

Les éléments à télécharger sont disponibles à l'adresse suivante :

**<http://www.editions-eni.fr>**

Saisissez la référence ENI de l'ouvrage **RI28PHP** 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 .....	21
2.5 Mixer du PHP et du HTML .....	22
2.6 Règles de nommage .....	25
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	48
5.4	Tableaux	50
5.4.1	Définition	50
5.4.2	Création	51
5.4.3	Manipulation	57
5.4.4	Décomposer un tableau	62
5.4.5	Portée	64
5.5	Opérateurs	64
5.5.1	L'opérateur d'affectation par valeur	64
5.5.2	L'opérateur d'affectation par référence	66
5.5.3	Les opérateurs arithmétiques	67
5.5.4	L'opérateur de chaîne	68
5.5.5	Les opérateurs combinés	68
5.5.6	Les opérateurs de comparaison	69
5.5.7	Les opérateurs logiques	70
5.5.8	L'opérateur ternaire	71
5.5.9	L'opérateur de fusion NULL	72
5.5.10	L'opérateur d'affectation de fusion NULL	73
5.5.11	L'opérateur de comparaison combinée	74
5.5.12	Préférence des opérateurs	75
5.6	Structures de contrôle	76
5.6.1	La structure if	76
5.6.2	La structure switch	78
5.6.3	La structure while	81
5.6.4	La structure do ... while	83

5.6.5	La structure for .....	84
5.6.6	Les instructions continue et break .....	87
5.6.7	L'expression match .....	88
5.7	Inclure un fichier. ....	90
5.7.1	Fonctionnement .....	90
5.7.2	Utilisation .....	93
5.8	Interrompre le script. ....	94
5.9	Exercice 2 : variables et structures de contrôle .....	95

## **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 .....	140
6.	Utiliser les expressions rationnelles .....	159
6.1	Introduction .....	159
6.2	Structure d'une expression rationnelle .....	159
6.3	Fonctions .....	169
7.	Manipuler les dates .....	174
8.	Générer un identifiant unique .....	194
9.	Manipuler les fichiers sur le serveur .....	196
9.1	Fonctions utiles .....	196
9.2	Exemples d'utilisation. ....	204
10.	Manipuler les en-têtes HTTP .....	205

11. Exercices .....	207
11.1 Exercice 3 : manipuler les données.....	207
11.2 Exercice 4 : écrire et lire un fichier sur le serveur .....	211

## Chapitre 4

### Écrire des fonctions et des classes PHP

1. Fonctions .....	215
1.1 Introduction .....	215
1.2 Déclaration et appel .....	215
1.3 Paramètres .....	225
1.3.1 Syntaxe.....	225
1.3.2 Valeur par défaut .....	227
1.3.3 Déclaration du type de données .....	229
1.3.4 Passage par référence .....	233
1.3.5 Liste variable de paramètres .....	234
1.3.6 Utilisation du nom du paramètre dans l'appel.....	237
1.4 Considérations sur les variables utilisées dans les fonctions .....	239
1.4.1 Variables locales/globales .....	239
1.4.2 Variables statiques.....	241
1.5 Les constantes et les fonctions.....	243
1.6 Récursivité.....	244
1.7 Fonction anonyme .....	245
1.8 Fonction fléchée .....	247
1.9 Fonction générateur .....	248
1.10 Exercice 5 : écrire des fonctions .....	251
2. Classes .....	253
2.1 Concept .....	253
2.2 Définir une classe .....	253
2.3 Instancier une classe.....	259
2.4 Héritage .....	263
2.5 Autres fonctionnalités sur les classes.....	268
2.5.1 Classes ou méthodes abstraites.....	268
2.5.2 Classes ou méthodes finales .....	269
2.5.3 Interfaces .....	270
2.5.4 Propriétés ou méthodes statiques - Constantes de classes ..	272

2.5.5	Traits .....	276
2.5.6	Classes anonymes .....	278
2.6	Exceptions .....	279
2.7	Énumérations .....	283
2.8	Exercice 6 : écrire une classe .....	289
3.	Espaces de noms .....	293

## Chapitre 5

### Gérer les erreurs dans un script PHP

1.	Vue d'ensemble .....	299
2.	Les messages d'erreur PHP .....	300
3.	Les fonctions de gestion des erreurs .....	304
4.	Exercice 7 : gérer les erreurs .....	319

## Chapitre 6

### Gérer les formulaires et les liens

1.	Vue d'ensemble .....	323
1.1	Introduction .....	323
1.2	Les liens .....	323
1.3	Les formulaires .....	326
1.3.1	Petit rappel sur les formulaires .....	326
1.3.2	Construire un formulaire dynamiquement .....	328
1.3.3	Traiter un formulaire à l'aide d'un script PHP .....	333
1.4	Récupérer les données d'une URL ou d'un formulaire .....	337
2.	Récupérer les données passées par l'URL .....	340
2.1	Considérations .....	340
2.1.1	Que se passe-t-il si deux paramètres portent le même nom ?	340
2.1.2	Utiliser un tableau pour passer des données dans l'URL .....	340
2.2	Transmettre des caractères spéciaux .....	341
2.3	Exercice 8 : récupérer des données passées par l'URL .....	344

3.	Récupérer les données saisies dans le formulaire . . . . .	347
3.1	Considérations . . . . .	347
3.1.1	Que se passe-t-il si deux zones portent le même nom ? . . . . .	347
3.1.2	Que se passe-t-il s'il y a deux formulaires dans la page HTML ? . . . . .	347
3.1.3	Utiliser un tableau pour récupérer les données saisies . . . . .	347
3.1.4	Passer des informations dans une zone de formulaire cachée . . . . .	349
3.2	Les différents types de zones . . . . .	350
3.2.1	Vue d'ensemble . . . . .	350
3.2.2	Zones contenant du texte . . . . .	353
3.2.3	Groupes de boutons radio . . . . .	353
3.2.4	Cases à cocher . . . . .	354
3.2.5	Listes à sélection unique . . . . .	356
3.2.6	Listes à sélection multiple . . . . .	357
3.2.7	Boutons de validation . . . . .	359
3.2.8	Boutons image . . . . .	361
3.2.9	Boutons "reset" ou "button" . . . . .	361
3.3	Synthèse . . . . .	362
3.4	Exercice 9 : récupérer des données saisies dans un formulaire	364
4.	Contrôler les données récupérées . . . . .	367
4.1	Vue d'ensemble . . . . .	367
4.2	Vérifications classiques . . . . .	367
4.2.1	Nettoyage des espaces indésirables . . . . .	367
4.2.2	Données obligatoires . . . . .	368
4.2.3	Longueur maximum d'une chaîne . . . . .	368
4.2.4	Caractères autorisés pour une chaîne - Format . . . . .	368
4.2.5	Validité d'une date - Plage de valeurs . . . . .	369
4.2.6	Validité d'un nombre - Plage de valeurs . . . . .	371
4.2.7	Validité d'une adresse e-mail . . . . .	371
5.	Problèmes sur les données récupérées . . . . .	373
6.	Utilisation des filtres . . . . .	380
6.1	Principes . . . . .	380
6.2	Application aux formulaires . . . . .	389

6.3	Exercices . . . . .	391
6.3.1	Exercice 10 : contrôler des données passées par l'URL . . . . .	391
6.3.2	Exercice 11 : contrôler des données saisies dans un formulaire. . . . .	392
7.	Aller sur une autre page . . . . .	396
8.	Échanger un fichier entre le client et le serveur. . . . .	402
8.1	Vue d'ensemble . . . . .	402
8.2	Envoyer un fichier depuis le client (upload) . . . . .	402
8.3	Télécharger un fichier à partir du serveur (download) . . . . .	407

## Chapitre 7

### Accéder aux bases de données

1.	Introduction. . . . .	415
1.1	Vue d'ensemble . . . . .	415
1.2	La notion de fetch. . . . .	416
2.	Utilisation de MySQL. . . . .	417
2.1	Préambule . . . . .	417
2.2	Connexion et déconnexion . . . . .	418
2.2.1	Connexion . . . . .	418
2.2.2	Déconnexion . . . . .	419
2.2.3	Obtenir des informations sur le serveur MySQL. . . . .	420
2.2.4	Définir le jeu de caractères du client. . . . .	420
2.2.5	Obtenir des informations en cas d'erreur de connexion . . . . .	420
2.2.6	Mode de rapport d'erreur . . . . .	421
2.2.7	Exemple . . . . .	422
2.3	Sélectionner une base de données . . . . .	423
2.4	Utiliser des requêtes non préparées . . . . .	424
2.4.1	Vue d'ensemble . . . . .	424
2.4.2	Exécuter une requête . . . . .	425
2.4.3	Connaître le nombre de lignes dans le résultat d'une requête de lecture	426
2.4.4	Extraire le résultat d'une requête de lecture . . . . .	427
2.4.5	Obtenir des informations sur le résultat d'une requête de mise à jour . . . . .	437
2.4.6	Gérer les erreurs . . . . .	439

2.5	Utiliser des requêtes préparées . . . . .	441
2.5.1	Vue d'ensemble . . . . .	441
2.5.2	Préparer une requête . . . . .	443
2.5.3	Lier des variables PHP aux paramètres de la requête . . . . .	444
2.5.4	Exécuter la requête préparée . . . . .	446
2.5.5	Lier des variables PHP aux colonnes du résultat d'une requête de lecture . . . . .	447
2.5.6	Extraire le résultat d'une requête de lecture . . . . .	449
2.5.7	Utiliser un résultat stocké . . . . .	451
2.5.8	Obtenir des informations sur le résultat d'une requête de mise à jour . . . . .	453
2.5.9	Gérer les erreurs . . . . .	456
2.5.10	Fermer une requête préparée . . . . .	458
2.6	Gérer les transactions . . . . .	458
2.7	Appeler un programme stocké . . . . .	461
2.7.1	Procédure stockée . . . . .	461
2.7.2	Fonction stockée . . . . .	465
2.8	Exercice 12 : utiliser MySQL . . . . .	467
3.	Utilisation d'Oracle . . . . .	474
3.1	Préambule . . . . .	474
3.2	Environnement NLS . . . . .	475
3.3	Connexion et déconnexion . . . . .	475
3.3.1	Connexion . . . . .	475
3.3.2	Déconnexion . . . . .	477
3.3.3	Obtenir des informations sur le serveur Oracle . . . . .	478
3.3.4	Obtenir des informations en cas d'erreur de connexion . . . . .	478
3.3.5	Exemple . . . . .	479
3.4	Exécuter une requête . . . . .	480
3.4.1	Vue d'ensemble . . . . .	480
3.4.2	Analyser une requête . . . . .	481
3.4.3	Lier des variables PHP aux paramètres de la requête . . . . .	482
3.4.4	Exécuter une requête . . . . .	485
3.4.5	Extraire le résultat de la requête de lecture . . . . .	487
3.4.6	Mettre à jour des données et gérer les transactions . . . . .	501
3.4.7	Fermer un curseur . . . . .	506
3.5	Appeler une procédure stockée . . . . .	507
3.6	Illustration des problèmes liés à l'environnement NLS . . . . .	512

3.7	Gérer les erreurs .....	515
3.8	Exercice 13 : utiliser Oracle .....	517
4.	PHP Data Objects (PDO) .....	524
5.	Gestion des apostrophes dans le texte des requêtes .....	527
6.	Exemples d'intégration dans des formulaires .....	532
6.1	Vue d'ensemble .....	532
6.2	Construction d'une liste de sélection dans un formulaire .....	543
6.3	Affichage d'une liste .....	545
6.4	Formulaire de saisie en liste .....	549
6.5	Formulaire de recherche et de saisie .....	553

## Chapitre 8

### Gérer les sessions

1.	Description du problème .....	557
2.	Authentification .....	558
2.1	Vue d'ensemble .....	558
2.2	Saisie de l'identification .....	558
2.2.1	Identification par formulaire .....	558
2.2.2	Identification par authentification HTTP .....	561
2.3	Vérifier l'identification saisie .....	563
3.	Utiliser des cookies .....	564
3.1	Principe .....	564
3.2	Application à la gestion des sessions .....	571
4.	Utiliser la gestion des sessions de PHP .....	572
4.1	Principes .....	572
4.2	Mise en œuvre .....	572
4.3	Gérer soi-même la transmission de l'identifiant de session .....	586
4.3.1	Description du problème .....	586
4.3.2	Solution .....	590
4.4	Quelques directives de configuration supplémentaires .....	592
4.5	Exemples d'application .....	593
4.5.1	Principes .....	593
4.5.2	Avec authentification des utilisateurs .....	596
4.6	Remarques et conclusion .....	599

4.7	Exercice 14 : gérer les sessions . . . . .	603
5.	Conserver des informations d'une visite à une autre. . . . .	607
6.	Petite synthèse sur les variables Get/Post/Cookie/Session. . . . .	612

## **Chapitre 9**

### **Envoyer un courrier électronique**

1.	Vue d'ensemble . . . . .	617
2.	Envoyer un message texte sans pièce jointe. . . . .	617
3.	Envoyer un message au format MIME. . . . .	621
3.1	Préambule . . . . .	621
3.2	Message au format HTML . . . . .	621
3.3	Message avec pièce jointe. . . . .	624
4.	Exercice 15 : envoyer un courrier électronique . . . . .	627

## **Annexe**

1.	Variables PHP prédéfinies . . . . .	631
2.	Constantes PHP prédéfinies . . . . .	633
3.	Exemples complémentaires. . . . .	634
3.1	Introduction . . . . .	634
3.2	Lire un document XML . . . . .	635
3.3	Générer un document PDF. . . . .	640
3.4	Générer une image . . . . .	645
4.	Résumé des principales nouveautés des versions 8.0, 8.1 et 8.2 . . . . .	650
	Index . . . . .	655

Les éléments à télécharger sont disponibles à l'adresse suivante :

**<http://www.editions-eni.fr>**

Saisissez la référence ENI de l'ouvrage **EI6SYM** 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 .....	48
1.5.3 Visual Studio Code .....	51
1.5.4 Conclusion .....	53
2. Création d'un projet Symfony .....	54
2.1 Prérequis .....	54
2.2 Création via l'installateur Symfony .....	54
2.3 Création via Composer .....	55
2.4 Configurer son serveur web .....	57
2.4.1 Serveur web PHP .....	57
2.4.2 Apache et Nginx .....	58
3. Structure de l'application .....	62
3.1 Arborescence du projet .....	62
3.2 Règles et conventions d'organisation du projet .....	63
3.2.1 Le standard PSR-4 .....	63
3.2.2 Conventions de nommage .....	64

3.3	La configuration d'une application Symfony . . . . .	66
3.3.1	Les annotations . . . . .	67
3.3.2	Le format YAML . . . . .	69
3.3.3	Le format XML . . . . .	69
3.3.4	Le format PHP . . . . .	70
3.3.5	Les attributs PHP 8 . . . . .	70
3.3.6	Choisir son format de configuration . . . . .	71

## Chapitre 3

### Architecture du framework

1.	Le modèle de conception MVC . . . . .	73
1.1	Définitions et responsabilités . . . . .	73
1.1.1	La vue . . . . .	74
1.1.2	Le modèle . . . . .	74
1.1.3	Le contrôleur . . . . .	74
1.2	En pratique . . . . .	75
1.2.1	Le contrôleur frontal . . . . .	76
1.2.2	Le routage . . . . .	76
1.2.3	Le contrôleur et le modèle . . . . .	76
1.2.4	La vue . . . . .	77
1.2.5	En synthèse . . . . .	78
2.	Architecture de Symfony . . . . .	78
2.1	Présentation . . . . .	78
2.2	Le contrôleur frontal . . . . .	79
2.3	Le Service Container . . . . .	80
2.4	Le modèle MVC dans Symfony . . . . .	80
2.5	L'approche par composant . . . . .	81
3.	Symfony Flex . . . . .	82
3.1	Présentation . . . . .	82
3.2	Fonctionnement de Symfony Flex . . . . .	83
3.3	Les recettes Symfony . . . . .	84

4.	Les environnements . . . . .	85
4.1	Principe et apports . . . . .	85
4.2	Les fichiers de configuration . . . . .	85
4.3	Dans le contexte HTTP . . . . .	86
4.4	Dans le contexte CLI (Command Line Interface) . . . . .	86
5.	Le chargement automatique de classes . . . . .	87
5.1	Le standard PSR-4 . . . . .	87
5.2	Mécanismes alternatifs . . . . .	88
5.3	Application aux applications Symfony . . . . .	88
6.	La console . . . . .	89
6.1	Présentation . . . . .	89
6.2	Les commandes . . . . .	90
6.2.1	Lister les commandes disponibles . . . . .	90
6.2.2	Exécuter une commande . . . . .	90
6.3	Les options . . . . .	90
6.4	Les arguments . . . . .	91
6.5	L'aide sur les commandes . . . . .	92
6.6	Exécuter rapidement des commandes . . . . .	93
7.	Les outils pour le débogage . . . . .	94
7.1	Le profiler Symfony . . . . .	94
7.2	La fonction dump() . . . . .	96

## Chapitre 4

### Routage et contrôleur

1.	Fonctionnement du routage dans Symfony . . . . .	99
1.1	Définition . . . . .	99
1.2	Le répertoire public et le contrôleur frontal . . . . .	100
1.3	Une requête, une action . . . . .	100
2.	Définition des routes . . . . .	101
2.1	Les différents formats de définition . . . . .	101
2.2	Les options sur la définition des routes . . . . .	103

3. Configurer le path . . . . .	104
3.1 Illustration par l'exemple : /hello/world . . . . .	104
3.2 La notation du contrôleur . . . . .	108
3.3 Importer des routes depuis d'autres fichiers . . . . .	108
3.4 Comprendre l'ordre de chargement des routes . . . . .	109
3.5 Préfixer les routes . . . . .	110
3.6 Les paramètres de substitution des routes . . . . .	112
3.7 Les restrictions sur les paramètres . . . . .	114
3.8 Obtenir des informations sur le routage . . . . .	116
4. Routage par nom de domaine . . . . .	117
4.1 Prérequis . . . . .	117
4.2 Exemple de mise en œuvre . . . . .	118
4.3 Explications . . . . .	121
5. Le contrôleur . . . . .	122
5.1 Modèle de programmation et règles . . . . .	122
5.2 Travailler avec les services . . . . .	122
5.3 Utiliser les paramètres de substitution . . . . .	123
5.3.1 Paramètres de substitution des routes . . . . .	123
5.3.2 Exemples . . . . .	123
5.4 Travailler avec les URL . . . . .	125
5.5 Effectuer une redirection . . . . .	126
5.6 La délégation de requête . . . . .	126
5.7 La gestion des erreurs et des pages d'erreurs dans les contrôleurs . . . . .	127
5.7.1 Le contrôleur . . . . .	127
5.7.2 La vue . . . . .	130

## Chapitre 5

### L'injection de dépendances

1.	Le modèle de conception IoC : Inversion Of Control .....	133
1.1	Apports dans une architecture applicative .....	133
1.2	IoC et injection de dépendances .....	135
2.	L'injection de dépendances .....	135
2.1	Principes de base .....	135
2.2	Les différentes techniques d'injection de dépendances .....	135
2.2.1	L'injection de dépendances par le constructeur .....	135
2.2.2	L'injection de dépendances par setter (mutateur) .....	136
2.2.3	L'injection de dépendances par propriété .....	137
2.3	Les avantages .....	138
3.	Le Service Container .....	138
3.1	Les services .....	139
3.2	Explications au travers d'un service X .....	139
4.	Créer un service et configurer ses injections .....	140
4.1	Créer un service .....	140
4.2	Les différents types d'injections dans un service Symfony .....	141
4.2.1	Injection par constructeur .....	141
4.2.2	Injection par méthode .....	141
4.2.3	Injection par propriété .....	142
4.3	Injection automatique avec l'autowiring .....	143
4.4	Les services « lazy » .....	144
5.	Le chargement automatique de services .....	145
5.1	La configuration .....	145
5.2	Exemple d'utilisation .....	146
6.	Créer des services réutilisables et distribuables .....	148
6.1	Le concept de bundle .....	148
6.1.1	Créer un bundle .....	148
6.1.2	Arborescence du bundle .....	149
6.2	La définition de services dans un bundle .....	150

---

6.3	La configuration . . . . .	151
6.3.1	Définir une arborescence . . . . .	152
6.3.2	Les différentes étapes du traitement de la configuration . . . . .	154
6.3.3	Récupérer la configuration validée . . . . .	157
6.4	Les « Compiler Passes » . . . . .	160
6.4.1	Concept . . . . .	161
6.4.2	Les tags . . . . .	163
6.4.3	Le Compiler Pass . . . . .	164

## Chapitre 6

### Les templates avec Twig

1.	Présentation et concepts . . . . .	167
1.1	Le concept de Templating . . . . .	167
1.2	Templating et modèle MVC . . . . .	167
2.	Twig . . . . .	168
2.1	Présentation . . . . .	168
2.2	Pourquoi un nouveau langage ? . . . . .	169
2.3	Mise en pratique . . . . .	170
2.4	Remarques sur l'utilisation . . . . .	171
2.5	La notation des templates . . . . .	171
2.6	Extension du système de templates . . . . .	172
2.7	L'annotation @Template . . . . .	173
3.	Les gabarits de pages (layouts) et les blocks . . . . .	174
3.1	La composition de pages . . . . .	174
3.2	Définition des gabarits . . . . .	175
3.3	Les blocks . . . . .	177
4.	Le langage Twig . . . . .	180
4.1	Les différents types d'instructions . . . . .	180
4.2	Manipulation des variables . . . . .	180
4.2.1	Utilisation de variables dans les templates . . . . .	180
4.2.2	Utilisation des variables de type tableau ou objet . . . . .	182

4.3	Structures de contrôle et tags . . . . .	183
4.3.1	Les conditions . . . . .	183
4.3.2	Les boucles . . . . .	183
4.4	Les balises Twig (tags) . . . . .	186
4.4.1	Créer et modifier des variables . . . . .	186
4.4.2	Twig et l'échappement . . . . .	187
4.5	Inclure des templates . . . . .	190
5.	Les filtres et les fonctions. . . . .	191
5.1	Présentation des filtres . . . . .	191
5.2	Les principaux filtres Twig . . . . .	192
5.2.1	Chaînes de caractères . . . . .	192
5.2.2	Échappement . . . . .	193
5.2.3	L'encodage . . . . .	194
5.3	Les fonctions . . . . .	196
5.3.1	Twig et le routage . . . . .	196
5.3.2	Débogage avec la fonction dump . . . . .	198
6.	La gestion des ressources statiques (images, feuilles de style, scripts JS...) . . . . .	198
6.1	Les ressources statiques dans une application Symfony . . . . .	198
6.2	Le cas des ressources statiques externes . . . . .	199
6.3	Référencer les ressources publiques depuis un template . . . . .	200
6.4	Cas pratique avec le framework CSS Bootstrap . . . . .	201
7.	Étendre le frontend avec Webpack Encore . . . . .	202
7.1	Présentation . . . . .	202
7.2	Installation et mise en place d'Encore . . . . .	203
7.2.1	Prérequis . . . . .	203
7.2.2	Installation d'Encore . . . . .	203
7.3	Utilisation d'Encore . . . . .	204

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

1. Présentation et concepts .....	207
1.1 Les principes de l'ORM .....	207
1.2 Architecture de Doctrine .....	208
1.2.1 DBAL .....	208
1.2.2 Entité .....	209
1.2.3 ORM .....	209
1.3 La notion d'entité .....	210
2. Installer et configurer Doctrine .....	210
2.1 Mise en place de Doctrine .....	210
2.2 Relation avec PDO .....	211
2.3 Configuration .....	212
3. Définition des entités et de leur mapping .....	214
3.1 Règles de conception des entités .....	214
3.2 Les syntaxes pour le mapping des entités .....	216
3.3 Le mapping d'entités simples .....	219
3.3.1 Définir une entité avec #[ORM\Entity] .....	219
3.3.2 Gérer les colonnes de la table avec #[ORM\Column] .....	220
3.3.3 #[ORM\Table] .....	223
3.3.4 Les clés primaires .....	224
3.3.5 Configurer les index .....	225
3.4 Le mapping des relations (clés étrangères) .....	226
3.4.1 #[ORM\OneToOne] .....	226
3.4.2 #[ORM\ManyToOne] .....	228
3.4.3 #[ORM\ManyToMany] .....	229
3.4.4 Relations bidirectionnelles .....	230
3.5 Les outils de la console .....	234
3.5.1 Repérer les erreurs de mapping .....	234
3.5.2 Générer le schéma des données à partir des entités .....	235

4. Manipulation des entités avec l'EntityManager . . . . .	235
4.1 Le rôle de l'EntityManager . . . . .	235
4.2 Insertion de données . . . . .	236
4.3 Modification de données . . . . .	238
4.4 Suppression de données . . . . .	239
4.5 Autres opérations de l'EntityManager . . . . .	239
4.5.1 refresh() . . . . .	239
4.5.2 detach() . . . . .	240
4.6 Les opérations en cascade . . . . .	240
5. Récupérer des entités . . . . .	244
5.1 Le repository . . . . .	244
5.1.1 Un rôle de centralisateur . . . . .	244
5.1.2 Les méthodes de base du repository . . . . .	245
5.1.3 Les méthodes personnalisées du repository . . . . .	247
5.2 Le DQL . . . . .	248
5.2.1 SELECT . . . . .	249
5.2.2 FROM . . . . .	250
5.2.3 JOIN et LEFT JOIN . . . . .	251
5.2.4 WHERE . . . . .	254
5.2.5 ORDER BY . . . . .	258
5.2.6 Les limites . . . . .	258
5.2.7 Les limites et la pagination . . . . .	259
5.3 Le QueryBuilder . . . . .	261
6. Fonctionnalités avancées . . . . .	263
6.1 Les extensions Doctrine . . . . .	263
6.1.1 Installation . . . . .	264
6.1.2 Utilisation d'un slug sur une entité . . . . .	264

**Chapitre 8**  
**La gestion des événements applicatifs**

1.	Concepts et écoute d'événement applicatifs . . . . .	267
1.1	La propagation des événements. . . . .	268
1.2	L'écoute des événements . . . . .	270
2.	Les événements du Kernel . . . . .	274
2.1	Les différents type d'événements . . . . .	274
2.2	Applications . . . . .	275
3.	Les événements de la console. . . . .	277
3.1	Prérequis . . . . .	277
3.2	Les événements . . . . .	277

**Chapitre 9**  
**Les formulaires**

1.	Un composant MVC . . . . .	279
1.1	Le modèle . . . . .	279
1.2	Le contrôleur. . . . .	281
1.3	La vue . . . . .	282
2.	Fonctionnement du composant. . . . .	283
2.1	L'objet « Form » . . . . .	283
2.1.1	Soumission. . . . .	283
2.1.2	Validation. . . . .	284
2.1.3	Vue . . . . .	284
2.2	Les types . . . . .	284
2.3	Les options . . . . .	285
2.4	Les objets « Form » et « FormBuilder » . . . . .	286
2.4.1	Le FormBuilder. . . . .	286
2.4.2	Structure de l'objet Form . . . . .	286
2.5	Association avec l'objet de la couche Modèle. . . . .	288
2.6	Formulaires sans objet . . . . .	289

2.7	La représentation des valeurs . . . . .	290
2.7.1	Transformation des données . . . . .	290
2.7.2	Illustration avec le type date . . . . .	291
3.	Les types de champs de formulaire . . . . .	292
3.1	L'héritage . . . . .	292
3.2	FormType . . . . .	293
3.2.1	label . . . . .	293
3.2.2	label_attr . . . . .	293
3.2.3	data . . . . .	293
3.2.4	required . . . . .	294
3.2.5	disabled . . . . .	294
3.2.6	mapped . . . . .	294
3.2.7	property_path . . . . .	295
3.2.8	attr . . . . .	295
3.2.9	trim . . . . .	295
3.2.10	error_bubbling . . . . .	296
3.3	TextType . . . . .	296
3.4	PasswordType . . . . .	296
3.5	RepeatedType . . . . .	297
3.5.1	type . . . . .	297
3.5.2	first_options et second_options . . . . .	298
3.5.3	options . . . . .	298
3.5.4	first_name . . . . .	298
3.5.5	second_name . . . . .	298
3.5.6	invalid_message . . . . .	298
3.6	ChoiceType . . . . .	298
3.6.1	choices . . . . .	299
3.6.2	expanded et multiple . . . . .	299
3.6.3	placeholder . . . . .	300
3.6.4	preferred_choices . . . . .	300
3.6.5	Types similaires . . . . .	301

3.7	EntityType .....	301
3.7.1	class .....	301
3.7.2	choice_label .....	301
3.7.3	query_builder .....	302
3.7.4	group_by .....	302
3.7.5	em .....	302
3.8	DateType .....	303
3.8.1	widget .....	303
3.8.2	format .....	303
3.8.3	model_timezone .....	304
3.8.4	view_timezone .....	304
3.8.5	years .....	304
3.8.6	months .....	304
3.8.7	days .....	304
3.8.8	placeholder .....	305
3.8.9	Types similaires .....	305
3.9	FileType .....	305
3.9.1	multiple .....	305
3.9.2	Récupérer les fichiers .....	306
3.9.3	Traiter les fichiers .....	307
3.10	CheckboxType .....	308
3.11	SubmitType, ResetType etButtonType .....	308
4.	Créer des formulaires réutilisables .....	310
4.1	Définir un formulaire avec la classe AbstractType .....	310
4.2	Utiliser un formulaire défini dans une classe .....	317
4.2.1	Définition manuelle .....	317
4.2.2	Avec l'injection de dépendances .....	317
5.	Validation des données .....	319
5.1	Objectifs .....	319
5.2	La définition des contraintes de validation .....	319
5.2.1	Ajout des contraintes lors de la configuration d'un formulaire .....	320
5.2.2	Ajout des contraintes sur l'objet associé au formulaire	321

5.2.3	Les différents formats de configuration . . . . .	325
5.2.4	Les options . . . . .	329
5.3	Les contraintes et leurs options . . . . .	331
5.3.1	NotBlank et NotNull . . . . .	332
5.3.2	IsNull et Blank . . . . .	332
5.3.3	IsTrue, IsFalse . . . . .	332
5.3.4	Type . . . . .	333
5.3.5	Email, Url et Ip . . . . .	333
5.3.6	Regex . . . . .	335
5.3.7	Length, Count . . . . .	335
5.3.8	Range . . . . .	337
5.3.9	Comparaisons . . . . .	337
5.3.10	Dates . . . . .	338
5.3.11	File . . . . .	338
5.3.12	Image . . . . .	339
5.3.13	Choice . . . . .	339
5.3.14	UniqueEntity . . . . .	340
5.3.15	Données financières . . . . .	341
5.3.16	Callback . . . . .	341
5.3.17	All . . . . .	343
5.3.18	Valid . . . . .	343
5.4	Groupes de validation . . . . .	344
5.5	Validation d'un objet hors du contexte d'un formulaire . . . . .	346
6.	Personnaliser le rendu - thèmes de formulaires . . . . .	347
6.1	Afficher le formulaire manuellement . . . . .	347
6.1.1	form_start() . . . . .	348
6.1.2	form_end() . . . . .	349
6.1.3	form_widget() . . . . .	349
6.1.4	form_errors() . . . . .	350
6.1.5	form_label() . . . . .	350
6.1.6	form_row() . . . . .	350
6.1.7	form_rest() . . . . .	350
6.1.8	Arborescence des parties de formulaires . . . . .	351

6.2	Créer des thèmes . . . . .	352
6.2.1	Formulaire d'exemple . . . . .	352
6.2.2	Créer et associer un thème de formulaires . . . . .	353
6.2.3	Comprendre le nom des blocks . . . . .	357

## Chapitre 10

### La sécurité dans une application Symfony

1.	Présentation et concepts de sécurité . . . . .	359
1.1	Les challenges de la sécurité des applications web . . . . .	359
1.2	La sécurité dans Symfony . . . . .	360
2.	Authentification . . . . .	361
2.1	Pare-feu . . . . .	361
2.2	Authentification HTTP . . . . .	362
2.3	Authentification par formulaire de connexion . . . . .	363
2.4	Connexion automatique des utilisateurs . . . . .	366
2.5	Déconnexion des utilisateurs . . . . .	366
3.	Utilisateurs et rôles . . . . .	367
3.1	L'utilisateur . . . . .	367
3.2	Les fournisseurs d'utilisateurs . . . . .	369
3.2.1	En mémoire . . . . .	369
3.2.2	Fournisseur d'utilisateurs de bases de données . . . . .	370
3.2.3	Fournisseur d'utilisateurs personnalisé . . . . .	373
3.2.4	Notes additionnelles . . . . .	376
3.3	Cryptage des mots de passe . . . . .	377
3.3.1	Passwords hashers . . . . .	377
3.3.2	Crypter un mot de passe . . . . .	378
3.4	Les rôles . . . . .	379

4. Autorisations .....	382
4.1 Les rôles, au cœur du processus.....	382
4.2 Vérifier le rôle de l'utilisateur .....	382
4.3 Sécuriser une action .....	384
4.4 Sécuriser une section de l'application .....	385
4.5 Sécuriser selon d'autres critères.....	385
4.6 Pour aller plus loin .....	387

## Chapitre 11

### Développer une API REST avec Symfony

1. Introduction à REST et concepts fondamentaux.....	389
1.1 Les concepts de REST.....	389
1.1.1 Les ressources.....	390
1.1.2 Le changement d'état d'une ressource .....	390
1.2 Architecture et protocole HTTP .....	390
1.3 Les Single-Page Applications .....	391
2. La gestion du format JSON .....	392
2.1 Présentation du format JSON .....	392
2.1.1 Représentation des données en JSON .....	393
2.1.2 Types de données .....	393
2.1.3 Structures.....	393
2.2 Le support de JSON en PHP.....	395
2.3 Et dans Symfony ? .....	395
3. Mise en place d'une API REST.....	396
3.1 Le service serializer .....	396
3.1.1 Sérialiser des données .....	397
3.1.2 Désérialiser des données .....	397
3.2 Adaptation des contrôleurs .....	398
4. Les objets de requête et de réponse .....	398
4.1 Le contenu et les en-têtes de requête .....	399
4.2 Manipulation de la réponse avec Response et JsonResponse ..	400

---

4.3	Les codes de réponse HTTP dans une API REST . . . . .	402
4.3.1	Problématique de l'état de la réponse. . . . .	402
4.3.2	Expression de la réponse avec HTTP . . . . .	403
4.3.3	Mise en œuvre . . . . .	404
5.	Tester une API REST . . . . .	405
5.1	Les limites du navigateur web . . . . .	405
5.2	Les outils . . . . .	405
5.2.1	Postman . . . . .	406
5.2.2	SOAP UI. . . . .	408
6.	Créer une API REST avec API Platform . . . . .	409
6.1	Présentation . . . . .	409
6.2	Installation . . . . .	409
6.3	La configuration d'API Platform . . . . .	410
6.4	Définition de l'API . . . . .	411
6.4.1	Les opérations de l'API . . . . .	414
6.4.2	Personnaliser l'API. . . . .	415

## Chapitre 12

### Tester son application Symfony

1.	Introduction au test logiciel. . . . .	417
1.1	Les tests : un indispensable pour la qualité logicielle . . . . .	417
1.2	Les différentes catégories de tests . . . . .	418
1.2.1	Les tests unitaires . . . . .	418
1.2.2	Les tests d'intégration . . . . .	419
1.2.3	Les tests fonctionnels . . . . .	419
1.3	Analogie. . . . .	419
1.4	L'approche des tests en PHP . . . . .	420
2.	Les tests unitaires avec PHPUnit . . . . .	421
2.1	Mise en place des tests . . . . .	421
2.2	Règle d'écriture des tests . . . . .	421
2.3	Exécuter les tests . . . . .	423

3. Les tests fonctionnels . . . . .	424
3.1 Différence par rapport aux tests unitaires et d'intégration . . . . .	424
3.2 Tester une action . . . . .	424
3.3 Les objets pour l'écriture des tests . . . . .	425
3.3.1 L'objet Client . . . . .	425
3.3.2 L'objet Crawler . . . . .	428
3.4 Soumettre un formulaire . . . . .	430
3.5 Pour aller plus loin . . . . .	431

## Chapitre 13

### Journalisation et surveillance avec Symfony

1. Générer des journaux avec Monolog . . . . .	433
1.1 La journalisation . . . . .	433
1.2 La librairie Monolog . . . . .	434
1.3 Le service logger . . . . .	435
1.4 Le fichier journal . . . . .	436
1.4.1 Identifier la cause d'un bogue . . . . .	436
1.4.2 Le problème . . . . .	436
1.5 Les gestionnaires (handlers) . . . . .	437
1.5.1 Définir plusieurs gestionnaires . . . . .	438
1.5.2 Envoyer des logs par e-mail . . . . .	438
1.5.3 Utiliser un tampon (buffer) . . . . .	439
1.5.4 Ajouter des informations complémentaires . . . . .	439
1.6 Les canaux (channels) . . . . .	441
1.6.1 Ajouter ses propres canaux . . . . .	441
1.6.2 Envoyer un enregistrement sur un canal donné . . . . .	442
1.6.3 Configurer les gestionnaires par canaux . . . . .	443
1.6.4 Gestion des erreurs 404 . . . . .	443

2. Le monitoring avec Prometheus et Grafana .....	446
2.1 Un allié proactif au logging .....	446
2.2 Préparation d'une application Symfony pour Prometheus ..	447
2.3 Instrumenter les mesures.....	451
2.4 Pour aller plus loin .....	452

## Chapitre 14

### Amélioration des performances

1. La mise en cache de pages .....	453
1.1 Autour du protocole HTTP .....	453
1.2 Un serveur proxy inverse (ou « reverse proxy »).....	454
1.2.1 HttpCache .....	456
1.2.2 Nginx .....	456
1.2.3 Varnish.....	457
1.3 Les en-têtes .....	462
1.4 Les réponses publiques et privées .....	462
1.5 L'expiration.....	463
1.5.1 L'en-tête Expires .....	463
1.5.2 Les directives max-age et s-max-age.....	464
1.5.3 L'annotation @Cache .....	464
1.6 La validation .....	466
1.6.1 Par date avec Last-Modified .....	466
1.6.2 Par empreinte avec l'en-tête ETag .....	468
1.7 Les ESI.....	469
1.7.1 Activation .....	469
1.7.2 Générer une balise ESI.....	470
2. L'autochargement des classes .....	471
3. Le cache avec Doctrine.....	472
3.1 Les différents types de cache .....	472
3.2 Configuration .....	473
4. Le cache d'annotations.....	474

5. Les sessions . . . . .	475
6. Autres optimisations . . . . .	476
6.1 Choix de sa SAPI PHP . . . . .	476
6.1.1 Qu'est-ce qu'une SAPI ? . . . . .	476
6.1.2 Module du serveur . . . . .	477
6.1.3 CGI . . . . .	477
6.1.4 FastCGI . . . . .	478
6.1.5 Conclusion . . . . .	478
6.2 Mise en cache d'OPCodes . . . . .	479
6.2.1 Les OPCodes . . . . .	479
6.2.2 Une étape lourde . . . . .	480
6.2.3 La mise en cache . . . . .	480
6.3 La compression des réponses . . . . .	480
6.3.1 Compression gzip . . . . .	481
6.3.2 Précompression . . . . .	481
6.4 Optimisation des images . . . . .	482
6.4.1 Validation . . . . .	482
6.4.2 Expiration . . . . .	482
6.4.3 Autres techniques . . . . .	482
7. Test des performances d'un site web . . . . .	483
7.1 Côté serveur . . . . .	483
7.1.1 Apache Bench . . . . .	483
7.1.2 Xhprof . . . . .	484
7.2 Côté client . . . . .	484

## Chapitre 15

### Internationalisation des applications Symfony

1. Introduction .....	485
1.1 Culture, internationalisation et régionalisation .....	485
1.1.1 La culture (Locale) .....	486
1.1.2 Internationalisation .....	486
1.1.3 Régionalisation .....	486
1.2 L'internationalisation dans Symfony .....	487
2. Détecter la culture d'un utilisateur .....	487
2.1 Les techniques .....	487
2.1.1 Négociation de contenu .....	487
2.1.2 Par l'URL .....	488
2.2 En pratique .....	488
3. Activation des traductions .....	489
3.1 Le composant translator .....	489
3.2 Configuration du framework .....	489
4. Les routes et les traductions .....	490
5. Les fichiers de traductions .....	491
5.1 Organisation et règles de nommage .....	491
5.2 Outilage pour la création des fichiers de traduction .....	492
5.2.1 Afficher la liste des traductions manquantes .....	492
5.2.2 Générer un fichier de traduction .....	493
6. Traduction d'un message .....	493
6.1 Le service translator .....	493
6.2 Les paramètres de substitution (placeholders) .....	494
6.3 Utilisation dans les templates Twig .....	495

## Annexes

1.	Créer une commande pour la console . . . . .	497
1.1	La configuration d'une commande . . . . .	497
1.2	Les objets input et output . . . . .	498
1.3	Le Service Container . . . . .	501
1.4	Commande d'exemple . . . . .	502
2.	Envoyer des e-mails grâce à Mailer . . . . .	503
2.1	Le protocole SMTP . . . . .	503
2.2	Le transport . . . . .	503
2.2.1	Le transport smtp . . . . .	504
2.2.2	Le transport sendmail . . . . .	505
2.3	Envoi d'un e-mail . . . . .	506
3.	Travailler avec les sessions . . . . .	507
3.1	Introduction . . . . .	507
3.2	Intégration des sessions dans Symfony . . . . .	507
3.3	Configuration du gestionnaire de sauvegarde . . . . .	508
3.3.1	Avec PHP . . . . .	508
3.3.2	Avec Symfony . . . . .	509
3.4	Les messages « flash » . . . . .	509
4.	Déployer une application symfony . . . . .	511
4.1	Le déploiement . . . . .	511
4.2	Faut-il déployer par FTP ? . . . . .	512
4.3	Les différentes étapes . . . . .	513
4.4	Capistrano et Capifony . . . . .	513
4.4.1	Installation . . . . .	513
4.4.2	Configuration . . . . .	514
4.4.3	Déploiement . . . . .	516
4.5	Fonctionnalités avancées . . . . .	516
	Index . . . . .	517