

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI8PHP** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1 Préambule

- 1. Objectif de l'ouvrage 11
- 2. Bref historique de PHP 12
- 3. Où se procurer PHP ? 13
- 4. Conventions d'écriture 14
- 5. À propos des exercices 15

Chapitre 2 Introduction à PHP

- 1. Qu'est-ce que PHP ? 17
- 2. Structure de base d'une page PHP 19
 - 2.1 Les balises PHP 19
 - 2.2 La fonction echo 19
 - 2.3 Séparateur d'instructions 21
 - 2.4 Commentaires 22
 - 2.5 Mixer du PHP et de l'HTML 22
 - 2.6 Règles de nommage 26
 - 2.7 Exercice 1 : mon premier script PHP 26
- 3. Configuration de PHP 27
 - 3.1 Le fichier de configuration php.ini 27
 - 3.2 Informations sur la configuration 28
 - 3.3 Jeu de caractères 31
- 4. Utiliser PHP en ligne de commande 31
- 5. Les bases du langage PHP 32
 - 5.1 Constantes 32
 - 5.1.1 Définition 32

5.1.2	Portée	35
5.2	Variables	35
5.2.1	Initialisation et affectation	35
5.2.2	Portée et durée de vie	37
5.2.3	Variables dynamiques (ou variables variables)	38
5.3	Types de données	38
5.3.1	Types de données disponibles	38
5.3.2	Types de données scalaires	39
5.3.3	Types de données spéciaux	47
5.3.4	Déclaration de type	47
5.4	Tableaux	49
5.4.1	Définition	49
5.4.2	Création	50
5.4.3	Manipulation	56
5.4.4	Décomposer un tableau	61
5.4.5	Portée	63
5.5	Opérateurs	63
5.5.1	L'opérateur d'affectation par valeur	63
5.5.2	L'opérateur d'affectation par référence	65
5.5.3	Les opérateurs arithmétiques	66
5.5.4	L'opérateur de chaîne	67
5.5.5	Les opérateurs combinés	67
5.5.6	Les opérateurs de comparaison	68
5.5.7	Les opérateurs logiques	69
5.5.8	L'opérateur ternaire	70
5.5.9	L'opérateur de fusion NULL	71
5.5.10	L'opérateur d'affectation de fusion NULL	72
5.5.11	L'opérateur de comparaison combinée	73
5.5.12	Précédence des opérateurs	74
5.6	Structures de contrôle	75
5.6.1	La structure if	75
5.6.2	La structure switch	77
5.6.3	La structure while	80
5.6.4	La structure do ... while	82
5.6.5	La structure for	83
5.6.6	Les instructions continue et break	86
5.6.7	L'expression match	87

- 5.7 Inclure un fichier 90
 - 5.7.1 Fonctionnement 90
 - 5.7.2 Utilisation 92
- 5.8 Interrompre le script. 93
- 5.9 Exercice 2 : variables et structures de contrôle 94

Chapitre 3
Utiliser les fonctions PHP

- 1. Préambule 101
- 2. Manipuler les constantes, les variables et les types de données 102
 - 2.1 Constantes 102
 - 2.2 Variables 103
 - 2.3 Types de données 109
 - 2.3.1 Conversions 109
 - 2.3.2 Fonctions utiles 114
- 3. Manipuler les tableaux 120
- 4. Manipuler les nombres 134
- 5. Manipuler les chaînes de caractères 139
- 6. Utiliser les expressions rationnelles 158
 - 6.1 Introduction 158
 - 6.2 Structure d’une expression rationnelle 158
 - 6.3 Fonctions. 168
- 7. Manipuler les dates 173
- 8. Générer un identifiant unique 189
- 9. Manipuler les fichiers sur le serveur 191
 - 9.1 Fonctions utiles. 191
 - 9.2 Exemples d'utilisation. 199
- 10. Manipuler les en-têtes HTTP 200
- 11. Exercices 202
 - 11.1 Exercice 3 : manipuler les données 202
 - 11.2 Exercice 4 : écrire et lire un fichier sur le serveur 207

Chapitre 4**Écrire des fonctions et des classes PHP**

1. Fonctions	211
1.1 Introduction	211
1.2 Déclaration et appel	211
1.3 Paramètres	221
1.3.1 Syntaxe	221
1.3.2 Valeur par défaut	222
1.3.3 Déclaration du type de données	224
1.3.4 Passage par référence	227
1.3.5 Liste variable de paramètres	229
1.3.6 Utilisation du nom du paramètre dans l'appel	231
1.4 Considérations sur les variables utilisées dans les fonctions	233
1.4.1 Variables locales/globales	233
1.4.2 Variables statiques	235
1.5 Les constantes et les fonctions	236
1.6 Récursivité	237
1.7 Fonction anonyme	239
1.8 Fonction fléchée	240
1.9 Fonction générateur	242
1.10 Exercice 5 : écrire des fonctions	245
2. Classes	246
2.1 Concept	246
2.2 Définir une classe	247
2.3 Instancier une classe	252
2.4 Héritage	255
2.5 Autres fonctionnalités sur les classes	259
2.5.1 Classes ou méthodes abstraites	259
2.5.2 Classes ou méthodes finales	260
2.5.3 Interfaces	261
2.5.4 Propriétés ou méthodes statiques - Constantes de classes	263
2.5.5 Traits	266
2.5.6 Classes anonymes	269
2.6 Exceptions	270
2.7 Exercice 6 : écrire une classe	274
3. Espaces de noms	277

Chapitre 5

Gérer les erreurs dans un script PHP

- 1. Vue d'ensemble 283
- 2. Les messages d'erreur PHP..... 284
- 3. Les fonctions de gestion des erreurs..... 288
- 4. Exercice 7 : gérer les erreurs 303

Chapitre 6

Gérer les formulaires et les liens

- 1. Vue d'ensemble 307
 - 1.1 Introduction 307
 - 1.2 Les liens 307
 - 1.3 Les formulaires 310
 - 1.3.1 Petit rappel sur les formulaires 310
 - 1.3.2 Construire un formulaire dynamiquement 312
 - 1.3.3 Traiter un formulaire à l'aide d'un script PHP..... 317
 - 1.4 Récupérer les données d'une URL ou d'un formulaire..... 321
- 2. Récupérer les données passées par l'URL..... 324
 - 2.1 Considérations 324
 - 2.1.1 Que se passe-t-il si deux paramètres portent le même nom ? 324
 - 2.1.2 Utiliser un tableau pour passer des données dans l'URL. 324
 - 2.2 Transmettre des caractères spéciaux..... 325
 - 2.3 Exercice 8 : récupérer des données passées par l'URL..... 328
- 3. Récupérer les données saisies dans le formulaire..... 330
 - 3.1 Considérations 330
 - 3.1.1 Que se passe-t-il si deux zones portent le même nom ? 330
 - 3.1.2 Que se passe-t-il s'il y a deux formulaires dans la page HTML ?..... 331
 - 3.1.3 Utiliser un tableau pour récupérer les données saisies 331
 - 3.1.4 Passer des informations dans une zone de formulaire cachée. 332
 - 3.2 Les différents types de zones 334
 - 3.2.1 Vue d'ensemble 334
 - 3.2.2 Zones contenant du texte..... 337
 - 3.2.3 Groupes de boutons radio 337

3.2.4	Cases à cocher	338
3.2.5	Listes à sélection unique	340
3.2.6	Listes à sélection multiple	341
3.2.7	Boutons de validation	343
3.2.8	Boutons image	345
3.2.9	Boutons "reset" ou "button"	346
3.3	Synthèse.	346
3.4	Exercice 9 : récupérer des données saisies dans un formulaire	348
4.	Contrôler les données récupérées	351
4.1	Vue d'ensemble	351
4.2	Vérifications classiques.	352
4.2.1	Nettoyage des espaces indésirables	352
4.2.2	Données obligatoires	352
4.2.3	Longueur maximum d'une chaîne.	352
4.2.4	Caractères autorisés pour une chaîne - Format	353
4.2.5	Validité d'une date - Plage de valeurs	353
4.2.6	Validité d'un nombre - Plage de valeurs	355
4.2.7	Validité d'une adresse e-mail	356
5.	Problèmes sur les données récupérées.	357
6.	Utilisation des filtres	364
6.1	Principes.	364
6.2	Application aux formulaires.	373
6.3	Exercices.	375
6.3.1	Exercice 10 : contrôler des données passées par l'URL	375
6.3.2	Exercice 11 : contrôler des données saisies dans un formulaire.	376
7.	Aller sur une autre page.	380
8.	Échanger un fichier entre le client et le serveur	386
8.1	Vue d'ensemble	386
8.2	Envoyer un fichier depuis le client (upload)	387
8.3	Télécharger un fichier à partir du serveur (download)	392

Chapitre 7
Accéder aux bases de données

- 1. Introduction 399
 - 1.1 Vue d'ensemble 399
 - 1.2 La notion de fetch 400
- 2. Utilisation de MySQL 401
 - 2.1 Préambule 401
 - 2.2 Connexion et déconnexion 402
 - 2.2.1 Connexion 402
 - 2.2.2 Déconnexion 403
 - 2.2.3 Obtenir des informations sur le serveur MySQL 404
 - 2.2.4 Obtenir des informations en cas d'erreur de connexion 404
 - 2.2.5 Exemple 404
 - 2.3 Sélectionner une base de données 406
 - 2.4 Utiliser des requêtes non préparées 407
 - 2.4.1 Vue d'ensemble 407
 - 2.4.2 Exécuter une requête 407
 - 2.4.3 Connaître le nombre de lignes
dans le résultat d'une requête de lecture 409
 - 2.4.4 Extraire le résultat d'une requête de lecture 410
 - 2.4.5 Obtenir des informations sur le résultat d'une requête
de mise à jour 418
 - 2.4.6 Gérer les erreurs 421
 - 2.5 Utiliser des requêtes préparées 422
 - 2.5.1 Vue d'ensemble 422
 - 2.5.2 Préparer une requête 424
 - 2.5.3 Lier des variables PHP aux paramètres de la requête 425
 - 2.5.4 Exécuter la requête préparée 427
 - 2.5.5 Lier des variables PHP aux colonnes du résultat
d'une requête de lecture 428
 - 2.5.6 Extraire le résultat d'une requête de lecture 430
 - 2.5.7 Utiliser un résultat stocké 431
 - 2.5.8 Obtenir des informations sur le résultat d'une requête
de mise à jour 434
 - 2.5.9 Gérer les erreurs 437
 - 2.5.10 Fermer une requête préparée 438
 - 2.6 Gérer les transactions 438

2.7	Appeler un programme stocké	441
2.7.1	Procédure stockée	441
2.7.2	Fonction stockée	445
2.8	Exercice 12 : utiliser MySQL	447
3.	Utilisation d'Oracle	455
3.1	Préambule	455
3.2	Environnement NLS	455
3.3	Connexion et déconnexion	456
3.3.1	Connexion	456
3.3.2	Déconnexion	458
3.3.3	Obtenir des informations sur le serveur Oracle	459
3.3.4	Obtenir des informations en cas d'erreur de connexion	459
3.3.5	Exemple	459
3.4	Exécuter une requête	461
3.4.1	Vue d'ensemble	461
3.4.2	Analyser une requête	462
3.4.3	Lier des variables PHP aux paramètres de la requête	463
3.4.4	Exécuter une requête	466
3.4.5	Extraire le résultat de la requête de lecture	468
3.4.6	Mettre à jour des données et gérer les transactions	481
3.4.7	Fermer un curseur	487
3.5	Appeler une procédure stockée	487
3.6	Illustration des problèmes liés à l'environnement NLS	493
3.7	Gérer les erreurs	495
3.8	Exercice 13 : utiliser Oracle	498
4.	PHP Data Objects (PDO)	505
5.	Gestion des apostrophes dans le texte des requêtes	508
6.	Exemples d'intégration dans des formulaires	513
6.1	Vue d'ensemble	513
6.2	Construction d'une liste de sélection dans un formulaire	524
6.3	Affichage d'une liste	526
6.4	Formulaire de saisie en liste	530
6.5	Formulaire de recherche et de saisie	534

Chapitre 8

Gérer les sessions

1. Description du problème	539
2. Authentification	540
2.1 Vue d'ensemble	540
2.2 Saisie de l'identification	540
2.2.1 Identification par formulaire	540
2.2.2 Identification par authentification HTTP	543
2.3 Vérifier l'identification saisie	545
3. Utiliser des cookies	546
3.1 Principe	546
3.2 Application à la gestion des sessions	553
4. Utiliser la gestion des sessions de PHP	554
4.1 Principes	554
4.2 Mise en œuvre	554
4.3 Gérer soi-même la transmission de l'identifiant de session	568
4.3.1 Description du problème	568
4.3.2 Solution	572
4.4 Quelques directives de configuration supplémentaires	574
4.5 Exemples d'application	575
4.5.1 Principes	575
4.5.2 Avec authentification des utilisateurs	578
4.6 Remarques et conclusion	581
4.7 Exercice 14 : gérer les sessions	585
5. Conserver des informations d'une visite à une autre	589
6. Petite synthèse sur les variables Get/Post/Cookie/Session	595

Chapitre 9**Envoyer un courrier électronique**

1. Vue d'ensemble 599
2. Envoyer un message texte sans pièce jointe..... 599
3. Envoyer un message au format MIME..... 603
 - 3.1 Préambule 603
 - 3.2 Message au format HTML..... 603
 - 3.3 Message avec pièce jointe..... 606
4. Exercice 15 : envoyer un courrier électronique 609

Annexe

1. Variables PHP prédéfinies 613
 2. Constantes PHP prédéfinies 615
 3. Exemples complémentaires..... 616
 - 3.1 Introduction 616
 - 3.2 Lire un document XML 617
 - 3.3 Générer un document PDF..... 622
 - 3.4 Générer une image 626
 4. Résumé des principales nouveautés de la version 8..... 631
- Index..... 635

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **EI5SYM** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1 Introduction

1. Avant-propos	23
2. Public visé	24
3. Prérequis	24
4. Objectifs du livre	25
5. Le développement avec les frameworks	26
5.1 Complexité des développements et productivité	26
5.1.1 Productivité et qualité logicielle	26
5.1.2 Intégration et livraison continues	27
5.2 Particularité des développements en PHP	27
5.2.1 Contexte historique de PHP	27
5.2.2 Évolutions	28
5.3 L'apport des frameworks	28
5.3.1 Éviter les problèmes techniques liés à l'organisation du code	29
5.3.2 Définir des responsabilités	29
5.3.3 Ne pas réinventer la roue	30
5.3.4 Utiliser des modèles de conception éprouvés	30
6. Symfony	31
6.1 Historique	31
6.2 Gouvernance et gestion des versions	32
6.2.1 Les versions	32
6.2.2 Le cycle de « release »	33
6.3 Choisir sa version pour un projet	35

Chapitre 2

Mise en place d'un projet Symfony

1. L'outillage nécessaire	37
1.1 Introduction	37
1.2 Symfony : Un projet PHP	38
1.2.1 Préconisation d'installation	38
1.2.2 Installation sous Linux	39
1.2.3 Installation sous Windows	41
1.3 Symfony CLI	42
1.4 Composer	43
1.4.1 Installer Composer	43
1.5 Les environnements de développement pour Symfony	47
1.5.1 Un IDE pour Symfony !	48
1.5.2 PhpStorm	49
1.5.3 Eclipse IDE for PHP Developers	51
1.5.4 Visual Studio Code	52
1.5.5 Conclusion	54
2. Création d'un projet Symfony	55
2.1 Prérequis	55
2.2 Création via l'installateur Symfony	55
2.3 Création via Composer	56
2.4 Configurer son serveur web	58
2.4.1 Serveur web PHP	58
2.4.2 Apache et Nginx	59
3. Structure de l'application	63
3.1 Arborescence du projet	63
3.2 Règles et conventions d'organisation du projet	64
3.2.1 Le standard PSR-4	64
3.2.2 Conventions de nommage	65

- 3.3 La configuration d’une application Symfony 67
 - 3.3.1 Les annotations 68
 - 3.3.2 Le format YAML 70
 - 3.3.3 Le format XML 70
 - 3.3.4 Le format PHP 71
 - 3.3.5 Les attributs PHP 8 71
 - 3.3.6 Choisir son format de configuration 72

Chapitre 3
Architecture du framework

- 1. Le modèle de conception MVC 75
 - 1.1 Définitions et responsabilités 75
 - 1.1.1 La vue 76
 - 1.1.2 Le modèle 76
 - 1.1.3 Le contrôleur 76
 - 1.2 En pratique 77
 - 1.2.1 Le contrôleur frontal 78
 - 1.2.2 Le routage 78
 - 1.2.3 Le contrôleur et le modèle 78
 - 1.2.4 La vue 79
 - 1.2.5 En synthèse 80
- 2. Architecture de Symfony 80
 - 2.1 Présentation 80
 - 2.2 Le contrôleur frontal 81
 - 2.3 Le Service Container 82
 - 2.4 Le modèle MVC dans Symfony ? 82
 - 2.5 L’approche par composant 83
- 3. Symfony Flex 84
 - 3.1 Présentation 84
 - 3.2 Fonctionnement de Symfony Flex 85
 - 3.3 Les recettes Symfony 86

4 Symfony 5

Développez des sites web PHP structurés et performants

4.	Les environnements	87
4.1	Principe et apports	87
4.2	Les fichiers de configuration	87
4.3	Dans le contexte HTTP	88
4.4	Dans le contexte CLI (Command Line Interface)	88
5.	Le chargement automatique de classes	89
5.1	Le standard PSR-4	89
5.2	Mécanismes alternatifs	90
5.3	Application aux applications Symfony	90
6.	La console	91
6.1	Présentation	91
6.2	Les commandes	92
6.2.1	Lister les commandes disponibles	92
6.2.2	Exécuter une commande	92
6.3	Les options	92
6.4	Les arguments	93
6.5	L'aide sur les commandes	94
6.6	Exécuter rapidement des commandes	95
7.	Les outils pour le débogage	96
7.1	Le profiler Symfony	96
7.2	La fonction dump()	98

Chapitre 4 Routage et contrôleur

1.	Fonctionnement du routage dans Symfony	101
1.1	Définition	101
1.2	Le répertoire public et le contrôleur frontal	102
1.3	Une requête, une action	102
2.	Définition des routes	103
2.1	Les différents formats de définition	103
2.2	Les options sur la définition des routes	105

3.	Configurer le path.	106
3.1	Illustration par l'exemple : /hello/world.	106
3.2	La notation du contrôleur	110
3.3	Importer des routes depuis d'autres fichiers.	110
3.4	Comprendre l'ordre de chargement des routes.	111
3.5	Préfixer les routes	112
3.6	Les paramètres de substitution des routes	114
3.7	Les restrictions sur les paramètres.	116
3.8	Obtenir des informations sur le routage.	118
4.	Routage par nom de domaine	119
4.1	Prérequis	119
4.2	Exemple de mise en œuvre.	120
4.3	Explications.	123
5.	Le contrôleur.	124
5.1	Modèle de programmation et règles	124
5.2	Travailler avec les services	124
5.3	Utiliser les paramètres de substitution	125
5.3.1	Paramètres de substitution des routes	125
5.3.2	Exemples	125
5.4	Travailler avec les URL.	127
5.5	Effectuer une redirection	128
5.6	La délégation de requête.	128
5.7	La gestion des erreurs et des pages d'erreurs dans les contrôleurs	129
5.7.1	Le contrôleur.	129
5.7.2	La vue	132

Chapitre 5

L'injection de dépendances

1. Le modèle de conception IoC : Inversion Of Control	135
1.1 Apports dans une architecture applicative	135
1.2 IoC et injection de dépendances	137
2. L'injection de dépendances	137
2.1 Principes de base	137
2.2 Les différentes techniques d'injection de dépendances	137
2.2.1 L'injection de dépendances par le constructeur	137
2.2.2 L'injection de dépendances par setter (mutateur)	138
2.2.3 L'injection de dépendances par propriété	139
2.3 Les avantages	140
3. Le Service Container	140
3.1 Les services	141
3.2 Explications au travers d'un service X	141
4. Créer un service et configurer ses injections	142
4.1 Créer un service	142
4.2 Les différents types d'injections dans un service Symfony	143
4.2.1 Injection par constructeur	143
4.2.2 Injection par méthode	143
4.2.3 Injection par propriété	144
4.3 Injection automatique avec l'autowiring	145
4.4 Les services « lazy »	146
5. Le chargement automatique de services	147
5.1 La configuration	147
5.2 Exemple d'utilisation	149
6. Créer des services réutilisables et distribuables	151
6.1 Le concept de bundle	151
6.1.1 Créer un bundle	151
6.1.2 Arborescence du bundle	152

6.2	La définition de services dans un bundle	152
6.3	La configuration	153
6.3.1	Définir une arborescence	155
6.3.2	Les différentes étapes du traitement de la configuration	156
6.3.3	Récupérer la configuration validée	160
6.4	Les « Compiler Passes »	163
6.4.1	Concept	163
6.4.2	Les tags	166
6.4.3	Le Compiler Pass	166

Chapitre 6

Les templates avec Twig

1.	Présentation et concepts	169
1.1	Le concept de Templating	169
1.2	Templating et modèle MVC	169
2.	Twig	170
2.1	Présentation	170
2.2	Pourquoi un nouveau langage ?	171
2.3	Mise en pratique	172
2.4	Remarques sur l'utilisation	173
2.5	La notation des templates	174
2.6	Extension du système de templates	175
2.7	L'annotation @Template	175
3.	Les gabarits de pages (layouts) et les blocks	177
3.1	La composition de pages	177
3.2	Définition des gabarits	177
3.3	Les blocks	179
4.	Le langage Twig	182
4.1	Les différents types d'instructions	182

4.2	Manipulation des variables	183
4.2.1	Utilisation de variables dans les templates	183
4.2.2	Utilisation des variables de type tableau ou objet	185
4.3	Structures de contrôle et tags	185
4.3.1	Les conditions	185
4.3.2	Les boucles	186
4.4	Les balises Twig (tags)	189
4.4.1	Créer et modifier des variables	189
4.4.2	Twig et l'échappement	190
4.5	Inclure des templates	193
5.	Les filtres et les fonctions	194
5.1	Présentation des filtres	194
5.1.1	Utilisation et syntaxe	194
5.2	Les principaux filtres Twig	195
5.2.1	Chaînes de caractères	195
5.2.2	Échappement	196
5.2.3	L'encodage	197
5.3	Les fonctions	198
5.3.1	Twig et le routage	199
5.3.2	Débogage avec la fonction dump	200
6.	La gestion des ressources statiques (images, feuilles de style, scripts JS...)	201
6.1	Les ressources statiques dans une application Symfony	201
6.2	Le cas des ressources statiques externes	202
6.3	Référencer les ressources publiques depuis un template	203
6.4	Cas pratique avec le framework CSS Bootstrap	203
7.	Étendre le frontend avec Webpack Encore	205
7.1	Présentation	205
7.2	Installation et mise en place d'Encore	205
7.2.1	Prérequis	205
7.2.2	Installation d'Encore	205
7.3	Utilisation d'Encore	206

Chapitre 7**Accéder aux bases de données avec Doctrine**

1. Présentation et concepts	209
1.1 Les principes de l'ORM	209
1.2 Architecture de Doctrine	210
1.2.1 DBAL	210
1.2.2 Entité	211
1.2.3 ORM	211
1.3 La notion d'entité	211
2. Installer et configurer Doctrine	212
2.1 Mise en place de Doctrine	212
2.2 Relation avec PDO	213
2.3 Configuration	214
3. Définition des entités et de leur mapping	215
3.1 Règles de conception des entités	215
3.2 Les syntaxes pour le mapping des entités	217
3.3 Le mapping d'entités simples	220
3.3.1 Définir une entité avec <code>@ORM\Entity</code>	220
3.3.2 Gérer les colonnes de la table avec <code>@ORM\Column</code>	222
3.3.3 <code>@ORM\Table</code>	225
3.3.4 Les clés primaires	227
3.3.5 Configurer les index	228
3.4 Le mapping des relations (clés étrangères)	229
3.4.1 <code>@ORM\OneToOne</code>	230
3.4.2 <code>@ORM\ManyToOne</code>	232
3.4.3 <code>@ORM\ManyToMany</code>	233
3.4.4 Relations bidirectionnelles	235
3.5 Les outils de la console	240
3.5.1 Repérer les erreurs de mapping	240
3.5.2 Générer le schéma des données à partir des entités	241
3.5.3 Générer les entités à partir du schéma des données	241

4.	Manipulation des entités avec l'EntityManager	243
4.1	Le rôle de l'EntityManager	243
4.2	Insertion de données	243
4.3	Modification de données	245
4.4	Suppression de données	246
4.5	Autres opérations de l'EntityManager	247
4.5.1	refresh()	247
4.5.2	detach()	247
4.6	Les opérations en cascade	248
5.	Récupérer des entités	251
5.1	Le repository	251
5.1.1	Un rôle de centralisateur	251
5.1.2	Les méthodes de base du repository	252
5.1.3	Les méthodes personnalisées du repository	254
5.2	Le DQL	255
5.2.1	SELECT	257
5.2.2	FROM	257
5.2.3	JOIN et LEFT JOIN	258
5.2.4	WHERE	261
5.2.5	ORDER BY	265
5.2.6	Les limites	266
5.2.7	Les limites et la pagination	266
5.3	Le QueryBuilder	269
6.	Fonctionnalités avancées	271
6.1	Les extensions Doctrine	271
6.1.1	Installation	271
6.1.2	Utilisation d'un slug sur une entité	272

Chapitre 8

La gestion des événements applicatifs

- 1. Concepts et écoute d'événement applicatifs 275
 - 1.1 La propagation des événements..... 276
 - 1.2 L'écoute des événements 278
- 2. Les événements du Kernel 282
 - 2.1 Les différents type d'événements 282
 - 2.2 Applications 283
- 3. Les événements de la console..... 285
 - 3.1 Prérequis 285
 - 3.2 Les événements..... 285

Chapitre 9

Les formulaires

- 1. Un composant MVC 289
 - 1.1 Le modèle 289
 - 1.2 Le contrôleur..... 291
 - 1.3 La vue 292
- 2. Fonctionnement du composant..... 293
 - 2.1 L'objet « Form » 293
 - 2.1.1 Soumission 293
 - 2.1.2 Validation 294
 - 2.1.3 Vue 294
 - 2.2 Les types 294
 - 2.3 Les options 295
 - 2.4 Les objets « Form » et « FormBuilder » 296
 - 2.4.1 Le FormBuilder 296
 - 2.4.2 Structure de l'objet Form..... 296
 - 2.5 Association avec l'objet de la couche Modèle..... 298
 - 2.6 Formulaire sans objet 299

2.7	La représentation des valeurs	300
2.7.1	Transformation des données	300
2.7.2	Illustration avec le type date	301
3.	Les types de champs de formulaire	302
3.1	L'héritage	302
3.2	FormType	303
3.2.1	label	303
3.2.2	label attr	303
3.2.3	data	303
3.2.4	required	304
3.2.5	disabled	304
3.2.6	mapped	304
3.2.7	property_path	305
3.2.8	attr	305
3.2.9	trim	305
3.2.10	error_bubbling	305
3.3	TextType	306
3.4	PasswordType	306
3.5	RepeatedType	307
3.5.1	type	307
3.5.2	first_options et second_options	307
3.5.3	options	307
3.5.4	first_name	308
3.5.5	second_name	308
3.5.6	invalid_message	308
3.6	ChoiceType	308
3.6.1	choices	308
3.6.2	expanded et multiple	309
3.6.3	placeholder	309
3.6.4	preferred_choices	310
3.6.5	Types similaires	310

3.7	EntityType	310
3.7.1	class	311
3.7.2	choice_label	311
3.7.3	query_builder	311
3.7.4	group_by	312
3.7.5	em	312
3.8	DateType	312
3.8.1	widget	312
3.8.2	format	313
3.8.3	model_timezone	313
3.8.4	view_timezone	313
3.8.5	years	314
3.8.6	months	314
3.8.7	days	314
3.8.8	placeholder	314
3.8.9	Types similaires	315
3.9	FileType	315
3.9.1	multiple	315
3.9.2	Récupérer les fichiers	315
3.9.3	Traiter les fichiers	316
3.10	CheckboxType	317
3.11	SubmitType, ResetType et ButtonType	318
4.	Créer des formulaires réutilisables	319
4.1	Définir un formulaire avec la classe AbstractType	320
4.2	Utiliser un formulaire défini dans une classe	326
4.2.1	Définition manuelle	326
4.2.2	Avec l'injection de dépendances	327
5.	Validation des données	328
5.1	Objectifs	328
5.2	La définition des contraintes de validation	329
5.2.1	Ajout des contraintes lors de la configuration d'un formulaire	329
5.2.2	Ajout des contraintes sur l'objet associé au formulaire	331

5.2.3	Les différents formats de configuration	336
5.2.4	Les options	339
5.3	Les contraintes et leurs options	342
5.3.1	NotBlank et NotNull	342
5.3.2	IsNull et Blank	343
5.3.3	IsTrue, IsFalse	343
5.3.4	Type	343
5.3.5	Email, Url et Ip	344
5.3.6	Regex	345
5.3.7	Length, Count	346
5.3.8	Range	347
5.3.9	Comparaisons	347
5.3.10	Dates	348
5.3.11	File	349
5.3.12	Image	350
5.3.13	Choice	350
5.3.14	UniqueEntity	351
5.3.15	Données financières	352
5.3.16	Callback	352
5.3.17	All	354
5.3.18	Valid	354
5.4	Groupes de validation	355
5.5	Validation d'un objet hors du contexte d'un formulaire	357
6.	Personnaliser le rendu - thèmes de formulaires	358
6.1	Afficher le formulaire manuellement	359
6.1.1	form_start()	359
6.1.2	form_end()	360
6.1.3	form_widget()	360
6.1.4	form_errors()	361
6.1.5	form_label()	361
6.1.6	form_row()	361
6.1.7	form_rest()	361
6.1.8	Arborescence des parties de formulaires	362

6.2	Créer des thèmes	363
6.2.1	Formulaire d'exemple	363
6.2.2	Créer et associer un thème de formulaires	364
6.2.3	Comprendre le nom des blocks	368

Chapitre 10

La sécurité dans une application Symfony

1.	Présentation et concepts de sécurité	371
1.1	Les challenges de la sécurité des applications web	371
1.2	La sécurité dans Symfony	372
2.	Authentification	373
2.1	Pare-feu	373
2.2	Authentification HTTP	375
2.3	Authentification par formulaire de connexion	375
2.4	Connexion automatique des utilisateurs	378
2.5	Déconnexion des utilisateurs	379
3.	Utilisateurs et rôles	380
3.1	L'utilisateur	380
3.1.1	Récupérer l'utilisateur courant	381
3.2	Les fournisseurs d'utilisateurs	382
3.2.1	En mémoire	382
3.2.2	Fournisseur d'utilisateurs de bases de données	383
3.2.3	Fournisseur d'utilisateurs personnalisé	385
3.2.4	Notes additionnelles	389
3.3	Cryptage des mots de passe	390
3.3.1	Encodeurs	390
3.3.2	Le salage	391
3.3.3	Crypter un mot de passe	393
3.4	Les rôles	393

4. Autorisations	397
4.1 Les rôles, au cœur du processus	397
4.2 Vérifier le rôle de l'utilisateur	397
4.3 Sécuriser une action	399
4.4 Sécuriser une section de l'application	400
4.5 Sécuriser selon d'autres critères	400
4.6 Pour aller plus loin	402

Chapitre 11

Développer une API REST avec Symfony

1. Introduction à REST et concepts fondamentaux	403
1.1 Les concepts de REST	403
1.1.1 Les ressources	404
1.1.2 Le changement d'état d'une ressource	404
1.2 Architecture et protocole HTTP	404
1.3 Les Single-Page Applications	405
2. La gestion du format JSON	406
2.1 Présentation du format JSON	406
2.1.1 Représentation des données en JSON	407
2.1.2 Types de données	407
2.1.3 Structures	407
2.2 Le support de JSON en PHP	409
2.3 Et dans Symfony ?	409
3. Mise en place d'une API REST	410
3.1 Le service serializer	410
3.1.1 Sérialiser des données	411
3.1.2 Désérialiser des données	411
3.2 Adaptation des contrôleurs	412

4.	Les objets de requête et de réponse	412
4.1	Le contenu et les en-têtes de requête.	413
4.2	Manipulation de la réponse avec Response et JsonResponse .	414
4.3	Les codes de réponse HTTP dans une API REST	416
4.3.1	Problématique de l'état de la réponse	416
4.3.2	Expression de la réponse avec HTTP.	417
4.3.3	Mise en œuvre	418
5.	Tester une API REST	419
5.1	Les limites du navigateur web	419
5.2	Les outils	419
5.2.1	Postman.	420
5.2.2	SOAP UI	422
6.	Créer une API REST avec API Platform	423
6.1	Présentation	423
6.2	Installation	423
6.3	La configuration d'API Platform	424
6.4	Définition de l'API	425
6.4.1	Les opérations de l'API.	428
6.4.2	Personnaliser l'API	429

Chapitre 12

Tester son application Symfony

1.	Introduction au test logiciel.	431
1.1	Les tests : un indispensable pour la qualité logicielle	431
1.2	Les différentes catégories de tests	432
1.2.1	Les tests unitaires	432
1.2.2	Les tests d'intégration	432
1.2.3	Les tests fonctionnels	433
1.3	Analogie.	433
1.4	L'approche des tests en PHP	434

2.	Les tests unitaires avec PHPUnit	435
2.1	Mise en place des tests	435
2.2	Règle d'écriture des tests	435
2.3	Exécuter les tests	437
3.	Les tests fonctionnels	438
3.1	Différence par rapport aux tests unitaires et d'intégration	438
3.2	Tester une action	438
3.3	Les objets pour l'écriture des tests	439
3.3.1	L'objet Client	439
3.3.2	L'objet Crawler	442
3.4	Soumettre un formulaire	444
3.5	Pour aller plus loin	445

Chapitre 13

Journalisation et surveillance avec Symfony

1.	Générer des journaux avec Monolog	447
1.1	La journalisation	447
1.2	La librairie Monolog	448
1.3	Le service logger	449
1.4	Le fichier journal	450
1.4.1	Identifier la cause d'un bogue	450
1.4.2	Le problème	450
1.5	Les gestionnaires (handlers)	451
1.5.1	Définir plusieurs gestionnaires	452
1.5.2	Envoyer des logs par e-mail	452
1.5.3	Utiliser un tampon (buffer)	453
1.5.4	Ajouter des informations complémentaires	453
1.6	Les canaux (channels)	455
1.6.1	Ajouter ses propres canaux	455
1.6.2	Envoyer un enregistrement sur un canal donné	456
1.6.3	Configurer les gestionnaires par canaux	457
1.6.4	Gestion des erreurs 404	457

2. Le monitoring avec Prometheus et Grafana	460
2.1 Un allié proactif au logging	460
2.2 Préparation d'une application Symfony pour Prometheus	461
2.3 Instrumenter les mesures	465
2.4 Pour aller plus loin	466

Chapitre 14

Amélioration des performances

1. La mise en cache de pages	467
1.1 Autour du protocole HTTP	467
1.2 Un serveur proxy inverse (ou « reverse proxy »)	468
1.2.1 HttpCache	470
1.2.2 Nginx	470
1.2.3 Varnish	471
1.3 Les en-têtes	476
1.4 Les réponses publiques et privées	476
1.5 L'expiration	477
1.5.1 L'en-tête Expires	477
1.5.2 Les directives max-age et s-max-age	478
1.5.3 L'annotation @Cache	478
1.6 La validation	480
1.6.1 Par date avec Last-Modified	481
1.6.2 Par empreinte avec l'en-tête ETag	482
1.7 Les ESI	483
1.7.1 Activation	483
1.7.2 Générer une balise ESI	484
2. L'autochargement des classes	485

3.	Le cache avec Doctrine	486
3.1	Les différents types de cache	486
3.2	Configuration	487
4.	Le cache d'annotations	488
5.	Les sessions	489
6.	Autres optimisations	490
6.1	Choix de sa SAPI PHP	490
6.1.1	Qu'est-ce qu'une SAPI ?	490
6.1.2	Module du serveur	491
6.1.3	CGI	491
6.1.4	FastCGI	492
6.1.5	Conclusion	492
6.2	Mise en cache d'OPCodes	493
6.2.1	Les OPCodes	493
6.2.2	Une étape lourde	494
6.2.3	La mise en cache	494
6.3	La compression des réponses	494
6.3.1	Compression gzip	495
6.3.2	Précompression	495
6.4	Optimisation des images	496
6.4.1	Validation	496
6.4.2	Expiration	496
6.4.3	Autres techniques	496
7.	Test des performances d'un site web	497
7.1	Côté serveur	497
7.1.1	Apache Bench	497
7.1.2	Xhprof	498
7.2	Côté client	498

Chapitre 15
Internationalisation des applications Symfony

- 1. Introduction 499
 - 1.1 Culture, internationalisation et régionalisation..... 499
 - 1.1.1 La culture (Locale) 500
 - 1.1.2 Internationalisation 500
 - 1.1.3 Régionalisation 500
 - 1.2 L'internationalisation dans Symfony 501
- 2. Détecter la culture d'un utilisateur 501
 - 2.1 Les techniques..... 501
 - 2.1.1 Négociation de contenu 501
 - 2.1.2 Par l'URL 502
 - 2.2 En pratique 502
- 3. Activation des traductions..... 503
 - 3.1 Le composant translator 503
 - 3.2 Configuration du framework..... 503
- 4. Les routes et les traductions..... 504
- 5. Les fichiers de traductions 505
 - 5.1 Organisation et règles de nommage 505
 - 5.1.1 Règle de nommage 505
 - 5.2 Outillage pour la création des fichiers de traduction 506
 - 5.2.1 Afficher la liste des traductions manquantes 506
 - 5.2.2 Générer un fichier de traduction 507
- 6. Traduction d'un message 507
 - 6.1 Le service translator 507
 - 6.2 Les paramètres de substitution (placeholders) 508
 - 6.3 Utilisation dans les templates Twig 509

Annexes

1. Créer une commande pour la console	511
1.1 La configuration d'une commande	511
1.2 Les objets input et output	512
1.3 Le Service Container.	514
1.4 Commande d'exemple	516
2. Envoyer des e-mails grâce à Mailer	517
2.1 Le protocole SMTP	517
2.2 Le transport.	517
2.2.1 Le transport smtp	518
2.2.2 Le transport sendmail.	519
2.3 Envoi d'un e-mail	520
3. Travailler avec les sessions	521
3.1 Introduction	521
3.2 Intégration des sessions dans Symfony	521
3.3 Configuration du gestionnaire de sauvegarde	522
3.3.1 Avec PHP	522
3.3.2 Avec Symfony.	523
3.4 Les messages « flash »	523
4. Déployer une application symfony	525
4.1 Le déploiement	525
4.2 Faut-il déployer par FTP ?	526
4.3 Les différentes étapes	527
4.4 Capistrano et Capifony	527
4.4.1 Installation	527
4.4.2 Configuration	528
4.4.3 Déploiement	530
4.5 Fonctionnalités avancées	530
Index	531