

Version en ligne

OFFERTE !


pendant 1 an

+ QUIZ 

Phalcon 3

Développez des applications web
complexes et performantes
en **PHP**
(nouveautés de la version 4 incluses)

En téléchargement

 le code source

 des scripts

 Informatique technique


Collection

epsilon



Jérémy PASTOURET



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

Avant-propos

- 1. Au commencement 37
- 2. Les raisons d'être de ce livre 38
- 3. Le public visé. 38
- 4. Le contenu du livre 39
- 5. Remerciements 41

Chapitre 1

Présentation de Phalcon

- 1. Introduction 43
- 2. Historique du framework. 44
- 3. Caractéristiques techniques de Phalcon 45
- 4. Versions de Phalcon 47
- 5. Pourquoi choisir Phalcon ? 48

Chapitre 2

Installation

- 1. Présentation des éléments à installer. 49
 - 1.1 Serveur web. 49
 - 1.2 Langage PHP 50
 - 1.3 Choix de la version de Phalcon 50
 - 1.4 Gestionnaire de dépendances PHP. 52
 - 1.5 Moteur de base de données 53

2.	Configuration sous Windows	53
2.1	Installation d'Apache 2.	53
2.2	Installation de PHP.	57
2.3	Compilation depuis les sources	57
2.4	Ajout de l'extension	59
2.5	Installation de Composer.	62
2.6	Installation de PostgreSQL	63
2.7	Installation de DBeaver	64
3.	Configuration sous Ubuntu.	66
3.1	Installation d'Apache 2.	66
3.2	Installation de PHP.	68
3.3	Compilation depuis les sources de Phalcon	72
3.4	Ajout de l'extension	75
3.5	Installation de Composer.	76
3.6	Installation de PostgreSQL	77
3.7	Installation de DBeaver	79
4.	Docker	80
4.1	Création d'un fichier Dockerfile	80
4.2	Création d'un fichier docker-compose.yml	82
5.	Initialisation d'une base de données PostgreSQL	84
5.1	Connexion à une base de données.	84
5.2	Import d'un schéma avec des données	88
6.	Nouveautés de la version 4	91
7.	Migration vers la version 4.	92
7.1	Configuration sous Windows	92
7.1.1	Mise à jour de Phalcon	92
7.1.2	Téléchargement et installation de l'extension psr	93
7.2	Configuration sous Ubuntu.	94
7.2.1	Téléchargement, compilation et installation de l'extension psr	94
7.2.2	Mise à jour de Phalcon	95

- 7.3 Docker 97
 - 7.3.1 Modification du Dockerfile. 97
 - 7.3.2 Modification du docker-compose. 97

Chapitre 3

Création d'un premier projet

- 1. Initialisation avec Phalcon DevTools 99
- 2. Installation avec Composer 100
- 3. Création du projet 102
 - 3.1 Exploration du Hello World. 104
 - 3.1.1 Présentation des fichiers à la racine du projet 105
 - 3.1.2 Présentation des répertoires à la racine 107
- 4. Compréhension du squelette de base de Phalcon. 108
 - 4.1 Configuration 108
 - 4.1.1 Utilisation des constantes de base 109
 - 4.1.2 Configuration d'une base de données. 109
 - 4.1.3 Intégration des répertoires clés d'un projet Phalcon ... 111
 - 4.1.4 Ajout de données complémentaires 113
 - 4.2 Routage des pages. 114
 - 4.2.1 Description du fichier de base. 115
 - 4.2.2 Gestion par défaut des liens 116
 - 4.2.3 Définition d'URL. 116
 - 4.2.4 Gestion des paramètres d'URL 117
 - 4.3 Contrôleurs 119
 - 4.3.1 Récupération de paramètres provenant du router.php. 120
 - 4.4 Modèles 121
 - 4.4.1 Création d'une classe Modèle 122
 - 4.4.2 Description du modèle généré 122
 - 4.4.3 Récupération de données 124

4.5	Vues	127
4.5.1	Passage des données à la vue	127
4.5.2	Affichage des données dans une vue	128
4.6	Zone publique	129
4.6.1	Présentation	129
4.6.2	Import des fichiers	131
4.6.3	Intégration des fichiers de la zone publique dans une vue	131
5.	Débogage d'un projet	132
5.1	Installation de Xdebug	132
5.1.1	Installation sous Windows	132
5.1.2	Installation sous Linux	133
5.1.3	Installation sous Docker	133
5.1.4	Vérification d'installation de l'extension	135
5.1.5	Installation de l'extension pour navigateur	135
5.2	Configuration sur PHPStorm	136
5.2.1	Paramétrage de Xdebug	136
5.2.2	Intégration du code de Phalcon	139
6.	Nouveautés de la version 4	142
6.1	Fonction handle	142
6.1.1	Changement dans l'objet Application	142
6.1.2	Changement dans l'objet router	143
6.2	Changement de l'espace de nom de la classe Url	143
6.3	Changement dans les options de Volt	144
6.4	Modification au niveau des modèles de bases de données	144
6.4.1	La fonction find	144
6.4.2	La fonction getSource	145
6.5	Création d'un plugin pour PHPStorm	145
7.	Migration vers la version 4	146
7.1	Phalcon DevTools	146
7.1.1	Mise à jour classique	146
7.1.2	Autre méthode d'installation	147
7.2	Mise à jour de phalcon/ide-stubs	153

7.3	Modification de la fonction handle	154
7.3.1	Modification du point d'entrée	154
7.3.2	Modification du fichier de routage	155
7.4	Modification de l'espace de nom de la classe Url	155
7.5	Modification des options de Volt	156
7.6	Modification de la fonction find	156

Chapitre 4

Services de Phalcon

1.	Injection de dépendances (DI)	157
1.1	Une dépendance simple	158
1.1.1	Explication	158
1.1.2	Mise en œuvre d'un service avec une simple instanciation	159
1.1.3	Mise en œuvre d'un service avec une fonction anonyme	160
1.2	Une dépendance partagée	163
1.2.1	Explication	163
1.2.2	Mise en œuvre	163
1.2.3	Liste des dépendances de Phalcon	164
2.	Inclusion de nouveaux répertoires	167
2.1	Ajout du dossier dans la configuration	167
2.2	Chargement du contenu d'un répertoire	168
2.3	Ajout d'un espace de noms pour un répertoire	168
3.	Nouveautés de la version 4	170
3.1	Les collections	170
3.1.1	Construction d'une collection	170
3.1.2	Récupérer le nombre d'éléments d'une collection/tableau	170
3.1.3	Réutilisation d'une collection	171
3.1.4	Récupération d'un élément	172
3.1.5	Vérification de la présence d'un élément	173

3.1.6	Modification d'un élément	174
3.1.7	Ajout d'un élément	174
3.1.8	Suppression d'un élément	174
3.1.9	Parcours d'une collection	175
3.1.10	Sérialisation d'une collection	176
3.1.11	Désérialisation d'une collection	176
3.1.12	Transformation en un tableau standard	176
3.1.13	Transformation en JSON	177
3.2	La lecture seule	177
3.3	La classe Registry	178
3.4	La classe d'aide pour les tableaux	179
3.4.1	Récupération de la première valeur	179
3.4.2	Récupération de la première valeur en suivant une fonction	180
3.4.3	Récupération de la première clé	181
3.4.4	Récupération de la première clé en suivant une fonction	181
3.4.5	Récupération de la dernière valeur	182
3.4.6	Récupération de la dernière valeur en suivant une fonction	182
3.4.7	Récupération de la dernière clé	183
3.4.8	Récupération de la dernière clé en suivant une fonction	183
3.4.9	Récupération d'un élément	184
3.4.10	Ajout d'un élément	185
3.4.11	Modification d'un élément	187
3.4.12	Vérification de la présence d'une clé	188
3.4.13	Groupement de données	188
3.4.14	Groupement de données en suivant une fonction	189
3.4.15	Groupement de données avec une liste d'objets	190
3.4.16	Trier un tableau	191
3.4.17	Trier un tableau en précisant le sens	191
3.4.18	Vérification de l'unicité des valeurs d'un tableau	192
3.4.19	Récupération des valeurs d'une clé spécifique	193

3.4.20	Suppression d'un niveau de profondeur dans un tableau	194
3.4.21	Mise à plat d'un tableau	195
3.4.22	Séparation d'un tableau en tronçon	195
3.4.23	Extraction d'une partie d'un tableau	197
3.4.24	Séparation des clés et des valeurs d'un tableau	198
3.4.25	Transformation en objet	199
3.4.26	Validation des données	199
3.4.27	Validation partielle d'un tableau	201
3.4.28	Filtrage des données avec une liste d'inclusion	201
3.5	La classe d'aide pour les fichiers	202
3.5.1	Récupération du nom du fichier à partir d'un chemin	202
3.6	La classe d'aide pour les nombres	203
3.6.1	Vérification de la présence d'un chiffre dans un intervalle donné	203
3.7	La classe d'aide pour les chaînes de caractères	203
3.7.1	Transformation d'une chaîne de caractères en CamelCase (Casse de chameau)	203
3.7.2	Transformation d'une chaîne de caractères en SnakeCase	204
3.7.3	Remplacement d'espaces par des tirets du bas	205
3.7.4	Concaténation de chaînes de caractères	206
3.7.5	Compter le nombre de voyelles	206
3.7.6	Transformation de la première lettre en minuscule	206
3.7.7	Incrémement d'un nombre dans une chaîne de caractères	207
3.7.8	Décrémement d'un nombre dans une chaîne de caractères	208
3.7.9	Génération d'une structure de dossier à partir d'un nom de fichier	209
3.7.10	Contrôle et mise à jour du chemin d'un répertoire	209
3.7.11	Génération de chaînes aléatoires	209
3.7.12	Génération de caractères aléatoires	211

3.7.13	Vérification des caractères présents au début d'une chaîne	212
3.7.14	Vérification des caractères présents à la fin d'une chaîne	213
3.7.15	Vérification de la présence d'un mot ou d'une expression dans une chaîne	213
3.7.16	Vérification de la présence de majuscules sur l'ensemble de la chaîne	214
3.7.17	Vérification de la présence de minuscules sur l'ensemble de la chaîne	214
3.7.18	Vérification du fait que deux chaînes de caractères sont des anagrammes	214
3.7.19	Vérification du fait qu'une chaîne de caractères est un palindrome	215
3.7.20	Récupération de la première occurrence d'une chaîne comprise entre deux délimiteurs	215
3.7.21	Rendre plus lisible une chaîne de caractères	216
3.7.22	Transformation d'une chaîne de caractères en minuscules	216
3.7.23	Transformation d'une chaîne de caractères en majuscules	216
3.7.24	Suppression de slashes en trop	217

Chapitre 5

Gestion de requête HTTP

1.	Récupération des données d'une requête	219
1.1	Méthode GET	219
1.2	Méthode POST	221
1.2.1	Envoi de données avec un formulaire	221
1.2.2	Analyse de la requête émise	222
1.2.3	Récupération des données émises en POST	223
1.2.4	Nettoyage des données avec des filtres	223
1.2.5	Récupération de fichiers	228

2.	Gestion des cookies	230
2.1	Présentation	230
2.2	Usage	231
2.2.1	Ajout du service de cryptage	231
2.2.2	Création et envoi d'un cookie	232
2.2.3	Vérification de la présence d'un cookie	233
2.2.4	Récupération d'un cookie	234
2.2.5	Suppression d'un cookie	234
2.2.6	Désactivation du cryptage des cookies	235
3.	Gestion des sessions	236
3.1	Présentation	236
3.2	Usage	238
3.2.1	Gestion des sessions avec un moteur tiers	239
3.2.2	Enregistrement d'information en session	240
3.2.3	Vérification de la présence d'une propriété en session	240
3.2.4	Récupération d'information en session	240
3.2.5	Suppression d'un élément en session	241
3.2.6	Suppression de tous les éléments en session	241
3.2.7	Séparation des sessions par application	241
4.	Gestion des réponses	243
4.1	Envoi d'une réponse simple	243
4.2	Envoi d'une réponse avec un contenu JSON	244
4.3	Envoi d'une réponse d'un type particulier	246
4.4	Modification de l'en-tête de la requête HTTP	247
4.5	Gestion du cache sur les requêtes HTTP	248
4.5.1	Utilisation d'identifiant par donnée envoyée	248
4.5.2	Expiration temporelle du cache	249
4.5.3	Expiration du cache en précisant une date	250
4.6	Redirection de page HTTP	253
4.6.1	Redirection sur une URL interne	253
4.6.2	Redirection sur le nom d'une route	253
4.6.3	Redirection sur une URL externe	254

5. Nouveautés de la version 4	254
5.1 Changement dans la gestion des filtres	254
5.1.1 Changement dans le code de Phalcon	254
5.1.2 Utilisation des filtres avec la version 4	255
5.1.3 Changement dans les types de filtres	256
5.2 Changement dans la gestion de session	258
5.2.1 Changement dans les classes de session	258
5.2.2 Création d'un service de session en version 4	260
6. Migration vers la version 4	260
6.1 Les filtres	260
6.2 Les sessions	261
6.3 Les cookies	263

Chapitre 6

Mise en place de formulaires

1. Présentation	265
2. Élaboration d'un formulaire	266
2.1 Création du répertoire contenant les formulaires	266
2.2 Contenu basique d'un formulaire	267
3. Présentation de la classe mère Element	268
3.1 Attributs communs à tous les éléments	268
3.2 Initialisation d'un élément	269
3.3 Mise en place d'une valeur par défaut	270
3.4 Création d'un libellé	270
3.5 Modification des attributs	270
4. Gestion des types de champs d'un formulaire	271
4.1 Types basiques	271
4.1.1 Texte	271
4.1.2 Mot de passe	272
4.1.3 Liste déroulante simple	272
4.1.4 Liste déroulante à valeur multiple	273

4.1.5	Case unique à choisir	275
4.1.6	Case à cocher multiple	276
4.1.7	Zone de texte	277
4.1.8	Champ caché	278
4.1.9	Fichier	278
4.1.10	Date	279
4.1.11	Numérique	280
4.1.12	Bouton de soumission de formulaire	281
4.2	Création de types personnalisés	282
4.2.1	Création d'une nouvelle classe	282
4.2.2	Ajout de paramètre à l'initialisation du champ	283
4.2.3	Création du rendu de l'élément HTML	284
4.2.4	Utilisation du nouveau champ dans un formulaire	286
5.	Gestion des validateurs	287
5.1	Fonctionnement	287
5.2	Usage des validateurs existants	288
5.2.1	Alphanumérique	289
5.2.2	Alphabétique	290
5.2.3	Intervalle de valeur	291
5.2.4	Validateur personnalisé	292
5.2.5	Confirmation de valeur	294
5.2.6	Numéro de carte de paiement	295
5.2.7	Date	296
5.2.8	Numérique	297
5.2.9	E-mail	298
5.2.10	Exclusion de valeur	299
5.2.11	Fichier	300
5.2.12	Valeur identique	303
5.2.13	Inclusion de valeur	304
5.2.14	Valide numériquement	305
5.2.15	Non nulle	306
5.2.16	Expression régulière	307
5.2.17	Nombre de caractères	308

5.2.18	Valeur unique	310
5.2.19	URL	311
6.	Gestion des formulaires	312
6.1	Ajout d'un élément au formulaire	312
6.2	Passage et intégration des champs dans la vue	313
6.2.1	Instanciation du formulaire	313
6.2.2	Passage des données à la vue	313
6.2.3	Affichage des champs dans une vue	313
6.3	Passage des valeurs dans le formulaire	315
6.3.1	Utilisation d'une entité	315
6.3.2	Utilisation d'une classe d'entité	316
6.3.3	Passage de données diverses	317
6.4	Validation du formulaire	318
6.5	Renvoi des erreurs à la vue	319
6.6	Enregistrement des données	320
6.6.1	Nouvelle entrée	320
6.6.2	Mise à jour d'une entrée	321
6.7	Affichage personnalisé des variables	322
7.	Astuces avec PHPStorm	324
7.1	Utilisation de file and code templates	324
7.1.1	Création de modèles de fichiers dans PHPStorm	324
7.1.2	Utilisation du modèle de fichiers avec PHPStorm	327
7.2	Utilisation des live templates	328
7.2.1	Les live templates disponibles dans PHPStorm	328
7.2.2	Association d'un modèle de code à un type de fichier/contexte	329
7.2.3	Utilisation d'un modèle de code	329
7.2.4	Création d'un modèle de code	330
7.3	Import des modèles	332
8.	Nouveautés de la version 4	333
8.1	Changement dans la création d'éléments personnalisés	333
8.1.1	Changement de classe	333
8.1.2	Ajout de PSR	333

- 9. Migration vers la version 4. 333
 - 9.1 Changement de nom de la classe Element 333
 - 9.2 Typage de la fonction rendue 334

Chapitre 7

Création de vues HTML

- 1. Présentation générale des vues. 335
- 2. Hiérarchisation des vues 335
- 3. Volt ou PHTML ? 337
- 4. Utilisation de PHTML 338
 - 4.1 Affichage des variables 338
 - 4.2 Utilisation générale de PHTML. 338
- 5. Utilisation de Volt 339
 - 5.1 Gestion des commentaires 339
 - 5.2 Gestion des variables 339
 - 5.2.1 Affichage de variables simples 339
 - 5.2.2 Création et initialisation des variables 340
 - 5.2.3 Expressions mathématiques 343
 - 5.2.4 Expression globale 344
 - 5.3 Utilisations de conditions 345
 - 5.3.1 Structure de condition Si (If) 345
 - 5.3.2 Conditions préconstruites par Phalcon 347
 - 5.3.3 Structure de condition ternaire 352
 - 5.3.4 Structure selon le cas (Switch case) 352
 - 5.3.5 Opérateurs de comparaison et logiques 354
 - 5.4 Gestion des tableaux et objets 355
 - 5.4.1 Parcours de tableaux 355
 - 5.4.2 Mot-clé loop. 356
 - 5.4.3 Conditions dans les tableaux 358

5.5	Fonctions de base de Volt	359
5.5.1	Fonctions pour des nombres	360
5.5.2	Fonctions pour des chaînes de caractères	360
5.5.3	Fonctions pour des tableaux	367
5.5.4	Fonctions pour tout type de variable	369
6.	Gestion des ressources	369
6.1	Intégration de ressources lien par lien	370
6.1.1	Au niveau d'une action d'une page	370
6.1.2	Au niveau d'un contrôleur	372
6.1.3	Au niveau du contrôleur principal	373
6.1.4	Intégration en PHTML	374
6.1.5	Intégration en Volt	375
6.2	Gestion des collections	375
6.2.1	Création dans le contrôleur principal	375
6.2.2	Intégration dans les vues	376
6.2.3	Minification des ressources	377
7.	Nouveautés de la version 4	380
7.1	Changement dans la gestion de fichier JavaScript et CSS	380
8.	Migration vers la version 4	380
8.1	Suppression des filtres	380

Chapitre 8

Gestion avancée des vues

1.	Création de fonctions utilisables dans les vues	383
1.1	Ajout de fonctions provenant de PHP	384
1.2	Ajout d'une fonction personnalisée	385
2.	Création de filtres utilisables dans les vues	387
2.1	Ajout de filtre provenant d'une fonction PHP	387
2.2	Ajout d'un filtre personnalisé	387

3.	Utilisation des tags HTML	389
3.1	Génération du Doctype	389
3.1.1	Écriture dans la vue	391
3.2	Génération du titre (title)	391
3.2.1	Écriture dans la vue	393
3.3	Déclaration de style CSS	394
3.3.1	Écriture dans la vue	394
3.4	Déclaration de script JavaScript	395
3.4.1	Écriture dans la vue	396
3.5	Génération d'image	397
3.5.1	Écriture dans la vue	397
3.6	Génération de lien hypertexte	399
3.6.1	Écriture dans la vue	399
3.7	Génération de formulaire	403
3.7.1	Écriture dans la vue	403
3.8	Génération de champs de texte	404
3.8.1	Écriture dans la vue	404
3.9	Génération de mots de passe	406
3.9.1	Écriture dans la vue	406
3.10	Génération de champs cachés	407
3.10.1	Écriture dans la vue	407
3.11	Génération de zones de texte	408
3.11.1	Écriture dans la vue	408
3.12	Génération de listes déroulantes statiques	409
3.12.1	Écriture dans la vue	409
3.13	Génération de listes déroulantes dynamiques	411
3.13.1	Écriture dans la vue	411
3.14	Génération d'un bouton soumission de formulaire	414
3.14.1	Écriture dans la vue	414
4.	Création de macros	415
4.1	Génération de code HTML avec des paramètres	415
4.2	Génération de code HTML avec des paramètres nommés	417
4.3	Création de macro avec des paramètres par défaut	418

4.4	Retour de données avec une macro	419
5.	Gestion de l'internationalisation	420
5.1	Installation	421
5.1.1	Pour Windows	421
5.1.2	Pour Linux	421
5.1.3	Extension PHP	422
5.1.4	Pour Docker	422
5.2	Gestion du logiciel de traduction.	423
5.2.1	Téléchargement et installation de Poedit	423
5.2.2	Ajout de l'extracteur Volt	423
5.2.3	Création du dictionnaire de traduction pour le projet .	425
5.2.4	Paramétrage du projet	426
5.2.5	Indication des fichiers sources à analyser	427
5.2.6	Ajout des mots-clés de reconnaissance de traduction .	428
5.2.7	Mise à jour et traduction de texte	429
5.3	Mise en place du service sur un projet	429
5.4	Intégration du service de traduction dans toutes les vues . . .	431
5.5	Utilisation de la fonction de traduction dans les vues.	432
5.5.1	Traduction simple	432
5.5.2	Traduction avec variable.	432
6.	Création de plans (layouts) avancés	433
6.1	Gestion des modèles	433
6.2	Inclusion de vues multiples	436
6.3	Gestion des layouts par extends	437
7.	Affichage de différents niveaux de rendu	440
7.1	Désactivation de certains niveaux de vues	441
7.2	Désactivation de la génération de vue.	443
8.	Nouveautés de la version 4	443
8.1	Changement dans les tags	443
8.1.1	L'affichage du titre HTML	443
8.1.2	La liste déroulante dynamique	443
8.2	Changement dans le module de traduction	444

8.3	Ajout de classe HTML\Helper	444
8.3.1	Création d'un lien hypertexte	446
8.3.2	Création d'un lien hypertexte brut	447
8.3.3	Création d'un bouton	448
8.3.4	Création d'une image	449
8.3.5	Création d'une zone de texte	449
8.3.6	Création de balise de fin	450
8.3.7	Création d'un formulaire	451
8.3.8	Création du corps (body)	452
8.3.9	Création d'un libellé (label)	452
8.3.10	Création d'un champ générique	453
8.3.11	Création d'un champ générique brut	454
8.3.12	L'usine à éléments HTML	454
8.3.13	Création d'un fil d'Ariane	456
9.	Migration vers la version 4	457
9.1	Modification des tags	457
9.1.1	L'affichage du titre HTML	457
9.1.2	L'affichage de liste déroulante dynamique	457
9.2	Modification du module de traduction	458

Chapitre 9

Bases de données - Les modèles Phalcon

1.	Préambule : connexion aux différents types de bases de données gérés	461
1.1	Modification de la configuration	462
1.2	Modification du service de connexion à la base de données	463
1.3	Ajout d'options complémentaires au service de base de données	464
2.	Gestion des modèles	464
2.1	Création de modèles	465
2.1.1	Avec Phalcon DevTools	465
2.1.2	Manuellement	467

2.2	Récupération simple des données	468
2.2.1	Liste des éléments d'une table.	468
2.2.2	Récupération de la première ligne	469
2.2.3	Récupération de la dernière ligne	470
2.3	Sélection de colonnes	470
2.4	Conditions	471
2.4.1	Simple.	471
2.4.2	Personnalisée	471
2.4.3	Avec typage des valeurs.	473
2.5	Limitation du nombre de lignes.	474
2.6	Saut de ligne	475
2.7	Tri des données.	475
2.8	Groupement de données	476
2.9	Récupération du nombre de lignes	476
2.10	Gestion du curseur	477
2.10.1	Accès à la position courante du curseur.	477
2.10.2	Déplacement du curseur	477
2.11	Différents formats de résultats	477
2.11.1	Changement du format de résultat après la requête.	478
2.11.2	Changement du format de résultat par requête	479
2.12	Opérations mathématiques sur les tables.	479
2.13	Insertion de lignes.	481
2.13.1	Méthode classique.	481
2.13.2	Sauvegarde par tableau	481
2.13.3	Sauvegarde par formulaire	482
2.13.4	Forçage d'insertion d'une ligne	483
2.13.5	Gestion d'erreur à la sauvegarde	484
2.14	Mise à jour d'une ligne	484
2.14.1	Méthode classique.	484
2.14.2	Forçage de la mise à jour d'une valeur	485
2.15	Suppression d'une ligne	485

2.16	Validation des données.	486
2.16.1	Ajout de validateurs	486
2.16.2	Modification des messages d'erreurs	487
2.17	Pagination des résultats	489
2.18	Mise en place de relations entre les tables	491
2.18.1	Relation 1 à 1 (1-1)	491
2.18.2	Relation 1 à plusieurs (1-N)	493
2.18.3	Relation plusieurs à 1 (N-1)	494
2.18.4	Relation plusieurs à plusieurs (N-N)	496
2.18.5	Accès à une table grâce aux relations	498
2.18.6	Utilisation d'alias pour des requêtes récurrentes	500
2.18.7	Message d'erreur sur les clés étrangères	501
2.19	Conservation des changements d'un objet du modèle (Snapshots)	503
2.19.1	Initialisation du système de conservation des changements	503
2.19.2	Les fonctions associées aux Snapshots	503
2.20	Mise en place d'un préfixe sur toutes les tables	505
2.21	Connexion à différentes bases de données	506
2.21.1	Ajout de nouvelles bases de données	506
2.21.2	Connexion d'une table à une base de données	508
2.21.3	Gestion lecture/écriture par table	508
2.22	Mise en place d'un comportement spécifique	509
2.22.1	Gestion du temps	510
2.22.2	Gestion des suppressions	513
2.23	Mise en cache des résultats d'une requête	514
2.23.1	Mise en place du cache	514
2.23.2	Mise en cache simple d'une requête	515
2.23.3	Mise en cache d'une requête avec un temps d'expiration	516
2.23.4	Mise en cache d'une requête sur un service différent	517

2.24	Gestion des événements du modèle	518
2.24.1	Liste des événements existants associés au modèle.	518
2.24.2	Intégration et utilisation des événements dans une classe.	520
3.	Nouveautés de la version 4	522
3.1	Changement dans les classes d'interactions avec la base de données	522
3.2	Changement dans l'assignation des valeurs	522
3.3	Ajout du PSR.	523
3.4	Changement dans le système de pagination	523
4.	Migration vers la version 4.	524
4.1	Modification des modes de récupération de données.	524
4.2	Modification dans l'assignation des valeurs	525
4.2.1	Ajout/modification de ligne	525
4.2.2	Ajout/modification de ligne en spécifiant les colonnes	526
4.2.3	Création d'une nouvelle ligne	526
4.3	Prise en compte du PSR	527
4.4	Modification dans le système de pagination	527
4.5	Modification dans les classes de gestion d'événements	528

Chapitre 10

Bases de données - Utilisation de PHQL

1.	PHQL - Phalcon Query Language	529
2.	Récupération de données	529
2.1	Requête simple	529
2.2	Utilisation de mots-clés réservés	530
3.	Requêtes avec paramètres	530
3.1	Empêchement de l'intégration d'un paramètre sans contrôle	530
3.2	Construction d'une requête avec des paramètres simples	531
3.3	Construction d'une requête avec des paramètres typés.	532

4.	Mise en cache des requêtes PHQL	534
5.	Constructeur de requête	535
5.1	Récupération de plusieurs lignes d'une table	535
5.2	Récupération d'une ligne	536
5.3	Limitation du nombre de résultats	536
5.4	Limitations et sauts de lignes	536
5.5	Jointure naturelle entre deux tables	537
5.6	Sélection de colonnes	538
5.7	Mise en place d'alias sur les tables	539
5.8	Jointure manuelle	541
5.9	Conditions	542
5.9.1	Condition simple	542
5.9.2	Combinaison de conditions avec l'opérateur ET	543
5.9.3	Combinaison de conditions avec l'opérateur OU	543
5.9.4	Conditionnement d'une colonne dans un intervalle de valeurs	544
5.9.5	Conditionnement d'une colonne avec une liste de valeurs inclusive	545
5.9.6	Conditionnement d'une colonne avec une liste de valeurs exclusive	545
5.10	Groupement de données	546
5.10.1	Sans condition	546
5.10.2	Avec une condition	547
5.11	Passage de paramètres dans la fonction d'exécution de requête	548
6.	Nouveautés de la version 4	549
6.1	Changement sur la sélection des colonnes du QueryBuilder	549
7.	Migration vers la version 4	549
7.1	Modification sur la sélection des colonnes de QueryBuilder	549
7.2	Modification dans le système de cache	550

Chapitre 11**Bases de données - Abstraction Layer**

1.	Manipulation directe de la base de données	551
1.1	Récupération de données - de ligne à ligne	551
1.1.1	Lancement d'une requête	551
1.1.2	Récupération du nombre de lignes retournées	552
1.1.3	Lecture ligne à ligne	552
1.2	Récupération de données - toutes les lignes	553
1.2.1	Fonctionnement basique	553
1.2.2	Les différents modes de récupération	554
1.2.3	Spécification du mode de récupération des données	555
1.2.4	Paramétrage du mode de récupération par défaut	556
1.3	Récupération de la première ligne	557
1.4	Passage de paramètres	557
1.4.1	Construction d'une simple requête	557
1.4.2	Construction d'une requête préparée	559
1.5	Ajout/Modification/Suppression des données	561
1.5.1	Par requête SQL	561
1.5.2	Avec une fonction simple	562
1.5.3	Avec un dictionnaire de données	566
1.6	Gestion des événements	568
1.6.1	Liste des événements disponibles	568
1.6.2	Mise en place d'une classe d'événements	569
1.7	Gestion des transactions	570
2.	Manipulation du schéma d'une base de données	572
2.1	Description du schéma	572
2.1.1	Liste des tables d'un schéma	572
2.1.2	Vérification de l'existence d'une table	572
2.1.3	Description des colonnes d'une table	573
2.1.4	Récupération des index d'une table	575
2.1.5	Récupération des clés étrangères d'une table	575
2.1.6	Récupération des vues d'un schéma	576
2.1.7	Vérification de l'existence d'une vue	576

- 2.2 Modification de la structure de la base de données 576
 - 2.2.1 Création de colonnes 576
 - 2.2.2 Création d'index 578
 - 2.2.3 Création de clés étrangères 579
 - 2.2.4 Création d'une table 580
 - 2.2.5 Ajout d'une colonne à une table 580
 - 2.2.6 Suppression d'une colonne 581
 - 2.2.7 Suppression d'une table 582
- 3. Migration vers la version 4 582
 - 3.1 Modification des constantes de PDO 582
 - 3.2 Modification de la classe d'événement 583
 - 3.3 Modification du système de log 583

Chapitre 12
Gestion des logs et des erreurs

- 1. Gestion des logs 585
 - 1.1 Création du service de log 585
 - 1.2 Les différents stockages de logs disponibles 586
 - 1.2.1 Log classique par fichier 586
 - 1.2.2 Log par flux pour Docker 587
 - 1.2.3 Log système 588
 - 1.3 Adaptation du formatage des logs 588
 - 1.3.1 Formatage par ligne 589
 - 1.3.2 Formatage JSON 590
 - 1.3.3 Formatage Syslog 590
 - 1.4 Gestion du niveau de logs 591
 - 1.5 Création d'un log 592
 - 1.5.1 Fonctionnement 592
 - 1.5.2 Liste des fonctions utilisables 593

2.	Gestion des erreurs	595
2.1	Surcharge du service Dispatcher	595
2.2	Gestion de la classe ExceptionPlugin.	596
2.2.1	Création de la classe ExceptionPlugin	596
2.2.2	Surcharge de l'évènement	596
2.3	Création d'une exception.	599
2.3.1	Erreur 404.	599
2.3.2	Erreur 500.	599
3.	Nouveautés de la version 4	600
3.1	Changement dans la gestion des logs	600
3.1.1	Les adaptateurs	600
3.1.2	La nouvelle de classe Logger	600
3.1.3	Le niveau de log	602
3.1.4	Le formatage	602
3.2	Changement dans la gestion des exceptions	603
3.3	Changement dans les constantes d'erreur du dispatcher	603
4.	Migration vers la version 4.	603
4.1	Modification dans la création d'un service de logs.	603
4.1.1	Modification de l'adaptateur.	603
4.1.2	Création d'un Logger.	604
4.1.3	Modification du niveau de log	605
4.1.4	Modification du formatage.	605

Chapitre 13

Gestion des événements

1.	Présentation	607
2.	Liste des événements de Phalcon.	607
3.	Mise en place.	612
3.1	Assignment d'une classe de gestion d'événements.	612
3.1.1	Création de la classe	613
3.1.2	Création d'une fonction d'évènement	614

3.1.3	Intégration du gestionnaire d'événements dans le service	615
3.2	Attribution d'une fonction anonyme par événement	616
3.3	Attribution d'une fonction anonyme pour tous les événements.	618
4.	Cas concret - Sécurisation de l'accès aux pages	620
5.	Nouveautés de la version 4	622
5.1	Changement dans les classes de gestion d'événements	622
6.	Migration vers la version 4.	623
6.1	Modification dans les classes de gestion d'événements	623

Chapitre 14

Gestion des droits d'accès (ACL)

1.	Présentation	625
1.1	Les cas traditionnels	625
1.2	Le service ACL (Access Control Lists) de Phalcon	626
1.2.1	Fonctionnement de base	626
1.2.2	Technologie	626
2.	Mise en place d'une stratégie de gestion de droits	627
2.1	Création de l'objet de gestion des droits	627
2.1.1	Création de la fonction dans SecurityPlugin	627
2.1.2	Instanciation de la classe ACL	628
2.1.3	Définition de la stratégie par défaut.	628
2.2	Gestion des rôles	629
2.2.1	Création d'un rôle	629
2.2.2	Ajout du rôle à la stratégie de droits	629
2.2.3	Structure de rôle	629
2.3	Gestion des ressources	630
2.3.1	Création d'une ressource	630
2.3.2	Ajout de la ressource à la stratégie de droits	631
2.3.3	Structure de ressource	631

2.4	Autorisation/blocage d'élément(s)	632
2.5	Structure d'autorisation	634
2.6	Optimisation de stockage	635
2.6.1	Session	635
2.6.2	Sérialisation	635
3.	Accès aux informations	636
3.1	Existence d'un rôle	636
3.2	Existence d'une ressource	637
3.3	Vérification de l'accès	637
3.3.1	Usage	637
3.3.2	Intégration	637
3.3.3	Fonction de vérification avancée	639
3.4	Héritage de rôle	641
3.4.1	En ajoutant un nouveau rôle	641
3.4.2	En liant deux rôles déjà créés	642
3.5	Evénements	643
3.5.1	Avant la vérification de l'accès	643
3.5.2	Après vérification de l'accès	644
3.5.3	Logs après une opération de contrôle d'accès	644
4.	Nouveautés de la version 4	645
4.1	Changement dans les constantes	645
5.	Migration vers la version 4	646
5.1	Modification des constantes	646
5.2	Modification de nom de la classe Resource	646
5.2.1	Définition des objets	646
5.2.2	Modification des fonctions	647

Chapitre 15

Gestion de la sécurité

1. Cryptage/décryptage de données	649
1.1 Installation de l'extension	649
1.2 Usage de la cryptographie	650
1.2.1 Avec les valeurs par défaut	650
1.2.2 Cryptage/décryptage en base64	652
1.2.3 En choisissant un cipher	652
1.2.4 Spécification de la clé et du cipher dans les services ...	653
1.2.5 Signature d'une donnée cryptée	654
2. Hachage de données	656
2.1 Hachage simple	657
2.2 Vérification d'une donnée hachée	658
2.3 Cas pratique	658
2.4 Accroissement de la sécurité du hachage	659
3. Protection CRSF (Cross-Site Request Forgery)	660
3.1 Présentation de l'attaque	660
3.2 Principe du CRSF	660
3.3 Mise en place	661
4. Les valeurs aléatoires	662
4.1 Génération d'un nombre aléatoire	662
4.2 Génération d'un UUID	663
4.3 Génération d'une valeur binaire	663
4.4 Génération d'une valeur hexadécimale	663
4.5 Génération d'une valeur en base 58	664
4.6 Génération d'une valeur en base 62	664
4.7 Génération d'une valeur en base 64	664
4.8 Génération d'une valeur en base 64 pour une URL	665

Chapitre 16

Gestion du cache

1. Présentation	667
2. Configuration	667
2.1 Les adaptateurs Front-End proposés par Phalcon	670
2.2 Les adaptateurs Back-End proposés par Phalcon	671
3. Usage	672
3.1 Utilisation basique du cache	672
3.2 Utilisation du cache avec un temps d'expiration défini	674
3.3 Liste des clés de cache disponibles	675
3.4 Recherche de clés de cache	676
3.5 Vérification de l'existence d'une clé en cache	676
3.6 Suppression du cache d'une clé en particulier	677
4. Nouveautés de la version 4	677
4.1 Mise en conformité avec la PSR-16	677
4.2 Changement dans les classes	678
4.2.1 Classes ajoutées	678
4.2.2 Classes renommées	678
4.2.3 Classes supprimées	679
4.3 Changement dans la création d'un service de cache	680
4.4 Changement dans les fonctions	681
4.4.1 Fonction modifiée	681
4.4.2 Fonction supprimée	681
5. Migration vers la version 4	682
5.1 Modification de classes	682
5.2 Modification de la création du service	682
5.3 Modification dans les fonctions	685
5.3.1 La fonction queryKeys	685
5.3.2 La fonction save	685
5.3.3 La fonction get	686
5.3.4 La fonction exists	686

Chapitre 17

Manipulation d'image

1. Présentation	687
2. Installation	687
3. Utilisation	688
3.1 Chargement d'une image	688
3.2 Accès aux données de l'image	689
3.2.1 Le fichier	689
3.2.2 Le MIME (type)	689
3.2.3 Le chemin de l'image	689
3.2.4 Le type détecté par la librairie	689
3.2.5 La largeur	690
3.2.6 La hauteur	690
3.3 Sauvegarde d'une image	690
3.4 Conversion d'image	691
3.5 Réduction de qualité	691
3.6 Redimensionnement	692
3.6.1 En largeur	692
3.6.2 En hauteur	692
3.6.3 En hauteur/largeur avec la proportion la plus petite	693
3.6.4 En hauteur/largeur avec la proportion la plus grande	694
3.6.5 En hauteur/largeur sans proportion	694
3.7 Rognage d'image	695
3.8 Rotation d'image	696
3.9 Retournement d'image	697
3.9.1 Vertical	697
3.9.2 Horizontal	698
3.10 Ajout d'un filigrane	698
3.11 Floutage d'une image	699
3.12 Ajout d'un reflet sur l'image	700
3.13 Pixelisation d'une image	701
3.14 Ajout de texte sur une image	701

3.15	Création d'une image	702
4.	Nouveautés de la version 4	702
4.1	Ajout d'une classe de constantes	702
5.	Migration vers la version 4.	703
5.1	Modification de la classe des constantes.	703

Chapitre 18

Amélioration des performances

1.	Ajout du profileur.	705
1.1	SQL	705
1.2	PHP	709
1.2.1	Mise en place d'une solution de profilage	709
1.2.2	Analyse du fichier avec PHPStorm	710
2.	Lancement des tâches en arrière-plan	712
2.1	Présentation	712
2.1.1	Les tâches longues	712
2.1.2	Les tâches planifiées	713
2.2	Création d'un projet CLI (Command Line)	713
2.2.1	Présentation de la structure d'un projet CLI	714
2.2.2	Présentation d'une classe de tâches	714
2.2.3	Lancement de la tâche par défaut.	715
2.2.4	Lancement d'une tâche spécifique	716
2.2.5	Passage de paramètres.	717
2.3	Fusion d'un projet simple avec un projet CLI.	717
3.	Nouveautés de la version 4	719
3.1	Changement dans le mode Cli de Phalcon	719
3.1.1	Passage de paramètre.	719
4.	Migration vers la version 4.	719
4.1	Modification dans le mode Cli de Phalcon	719
4.1.1	Modification du passage de paramètre	719

Chapitre 19

Architecture de projet

1. Présentation	721
2. Gestion d'un projet Micro	722
2.1 Présentation	722
2.2 Création d'un projet Micro	722
2.3 Présentation de la structure d'un projet Micro	723
2.4 Gestion des routes	724
2.4.1 Définition d'une route	724
2.4.2 Passage de la variable \$app	725
2.4.3 Passage de paramètres sans contrôle	726
2.4.4 Passage de paramètres avec contrôle	726
2.4.5 Les différentes méthodes HTTP	727
2.5 Gestion des contrôleurs	728
2.5.1 Création d'un contrôleur	728
2.5.2 Intégration de la collection dans l'application	730
2.5.3 Chargement optimisé d'une collection	732
2.5.4 Redirection	733
2.6 Gestion des vues	734
2.6.1 Spécification de la vue à utiliser	735
2.6.2 Passage de paramètres	736
2.6.3 Génération d'une URL	737
2.7 Gestion des réponses	737
2.7.1 Réponse JSON	737
2.7.2 Réponse typée	738
2.8 Gestion des événements avec une fonction anonyme	740
2.9 Gestion des événements avec un Middleware	742
2.9.1 Présentation	742
2.9.2 Mise en place de la structure	742
2.9.3 Création d'un Middleware simple	742
2.9.4 Création d'un Middleware avec surcharge d'événement existant	744
2.9.5 Intégration d'un Middleware dans l'application	745

3.	Gestion d'une API Rest avec une architecture Micro	746
3.1	Gestion des événements	747
3.1.1	Création de SecurityMiddleware	747
3.1.2	Contrôler l'accès à l'API	747
3.1.3	CORS - Cross-Origin resource sharing	748
3.1.4	Vérification des données reçues par l'API	749
3.1.5	Page introuvable	749
3.1.6	Architecture complète de SecurityMiddleware	750
3.1.7	Intégration de SecurityMiddleware dans l'application	751
3.2	Gestion des exceptions	751
3.2.1	Présentation	751
3.2.2	Mise en place	752
3.2.3	Les exceptions générales de Phalcon	753
3.3	Gestion automatique des réponses	754
3.3.1	Création de ReponseMiddleware	755
3.3.2	Intégration de ReponseMiddleware	756
3.4	Intégration des routes d'une API	756
3.4.1	Récupération de données	756
3.4.2	Ajout d'un nouvel élément en base de données	757
3.4.3	Modification d'un élément en base de données	757
3.4.4	Suppression d'un élément en base de données	758
3.5	Test de l'API	759
3.5.1	Création de requête	759
3.5.2	Test avec PHPStorm	760
3.6	Documentation/annuaire des routes de l'API	762
3.6.1	Installation d'ApiDoc	762
3.6.2	Commentaire dans le code PHP	763
3.6.3	Fichier de configuration pour ApiDoc	764
3.6.4	Génération de la documentation	764
3.6.5	Affichage de la documentation depuis l'index de l'API	765
3.6.6	Démonstration de la documentation	765

- 4. Gestion d'un projet API Rest avec une architecture complète 766
 - 4.1 Gestion des événements. 766
 - 4.1.1 Création de SecurityPlugin 767
 - 4.1.2 Création de ReponsePlugin. 769
 - 4.1.3 Intégration des événements dans le dispatcher 770
 - 4.2 Gestion des routes 770
 - 4.3 Récupération des données issues de la requête. 771
 - 4.3.1 Paramètre d'URL 771
 - 4.3.2 Données de la requête 772
 - 4.4 Documentation. 772
- 5. Nouveautés de la version 4 773
 - 5.1 Changement dans les projets Micro 773
 - 5.1.1 Changement dans la fonction handle 773
- 6. Migration vers la version 4. 774
 - 6.1 Modification dans les projets Micro 774
 - 6.1.1 Ajout d'un paramètre dans la fonction handle 774

Chapitre 20
Projet de cours en ligne

- 1. Présentation du projet 775
 - 1.1 Description du projet 775
 - 1.2 Architectures possibles. 775
 - 1.2.1 Premier cas : API et interface utilisateurs distincts 775
 - 1.2.2 Deuxième cas : application en un seul bloc 776
- 2. Caractéristique du projet 777
- 3. Organisation du projet 777
 - 3.1 Gestion des répertoires. 777
 - 3.2 Structure de la base de données. 778
 - 3.2.1 Description des colonnes des tables 778
 - 3.2.2 Script SQL de création de table
 et d'initialisation des données. 778

3.3	Structure Docker	780
3.3.1	Dockerfile	780
3.3.2	Docker-compose	780
4.	Préparation de l'API - Une route de santé	782
4.1	Suppression des vues	783
4.2	Présentation	783
4.3	Résultat attendu	783
4.4	Solution possible	784
4.4.1	Création de la route	784
4.4.2	Création de l'action santé	784
4.4.3	Création automatique d'une réponse JSON	785
4.5	Test de la route	786
5.	Le Front-End interroge l'API	788
5.1	Présentation de la librairie de requête	788
5.1.1	Téléchargement de Guzzle dans le projet	788
5.1.2	Intégration du vendor dans le projet	789
5.2	Lancement d'une requête à l'API	789
5.2.1	Création d'un Client HTTP	789
5.2.2	Lancement d'une requête API	790
5.2.3	Récupération des données de la réponse HTTP	792
6.	Gestion des appels API	793
6.1	Création de la classe ClientApi	793
6.2	Création du constructeur de ClientApi	794
6.3	Création d'une méthode d'envoi de requête HTTP	796
6.3.1	Méthodologie	796
6.3.2	Mise en place	796
6.4	Utilisation du ClientApi	798
7.	Un service API	798
7.1	Création d'un service API	799
7.2	Utilisation du nouveau service api	799

7.3	Intégration de la configuration Phalcon	800
7.3.1	Modification du fichier de configuration.	800
7.3.2	Modification du fichier de service	800
8.	Connexion d'un utilisateur au service web.	801
8.1	Interaction simple.	801
8.1.1	Présentation du processus.	801
8.1.2	Connexion à la base de données côté API	802
8.1.3	Création de l'espace de noms pour les modèles de l'API.	802
8.1.4	Création des modèles Utilisateurs et Cours dans l'API	803
8.1.5	Création d'une route API de connexion	804
8.1.6	Création d'une fonction API pour une connexion simple.	804
8.1.7	Envoi d'une requête de connexion du Web à l'API	805
8.2	Connexion avec un formulaire	806
8.2.1	Présentation du processus.	807
8.2.2	Création et intégration du répertoire forms	807
8.2.3	Création du formulaire web	808
8.2.4	Intégration simple du formulaire dans le contrôleur	810
8.2.5	Intégration des champs du formulaire dans une vue	811
8.2.6	Validation du formulaire.	812
8.2.7	Utilisation des identifiants de connexion	814
8.2.8	Stockage en session et redirection de l'enseignant connecté	815
8.3	Protection des échanges entre l'utilisateur connecté et l'API.	817
8.3.1	Présentation du processus.	817
8.3.2	Création de la classe SecurityPlugin dans l'API	818
8.3.3	Importation d'une librairie de gestion de jetons dans l'API	819
8.3.4	Création d'un générateur de jetons utilisateurs dans l'API	819
8.3.5	Génération d'un jeton API à la connexion d'un utilisateur	820

8.3.6	Stockage du jeton en session du Web	821
8.3.7	Envoi automatique du jeton par le Web	822
8.3.8	Récupération des données présentes dans le jeton de l'API	824
8.3.9	Création d'une route API pour récupérer les cours d'un enseignant	824
8.3.10	Récupération des cours d'un enseignant connecté avec le Web	826
9.	Simplification de la récupération de l'utilisateur connecté	827
9.1	Présentation du processus	827
9.2	Création du service API	828
9.3	Utilisation du nouveau service	828
10.	Une fondation solide	829
	Index	833



Chapitre 4

Services de Phalcon

1. Injection de dépendances (DI)

Le DI est un élément très important de la structure de ce framework. L'objectif principal de l'injecteur de dépendances : avoir un accès à différents services définis. Le principe consiste à définir des services (interaction avec une base de données, système de traduction, système de cache, etc.) qui peuvent être utilisés, peu importe où l'on se trouve dans l'application.

Si l'injecteur de dépendances n'existait pas, il faudrait rouvrir et refermer une connexion à une base de données chaque fois que l'on souhaite interagir avec elle, ce qui se traduirait par de nombreux copier-coller et une maintenance du code plus complexe.

Le système d'injection de dépendances propose deux modes :

- simple : appel complet du code ;
- partagé : fonctionne comme un Singleton.

1.1 Une dépendance simple

1.1.1 Explication

Une dépendance simple permet d'initialiser et de préparer un objet. Cette dépendance renverra toujours un nouvel objet. L'objectif principal, c'est d'avoir un objet avec une durée de vie limitée.

Présentation d'un cas concret

Dans le cadre d'un projet, vous devez créer différents PDF avec la même mise en forme (logo, en-tête, pied de page, police...). Afin d'éviter la duplication de code dans toute l'application, il est recommandé de créer une dépendance qui initialise un objet avec ces paramètres de style.

Si la dépendance est partagée

- L'utilisateur demande un export PDF de ses informations personnelles.
- Le développeur fait appel au service de création de PDF ; un objet préparé est envoyé.
- Le développeur ajoute les données personnelles dans le PDF et renvoie le PDF complet à l'utilisateur.
- L'utilisateur demande l'export PDF de ces dossiers.
- Le développeur fait de nouveau appel au service de création de PDF et obtient le même objet que précédemment.
- Le développeur ajoute les données relatives aux dossiers dans le PDF et renvoie le PDF complet à l'utilisateur.
- L'utilisateur obtient un PDF contenant ses données personnelles, ainsi que celles relatives à ses dossiers.

Si la dépendance n'est pas partagée

- L'utilisateur demande un export PDF de ses informations personnelles.
- Le développeur fait appel au service de création de PDF ; un objet préparé est envoyé.

- Le développeur ajoute les données personnelles dans le PDF et renvoie le PDF complet à l'utilisateur.
- L'utilisateur demande l'export PDF de ses dossiers.
- Le développeur fait de nouveau appel au service de création de PDF et obtient un objet tout neuf.
- Le développeur ajoute les données relatives aux dossiers dans le PDF et renvoie le PDF complet à l'utilisateur.
- L'utilisateur obtient un PDF contenant uniquement les informations relatives à son dossier.

1.1.2 Mise en œuvre d'un service avec une simple instanciation

Les dépendances sont définies dans le fichier `app/config/services.php`.

Syntaxe

```
■ $di->set('<nom du service>', new <classe>());
```

Exemple d'un service de PDF

```
■ $di->set('fpdf', new Fpdf());
```

Cet exemple permet de définir un service de création de PDF. Pour appeler le service depuis un contrôleur, par exemple, il suffit de procéder de la manière suivante.

Syntaxe de récupération d'un service

```
■ $this->di->get('<nom du service>');
```

Exemple avec le service de PDF

```
■ $oPDF = $this->di->get('fpdf');
```

Un nouvel objet *Fpdf* est renvoyé dans la variable `$oPDF`.

1.1.3 Mise en œuvre d'un service avec une fonction anonyme

Fonction simple

Syntaxe

```
$di->set('<nom du service>',function ()
{
    // code
    return <objet de service>
}
);
```

Exemple de création d'un service de PDF avec une fonction anonyme

```
$di->set('fpdf', function () {
    $oPdf = new Fpdf();

    $oPdf->AddPage();
    $oPdf->Image('logo_entete.png',8,5,25);

    return $oPdf;
}
);
```

Cet exemple permet d'initialiser un objet Fpdf avec un logo en en-tête à chaque fois que le service est appelé.

Passage de variable

Pour certains types de services, il est utile de passer des paramètres de données.

Syntaxe

```
$di->set('<nom du service>',function () use (<liste de variables>)
{
    // code
    return <objet de service>
}
);
```

Exemple de connexion à une base de données avec l'objet de configuration

```
$config = new \Phalcon\Config
(
    [
        'adapter'      => 'Postgresql',
        'host'         => '192.168.99.100',
        'username'     => 'postgres',
        ...
    ]
);

$di->set('db', function () use ($config) {
    $class = 'Phalcon\Db\Adapter\Pdo\' . $config->database->adapter;
    $params = [
        'host'        => $config->database->host,
        'username'    => $config->database->username
        ...
    ];

    $connection = new $class($params);

    return $connection;
});
```

Passage de données avec le DI

Il est aussi possible d'utiliser les données à l'intérieur du DI pour créer un service.

Typiquement, pour une connexion à une base de données, il est plus pratique d'utiliser l'objet de configuration. Cet objet contient traditionnellement les informations de connexion des différents services de l'application web.

Syntaxe

```
$di->set('<nom du service>', function () {
    $oService = $this->get('<nom du service>');

    // Ou

    $oService = $this->getNomService();

    ...
}
);
```

Exemple complet de connexion à une base de données

app/config/config.php

```
return new \Phalcon\Config([
    'database' => [
        'adapter' => 'Postgresql',
        'host'     => '192.168.99.100',
        ...
    ]
]);
```

app/config/services.php

```
$di->set('config', function () {
    return include APP_PATH . "/config/config.php";
});

$di->set('db', function () {
    $config = $this->getConfig();
    // ou $config = $this->get('config');

    $class = 'Phalcon\Db\Adapter\Pdo\\' . $config->database->adapter;
    $params = [
        'host'     => $config->database->host,
        'username' => $config->database->username,
        ...
    ];

    $connection = new $class($params);

    return $connection;
});
```

Dans cet exemple, un service **config** renvoie les informations présentes dans le fichier **app/config/config.php**. Ce service est ensuite utilisé pour créer la connexion à la base de données.