

Table des matières

AVANT-PROPOS	VII
1. INTRODUCTION À ZEND FRAMEWORK	1
Avantages et inconvénients de Zend Framework • 2	
Structure et principe • 3	
Les règles de développement • 4	
Les composants réutilisables • 4	
L'architecture • 4	
Conseils pour bien démarrer avec Zend Framework • 5	
Prérequis • 5	
État d'esprit • 6	
En résumé • 7	
2. CAHIER DES CHARGES DE L'APPLICATION EXEMPLE	9
Expression du besoin • 10	
L'objectif : votre application ! • 10	
Spécifications fonctionnelles et techniques • 11	
Maquettes • 12	
Mises en garde et conventions • 18	
Conventions • 18	
Plateforme technique • 18	
En résumé • 19	
3. INSTALLATION ET PRISE EN MAIN	21
Téléchargement du paquetage • 22	
Téléchargement sous Windows • 23	
Téléchargement sous Unix • 23	
Configuration du serveur Apache • 23	
Téléchargement par le dépôt Subversion • 24	
Première utilisation du framework • 25	
En résumé • 26	
4. COMPOSANTS DE BASE	29
Configuration de l'environnement • 30	
Zend_Loader • 30	
Exemple d'utilisation • 31	
Chargement manuel d'une classe • 31	
Chargement automatique d'une classe (autoload) • 32	
Aller plus loin avec Zend_Loader • 32	
Intégration dans l'application • 34	
Zend_Config • 34	
Exemples d'utilisation • 35	
Avec un fichier ini • 35	
Avec un fichier XML • 36	
Avec un fichier PHP • 37	
Intégration dans notre application • 38	
Zend_Log • 40	
Quelques notions • 40	
Exemple d'utilisation • 41	
Utilisation conjointe avec Zend_Config • 42	
Intégration dans notre application • 43	
Zend_Debug • 43	
Exemple d'utilisation • 43	
Utilisation conjointe avec Zend_Log • 44	
Zend_Exception • 44	
Zend_Registry • 45	
Exemple d'utilisation • 46	
Intégration dans l'application • 46	
En résumé • 47	
5. ACCÈS AUX BASES DE DONNÉES	49
Introduction • 50	
Utiliser les SGBD • 51	
Les SGBD utilisables par Zend Framework • 51	
Création d'une connexion • 51	
Requêtes sur une base de données • 53	
Envoyer des requêtes • 55	
Effectuer des requêtes de type SELECT avancées • 57	
Utiliser la passerelle vers les tables • 59	
Créer et exécuter des requêtes • 60	
Manipuler des données • 61	
Récupérer des enregistrements • 61	
Modifier et sauvegarder des enregistrements • 63	
Agir sur les tables dépendantes • 65	
Performances et stabilité • 68	
Les bons réflexes • 69	
Aller plus loin avec le composant Zend_Db • 69	
Créer ses requêtes personnalisées • 69	
Étendre Row et Rowset • 71	

En résumé • 74	
6. ARCHITECTURE MVC 77	
Zend_Controller : utilisation simple • 78	
Mettre en place l'architecture • 78	
Parcours d'une requête HTTP • 79	
Exemple simple d'utilisation de Zend_Controller • 80	
Mettre en place le squelette de l'application • 82	
Code du squelette • 83	
Attribuer des paramètres à la vue • 84	
Manipulation des données HTTP • 85	
Initialisation et postdispatch • 87	
Zend_Layout : créer un gabarit de page • 88	
Appel et contenu du gabarit principal • 89	
En-tête et pied de page • 91	
Déclaration du sous-menu • 92	
Gestion par défaut des erreurs • 93	
Les aides d'action • 94	
Utiliser une aide d'action existante • 95	
Créer une aide d'action utilisateur • 96	
En résumé • 99	
7. ARCHITECTURE MVC AVANCÉE..... 101	
Zend_Controller : utilisation avancée • 102	
Les différents objets de MVC • 102	
Fonctionnement global de MVC • 104	
Exécution du processus de distribution de la requête • 105	
Un processus flexible et avancé • 108	
Fonctionnement détaillé des objets du modèle MVC • 109	
Contrôleur frontal (FrontController) • 109	
Objet de requête • 113	
Objet de réponse • 115	
Routeur • 117	
Plug-ins de contrôleur frontal • 120	
Plug-ins inclus dans la distribution de Zend Framework • 123	
Le distributeur (dispatcheur) • 125	
Les contrôleurs d'action • 126	
Les aides d'action • 129	
La vue • 142	
Les aides de vue • 143	
Les filtres de vue • 147	
En résumé • 148	
8. SESSIONS, AUTHENTIFICATION ET AUTORISATIONS..... 151	
Notions élémentaires • 152	
Les sessions • 153	
Pourquoi choisir Zend_Session ? • 153	
Configurer sa session • 154	
Utiliser les espaces de noms • 155	
Gestion de l'authentification avec Zend_Auth • 156	
Pourquoi utiliser Zend_Auth ? • 157	
Les adaptateurs • 157	
Exemple d'utilisation • 158	
Zend_Acl : liste de contrôle d'accès • 160	
Pourquoi utiliser Zend_Acl ? • 160	
Un peu de théorie sur les ACL • 160	
Exemple pratique • 161	
En résumé • 165	
9. INTERNATIONALISATION 167	
Avant de commencer... • 168	
Les composants Zend et leurs équivalents PHP • 168	
Attention aux jeux de caractères • 169	
Zend_Local : socle de base de l'internationalisation • 169	
Zend_Translate : gérer plusieurs langues • 170	
Pourquoi utiliser Zend_Translate ? • 171	
Les adaptateurs • 171	
Exemple simple d'utilisation • 172	
Exemple de changement d'adaptateur • 174	
Internationalisation avancée • 175	
Règles d'architecture • 175	
Mettre en place l'adaptateur gettext • 176	
Mettre en place les chaînes à traduire • 176	
Créer les fichiers de traduction gettext (*.mo) • 177	
Modifier la langue • 178	
Zend_Currency : gestion des monnaies • 179	
Pourquoi utiliser Zend_Currency ? • 179	
Affichage des monnaies • 179	
Informations sur les monnaies • 180	
Zend_Date : gestion de la date et de l'heure • 180	
Pourquoi utiliser Zend_Date ? • 180	
En résumé • 182	
10. PERFORMANCES 185	
Qu'est-ce que la gestion de cache ? • 186	
Pourquoi utiliser un cache ? • 186	
Mises en garde concernant la gestion du cache • 186	
Zend_Cache : gestion du cache • 187	
Choisir son frontal et son support de cache • 188	
Utilisation de Zend_Cache dans l'application • 190	
Implémentation de Zfbok_Cache • 190	
Utilisation du cache dans l'application • 192	
Amélioration des performances des composants Zend • 193	
Zend_Memory : gestion de la mémoire • 194	

Exemple pratique • 194	Les composants (library) • 238
Améliorer les performances générales de l'application • 195	MVC • 242
Les bons réflexes • 195	Zend_Mail : envoi d'e-mails • 243
Compiler Zend Framework dans APC • 196	Envoyer un simple e-mail • 243
En résumé • 199	Envoyer un e-mail complet • 245
11. SÉCURITÉ..... 201	Zend_Pdf : créer des fichiers PDF • 247
En quoi consiste la sécurité sur le Web ? • 202	Zend_Form : génération et gestion de formulaires • 250
Règles de sécurité élémentaires • 203	Créer un formulaire • 251
Solutions de sécurité de Zend Framework • 203	Assigner des filtres ou des validateurs • 255
Les validateurs • 203	Tous les composants de Zend Framework • 256
Les filtres • 204	En résumé • 258
Les attaques courantes • 204	14. OUTILS ET MÉTHODOLOGIE 261
Le Cross Site Scripting (XSS) • 205	L'éditeur : Zend Studio pour Eclipse • 262
Attaque XSS • 206	Un environnement intégré pour optimiser ses développements • 262
Les protections • 206	Intégrer Zend Framework dans l'IDE • 262
Le Cross Site Request Forgery (CSRF) • 207	Personnaliser ses composants • 265
Attaque CSRF • 208	Un code source de meilleure qualité grâce au formateur • 266
Les protections • 209	Le débogueur • 266
Sessions et Cookies • 209	Analyse des performances avec le profileur • 268
Attaque d'une session • 210	Tests fonctionnels avec Zend_Test • 270
Les protections • 210	Zend_Test, pour quoi faire ? • 270
L'injection SQL • 212	Prise en main de Zend_Test • 271
Attaque par injection SQL • 212	Templates Zend Studio For Eclipse • 271
Les protections • 212	Gestion du bootstrap • 272
En résumé • 213	Écrire des tests • 274
12. INTEROPÉRABILITÉ ET SERVICES WEB 215	Faut-il tout tester ? • 276
L'interopérabilité, qu'est-ce que c'est ? • 216	En résumé • 277
Les solutions existantes • 217	15. UTILISATION AVANCÉE DES COMPOSANTS..... 279
REST • 217	MVC et les bibliothèques • 280
Avantages • 217	Créer un composant utilisateur • 281
Inconvénients • 217	Règles fondamentales et conventions • 281
SOAP • 218	Principe et organisation • 282
Avantages • 218	Exemple • 282
Inconvénients • 218	Modélisation minimale • 283
XML-RPC • 218	Implémentation du composant • 284
RSS et Atom • 218	Dériver un composant existant • 288
Préparer le terrain • 219	Règles fondamentales • 288
Zend_Rest : l'interopérabilité simplifiée • 222	Ajouter une fonctionnalité à un composant • 289
Principe de REST • 222	Modifier le comportement d'un composant • 289
Zend_Rest : REST, version Zend Framework • 223	Simplifier l'accès à un ou plusieurs composants • 290
Zend_Soap : l'interopérabilité par définition • 227	Intégrer un composant externe • 292
Zend_Feed : pour les protocoles simples RSS et Atom • 230	En résumé • 293
En résumé • 235	
13. AUTRES COMPOSANTS UTILES..... 237	
Préparation de l'architecture • 238	

A. QU'EST-CE QU'UN FRAMEWORK ?.....	295
Définition et objectifs • 296	
Le framework au service du développement web • 297	
Risques et périls des pratiques courantes • 297	
Le framework à la rescousse • 298	
Inconvénients du framework • 299	
En résumé • 300	
B. BASES DE DONNÉES.....	301
Qu'est-ce qu'un SGBD ? • 302	
Architecture d'un SGBD • 302	
La base de données • 303	
Exemple simple • 303	
Notions techniques • 303	
Représentation graphique • 304	
Types de données • 304	
Clés et contraintes d'intégrité • 305	
Les principaux SGBD du marché • 305	
MySQL • 305	
Oracle • 306	
SQLite • 307	
Connexion à PHP • 307	
Notions avancées • 309	
Les ORM • 309	
Couches d'abstraction • 310	
Réplication et clustering • 310	
C. PROGRAMMATION ORIENTÉE OBJET.....	311
Concepts de base • 312	
Tout est question d'organisation • 312	
Ranger ses procédures dans les bons rayons • 313	
Qu'est-ce qu'une classe ? • 313	
Déclarer une classe • 313	
Des classes et des objets • 314	
Implémentation en PHP • 315	
Visibilité • 316	
Construction et destruction • 317	
Héritage • 318	
Variables et méthodes statiques • 320	
Constantes de classe • 321	
Classes, méthodes abstraites et interfaces • 322	
Abstract • 322	
Interfaces • 324	
Final • 325	
Modélisation et génie logiciel • 326	
Les relations entre classes • 326	
L'héritage • 326	
L'association • 326	
L'agrégation • 327	
La composition • 328	
La dépendance • 328	
Les diagrammes UML • 329	
Le diagramme de cas d'utilisation • 329	
Le diagramme de classes • 330	
Le diagramme de séquence • 331	
La rétro-ingénierie • 331	
Les logiciels de modélisation • 332	
ArgoUML • 333	
Umbrello • 333	
StarUML • 334	
Dia • 334	
Concepts objet PHP avancés • 335	
Les exceptions • 335	
La gestion des objets et les opérateurs • 340	
Références et clonage • 340	
Opérateurs et fonctions relatives aux objets • 341	
Typage d'argument • 343	
Les méthodes magiques • 344	
__get() et __set() • 345	
__call() • 346	
__isset(), __unset() • 347	
__clone() • 348	
__toString() • 349	
__sleep(), __wakeup() • 350	
L'interface de Réflexion • 352	
SPL : Standard PHP Library • 356	
Iterator • 357	
RecursiveIterator • 358	
Autres itérateurs • 360	
L'autoload • 361	
D. DESIGN PATTERNS.....	365
Comprendre les motifs de conception • 366	
Motif Singleton • 367	
Exemple de motif Singleton en PHP • 367	
Un Singleton dans Zend Framework ? • 368	
Motif Fabrique • 368	
Exemple de motif Fabrique en PHP • 368	
Une Fabrique dans Zend Framework ? • 370	
Motif Proxy • 370	
Exemple de motif Proxy dynamique • 370	
Un Proxy dans Zend Framework ? • 373	
Motif Observateur/Sujet • 374	
Exemple de motif Observateur • 374	
Un Observateur/Sujet dans Zend Framework ? • 378	

Motif Registre • 378	Fonctions d'insertion • 407
Exemple de motif Registre • 378	Fonctions d'ajout • 407
Un Registre dans Zend Framework ? • 380	Fonctions de suppression • 407
Et bien d'autres encore... • 380	Fonction d'interrogation • 408
E. LE PATTERN MVC EN THÉORIE 383	G. UTILISER SUBVERSION 409
Pourquoi utiliser MVC ? • 384	Subversion dans un projet • 410
Des avantages pour le travail en équipe • 384	Subversion pour Zend Framework • 411
Des avantages pour le développement et la maintenance • 385	Prise en main d'un client Subversion • 412
MVC schématisé • 386	Installation d'un client (Windows) • 412
Une implémentation « intuitive » • 386	Quelques commandes Subversion • 413
L'implémentation MVC • 388	Liaison du dépôt Subversion avec un serveur web • 416
Et les bibliothèques ? • 390	
F. COMMENT FONCTIONNE PHP ? 393	H. PRATIQUE DES TESTS AVEC PHPUNIT 417
PHP, qu'est-ce que c'est ? • 394	Présentation du concept de test • 418
Principe de fonctionnement • 394	Les tests unitaires • 419
Utilisation autonome • 394	Un exemple simple • 419
Utilisation avec un serveur HTTP • 396	Aller plus loin avec les tests : PHPUnit • 422
Composition • 397	Installer PHPUnit • 422
Environnement • 398	Écrire des tests avec PHPUnit • 423
Conseils pour paramétrer son environnement • 399	Exemple de test unitaire • 423
Paramétrer le fichier php.ini • 400	Concept du développement piloté par les tests • 426
Comment optimiser PHP ? • 402	Organisation du projet • 426
Réduire les accès disque • 403	Définition de la classe de tests des Produits • 426
Réduire les phases de compilation • 403	Définition de la classe de tests du Panier • 428
Aperçu du cache APC • 404	Encore plus loin avec PHPUnit... • 433
Fonctionnement d'APC • 405	Les tests de Zend Framework • 435
Configuration d'APC • 406	Tests fonctionnels avec Zend_Test • 436
Fonctions et cache utilisateur • 407	
	INDEX 437