

Editions ENI

PHP 7

Développez un site web dynamique et interactif (2^e édition)

Collection
Ressources Informatiques

Table des matières

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI27PHP** 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 20
 - 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	36
5.2.3 Variables dynamiques (ou variables variables)	37
5.3 Types de données	38
5.3.1 Types de données disponibles	38
5.3.2 Types de données scalaires	38
5.3.3 Types de données spéciaux	45
5.4 Tableaux	46
5.4.1 Définition	46
5.4.2 Création	47
5.4.3 Manipulation	52
5.4.4 Portée	57
5.5 Opérateurs	57
5.5.1 L'opérateur d'affectation par valeur	57
5.5.2 L'opérateur d'affectation par référence	59
5.5.3 Les opérateurs arithmétiques	60
5.5.4 L'opérateur de chaîne	61
5.5.5 Les opérateurs combinés	61
5.5.6 Les opérateurs de comparaison	62
5.5.7 Les opérateurs logiques	63
5.5.8 L'opérateur ternaire	63
5.5.9 L'opérateur d'union NULL	64
5.5.10 L'opérateur de comparaison combinée	65
5.5.11 Précédence des opérateurs	66
5.6 Structures de contrôle	66
5.6.1 La structure if	66
5.6.2 La structure switch	69
5.6.3 La structure while	72
5.6.4 La structure do ... while	74
5.6.5 La structure for	75
5.6.6 Les instructions continue et break	77

5.7	Inclure un fichier	78
5.7.1	Fonctionnement	78
5.7.2	Utilisation	80
5.8	Interrompre le script.	82
5.9	Exercice 2 : variables et structures de contrôle	83

Chapitre 3

Utiliser les fonctions PHP

1.	Préambule	89
2.	Manipuler les constantes, les variables et les types de données	89
2.1	Constantes	89
2.2	Variables	91
2.3	Types de données	97
2.3.1	Conversions	97
2.3.2	Fonctions utiles	101
3.	Manipuler les tableaux	107
4.	Manipuler les nombres	120
5.	Manipuler les chaînes de caractères	125
6.	Utiliser les expressions rationnelles	142
6.1	Introduction	142
6.2	Structure d'une expression rationnelle	142
6.3	Fonctions.	151
7.	Manipuler les dates	156
8.	Générer un identifiant unique	170
9.	Manipuler les fichiers sur le serveur	171
9.1	Fonctions utiles.	171
9.2	Exemples d'utilisation.	179
10.	Manipuler les en-têtes HTTP	181
11.	Exercices	183
11.1	Exercice 3 : manipuler les données	183
11.2	Exercice 4 : écrire et lire un fichier sur le serveur	187

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

1. Fonctions	191
1.1 Introduction	191
1.2 Déclaration et appel	191
1.3 Paramètres	200
1.3.1 Syntaxe	200
1.3.2 Valeur par défaut	200
1.3.3 Déclaration du type de données	202
1.3.4 Passage par référence	204
1.3.5 Liste variable de paramètres	206
1.4 Considérations sur les variables utilisées dans les fonctions	209
1.4.1 Variables locales/globales	209
1.4.2 Variables statiques	210
1.5 Les constantes et les fonctions	212
1.6 Récursivité	213
1.7 Fonction anonyme	214
1.8 Fonction générateur	216
1.9 Exercice 5 : écrire des fonctions	219
2. Classes	221
2.1 Concept	221
2.2 Définir une classe	221
2.3 Instancier une classe	224
2.4 Héritage	226
2.5 Autres fonctionnalités sur les classes	228
2.5.1 Classes ou méthodes abstraites	228
2.5.2 Classes ou méthodes finales	230
2.5.3 Interfaces	231
2.5.4 Attributs ou méthodes statiques - Constantes de classes	232
2.5.5 Traits	235
2.5.6 Classes anonymes	237
2.6 Exceptions	238
2.7 Exercice 6 : écrire une classe	241
3. Espaces de noms	244

Chapitre 5
Gérer les erreurs dans un script PHP

- 1. Vue d'ensemble 249
- 2. Les messages d'erreur PHP..... 250
- 3. Les fonctions de gestion des erreurs 253
- 4. Exercice 7 : gérer les erreurs 268

Chapitre 6
Gérer les formulaires et les liens

- 1. Vue d'ensemble 271
 - 1.1 Introduction 271
 - 1.2 Les liens 271
 - 1.3 Les formulaires 274
 - 1.3.1 Petit rappel sur les formulaires 274
 - 1.3.2 Construire un formulaire dynamiquement 277
 - 1.3.3 Traiter un formulaire à l'aide d'un script PHP..... 281
 - 1.4 Récupérer les données d'une URL ou d'un formulaire..... 286
- 2. Récupérer les données passées par l'URL..... 288
 - 2.1 Considérations 288
 - 2.1.1 Que se passe-t-il si deux paramètres portent le même nom ? 288
 - 2.1.2 Utiliser un tableau pour passer des données dans l'URL. . . . 288
 - 2.2 Transmettre des caractères spéciaux..... 289
 - 2.3 Exercice 8 : récupérer des données passées par l'URL..... 292
- 3. Récupérer les données saisies dans le formulaire..... 295
 - 3.1 Considérations 295
 - 3.1.1 Que se passe-t-il si deux zones portent le même nom ? 295
 - 3.1.2 Que se passe-t-il s'il y a deux formulaires
dans la page HTML ?..... 295
 - 3.1.3 Utiliser un tableau pour récupérer les données saisies 295
 - 3.1.4 Passer des informations dans une zone de formulaire cachée. 297
 - 3.2 Les différents types de zones 299
 - 3.2.1 Vue d'ensemble 299
 - 3.2.2 Zones contenant du texte..... 301
 - 3.2.3 Groupes de boutons radio..... 302

3.2.4	Cases à cocher	302
3.2.5	Listes à sélection unique	305
3.2.6	Listes à sélection multiple	306
3.2.7	Boutons de validation	308
3.2.8	Boutons image	309
3.2.9	Boutons "reset" ou "button".	310
3.3	Synthèse.	310
3.4	Exercice 9 : récupérer des données saisies dans un formulaire.	312
4.	Contrôler les données récupérées	315
4.1	Vue d'ensemble	315
4.2	Vérifications classiques.	316
4.2.1	Nettoyage des espaces indésirables	316
4.2.2	Données obligatoires	316
4.2.3	Longueur maximum d'une chaîne.	316
4.2.4	Caractères autorisés pour une chaîne - Format	317
4.2.5	Validité d'une date - Plage de valeurs	317
4.2.6	Validité d'un nombre - Plage de valeurs	319
4.2.7	Validité d'une adresse e-mail	320
5.	Problèmes sur les données récupérées.	321
6.	Utilisation des filtres	328
6.1	Principes.	328
6.2	Application aux formulaires.	337
6.3	Exercices.	339
6.3.1	Exercice 10 : contrôler des données passées par l'URL	339
6.3.2	Exercice 11 : contrôler des données saisies dans un formulaire.	340
7.	Aller sur une autre page.	344
8.	Échanger un fichier entre le client et le serveur	350
8.1	Vue d'ensemble	350
8.2	Envoyer un fichier depuis le client (upload)	350
8.3	Télécharger un fichier à partir du serveur (download).	355

Chapitre 7
Accéder aux bases de données

- 1. Introduction 363
 - 1.1 Vue d'ensemble 363
 - 1.2 La notion de fetch 364
- 2. Utilisation de MySQL 365
 - 2.1 Préambule 365
 - 2.2 Connexion et déconnexion 367
 - 2.2.1 Connexion 367
 - 2.2.2 Déconnexion 368
 - 2.2.3 Obtenir des informations sur le serveur MySQL 368
 - 2.2.4 Obtenir des informations en cas d'erreur de connexion 368
 - 2.2.5 Exemple 369
 - 2.3 Sélectionner une base de données 370
 - 2.4 Utiliser des requêtes non préparées 371
 - 2.4.1 Vue d'ensemble 371
 - 2.4.2 Exécuter une requête 371
 - 2.4.3 Connaître le nombre de lignes dans le résultat d'une requête de lecture 373
 - 2.4.4 Extraire le résultat d'une requête de lecture 374
 - 2.4.5 Obtenir des informations sur le résultat d'une requête de mise à jour 382
 - 2.4.6 Gérer les erreurs 385
 - 2.5 Utiliser des requêtes préparées 386
 - 2.5.1 Vue d'ensemble 386
 - 2.5.2 Préparer une requête 388
 - 2.5.3 Lier des variables PHP aux paramètres de la requête 389
 - 2.5.4 Exécuter la requête préparée 390
 - 2.5.5 Lier des variables PHP aux colonnes du résultat d'une requête de lecture 391
 - 2.5.6 Extraire le résultat d'une requête de lecture 393
 - 2.5.7 Utiliser un résultat stocké 395
 - 2.5.8 Obtenir des informations sur le résultat d'une requête de mise à jour 397
 - 2.5.9 Gérer les erreurs 400
 - 2.5.10 Fermer une requête préparée 402
 - 2.6 Gérer les transactions 402

2.7	Appeler un programme stocké	405
2.7.1	Procédure stockée	405
2.7.2	Fonction stockée	409
2.8	Migrer de l'extension MySQL à l'extension MySQLi	411
2.9	Exercice 12 : utiliser MySQL	413
3.	Utilisation d'Oracle	420
3.1	Préambule	420
3.2	Environnement NLS	420
3.3	Connexion et déconnexion	421
3.3.1	Connexion	421
3.3.2	Déconnexion	423
3.3.3	Obtenir des informations sur le serveur Oracle	424
3.3.4	Obtenir des informations en cas d'erreur de connexion	424
3.3.5	Exemple	424
3.4	Exécuter une requête	426
3.4.1	Vue d'ensemble	426
3.4.2	Analyser une requête	427
3.4.3	Lier des variables PHP aux paramètres de la requête	428
3.4.4	Exécuter une requête	431
3.4.5	Extraire le résultat de la requête de lecture	432
3.4.6	Mettre à jour des données et gérer les transactions	446
3.4.7	Fermer un curseur	451
3.5	Appeler une procédure stockée	452
3.6	Illustration des problèmes liés à l'environnement NLS	457
3.7	Gérer les erreurs	460
3.8	Exercice 13 : utiliser Oracle	462
4.	Utilisation de SQLite	470
4.1	Préambule	470
4.2	Ouvrir et fermer une base de données	470
4.2.1	Ouvrir une base de données	470
4.2.2	Fermer une base de données	471
4.3	Exécuter une requête	472
4.3.1	Vue d'ensemble	472
4.3.2	Préparer une requête	473
4.3.3	Lier des variables PHP aux paramètres d'une requête préparée	474
4.3.4	Exécuter une requête préparée	476

- 4.3.5 Exécuter une requête non préparée 477
- 4.3.6 Extraire le résultat d'une requête de lecture 478
- 4.3.7 Mettre à jour des données et gérer les transactions 483
- 4.3.8 Fermer une requête préparée 488
- 4.4 Gérer les erreurs 488
- 4.5 Exercice 14 : utiliser SQLite 490
- 5. PHP Data Objects (PDO) 497
- 6. Gestion des apostrophes dans le texte des requêtes 500
- 7. Exemples d'intégration dans des formulaires 505
 - 7.1 Vue d'ensemble 505
 - 7.2 Construction d'une liste de sélection dans un formulaire 520
 - 7.3 Affichage d'une liste 522
 - 7.4 Formulaire de saisie en liste 526
 - 7.5 Formulaire de recherche et de saisie 531

Chapitre 8
Gérer les sessions

- 1. Description du problème 535
- 2. Authentification 536
 - 2.1 Vue d'ensemble 536
 - 2.2 Saisie de l'identification 536
 - 2.2.1 Identification par formulaire 536
 - 2.2.2 Identification par authentification HTTP 539
 - 2.3 Vérifier l'identification saisie 542
- 3. Utiliser des cookies 543
 - 3.1 Principe 543
 - 3.2 Application à la gestion des sessions 549
- 4. Utiliser la gestion des sessions de PHP 549
 - 4.1 Principes 549
 - 4.2 Mise en œuvre 550
 - 4.3 Gérer soi-même la transmission de l'identifiant de session 564
 - 4.3.1 Description du problème 564
 - 4.3.2 Solution 568
 - 4.4 Quelques directives de configuration supplémentaires 570

4.5	Exemples d'application	571
4.5.1	Principes	571
4.5.2	Avec authentification des utilisateurs	574
4.6	Remarques et conclusion	577
4.7	Exercice 15 : gérer les sessions	581
5.	Conserver des informations d'une visite à une autre	585
6.	Petite synthèse sur les variables Get/Post/Cookie/Session	591

Chapitre 9

Envoyer un courrier électronique

1.	Vue d'ensemble	595
2.	Envoyer un message texte sans pièce jointe	595
3.	Envoyer un message au format MIME	599
3.1	Préambule	599
3.2	Message au format HTML	599
3.3	Message avec pièce jointe	602
4.	Exercice 16 : envoyer un courrier électronique	605

Annexe

1.	Variables PHP prédéfinies	609
2.	Constantes PHP prédéfinies	611
3.	Exemples complémentaires	612
3.1	Introduction	612
3.2	Lire un document XML	613
3.3	Générer un document PDF	617
3.4	Générer une image	622
	Index	629

Editions ENI

Laravel

**Un framework efficace pour développer
vos applications PHP**

Collection
Expert IT

Table des matières

Avant-propos

- 1. Introduction 19
- 2. À qui s'adresse ce livre ? 20
- 3. Prérequis 20
- 4. Pourquoi choisir Laravel ? 21

Chapitre 1 Installation

- 1. Introduction 23
- 2. macOS 24
 - 2.1 À propos de Valet 24
 - 2.2 Installation de Valet 24
 - 2.2.1 Installer Homebrew 24
 - 2.2.2 Installer PHP7 25
 - 2.2.3 Installer Composer 25
 - 2.2.4 Installer Valet 26
 - 2.2.5 Créer un dossier de travail 26
 - 2.3 Créer une application Laravel 27
 - 2.3.1 Télécharger Laravel 27
 - 2.3.2 Créer un premier site 27
 - 2.4 Accéder au site 27
 - 2.4.1 Tester en local 27
 - 2.4.2 Partager un site 28
 - 2.4.3 HTTPS 28
- 3. Windows 29
 - 3.1 Choisir une méthode d'installation 29
 - 3.1.1 Difficultés d'installation avec Windows 29
 - 3.1.2 Homestead : une VM Ubuntu sous Windows 29
 - 3.2 Installation d'Homestead 30
 - 3.2.1 À propos de la virtualisation 30
 - 3.2.2 VirtualBox et Vagrant 31

3.2.3	Terminal	31
3.2.4	Git	32
3.2.5	Installer et initialiser Homestead	32
3.2.6	Créer une clé SSH	33
3.2.7	Créer un dossier de travail	33
3.2.8	Modifier le fichier Homestead.yaml	34
3.3	Créer une application Laravel	35
3.3.1	Installer Composer	35
3.3.2	Télécharger Laravel	35
3.3.3	Créer un site	36
3.3.4	Configurer le site pour Homestead	36
3.4	Accéder au site	37
3.5	Utiliser Homestead	37
4.	Linux	38
4.1	Installation rapide	38
5.	Autres méthodes d'installation	40
5.1	Artisan Serve	40
5.2	WampServer, XAMPP, MAMP et EasyPHP	41
5.3	Laragon	41
5.4	Valet pour Windows	42
5.5	Docker	42
5.6	Homestead pour macOS et Linux	42
5.7	Installation manuelle	42

Chapitre 2

Premiers pas

1.	Introduction	43
2.	Structure de l'application	44
3.	Configuration	45
3.1	Environnement	45
3.2	Quelques éléments à configurer	46

- 4. Choisir une base de données 47
 - 4.1 Quelle base de données ? 47
 - 4.2 PostgreSQL 48
 - 4.2.1 Installer PostgreSQL 48
 - 4.2.2 Configuration 48
 - 4.3 MySQL 49
 - 4.3.1 Installer MySQL 49
 - 4.3.2 Configuration 49
- 5. Utiliser Artisan 50
- 6. Commandes utiles pour déboguer 50
 - 6.1 dd 50
 - 6.2 dump 51
 - 6.3 Log 51

Chapitre 3
Créer des routes

- 1. Prise en main 53
 - 1.1 Définir une route 53
 - 1.2 Les méthodes HTTP 54
 - 1.3 Choisir les adresses des routes 55
 - 1.4 La fonction de rappel 56
- 2. Paramètres des routes 57
 - 2.1 Déclarer des paramètres 57
 - 2.2 Utiliser les paramètres capturés 58
 - 2.3 Utiliser plusieurs paramètres 58
 - 2.4 Paramètres optionnels 59
- 3. Réponses 60
 - 3.1 Les types de réponses 60
 - 3.2 Les réponses textuelles 61
 - 3.3 Les réponses JSON 62
 - 3.4 Les réponses HTML 63

3.5	Les redirections	64
4.	Groupes de routes	65
4.1	Préfixes de routes	65
4.2	Sous-domaines	67
4.3	Autres usages	68
5.	Routes et vues	69

Chapitre 4 Utiliser des vues

1.	Créer des vues	71
1.1	Afficher du HTML	71
1.2	Organisation en sous-dossiers	72
1.3	Insérer du code PHP dans les vues	73
2.	Premiers pas avec le moteur de gabarit Blade	74
2.1	Pourquoi un système de gabarit ?	74
2.2	Syntaxe de base	76
3.	Les variables	77
3.1	Afficher des données	77
3.2	Passer des variables aux vues	78
3.3	Afficher des données non protégées	80
4.	Directives	80
4.1	Définition	80
4.2	Boucles	81
4.2.1	Afficher une liste	81
4.2.2	Autres types de boucles	82
4.2.3	La variable <code>\$loop</code>	82
4.3	Conditions	84
5.	Organiser les vues	85
5.1	Héritage de gabarit et mise en page	85
5.1.1	Créer une base	86
5.1.2	Hériter de la base et créer des sections	88

- 5.1.3 En résumé 90
- 5.2 Les fichiers statiques 90
 - 5.2.1 Le dossier public 90
 - 5.2.2 Introduction à Laravel Mix 91
- 5.3 Inclusions 93
- 5.4 Piles 95
- 5.5 Composants 96
- 5.6 À propos du cache 97
- 6. Internationalisation 98
 - 6.1 Fichiers de traductions 98
 - 6.2 Définition de la langue courante 99
 - 6.3 Utilisation dans les vues 100
 - 6.3.1 Variables 100
 - 6.3.2 Pluriel 101
 - 6.4 Groupes de routes pour l'internationalisation 101
- 7. En résumé 103

Chapitre 5
Eloquent ORM

- 1. Object Relational Mapping 105
 - 1.1 À propos des modèles 105
 - 1.2 Manipuler des objets 106
 - 1.3 Active Record 106
- 2. Construire des modèles 107
 - 2.1 Utiliser Artisan 107
 - 2.1.1 Initialisation d'un modèle 107
 - 2.1.2 Modèles et migrations 108
 - 2.2 Créer une migration 109
 - 2.2.1 Initialisation d'une table 109
 - 2.2.2 Créer des colonnes 110
 - 2.2.3 Exemples et conseils de création de tables 112

2.2.4	Mise à jour d'une table	115
2.2.5	Lancer les migrations	116
2.2.6	Rollback	117
2.3	Lier les propriétés aux colonnes	117
2.3.1	Champs dates	118
2.3.2	Autres propriétés	119
3.	Utiliser les modèles	120
3.1	Créer et enregistrer une instance de modèle	120
3.1.1	Instancier un modèle	120
3.1.2	Enregistrement d'un modèle	121
3.1.3	Créer et enregistrer un modèle en une ligne	121
3.2	Récupérer un modèle	123
3.2.1	Récupérer par l'identifiant	123
3.2.2	Récupérer par une requête	124
3.2.3	Récupérer ou créer	125
3.3	Modifier un modèle	125
3.3.1	Mise à jour d'un modèle	125
3.3.2	Mettre à jour plusieurs modèles	126
3.4	Supprimer un modèle	126
3.4.1	Supprimer un modèle	126
3.4.2	Supprimer plusieurs modèles	127
3.4.3	Soft delete	127
3.5	Utiliser une collection de modèles	129
3.5.1	Récupérer une liste	129
3.5.2	Requêtes complexes (Query Builder)	130
3.6	Utilisation d'un modèle : mise en pratique	131
3.6.1	Création du modèle	132
3.6.2	Création des routes	134
3.6.3	Création de la vue	135
3.6.4	Test	137
4.	Organiser les modèles	137
4.1	Espace de noms par défaut	137
4.2	Choisir un autre espace de noms	138

- 5. Les relations 139
 - 5.1 Relations 1..1. 139
 - 5.1.1 Création des modèles 140
 - 5.1.2 Structure des tables 140
 - 5.1.3 Déclarer les relations 142
 - 5.1.4 Accéder aux relations 144
 - 5.2 Relations 1..n 145
 - 5.2.1 Création des modèles 145
 - 5.2.2 Structure des tables 145
 - 5.2.3 Déclarer les relations 147
 - 5.2.4 Accéder aux relations 149
 - 5.3 Relations n..n 150
 - 5.3.1 Création des modèles 151
 - 5.3.2 Structure des tables 151
 - 5.3.3 Déclarer les relations 155
 - 5.3.4 Accéder aux relations 156
- 6. Mutation, accesseurs et transformation 158
 - 6.1 Accesseurs 158
 - 6.2 Mutateurs 160
 - 6.3 Transformation. 161
- 7. Requêtes manuelles 162

Chapitre 6
Organiser grâce aux contrôleurs

- 1. Définition et usage 163
 - 1.1 Qu'est-ce qu'un contrôleur ? 163
 - 1.2 Déplacer le code des fonctions de rappel des routes. 163
- 2. Créer et utiliser un contrôleur 164
 - 2.1 Créer un contrôleur avec Artisan. 164
 - 2.2 Structure d'un contrôleur 165
 - 2.3 Lier une route à un contrôleur 166

Un framework efficace pour développer vos applications PHP

2.4	Variables	166
2.4.1	Passer un paramètre à un contrôleur	166
2.4.2	Valeurs optionnelles	167
3.	Dépendances	168
3.1	Injection de dépendance	168
3.1.1	Fonctionnement	168
3.1.2	Ordre des arguments	169
3.2	La classe Request	170
3.2.1	Obtenir les informations de la requête	170
3.2.2	Données des formulaires	171
3.2.3	URI et méthodes HTTP	172
3.3	Autres instances injectables	173
4.	Organiser une application	175
4.1	Lister des routes	175
4.2	Espaces de noms	176
4.3	Des contrôleurs CRUD de ressources	177
4.3.1	Rappels sur le CRUD	177
4.3.2	Séparer les contrôleurs par ressource	177
4.3.3	Contrôleur de ressources	178
5.	Bilan des premiers chapitres	179

Chapitre 7

Filtrer et décorer avec les middlewares

1.	À propos des middlewares	181
1.1	Cycle de vie de l'application	181
1.2	Exemple du middleware d'authentification	183
1.3	Autres exemples	184
2.	Créer un middleware	184
2.1	Utiliser Artisan	184
2.2	Structure d'un middleware	185

- 3. Utiliser un middleware 186
 - 3.1 Middleware global 186
 - 3.2 Middleware pour une route 188
 - 3.2.1 Déclaration 188
 - 3.2.2 Utilisation 189
 - 3.3 Middleware pour un groupe de routes 189
 - 3.3.1 Déclaration 189
 - 3.3.2 Utilisation 190
 - 3.4 Middleware pour un contrôleur 190

Chapitre 8
Traiter des formulaires

- 1. Sécuriser les formulaires 193
 - 1.1 Attaques CSRF 193
 - 1.2 Fonctionnement du CSRF Token 194
 - 1.3 Utilisation du csrf_field dans les formulaires 194
 - 1.4 Requêtes Ajax 195
 - 1.5 Désactiver la protection pour certaines routes 196
- 2. Valider les données 197
 - 2.1 Introduction 197
 - 2.2 Écrire les règles de validation 197
 - 2.2.1 Contexte 197
 - 2.2.2 Logique de validation dans les contrôleurs 199
 - 2.2.3 Règles disponibles 201
 - 2.2.4 Enregistrement en base de données 202
 - 2.3 Afficher les messages d’erreur 203
 - 2.3.1 La variable \$errors 203
 - 2.3.2 Affichage dans la vue 204
 - 2.3.3 Traduction des messages d’erreur 204
 - 2.3.4 Récupérer les anciennes valeurs du formulaire 205

Un framework efficace pour développer vos applications PHP

2.4	Afficher un message de succès	206
2.4.1	Contrôleur	206
2.4.2	Vue	207
2.5	Méthode de validation alternative	207
3.	Formulaires PATCH, PUT et DELETE	208
4.	Upload de fichiers	209
4.1	Création du formulaire	209
4.2	Validation des données	210
4.3	Récupérer les fichiers	211
4.4	Stocker les fichiers	211
4.4.1	Stocker sur le serveur	211
4.4.2	Paramétrer un autre espace de stockage	212
4.4.3	Dossier public	213

Chapitre 9

Authentification et autorisations

1.	Démarrer rapidement	215
1.1	Installation	216
1.2	Le modèle User et les migrations	216
1.3	Les vues, les routes et les contrôleurs	218
1.3.1	Les vues	218
1.3.2	Les routes	218
1.3.3	Les contrôleurs	220
1.4	Utiliser le système d'authentification	221
1.5	Espaces privés : protéger les routes	221
2.	Paramétrage avancé	222
2.1	Personnaliser le mail de réinitialisation de mot de passe	222
2.2	Événements et écouteurs : envoyer un mail de bienvenue	223
2.3	Ajouter un champ téléphone lors de l'inscription	226
2.3.1	Migration	226
2.3.2	Modification du modèle	228

- 2.3.3 Modification de la vue 228
- 2.3.4 Modification du contrôleur 229
- 3. Autres méthodes d'authentification 230
 - 3.1 Laravel Socialite 230
 - 3.2 Authentification manuelle 232
 - 3.3 Prise de contrôle (impersonation) 232
 - 3.4 Authentification HTTP 233
 - 3.5 Laravel Passport 234
- 4. Déconnexion 234
- 5. Rappels sur la sécurité 235
 - 5.1 Mots de passe 235
 - 5.2 Cryptage 237
- 6. Autorisations 238
 - 6.1 Portes 238
 - 6.1.1 Déclarer une porte 239
 - 6.1.2 Protéger à l'aide d'une porte 240
 - 6.2 Politiques d'autorisation 242
 - 6.2.1 Création d'une politique d'autorisation 242
 - 6.2.2 Protéger avec une politique d'autorisation 244
 - 6.3 Rôles 246
 - 6.3.1 Ajout d'une colonne « role » 247
 - 6.3.2 Politique d'autorisation 248
 - 6.3.3 Protéger les routes 250

Chapitre 10
Les sessions et le cache

- 1. Les sessions 251
 - 1.1 Choisir un pilote 251
 - 1.1.1 Stocker dans des fichiers pour le développement 252
 - 1.1.2 Utiliser Redis 252
 - 1.1.3 Utiliser une base de données 253

Un framework efficace pour développer vos applications PHP

1.2	Stocker des données	253
1.3	Retrouver des données	254
1.4	Supprimer des données	254
1.5	Messages flash	255
2.	Le cache	256
2.1	Choisir un pilote	256
2.1.1	Stockage dans des fichiers	257
2.1.2	Redis	257
2.1.3	Base de données	257
2.2	Stocker, retrouver et supprimer des données	257

Chapitre 11

Les files d'attente

1.	Introduction	261
1.1	Pourquoi utiliser des files d'attente ?	261
2.	Configuration	262
2.1	Pilote	262
2.2	Queue Worker	262
3.	Créer et expédier des tâches	263
3.1	Créer une tâche	263
3.2	Expédier une tâche	265
4.	Monitoring	267
4.1	Traitement des erreurs	267
4.2	Horizon	268

Chapitre 12

Envoyer des e-mails

1. Développement local	269
1.1 Mailtrap	270
1.1.1 Configuration	270
1.1.2 Utilisation	271
1.2 Autres méthodes	271
1.2.1 E-mails dans les logs	271
1.2.2 Destinataire global	272
2. Préparer des e-mails	272
2.1 Création de la classe Mailable	272
2.2 Création des vues	274
2.2.1 HTML	274
2.2.2 Markdown	274
2.3 Configuration de l'e-mail	276
2.3.1 Passer des données	276
2.3.2 Expéditeur	278
2.3.3 Sujet des e-mails	278
2.4 Pièces jointes	279
3. Envoyer des e-mails	280
3.1 Envoyer un e-mail immédiatement	280
3.2 Mise en file d'attente	281
3.3 Envoyer un e-mail plus tard	282
3.4 Afficher un e-mail sans l'envoyer	283
4. Utiliser des notifications	284
4.1 Créer une notification	284
4.1.1 Commande de création	284
4.1.2 Définir la liste des canaux	285
4.2 Paramétrage des différents canaux	286
4.2.1 E-mail	286
4.2.2 Slack	288
4.2.3 Base de données	289

Un framework efficace pour développer vos applications PHP

4.2.4	SMS	291
4.3	Envoyer une notification	292
4.3.1	Envoyer à un utilisateur	292
4.3.2	Envoyer à plusieurs utilisateurs	293
4.3.3	Mettre en file d'attente	293
5.	Choisir un pilote en environnement de production	294
5.1	Amazon SES	295
5.2	Mailgun	296
5.3	SMTP	297

Chapitre 13

Tests automatisés

1.	Introduction	299
2.	Tests unitaires	301
2.1	Introduction	301
2.2	Créer un test unitaire	301
2.3	Lancer les tests	304
3.	Base de données	304
3.1	Configuration	305
3.1.1	Modifier le fichier de configuration	305
3.1.2	Mettre à jour le fichier phpunit.xml	307
3.2	Migrations et transactions	307
3.3	Assertions	309
3.4	Fabriques	311
3.4.1	Initialisation	311
3.4.2	Utiliser les fabriques dans les tests	313
4.	Tester des fonctionnalités	314
4.1	Introduction	314
4.2	Créer un test de fonctionnalité	315
4.3	Lancer les tests	316

- 5. Laravel Dusk 316
 - 5.1 Introduction 316
 - 5.2 Installation 317
 - 5.3 Créer un test 318
 - 5.3.1 Créer la classe 318
 - 5.3.2 Interagir avec le navigateur 319
 - 5.4 Lancer les tests 322
 - 5.5 Pour aller plus loin 323
 - 5.5.1 Utiliser un autre navigateur 323
 - 5.5.2 Utilisation des pages 324
- 6. Simulation 326
- 7. Automatiser le lancement des tests 327

Chapitre 14
Créer et utiliser des composants

- 1. Intégrer un composant 329
 - 1.1 Introduction 329
 - 1.2 Choisir un composant 330
 - 1.3 Ajouter un composant 332
 - 1.4 Mettre à jour les composants 333
 - 1.5 Version partagée des composants 333
 - 1.5.1 Partager le fichier composer.lock 333
 - 1.5.2 Installer les dépendances 334
 - 1.5.3 Mettre à jour les dépendances 334
 - 1.5.4 En résumé 334
- 2. Composants utiles et de qualité 335
- 3. Créer ses propres composants 337
 - 3.1 Introduction 337
 - 3.2 Composant Laravel ou PHP 338

3.3	Réaliser un composant pour Laravel	338
3.3.1	ServiceProvider	339
3.3.2	Routes	340
3.3.3	Contrôleurs	341
3.3.4	Vues	341
3.3.5	Configuration	342
3.3.6	Migrations	343
3.4	Mettre à disposition un composant (Packagist)	344
3.4.1	Packagist et Composer	344
3.4.2	Fichier composer.json	345
3.4.3	Package Discovery pour les composants Laravel	346
3.4.4	Publier un composant	348

Chapitre 15

Assistants

1.	Introduction	349
2.	Manipuler des chaînes de caractères	350
2.1	Transformer des chaînes	350
2.1.1	Casse : Serpents, Kebabs et Chameaux	350
2.1.2	Slug	351
2.1.3	Limiter une chaîne à un certain nombre de caractères	351
2.2	Manipuler des portions de chaînes	352
2.2.1	Extraire une partie	352
2.2.2	Remplacer une partie d'une chaîne	352
2.3	Singulier et pluriel	353
2.4	Vérifier des chaînes	353
2.4.1	Commence ou finit par	353
2.4.2	Vérifier si une chaîne en contient une autre	354
2.4.3	Vérifier si une chaîne est similaire	354
2.5	Fonctions diverses	354

- 3. Fonctions sur les tableaux 355
 - 3.1 Récupérer des éléments 355
 - 3.2 Transformer un tableau 357
 - 3.3 Vérifier la présence d'éléments 358
- 4. Autres fonctions assistantes 358
 - 4.1 Chemins et dossiers 358
 - 4.2 URL 359
 - 4.3 Arrêter le programme 359
 - 4.4 Requêtes et réponses 360
 - 4.5 Sécurité 360
- 5. Les collections 361
 - 5.1 Introduction 361
 - 5.2 Transformer un tableau en collection 362
 - 5.3 Manipuler des collections 363
- 6. Tinker 366

Chapitre 16
Déployer un site en production

- 1. Considérations 367
 - 1.1 Choisir son environnement de production 368
 - 1.2 Les solutions à disposition 370
- 2. Laravel Forge 371
- 3. Préparer un serveur manuellement 373
 - 3.1 Prérequis 373
 - 3.1.1 Ce qu'il faut avoir 373
 - 3.1.2 Ce qui sera installé 373
 - 3.2 Préparer le serveur 374
 - 3.2.1 Initialisation 374
 - 3.2.2 Composer 374
 - 3.2.3 PostgreSQL 374
 - 3.2.4 Firewall 375

Un framework efficace pour développer vos applications PHP

3.2.5 Utilisateur	375
3.3 Installer l'application	376
3.4 Configurer Nginx	377
3.5 Désactiver l'authentification par mot de passe	379
3.5.1 Créer un utilisateur pour se connecter	379
3.5.2 Protéger l'authentification par clé SSH uniquement	380
3.6 HTTPS	380
3.7 Mise à jour	382
3.8 Conclusion	382
4. Intégration continue et déploiement	383
Index	385