans les instances de la classe fille les mêmes méthodes et les mêmes attributs que ceux qu'aurait une instance de la classe mère. Vous pourrez néanmoins définir pour la classe fille des propriétés et des méthodes supplémentaires, et également redéfinir les méthodes récupérées de la classe mère (on parle alors de surcharge). Dans le contexte de surcharge d'une méthode de la classe mère au sein la classe fille, cette méthode n'est plus accessible telle que définie dans la classe mère à partir de l'instance créée à partir de la classe fille puisque redéfinie, en revanche elle reste accessible dans la classe fille par le mot-clé *parent* ou le nom de la classe mère, suivi de l'opérateur `::`, combinaison grâce à laquelle vous pouvez appeler n'importe quelle méthode de la classe mère : le Listing 6 vous en montre un exemple.

### L'opérateur de résolution de portée (::)

Vous venez de découvrir cet opérateur dans sa combinaison avec le mot-clé *parent* afin

d'appeler l'une des méthodes d'une classe mère, sachez que ce n'est pas sa seule utilisation possible. Il sert également à appeler ce que l'on nomme *les fonctions de classe*, à savoir des méthodes utilisables sans qu'aucune instance ne soit créée, comme le montre le Listing 7, inutile de préciser que les méthodes en question ne doivent pas faire appel à la variable `$this`, référence à l'instance... qui n'existe pas.

Vous avez réalisé ce qu'on appelle des accès statiques, à savoir des appels directs à la classe,

sans instancier d'objet auparavant : vous pouvez réaliser ces accès à des méthodes (comme déjà vu), et des attributs.

Pour déclarer un attribut comme statique, vous devrez obligatoirement précéder sa déclaration du mot-clé *static*: vous pourrez alors utiliser cet attribut comme une variable classique portant comme nom `nomClasse::$nomAttribut`. Pour les méthodes par contre, ne pas les déclarer comme statiques ne vous empêchera pas de les utiliser comme telles, mais notez qu'une fois déclarées statiques,

**Listing 1. Première Instance**

```
<?php
class maClasse {
}
$instance = new maClasse();
echo gettype($instance); // affichera: object
?>
```

**Listing 2. Spécification de visibilité**

```
<?php
class maClasse {
    public $publicVar;
    protected $protectedVar;
    private $privateVar;
    public function helloWorld() {
        echo "Bonjour";
    }
}
$instance = new maClasse();
$instance->publicVar = "valeur"; // attribution d'une valeur à un attribut public,
// pas d'erreur.
$instance->protectedVar = "valeur"; // attribution d'une valeur à un attribut protégé
// en dehors de sa classe, erreur !
$instance->privateVar = "valeur"; // attribution d'une valeur à un attribut privé
// en dehors de sa classe ou de ses classes dérivées, erreur :
$instance->helloWorld(); // affichera : Bonjour
?>
```

**Tableau 1. Quelques constantes magiques**

Nom de la constante	Valeur
<code>__CLASS__</code>	Nom de la classe
<code>__DIR__</code>	Le nom du répertoire du script (disponible à partir de la version 5.3 de PHP)
<code>__FILE__</code>	Le nom complet du script
<code>__FUNCTION__</code>	Nom de la fonction
<code>__LINE__</code>	Numéro de la ligne de code
<code>__METHOD__</code>	Nom de la méthode

**Tableau 2. Descriptif des mot-clés associés à la visibilité des attributs et méthodes**

Mot-clé	Description
<code>public</code>	définit un attribut ou une méthode accessible à partir du script en utilisant l'instance de l'objet.
<code>protected</code>	définit un attribut ou une méthode accessible seulement au sein de la classe et ses classes dérivées.
<code>private</code>	définit un attribut ou une méthode accessible uniquement dans la définition de la classe.

vous ne pourrez plus les utiliser autrement; une fonction statique se comporte comme une fonction classique portant comme nom `nomClasse::nomFonction()`.

### Constructeurs et destructeurs

Afin de maîtriser au mieux vos objets, vous pouvez assigner des actions à la création et à la destruction de ses instances par la surcharge

de certaines méthodes un peu spéciales appelées *constructeur* et *destructeur*; les constructeurs ne sont pas propres à la version 5 de PHP puisqu'ils étaient déjà présents dans sa version antérieure, mais ils se doivent maintenant d'être nommés `__construct`. La méthode portant ce nom sera automatiquement appelée à chaque instanciation d'un nouvel objet, et la méthode destructeur nommée `__destruct` dès que l'instanciation est détruite.

La définition de la méthode `__construct` permet de passer des paramètres lors de la création de votre objet, en effet, cette fonction peut recevoir des arguments (dont des valeurs par défaut peuvent être précisées) que vous pouvez utiliser dans le processus d'instanciation, vous en retrouverez un exemple dans le Listing 8. Ces constructeurs et destructeurs peuvent être pratiques si votre objet a besoin de réaliser des opérations initiales comme la création d'un lien avec une base de donnée par exemple, on peut alors très bien imaginer un objet servant à manipuler un fichier ! Son constructeur s'occupant de l'ouverture du fichier et le destructeur de la fermeture, une fois ces deux méthodes définies, vous n'aurez qu'à définir d'autres méthodes pour écrire et lire dans le fichier, et vous aurez au final une classe permettant de gérer des fichier en toute transparence !

**Listing 3. Attributs et Méthodes**

```
<?php
class maClasse {
    public $compteur = 0;

    public function incrementerCompteur($pas = 1){
        $this->compteur += $pas; // on incrémente le compteur
        // veuillez noter la différence entre la variable $pas interne à la fonction,
        // et la variable
        // $compteur devant être appelée par la référence
        echo "incrémentation du compteur de $pas, nouvelle valeur du compteur:
            $this->compteur <br />";
    }
}

$instance = new maClasse(); // on crée une instance de l'objet défini par la
// classe maClasse
$instance->incrementerCompteur(); // affiche: incrémentation du compteur de 1,
// nouvelle valeur du compteur: 1
$instance->compteur = 10; // on attribue une nouvelle valeur à l'attribut compteur
$instance->incrementerCompteur(3); // affiche: incrémentation du compteur de 3,
// nouvelle valeur du compteur: 13
?>
```

**Tableau 3. Descriptif des fonctions associées aux objets**

Nom de la fonction	Prototypes	Paramètres	Description
<code>class_exists</code>	<code>bool class_exists (string \$nomClasse [, bool \$autoload ])</code>	<code>\$nomClasse</code> : nom de la classe <code>\$autoload</code> : si <code>__autoload</code> doit être appelée ou non (TRUE par défaut).	Vérifie qu'une classe a bien été définie.
<code>get_class</code>	<code>string get_class ([ object \$instance ])</code>	<code>\$instance</code> : instance à tester	Retourne le nom de la classe de l'instance.
<code>get_class_methods</code>	<code>array get_class_methods (mixed \$nomClasse)</code>	<code>\$nomClasse</code> : nom de la classe	Retourne l'ensemble des méthodes d'une classe sous forme de tableau.
<code>get_class_vars</code>	<code>array get_class_vars (string \$nomClasse)</code>	<code>\$nomClasse</code> : nom de la classe	Retourne les valeurs par défaut des attributs d'une classe sous forme de tableau.
<code>get_declared_classes</code>	<code>array get_declared_classes (void)</code>		Retourne l'ensemble des classes déclarées sous forme de tableau.
<code>get_declared_interfaces</code>	<code>array get_declared_interfaces (void)</code>		Retourne l'ensemble des interfaces déclarées sous forme de tableau.
<code>get_objects_vars</code>	<code>array get_object_vars (object \$instance)</code>	<code>\$instance</code> : instance à tester	Retourne les propriétés d'un objet sous forme de tableau associatif.
<code>get_parent_class</code>	<code>string get_parent_class ([ mixed \$object ])</code>	<code>\$object</code> : instance ou nom de la classe	Retourne le nom de la classe parente d'un objet ou d'une classe.
<code>interface_exists</code>	<code>bool interface_exists (string \$nomInterface [, bool \$autoload ])</code>	<code>\$nomInterface</code> : nom de l'interface <code>\$autoload</code> : si <code>__autoload</code> doit être appelée ou non par défaut.	Vérifie si une interface existe.
<code>is_a</code>	<code>bool is_a (object \$instance, string \$nomClasse)</code>	<code>\$instance</code> : instance d'objet <code>\$nomClasse</code> : nom de la classe	Vérifie qu'un objet a pour parent une classe donnée.
<code>is_subclass_of</code>	<code>bool is_subclass_of (mixed \$object, string \$nomClasse)</code>	<code>\$object</code> : instance d'objet ou nom de classe <code>\$nomClasse</code> : nom de la classe	Vérifie si un objet ou une classe dérive d'une classe.
<code>method_exists</code>	<code>bool method_exists (object \$instance, string \$nomMethode)</code>	<code>\$instance</code> : instance de l'objet <code>\$nomMethode</code> : nom de la méthode	Vérifie si une méthode d'un objet est définie.
<code>property_exists</code>	<code>bool property_exists (mixed \$object, string \$nomAttribut)</code>	<code>\$object</code> : instance d'objet ou nom de classe <code>\$nomAttribut</code> : nom de l'attribut	Vérifie si la propriété d'une instance existe ou est définie dans une classe.

Au même titre que toutes les variables sont automatiquement détruites à la fin du script, les instances d'objet le sont aussi, et il vous est donc possible grâce aux destructeurs d'associer des actions à la fin d'exécution d'un script. Un exemple d'application de la combinaison de cette propriété avec les constructeurs et destructeurs pourrait être une classe de gestion des entêtes et `__construct` et `__destruct` appelant elles même la fonction `include`, vous n'aurez alors qu'à instancier cette classe pour que vos portions de codes HTML soient automatiquement intégrées par le script.

### Le clonage d'objets

Au même titre que vous pouvez copier le contenu d'une variable, vous pouvez également faire une copie d'un objet. Dire que `$objet2 = $objet1`; ne vous fera copier que la référence, aussi lorsque vous appellerez `$objet2`, c'est `$objet1` que vous récupérerez. Pour obtenir 2 objets distincts, vous devrez utiliser le mot-clé *clone*: `$objet2 = clone $objet1`; ce qui vous permettra d'obtenir deux objets non seulement identiques mais également indépendants, vous pourrez très bien modifier les attributs du premier sans que le deuxième en soit affecté, le Listing 9 vous montre la différence entre la copie et le clonage d'une instance.

### Les fonctions magiques

`__construct` et `__destruct` sont des fonctions dites magiques en raison des immenses possibilités qu'elles vous offrent, mais il existe d'autres fonctions tout aussi magiques : `__get`, `__set`, `__call`, `__isset`, `__unset`, `__sleep`, `__wakeup`, `__toString`, `__clone` et `__autoload`!

`__get` et `__set` sont deux fonctions introduites à partir de la version 5 de PHP, capables d'intercepter les opérations de lecture (pour `__get`) et d'affectation (pour `__set`) d'un attribut non défini dans la classe. Si vous les implémentez, elles seront appelées à chaque fois qu'un attribut inexistant sera amené à subir une opération de lecture ou d'affectation. Elles doivent toutes les deux accepter comme premier argument le nom de l'attribut appelé, `__set` doit quant à lui accepter comme second argument la valeur de l'attribut. Ces fonctions vous permettent d'étendre le nombre d'attributs d'une instance à l'infini, puisque vous pourrez alors gérer ceux qui existent initialement, mais également ceux qui n'existent pas. Vous trouverez dans le Listing 10 une application de ces fonctions : en affectant des valeurs aux attributs d'une instance de la classe `registry`, vous sauvegarderez ces valeurs dans votre base de données, et il vous suffira d'appeler ces attributs pour qu'ils soient récupérés de votre base de données. La gestion

**Listing 4. Constantes et typages d'arguments**

```
<?php
class calcul {
    const PI = 3.14957;

    public function diametre(cercle $cercle) {
        return self::PI*2*$cercle->rayon;
    }
}

class cercle {
    public $rayon;
}

echo calcul::PI."<br />"; // affichera: 3.14957
$instanceA = new cercle();
$instanceA->rayon = 10;
$instanceB = new calcul();
echo $instanceB->diametre($instanceA); // affichera: 62.9914
?>
```

**Listing 5. Héritage et Visibilités**

```
<?php

class compteur {
    protected $compteur = 0; // une variable avec une visibilité private provoquerait
                          // une erreur dans la suite du script
}

class gestionCompteur extends compteur {
    public function incrementerCompteur($pas = 1) {
        $this->compteur += $pas;
    }

    public function retourCompteur() {
        return $this->compteur;
    }
}

$instance = new gestionCompteur();

echo $instance->compteur; // erreur ! la visibilité privée ne le permet pas : Fatal
// error: Cannot access protected property gestionCompteur::$compteur in ...
echo $instance->retourCompteur(); // à la place, ceci marche ! affichera: 1
?>
```

**Listing 6. Héritage et surcharge**

```
<?php

class classeMere {
    public function quiSuisJe() {
        echo "la classe classeMere ! <br />";
    }
}

class classeFille extends classeMere {
    public function quiSuisJe() {
        echo "la classe classeFille ! <br />";
    }

    public function quiEstMaMere() {
        parent::quiSuisJe();
    }
}

$instance = new classeFille();
$instance->quiSuisJe(); // affichera: la classe classeFille !
$instance->quiEstMaMere(); // affichera: la classe classeMere !
?>
```

d'un registre de valeurs devient totalement transparente !

`__call` est aux méthodes ce que `__get` est aux attributs, cette fonction vous permettra d'intercepter les appels de méthodes non définies dans votre classe, elle reçoit comme paramètres le nom de la méthode appelée, et les arguments reçus sous forme d'un tableau. Encore une fois cette fonction vous permet d'étendre le nombre de méthodes à l'infini ! Libre à vous de créer par exemple grâce à cette méthode magique un gestionnaire de

commandes capable d'analyser le nom de la méthode inexistante appelée et de comprendre par cette analyse le but de la méthode appelée pour en suite réaliser l'opération souhaitée.

La méthode `__unset` permet de surcharger la fonction `unset`, et la méthode `__isset` permet de surcharger les fonctions `isset` et `empty` : lorsque vous appellerez l'une de ces 3 fonctions sur un attribut non défini d'une instance, les méthodes surchargeantes correspondantes seront appelées (voir Listing 11).

#### Listing 7. Accès statiques

```
<?php
class cercle {
    const PI = 3.14957;

    public function diametreCercle($rayon) {
        return self::PI*2*$rayon;
    }
}
echo cercle::diametreCercle(10);
?>
```

#### Listing 8. Constructeurs et Destructeurs

```
<?php

class maClasse {
    function __construct() { // méthode appelée lors de la création de l'instance
        echo "Construction en cours <br />";
    }
    function __destruct() { // méthode appelée lors de la destruction de l'instance
        echo "Destruction<br />";
    }
}

$obj = new maClasse();
echo "actions du script <br />";
// Rendu HTML:
// Construction en cours
// actions du script
// Destruction
?>
```

#### Listing 9. Copie et clonage d'instances

```
<?php

class maClasse {
    public $nom;
}

$objet1 = new maClasse();
$objet1->nom = "objet1";
$objet1bis = $objet1;
$objet2 = clone $objet1;
echo $objet2->nom."<br />"; // affiche: objet1
$objet2->nom = "objet2"; // affiche: objet1
$objet1->nom = "nouveau nom";
echo $objet1bis->nom."<br />"; // affiche: nouveau nom
echo $objet2->nom."<br />"; //affiche: objet2
?>
```

Petit rappel : la fonction `unset` vous permet de détruire une variable et ainsi libérer la mémoire qu'elle occupe inutilement, l'utilisation de la méthode `__unset` pourrait vous servir à sauvegarder l'objet dans un fichier texte afin de pouvoir le réutiliser plus tard tout en optimisant la gestion des performances du script. La fonction `isset` quant à elle vous permet de vérifier si une variable est définie, si oui elle renvoie `TRUE`, la fonction `empty` de son côté retourne `FALSE` si la variable a une valeur non vide et différente de zéro.

`__sleep` et `__wakeup` sont appelées lors des appels respectifs des fonctions de linéarisation `serialize` et `unserialize` sur l'objet. Comme vous le savez déjà peut être, la fonction `serialize` transforme votre objet en une chaîne de caractères, et la fonction `unserialize` fait l'inverse et vous restitue votre objet à partir de cette même chaîne de caractères, l'intérêt de ces fonctions étant leur capacité de stockage dans une session. Une des utilités de ces fonctions magiques est par exemple le nettoyage de l'objet ou la clôture de la liaison avec la base de données que l'objet en question utilise lors de sa mise en sommeil, puis de la recréer lors de son réveil.

`__toString` est présentée dans le Listing 12. Cette méthode détermine le comportement que `echo` ou `print`. La méthode `__clone` est une fonction simple d'utilisation, elle sera appelée juste après la création de l'instance par clonage. Vous trouverez un exemple de sa syntaxe dans le Listing 16.

La fonction `__autoload` ayant déjà été abordée dans un article précédent : *Les nouveautés de PHP 5.3* du numéro 28, nous passerons brièvement dessus. Cette fonction est appelée si vous tentez d'utiliser une classe non définie, elle prend comme seul paramètre le nom de la classe en question. Son utilisation classique est l'inclusion à la dernière minute du fichier contenant la classe que vous essayez d'utiliser.

## Classes Abstraites et Interfaces

Il est possible de définir des classes et des méthodes ne pouvant être directement instanciées, le seul moyen de les utiliser étant alors de les faire hériter : on parle alors de classes abstraites, que l'on déclare par le mot-clé *abstract* (que vous pouvez combiner avec les mot-clés de définition de la visibilité). Dans la définition d'une méthode abstraite vous n'aurez que le prototype de la fonction à définir, laissant le corps de la fonction à définir par les classes dérivées. Concernant la visibilité, vous devez définir la méthode redéfinie avec une visibilité équivalente ou inférieure à celle de la méthode abstraite, par exemple si la méthode abstraite est protégée, vous pourrez redéfinir la méthode avec une visibilité

protégée ou publique, mais pas privée (remarque idiote : faire une méthode abstraite privée provoquerait une erreur et serait de toute façon aussi inutile que cette parenthèse). Notez que si vous définissez une méthode abstraite, vous devrez obligatoirement définir la classe à laquelle il appartient comme abstraite; dans le cas contraire PHP générera automatiquement une erreur. À l'inverse, le mot-clé final vous permettra de définir des classes et des méthodes, dites finales, qui ne pourront être redéfinies au sein des classes dont elles héritent. Vous retrouverez la syntaxe de déclaration d'une classe abstraite dans le Listing 13.

Les interfaces ont été intégrées dans PHP à partir de sa version 5, elles sont semblables aux classes abstraites dans le fait qu'elles intègrent seulement les prototypes des méthodes et qu'elles ne sont pas directement instanciables, elles doivent d'abord être implémentées (équivalent de *étendues* pour les interfaces) dans des classes où ces méthodes seront surchargées. Remarque importante : vous ne pouvez déclarer que des méthodes publiques dans les interfaces. La déclaration d'une interface se fait avec le mot-clé *interface* au lieu du mot-clé *class*, et l'implémentation avec le mot-clé *implements* au lieu de *extends*, notez qu'une classe peut implémenter autant d'interfaces que vous le souhaitez, vous n'avez qu'à énumérer les noms des interfaces séparés par des virgules après le mot-clé *implements*.

La partie de théorie pure concernant les interfaces se conclut avec le Listing 14. Avant de passer à la suite il serait intéressant pour vous d'en apprendre un peu plus sur l'utilité de ces classes abstraites mais surtout sur les interfaces. Les classes abstraites servent si on peut le dire, de structure pour vos objets, vous assurerez la présence de certaines méthodes au sein d'un objet par le fait que sa classe hérite de telle classe abstraite. Les interfaces peuvent définir une API (*Application Programming Interface*), sans même savoir comment marchent les classes implémentant une certaine interface, vous saurez comment les faire communiquer avec d'autres composants, il semble alors logique que les interfaces acceptent seulement les prototypes des méthodes publiques puisque les méthodes privées et protégées ne sont pas accessibles à l'extérieur des classes. Les classes abstraites et les interfaces permettent l'adoption de la programmation modulaire par la compatibilité garantie qu'elles procurent, à vous le développement de modules complexes et les joies du travail en équipe !

Bien saisir le concept et l'utilité des interfaces n'est pas chose aisée aux premiers abords, aussi je vous conseille fortement la lecture du tutoriel de *Alain Sahli* sur l'utilisation des interfaces en PHP.

#### Listing 10. \_\_set et \_\_get

```
<?php
// Pour utiliser ce script vous aurez besoin d'une table registry contenant 2 champs:
// key et value
class registry {
    private $properties = array();
    public function __set($key, $value) {
        mysql_query("UPDATE 'registry' SET 'value'='".addslashes($value)."' WHERE
            'key'='$key' LIMIT 1;");
        $this->properties[$key] = $value;
    }
    public function __get($key) {
        if(!isset($this->properties[$key])) {
            $sql = mysql_query("SELECT 'value' FROM 'registry' WHERE 'key'='$key'
                LIMIT 1;");
            $result = mysql_fetch_object($sql);
            $this->properties[$key] = stripslashes($result->value);
        } return $this->properties[$key];
    }
}
$registry = new registry();
$registry->adminPassword = 'toto';
echo ++$registry->visites;
?>
```

#### Listing 11. \_\_isset

```
<?php
class maClasse {
    public $variableExistante;
    function __isset($nomVariable) {
        echo "Test d'existence de la variable '". $nomVariable. "', resultat: ";
        return true;
    }
}
$instance = new maClasse();
var_dump(isset($instance->variableExistante)); // affiche: bool(false)
echo "<br />";
var_dump(isset($instance->variableInexistante)); // affiche: Test d'existence
// de la variable 'variableInexistante', resultat: bool(true)
echo "<br />";
// affichages identiques
var_dump(!empty($instance->variableExistante));
echo "<br />";
var_dump(!empty($instance->variableInexistante));
echo "<br />";
?>
```

#### Listing 12. \_\_toString

```
<?php
// Déclaration d'une classe simple
class maClasse {
    public $message;
    public function __construct($message) {
        $this->message = $message;
    } public function __toString() {
        return $this->message;
    }
}
$instance = new maClasse('Bonjour');
echo $instance; // affichera: Bonjour
?>
```

**Listing 13. Abstraction de classes**

```
<?php
abstract class maClasseAbstraite {
// cette méthode devra être redéfinie
    abstract protected function methodeAbstraite();

    public function methodeCommune() {
        /*
         corps de la fonction
        */
    }
}
class maClasse extends maClasseAbstraite {
// "public" étant une visibilité inférieure à "protected", aucune erreur ne sera
// provoquée
    public function methodeAbstraite() {
        // corps de la fonction
    }
}
?>
```

**Listing 14. Interfaces**

```
<?php
interface monInterfaceA {
    public function methode1($argument1, $argument2='valeur par défaut');
}
interface monInterfaceB {
    public function methode2();
}
class maClasse implements monInterfaceA, monInterfaceB {
// les prototypes des méthodes de l'interface et de la classe qui l'implémente
// doivent être identiques, jusqu'aux valeurs par défaut pour les arguments
    public function methode1($argument1, $argument2='valeur par défaut') {
        // corps de la fonction
    }
    public function methode2() {
        // corps de la fonction
    }
}
$instance = new maClasse();
$instance->methode1("bonjour");
?>
```

**Listing 15. Masque factory**

```
<?php
class factory {

    public static function create($nomClasse) {

        if (include_once "classes/$nomClasse.php") {
            // création de l'instance et retour
            return new $nomClasse;
        } else {
            return false;
        }
    }
}
// création de l'instance à partir de la classe driver_mysql contenue dans le fichier
// classes/driver_mysql.php
$driver_mysql = factory::create('driver_mysql');
?>
```

**Les masques**

Félicitations à vous, vous en avez fini avec la partie théorique de cet article ! Passons maintenant à autre chose d'un peu plus réjouissant, deux études de cas. Les masques sont des modèles reconnus d'astuces qui, en plus de créer de nouveaux outils de développement, utilisent des notions avancées

En premier lieu, le masque d'usine permettant la fabrication d'un objet (d'où son nom) : il peut gérer l'instanciation pendant l'exécution, l'opération la plus intéressante du code est l'instanciation dynamique d'une classe : au même titre que vous pouvez utiliser une variable de façon dynamique grâce à l'opérateur \$\$ ou appeler une fonction dynamiquement (\$nomFonction());, \$instance = new \$nomClasse vous permettra de créer une instance de la classe dont le nom se trouve dans la variable \$nomClasse. Le Listing 15 vous présente un exemple de cette classe ainsi que son utilisation. Ce masque peut avoir une utilisation intéressante lorsque vous aurez besoin d'un objet devant être créé à partir de telle ou telle classe selon un choix antérieur, comme la création d'un lien avec une base de données en ne sachant pas lors de la réalisation du script sur quel moteur votre application devra se baser.

Second masque : le *singleton* : si vous créez deux instances à partir de la même classe, elles seront forcément distinctes et indépendantes l'une de l'autre, or il se peut que pour certaines raisons vous ayez besoin de n'avoir qu'une seule et unique instance, quel que soit le nombre de fois que vous la créez et le nombre de variables lui faisant référence, le masque *singleton* vous permet de faire cela. Le discours peut vous paraître pour l'instant assez contradictoire par rapport à ce que nous avons vu précédemment dans la partie théorique, aussi je vous conseille de jeter un œil au Listing 16 : le principe de la classe du masque est de posséder une variable statique \$instance dans laquelle on sauvegarde l'instance une fois créée, ainsi lorsqu'on cherche à instancier cette classe on rencontre une condition : si l'instance n'a pas déjà été créée alors elle l'est, puis elle est retournée.

Nous devons donc créer une méthode s'occupant de l'instanciation : évidemment on ne peut confier cette condition à la méthode `__construct` puisqu'elle est appelée une fois l'instance créée : cette méthode n'a absolument aucun contrôle sur l'opération d'instanciation en elle-même. On ne peut pas non plus utiliser une méthode classique ayant besoin d'une instance pour être utilisée puisque c'est cette méthode qui est censée créer cette même instance, notre seule solution est donc le recours à une fonction de classe (méthode utilisable sans création d'instance) : méthode que nous appellerons `create()`. Par sûreté la visibilité du constructeur est

### Listing 16. Masque singleton

```
<?php
class singleton {
    private static $instance; // instance de la classe
    public $nom = '';

    private function __construct() { // Un constructeur privé ; empêche la création
        // directe d'objet

    }

    public static function create() { // La méthode singleton
        if (!isset(self::$instance)) {
            $c = __CLASS__;
            self::$instance = new $c;
        }
        return self::$instance;
    }

    public function __clone() { // Préviend les utilisateurs sur le clonage
        // de l'instance

    }
}

// Ceci récupère toujours une seule instance de la classe
$premiereInstance = singleton::create();
$secondeInstance = singleton::create();
$premiereInstance->nom = 'instance unique';
echo $secondeInstance->nom; // affichera: instance unique
?>
```

réduite à *private*, il ne sera donc pas accessible depuis l'extérieur de la classe, mais le restera de l'intérieur, et effectivement il sera appelé lors de l'instanciation à partir de la fonction de classe *create*, notez qu'il ne sera appelé qu'une seule fois dans tout le script. Maintenant que nous avons bloqué la création d'une autre instance à partir de la même classe, il ne nous reste plus qu'une chose à bloquer : le clonage. Comme vous l'avez vu plus haut, le clonage permet d'obtenir une copie conforme et indépendante d'une instance, ce que nous ne voulons pas ici.

La méthode de surcharge *\_\_clone* nous servira à intercepter le clonage, seul problème : cette méthode n'est appelée qu'après le clonage en question, quand il est déjà trop tard, malheureusement il n'y aucun moyen d'intercepter le clonage avant qu'il ne soit opéré, notre seule issue est de provoquer une erreur avec la fonction *trigger\_error()*. Nous avons donc finalement une classe qui n'aura en tout et pour tout qu'une seule instance, elle aura peut-être différents noms, mais au final ce sera toujours la même variable ! Vous pouvez très bien utiliser ce modèle comme classe mère pour de nombreuses autres classes.

Ces deux classes ne sont pas véritablement fondamentales, mais souvenez-vous d'elles car elles peuvent vous être d'un grand secours. Encore une fois ces masques ne sont

que des exemples, des modèles, vous pouvez toujours agrémente les classes avec des attributs et des méthodes propres à votre utilisation.

### Conclusion

Il ne s'agit ici que des bases de la programmation orientée objet sous PHP5, les exemples restent minimes et les applications peu nombreuses. Mais ce sont les bases, par lesquelles il faut bien passer, et avec lesquelles vous pourrez commencer à construire vos projets. Cet article n'avait aucunement l'intention de tout vous apprendre à ce sujet (de toute façon il aurait fallu bien plus qu'une dizaine de pages pour le faire de façon sérieuse), mais plutôt de faire une introduction au vaste domaine des possibilités qui s'ouvrent maintenant à vous. Nous espérons qu'il vous aura donné envie d'en connaître plus sur le sujet malgré son caractère indigeste, et que vous rallierez *PHP accros* au modèle objet.

### JEROME MANCHON

*Étudiant en classe préparatoire en vue des grandes écoles d'ingénieur, l'auteur travaille également depuis quelques années comme consultant dans la webagency LoryStudio, les divers projets qui lui ont été confiés l'ont amené à enrichir ses connaissances en matière de conception et de développement dans divers langages.*



# Formation PHP - MySQL pour débutants

Avec notre formation  
en ligne,

débutez facilement la  
programmation  
en PHP-MySQL.

[www.agie-formation.com](http://www.agie-formation.com)

Réduction\*  
de 15% avec  
le code  
AG88PH

Inscription en ligne

Démarrage dans la journée

The screenshot shows a webpage for 'Formations en ligne'. The main heading is 'Formation en ligne tutorée PHP - MySQL'. Below this, there is a 'Description' section with text: 'Vous souhaitez mettre en place des sites interactifs et performants, capables de communiquer avec des bases de données? Cette formation en ligne tutorée vous propose de découvrir rapidement la programmation pour le web avec PHP et MySQL. Des cas concrets pour progresser par la pratique. Des exercices et des conseils pour travailler et tester vos connaissances. Vous apprenez le développement des scripts PHP "petit à petit", vous maîtrisez rapidement toutes les techniques de base. Les sources des scripts PHP sont téléchargeables et modifiables pour vos sites.' There is also a 'Catégorie des formations' section and a 'CONTACT' section at the bottom.

[www.agie-formation.com](http://www.agie-formation.com)

\* Offre valable du 1er janvier au 31 janvier 2009

# Communication

## avec authentification avec PHP/CURL

L'extension PHP/CURL permet à un script PHP d'établir des communications avec authentification, chiffrées ou non, avec un serveur distant en utilisant les protocoles HTTP, HTTPS et FTP.

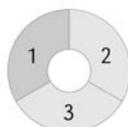
### Cet article explique :

- Comment s'authentifier avec CURL.
- Comment envoyer et enregistrer des cookies.
- Comment transférer des fichiers.

### Ce qu'il faut savoir :

- Connaître les bases du langage PHP.
- Avoir des notions sur les fichiers.
- Connaître les bases de CURL.

### Niveau de difficulté



Dans le numéro précédent de *php-solutions* vous avez utilisé l'extension *PHP/CURL* pour échanger des données avec un serveur web distant par les méthodes HTTP `GET` et `POST`. Les ressources demandées étaient en accès libre. Vous allez apprendre dans cet article à accéder à des ressources protégées par une authentification HTTP ou par un formulaire de *login* dans une navigation par session. Vous verrez comment échanger des *cookies* entre *CURL* et le serveur distant et également comment gérer les *cookies* de session.

Vous étudierez ensuite le transfert de fichiers avec un serveur FTP avec ou sans authentification. Enfin il vous sera montré comment gérer les erreurs intervenant lors d'une session *CURL*.

### Accès à des ressources protégées par une authentification HTTP

Le protocole HTTP offre la possibilité de restreindre l'accès d'un ou plusieurs répertoires du serveur web à un utilisateur authentifié. Ce mécanisme d'authentification HTTP repose sur l'envoi de champs dans l'en-tête HTTP. *CURL* permet d'envoyer l'en-tête HTTP contenant les identifiants afin d'accéder à une ressource protégée.

### Authentification HTTP

Le protocole HTTP fournit un mécanisme d'authentification par nom d'utilisateur et mot de passe. Ceci permet de restreindre l'accès d'un ensemble de ressources du serveur web à un ou plusieurs utilisateurs authentifiés. Lorsqu'un navigateur demande une ressource protégée, le serveur HTTP retourne un code de statut 401 (*Authorization Required*). Ce code indique au navigateur qu'il doit envoyer des identifiants valides afin d'obtenir l'accès à la ressource. Le navigateur affiche une fenêtre demandant à l'internaute ses identifiants (Figure 1). Ceux-ci sont transmis au serveur dans le champ *Authorization* de l'en-tête HTTP. Le serveur vérifie la validité du nom d'utilisateur et du mot de passe. En cas d'erreur d'authentification, le code 401 est envoyé à nouveau, sinon la ressource protégée est transmise au navigateur.

HTTP propose deux modes d'authentification : Basic et Digest. Le premier transmet les identifiants du client vers le serveur sans les chiffrer. La paire *utilisateur:mot\_de\_passe* est transmise encodée en base 64. Le second

mode d'authentification *chiffre* les identifiants. Afin d'éviter les attaques par rejeu, le chiffrement utilise une donnée transmise par le serveur. Il faut noter que, dans ces deux méthodes, les données elles-mêmes ne sont pas chiffrées. Lorsque la communication est sécurisée (*https*), le mode Basic peut être utilisé. Il faut cependant s'assurer dans ce cas que l'accès à la ressource n'est possible qu'avec *https*.

La protection des ressources par authentification est obtenue en utilisant des directives Apache (*AuthType*, *AuthName*, *AuthUserFile*, *AuthDigestFile*, *AuthGroupFile*, *AuthDigestGroupFile*, *Require...*). Ces directives, décrites sur le site d'Apache (cf. cadre Sur Internet), peuvent être placées dans une section *Directory* du fichier de configuration d'Apache *httpd.conf*, ou dans un fichier placé dans le répertoire du serveur à protéger. Généralement ce fichier est nommé *.htaccess* (le nom du fichier est défini par la directive *AccessFileName* du fichier *httpd.conf*). Il faut noter que l'authentification avec un *.htaccess* n'est possible que si la directive *AllowOverride* du *httpd.conf* autorise les directives d'authentification (valeur *AuthConfig* ou *All*). Le Listing 1 présente un exemple de *.htaccess*. La directive *AuthType* indique qu'il faut utiliser le mode d'authentification de base. Le texte affiché dans la fenêtre d'authentification est *intranet* (directive *AuthName*). L'emplacement du fichier contenant les noms



Figure 1. Fenêtre d'authentification HTTP

d'utilisateurs et leurs mots de passe est défini dans la directive `AuthUserFile`. La directive `Require` indique que l'accès sera autorisé à toute personne qui fournira une paire nom d'utilisateur/mot de passe valide (option `valid-user`). Pour limiter l'authentification à un des utilisateurs enregistré dans le fichier de mots de passe, il faut utiliser `Require user nom_utilisateur`.

Les identifiants valides sont stockés sur le serveur web dans un fichier généralement nommé `.htpasswd` (le nom et le chemin du fichier sur le serveur web sont définis par la directive `AuthUserFile` pour le mode Basic et `AuthDigestFile` pour le mode Digest). Ce fichier est généré automatiquement par l'exécutable `htpasswd` (mode Basic) ou `htdigest` (mode Digest). Ces outils, fournis dans la distribution Apache, sont décrits dans la documentation Apache (voir cadre Sur Internet). Dans l'exemple, la commande `htpasswd -c /usr/local/apache2/pwd/.htpasswd cecile` a été utilisée pour créer le fichier de mots de passe. L'outil demande dans un premier temps le mot de passe à associer à l'utilisateur dont le nom est passé en dernier argument, puis il crée le fichier (option `-c`) à l'emplacement indiqué et stocke le nom d'utilisateur et le mot de passe chiffré. Le fichier peut contenir les identifiants d'un ou plusieurs utilisateurs. Pour ajouter d'autres utilisateurs dans le fichier il suffit d'exécuter la commande sans l'option `-c`. Pour des raisons de sécurité, le fichier contenant les identifiants devrait être placé en dehors du répertoire racine du serveur web.

## Accès aux ressources protégées avec curl

Afin d'accéder à une ressource sur le web protégée par une authentification HTTP, il faut indiquer à curl la méthode d'authentification utilisée et lui fournir les identifiants. Ceci est réalisé en paramétrant deux options : `CURLOPT_HTTPAUTH` et `CURLOPT_USERPWD`. Le Listing 2 donne un exemple d'envoi de requête avec authentification HTTP.

Une fois la session CURL ouverte avec la fonction `curl_init`, qui retourne une référence vers la ressource externe CURL, les options de transfert sont paramétrées avec la fonction `curl_setopt`. Celle-ci prend en arguments la référence retournée par `curl_init`, l'option de transfert parmi une liste de constantes prédéfinies, et la valeur à lui affecter. Dans ce listing quatre options sont utilisées. L'option `CURLOPT_URL` permet de définir l'adresse de la ressource distante (dans cet exemple, l'URL est fictive). L'option `CURLOPT_RETURNTRANSFER` a été paramétrée à `false` afin de renvoyer directement la réponse HTTP au navigateur car, dans cet exemple, aucun traitement n'est effectué sur la réponse.

### Listing 1. `.htaccess`

```
AuthName "intranet"
AuthType Basic
AuthUserFile /usr/local/apache2/pwd/.htpasswd
Require valid-user
```

### Listing 2. `authentification.php`

```
<?php
// initialiser la session CURL
$curl_ref = curl_init() or die('erreur init curl');
// fixer l'url de la ressource (site fictif)
$url = 'https://www.test_site_curl.fr/intranet/liste.php';
curl_setopt($curl_ref, CURLOPT_URL, $url);
// envoyer la reponse HTTP au navigateur
curl_setopt($curl_ref, CURLOPT_RETURNTRANSFER, false);
// définir le type d'authentification (HTTP basic)
curl_setopt($curl_ref, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
// login/mot de passe pour accéder à la ressource
curl_setopt($curl_ref, CURLOPT_USERPWD, "cecile:mot2passe");
// envoyer la requete CURL (methode GET par default)
curl_exec($curl_ref);
// liberer les ressources
curl_close($curl_ref);
?>
```

### Listing 3. `cookie.php`

```
<?php
$curl_ref = curl_init() or die('erreur init curl');
$url = 'https://www.test_site_curl.fr/test_cookie.php';
curl_setopt($curl_ref, CURLOPT_URL, $url);
curl_setopt($curl_ref, CURLOPT_RETURNTRANSFER, false);
// définir les cookies à envoyer sous forme nom=valeur
curl_setopt($curl_ref, CURLOPT_COOKIE, "langue=français;nom=magali");
curl_exec($curl_ref);
?>
```

### Listing 4. `cookie_session.php`

```
<?php
$curl_ref = curl_init() or die('erreur init curl');
// ne pas envoyer la reponse HTTP au navigateur
curl_setopt($curl_ref, CURLOPT_RETURNTRANSFER, true);
// sauvegarder les cookies de la reponse HTTP dans le fichier cookies.txt
curl_setopt($curl_ref, CURLOPT_COOKIEJAR, "cookies.txt");
// renvoyer les cookies du fichier cookies.txt dans la requete CURL
curl_setopt($curl_ref, CURLOPT_COOKIEFILE, "cookies.txt");
// définir la methode d'envoi à POST et les parametres à envoyer
curl_setopt($curl_ref, CURLOPT_POST, true);
curl_setopt($curl_ref, CURLOPT_POSTFIELDS,
    array('login' => 'magali', 'pass' => 'passWord'));
// envoyer une premiere requete CURL (authentification)
$url = 'https://www.test_site_curl.fr/intranet/login.php';
curl_setopt($curl_ref, CURLOPT_URL, $url);
curl_exec($curl_ref);
// envoyer une seconde requete CURL (affichage liste apres authentification)
$url = 'https://www.test_site_curl.fr/intranet/liste.php';
curl_setopt($curl_ref, CURLOPT_URL, $url);
echo curl_exec($curl_ref); curl_close($curl_ref);
?>
```

L'option `CURLOPT_HTTPAUTH` précise la méthode d'authentification HTTP à utiliser. Elle prend la valeur `CURLAUTH_BASIC` pour une authentification en mode Basic ou `CURLAUTH_DIGEST` lorsque le mode Digest est utilisé.

L'option `CURLOPT_USERPWD` définit le nom d'utilisateur et le mot de passe à envoyer au

serveur. Les identifiants sont placés dans une chaîne de caractères composée du nom d'utilisateur et du mot de passe, séparés par le caractère deux points.

La session CURL est exécutée avec l'option `curl_exec` à laquelle on passe en paramètre la référence de session CURL. Enfin la session CURL est fermée avec la fonction

`curl_close`, qui libère les ressources. Dans l'exemple du Listing 2, l'authentification de base est utilisée. Le nom d'utilisateur et le mot de passe ne circuleront pas en clair sur le réseau car la communication est chiffrée (utilisation de https et non pas de http dans l'URL).

## Transmission de cookies

Les *cookies* permettent d'enregistrer des informations sur le disque dur du client et de recevoir lors des connexions futures ces informations. Ils rendent possible le stockage des préférences (langue...), l'authentification d'un client et le suivi de la navigation. Le *cookie* est une paire `nom=valeur` qui est envoyée dans la réponse HTTP au client. Ce *cookie* est stocké sur le disque du client, il est valide pendant une période donnée et pour un domaine et des chemins donnés. Le *cookie* est renvoyé par le client lorsqu'il émet des requêtes vers le serveur (en fonction du domaine et du chemin spécifié).

### Envoi de cookie du client vers le serveur

L'envoi de *cookies* par CURL s'effectue très simplement en paramétrant l'option `CURLOPT_COOKIE`. Les *cookies* sont envoyés sous la forme de paires `nom=valeur`. Il est possible d'envoyer plusieurs *cookies*, en les séparant par le caractère point-virgule. Dans le Listing 3, deux *cookies* sont envoyés, le *cookie* nommé `langue` dont la valeur est la chaîne de caractères `français` et le *cookie* `nom` de valeur `magali`. Ces deux *cookies* sont envoyés au serveur web lors de la soumission de la requête HTTP.

### Enregistrer un cookie et le renvoyer au serveur avec CURL

PHP intègre en standard un mécanisme, très simple d'utilisation, permettant de gérer la persistance des données : les sessions. Lorsqu'une session est créée, un identifiant de session unique est attribué à l'internaute. Pendant la navigation, cet identifiant de session est utilisé pour reconnaître l'internaute. L'identifiant est généralement transmis par *cookie*. Pour accéder à une ressource avec CURL dans le cadre d'une session, il faut pouvoir enregistrer le *cookie* d'identifiant de session envoyé par le serveur et le renvoyer au serveur. Il existe deux options CURL qui permettent de gérer le stockage et le renvoi de *cookies*, comme les *cookies* de session. L'option `CURLOPT_COOKIEJAR` définit le nom du fichier dans lequel les *cookies* doivent être stockés pour la session courante. L'option `CURLOPT_COOKIEFILE` définit le nom du fichier contenant les *cookies* à envoyer au serveur. CURL permet également d'accéder à des ressources suivies par session, nécessitant une authentification préalable. L'authentification est réalisée par le biais d'un formu-

#### Listing 5. login.php

```
<?php
session_start(); // ouvrir session
/**
 * Cree un formulaire d'authentification HTML
 * @return $form (string)
 */

function creerForm() {
    $form = '<form action="'.$_SERVER['PHP_SELF'].'" method="post" >';
    $form .= '<input type="text" name="login" /><br />';
    $form .= '<input type="password" name="pass" /><br />';
    $form .= '<input type="submit" name="validez" />';
    $form .= '</form>';
    return $form;
}

/**
 * Fonction de verification du login/pass dans une base de donnees/fichier
 * @param $login (string) identifiant a verifier
 * @param $pass (string) mot de passe a verifier
 * @return boolean
 */
function verif($login, $pass) {
    // verification des champs passes en parametre
}

// si utilisateur deja authentifie, redirection

if (isset($_SESSION['login'])) {
    header('index.php');
    die;
}

if (!isset($_POST['login'])) {
    echo creerForm();
    // si formulaire poste
} else {
    // si authentification, accepter la session

    if (verif($_POST['login'], $_POST['pass'])) {
        session_regenerate_id();
        $_SESSION['login'] = $_POST['login'];
        header('index.php');
        die;
    } // echec authentification

} else {
    echo 'Erreur d\'authentification';
    echo creerForm();
}
?>
```

laire d'identification soumis par la méthode POST. Le Listing 4 présente un exemple d'accès à une ressource d'un intranet. L'accès à l'intranet est géré par le script *login.php* (Listing 5). Ce dernier affiche le formulaire permettant de saisir le nom de l'utilisateur et son mot de passe. Il effectue la validation des identifiants lorsque le formulaire est soumis et initialise la variable de session *login* en cas de succès.

Dans le Listing 4, le but est d'obtenir la page *liste.php* de l'intranet. Deux requêtes CURL sont donc envoyées dans le script, la première afin de s'authentifier sur l'intranet et la seconde pour accéder à la ressource désirée.

L'envoi des données avec CURL s'effectue avec la méthode GET par défaut. Dans cet exemple, l'envoi du formulaire est réalisé avec la méthode POST. L'utilisation de cette dernière est obtenue en donnant la valeur *true* à l'option *CURLOPT\_POST*. L'option *CURLOPT\_POSTFIELDS* permet d'envoyer les données du formulaire nécessaires à l'authentification dans un tableau associatif, en utilisant la méthode POST avec CURL, à l'instar d'un utilisateur ayant rempli et posté le formulaire directement sur le site.

La première requête CURL d'authentification du Listing 4 envoie par la méthode POST le *login* et le mot de passe. Le script *login.php* vérifie les identifiants et envoie un identifiant de session. Celui-ci est stocké dans le fichier *cookies.txt* défini par l'option *CURLOPT\_COOKIEJAR*. Ce cookie est renvoyé au serveur lors de la deuxième requête CURL demandant la ressource *liste.php*.

## Téléchargement et envoi de fichier avec FTP et CURL

L'extension PHP/CURL permet à un script PHP d'établir des communications, avec ou sans authentification, avec un serveur FTP distant.

### Télécharger un fichier sur un FTP anonyme

Le Listing 6 présente une fonction de téléchargement de fichier par FTP sans authentification. Le fichier téléchargé est stocké sur le disque du serveur où le script PHP/CURL est exécuté. Ceci est réalisé en utilisant le protocole FTP dans l'URL. Par défaut, le transfert est réalisé en mode binaire. Dans le cas d'un fichier texte, la valeur *true* affectée à l'option *CURLOPT\_TRANSFERTEXT* réalise le transfert en mode ASCII. Dans cet exemple le but est d'enregistrer le fichier sur le disque, l'option *CURLOPT\_RETURNTRANSFER* a la valeur *true* afin de bloquer l'envoi des données au navigateur. Les données retournées sont écrites dans un fichier dont le nom est le même que celui du fichier téléchargé,

ce fichier est enregistré dans le répertoire du script exécuté.

### Envoyer un fichier sur un FTP avec authentification

L'envoi de fichier vers un serveur FTP avec CURL demande le paramétrage de trois options supplémentaires par rapport au téléchargement de fichier : *CURLOPT\_UPLOAD* (valeur *true*), *CURLOPT\_INFILESIZE* (taille du fichier), *CURLOPT\_READDATA* (fichier à envoyer). Il faut noter que pour les versions de la bibliothèque *libcurl* antérieures à la 7.9.7, il faut utiliser *CURLOPT\_INFILE* au lieu de *CURLOPT\_READDATA*. Dans le Listing 7, l'envoi de fichier s'effectue vers un serveur FTP avec authentification. Les identifiants sont passés directement dans l'URL sous la forme d'une paire *login:mot\_de\_passe*. L'URL est composée du protocole FTP suivi des identifiants, du symbole @, du nom du serveur FTP

et du chemin du fichier stocké sur le disque distant.

Dans cet exemple les identifiants et les données transitent en clair sur le réseau, mais il est également possible d'utiliser un FTP sécurisé avec CURL. Il faut noter qu'une extension FTP est disponible pour PHP (décrite sur le site officiel de PHP).

### Gestion des erreurs

Des erreurs peuvent intervenir à plusieurs niveaux lors d'une session CURL : échec de la connexion (protocole non supporté par CURL, port non ouvert, nom d'hôte erroné, serveur non disponible), erreur dans la syntaxe de l'URL, erreur HTTP (ressource non disponible ou interdite). Lorsqu'une erreur survient au cours de la session, aucun message n'est envoyé vers le navigateur. Pour obtenir des informations sur les erreurs, vous devez utiliser les fonctions *curl\_errno*

#### Listing 6. *ftp\_anonyme.php*

```
<?php
/**
 * Fonction de telechargement de fichier en FTP avec CURL
 * @param $serveur (string) nom du serveur
 * @param $chemin (string) chemin du fichier sur le serveur
 * @param $fichier (string) nom du fichier
 * @param $binaire (booléen) fichier binaire ou non
 */

function telecharger($serveur, $chemin, $fichier, $binaire) {
    $curl_ref = curl_init() or die('echec init curl');
    $url = "ftp://$serveur/$chemin/$fichier";
    curl_setopt($curl_ref, CURLOPT_URL, $url);
    curl_setopt($curl_ref, CURLOPT_RETURNTRANSFER, true);
    // preciser mode de transfert (texte ou binaire)

    if ($binaire) {
        // mode d'écriture du fichier (write binary)
        $mode = 'wb';
    } else {
        // preciser a la session curl transfert ASCII
        curl_setopt($curl_ref, CURLOPT_TRANSFERTEXT, true);
        // mode d'écriture du fichier (write)
        $mode = 'w';
    }

    $resultat = curl_exec($curl_ref);
    curl_close($curl_ref);
    // creer un fichier avec donnees transferees (texte ou binaire)
    $file = fopen($fichier, $mode) or die('erreur fichier');
    $nb = fwrite($file, $resultat);

    if ($nb === false) {
        echo 'erreur ecriture';
    }
    fclose($file);
}
?>
```

**Listing 7. ftp\_authentifie.php**

```
<?php
/**
 * Fonction d'envoi de fichier en FTP avec CURL
 * @param $serveur (string) nom du serveur
 * @param $chemin_serveur (string) chemin du fichier a creer sur le serveur
 * @param $chemin_local (string) chemin du fichier a envoyer
 * @param $login (string) identifiant pour serveur FTP
 * @param $pass (string) mot de passe pour serveur FTP
 * @param $binaire (booléen) fichier binaire ou non
 */
function envoyer($serveur, $chemin_serveur, $chemin_local, $login, $pass, $binaire=false) {
    $curl_ref = curl_init() or die('echec init curl');
    $url = "ftp://$login:$pass@$serveur/$chemin_serveur";
    curl_setopt($curl_ref, CURLOPT_URL, $url);
    curl_setopt($curl_ref, CURLOPT_RETURNTRANSFER, true);
    if ($binaire) { $mode = 'rb';
    } else {
        curl_setopt($curl_ref, CURLOPT_TRANSFERTEXT, true);
        $mode = 'r';
    }
    $file = fopen($chemin_local, $mode) or die('erreur fichier');
    // preparer l'envoi par FTP
    curl_setopt($curl_ref, CURLOPT_UPLOAD, true);
    // fichier a lire pour le transfert
    curl_setopt($curl_ref, CURLOPT_READDATA, $file);
    // taille du fichier a transférer
    curl_setopt($curl_ref, CURLOPT_INFILESIZE, filesize($chemin_local));
    $resultat = curl_exec($curl_ref);
    curl_close($curl_ref);
    fclose($file);
}
?>
```

**Listing 8. affiche\_erreurs.php**

```
<?php
$url = 'https://www.test_site_curl.fr:9999';
$curl_ref = curl_init() or die('echec init curl');
curl_setopt($curl_ref, CURLOPT_URL, $url);
curl_setopt($curl_ref, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl_ref, CURLOPT_FAILONERROR, true);
$reponse = curl_exec($curl_ref);
if (curl_errno($curl_ref) != CURLE_OK) {
    echo curl_error($curl_ref);
} else {
    echo ($reponse === false) ? '' : $reponse;
}
curl_close($curl_ref);
?>
```

**Sur Internet**

- <http://php.net/manual/fr/book.curl.php> – Manuel PHP/CURL sur le site officiel de PHP,
- <http://curl.haxx.se/libcurl/> – Site officiel de la bibliothèque *libcurl*,
- [http://curl.haxx.se/libcurl/c/curl\\_easy\\_setopt.html](http://curl.haxx.se/libcurl/c/curl_easy_setopt.html) – Options *libcurl*,
- <http://www.ietf.org/rfc/rfc2617> – RFC 2617 sur l'authentification HTTP,
- <http://httpd.apache.org/docs/2.2/howto/auth.html> – Documentation Apache sur l'authentification HTTP,
- <http://httpd.apache.org/docs/2.2/programs> – Documentation des outils *htdigest* et *htpasswd*,
- <http://curl.haxx.se/libcurl/c/libcurl-errors.html> – Codes d'erreurs *libcurl*.

et `curl_error`. Ces fonctions prennent en argument la référence de session retournée par `curl_init`. Elles retournent respectivement un code d'erreur et un message d'erreur en anglais. Si aucun problème n'a été rencontré lors de la dernière opération de la session, le code d'erreur a la valeur 0 (constante `CURLE_OK`) et le message est une chaîne de caractères vide. La valeur 1 indique que le protocole n'est pas supporté (`CURLE_UNSUPPORTED_PROTOCOL`). La valeur 3 indique que l'URL n'est pas valide (`CURLE_URL_MALFORMAT`). La valeur 7 indique que CURL n'a pas réussi à se connecter au serveur (`CURLE_COULDNT_CONNECT`).

Par défaut les erreurs HTTP dont les codes sont supérieurs ou égaux à 400 sont ignorées. Si la ressource demandée n'est pas sur le serveur (erreur HTTP 404) ou qu'elle est interdite (erreur HTTP 403), la fonction `curl_errno` retourne 0 (aucune erreur). Pour que les fonctions `curl_errno` et `curl_error` prennent en compte les erreurs HTTP, il faut donner la valeur `true` à l'option `CURLOPT_FAILONERROR`. Avec ce réglage, `curl_errno` retournera la constante `HTTP_RETURNED_ERROR` quand la réponse HTTP a un code d'erreur supérieur ou égal à 400. Pour obtenir le code de la réponse HTTP, vous pouvez utiliser la fonction `curl_getinfo` en lui passant en paramètre la référence retournée par `curl_init` et la constante `CURLINFO_HTTP_CODE`.

Le script du Listing 8 affiche le corps de la réponse HTTP s'il n'y a pas eu d'erreur lors de la session, sinon il affiche un message d'erreur. Le port demandé dans l'URL n'est pas ouvert sur le serveur, un message d'erreur indiquant que CURL ne peut pas se connecter au serveur est donc affiché quand le script est exécuté (*couldn't connect to host*).

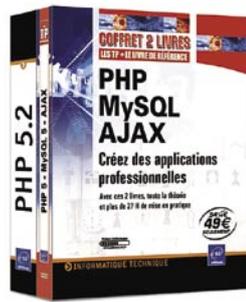
**Conclusion**

Vous savez à présent établir une communication authentifiée avec un serveur distant avec *PHP/CURL*. Vous avez appris à échanger des *cookies* entre le client *CURL* et le serveur web, ceci est utile notamment pour les navigations avec suivi de session. Vous savez envoyer des fichiers à un serveur FTP et en télécharger, avec ou sans authentification.

**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.odero@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 huit ans le développement d'applications Web à l'Université. Contact : http://magali.contensin.online.fr.*



## PHP 5, MySQL 5 et AJAX

Auteur : Brice Arnaud Guerin

Édition : ENI

ISBN : 978-2-7460-4131-8

Langue : français

Prix : 25,65 €

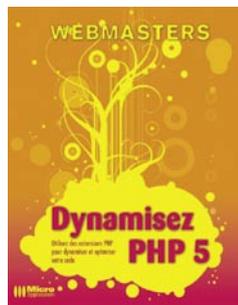
<http://www.eni-livres.com>

L'auteur Brice Arnaud Guerin présente dans ce livre sous forme de travaux pratiques les différents aspects de PHP, la programmation objet pour commencer avec quelques exercices dans chacun des chapitres afin de vous amener vers la création d'une plate-forme d'album photo avec AJAX ou encore pour finir avec un Tchat entièrement réalisé avec MySQL, PHP et AJAX. Avant de commencer chaque chapitre, une page avec quelques phrases où vous devez dire si les affirmations de ces phrases sont vraies ou fausses afin que vous puissiez voir si vous avez les pré-requis pour faire les exercices demandés dans les pages qui vont suivre. Chaque exercice est corrigé à la fin du livre afin de vous débloquent au cas où vous n'arriveriez pas à faire un exercice malgré les indices donnés. Le livre ne comporte malheureusement pas de cours, ce qui risque de vous obliger à aller vous documenter ailleurs, dans le cas d'une fonction mal comprise ou d'une notion manquante à la compréhension d'un exercice voir même d'un chapitre. C'est le

seul point négatif à tirer de ce livre. Les notions abordées sont bien sûr le couple PHP/MySQL mais aussi des bibliothèques comme SDL afin de créer un album photo, des fichiers PDF, manipulation des services web, les sessions et d'autres fonctions encore. AJAX est quant à lui utilisé à partir du deuxième chapitre puis au fil des chapitres vous serez amenés à la création de vraies applications dans les chapitres qui suivront votre progression.

En conclusion, je dirai que ce livre s'adresse à un public averti à cause du manque d'explication et de liens ou ressources permettant de se renseigner dans le cas où une partie serait trop difficile à assimiler. Pour le reste, ce livre est vraiment bien réalisé car en plus de son thème principal qui est la création d'application de type professionnel, vous allez apprendre dans la partie ressource à faire des petites applications pouvant être pratiques dans certains projets.

Critique faite par Mickael Puyfages



## Dynamisez PHP5

Auteur : David Drapeau & Frédéric Suire

Édition : MicroApplication

ISBN : 978-2-3000-1135-1

Langue : français

Prix : 23,75 €

<http://www.microapp.com>

Les auteurs de cet ouvrage, David Drapeau et Frédéric Suire, proposent un livre qui vous fera (re)découvrir certaines extensions PHP qui sont souvent bien utiles aux développeurs. Grâce à cet ouvrage vous découvrirez des extensions PHP qui vous permettront de rendre votre site plus efficace et votre code plus dynamique. Programmer du code dynamique vous permet de faire évoluer votre site en lui ajoutant de nouvelles fonctionnalités sans intervenir dans le codage.

L'efficacité de votre site en utilisant des extensions vous facilite aussi sa maintenance. Au programme, vous apprendrez la compression de données, la création de document office, la manipulation des images, la génération de documents PDF.

Vous saurez gérer vos annuaires LDAP, vos données avec PDO. Également, vous dynamiserez votre code PHP5 avec Classkit, Reflection.

Vous allégerez votre serveur avec APC ou la mise en cache et enfin vous saurez compiler votre code. Toutes ces extensions sont très détaillées et vous trouverez plus d'exemples concrets que de théorie sur PHP5.

Sans être la bible du développeur, ce livre est à posséder dans sa bibliothèque et à lire si vous voulez évoluer dans l'univers du web2. Considérez ce livre comme un guide d'approche : cela vous permettra sûrement de vous intéresser à d'autres extensions non développées dans cet ouvrage, mais tout aussi intéressantes pour vous et pour votre site.

Et si toutefois vous êtes débutants, initiez-vous directement à PHP5 et prenez de bonnes habitudes dès vos premières applications.

Critique faite par Valérie Viel

# 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 email.

Bonne lecture!



Le périodique *phpsolutions* est publié par  
Software-Wydawnictwo Sp. z o.o.  
Bokszerska 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 Marciniak  
**Directrice de la publication :** Dominika Baran

Imprimerie, photogravure : ArtDruk Zakład Poligraficzny,  
ul. Boh.Ossowa 10, 05-230 Kobylka, Pologne  
www.artdruk.com, Tél. +48 604 979 357  
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 :** Michał Szczypiek  
**Maquette :** Agnieszka Marchocka  
**Couverture :** Agnieszka Marchocka

**DTP Manager :** Andrzej Kuca  
**Composition :** Andrzej Kuca, Tomasz Tarchalski

**Correction :** Clément Quinton

**Bêta-testeurs :** Johan Denoyer, Valérie Viel, Aymeric Lagier, Christophe Milhau, Alain Ribault, Stéphane Guedon, Eric Boulet, Mickael Puyfages, David Drapeau, Jean-Michel Delhalle, Isabelle Lupi, Antoine Beluze, Claude Brulé, Timotée Neullas, Adrien Mogenet.

Les personnes intéressées par la coopération sont priées de nous contacter :  
editor@phpsolmag.org

**Abonnement :** abonnement@software.com.pl

**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.

La rédaction utilise le système PAO **AOPUS**  
Pour créer les diagrammes on a utilisé le programme **smartdraw.com** 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 mars !**

## POUR LES DÉBUTANTS

### ■ Débuter en PHP

Le langage PHP est l'un des plus populaires dans le domaine du Web. Il est incontournable de par ses évolutions constantes, sa communauté ultra développée et surtout, car il s'agit d'une solution gratuite pour les particuliers et entreprises. L'évidence est que, si vous lisez ces quelques lignes, c'est que vous vous y intéressez déjà. L'objectif de cet article est de vous montrer les bases du PHP pour que vous puissiez aller plus loin.

## OUTILS

### ■ Un moteur de template : Smarty

Qu'est-ce qu'un moteur de template ? Pourquoi utiliser un tel outil ? Cet article vous présente les moteurs de template et plus particulièrement Smarty. Vous apprendrez ce que sont les moteurs de template et plus particulièrement Smarty et vous verrez comment intégrer Smarty dans vos applications.

## PRATIQUE

### ■ Ma première application HELLO WORLD avec AMFPHP

AMFPHP est une solution simple et pratique pour faire communiquer une application Flash et PHP. Ce duo technologique permet d'accroître les possibilités graphiques et techniques d'un développement web. Vous apprendrez à écrire votre première application en utilisant la librairie

## SÉCURITÉ

### ■ Sécurité et PHP

La sécurité d'une application Web est un problème complexe par lequel il faut absolument passer si l'on veut éviter de voir son application détournée. C'est dommage à dire, mais beaucoup de scripts comportant des failles PHP peuplent la toile et viennent ainsi noircir la réputation de PHP. Cet article explique brièvement quelles sont les failles les plus courantes, comment s'en protéger, ainsi que différentes notions à ne jamais perdre de vue dans le développement d'une application.

**Et de nombreux autres articles à ne pas manquer !**

Contenus vidéo multilingues

Plateforme UGC

Services Live

Services de Caching

Services de DRM

Streaming sur Mobile

Solutions de Podcasting

Video On Demand



**ipercast**  
international

La voie vers le succès

 Ipercast Méta-CDN

Une solution de Caching intelligent disponible dans le monde entier.

Diffusion de vos Vidéos HD

Retrouvez toutes vos vidéos sur le web.

Mettez de la Vidéo sur Mobile

Soyez accessible sur toutes les plates-formes.

Ipercast Méta-Content Delivery Network (Méta-CDN), en connexion avec les plus grands CDN et fournisseurs d'IP au monde, vous garanti une disponibilité et une rapidité maximale d'accès aux contenus pour votre site web.

solutions  
**linux**  
opensource

Le Salon européen dédié à Linux et aux Logiciels Libres



**31 mars, 1<sup>er</sup> et 2 avril 2009**  
**Paris Expo - Porte de Versailles**

pour visiter le salon et obtenir votre badge d'accès gratuit,  
connectez-vous sur **[www.solutionslinux.fr](http://www.solutionslinux.fr)**

un événement

  
**Tarsus**

Solutions Linux/Open Source - 2/6 rue des Bourets - 92150 Suresnes - Tél : 33 (0) 1 41 18 63 33 - Fax : 33 (0) 1 41 18 60 68  
[www.solutionslinux.fr](http://www.solutionslinux.fr) - e-mail : [cjardon@tarsus.fr](mailto:cjardon@tarsus.fr)