

Avant-propos

1. Préambule	21
2. Introduction	24
3. Organisation du livre	26
4. Public visé	26
5. Pourquoi Spring	27
6. Prérequis pour aborder Spring et Java EE	27
7. Objectifs du livre	29

Éléments constitutifs du framework

1. Introduction	31
2. Historique	31
3. Utilité de Spring dans un projet	33
4. Vue d'ensemble et sujets abordés dans l'ouvrage	36

4.1 Les modules Spring	36
4.2 Sujets abordés	38
4.3 Version des briques Spring utilisées dans l'ouvrage	39
4.4 Version des ressources externes	39
4.5 Gestion des dépendances Maven	40
4.6 Utilisation d'un BOM (Bill Of Materials) Maven	41
5. Points clés	42
Spring et les design patterns	
1. Introduction	43
2. Le singleton en Java	45
2.1 Préoccupation	45
2.2 Solution Spring	47
2.3 Exemple	47
3. L'inversion de contrôle	49
4. Façade	51
5. Fabrique	52

6. Active record	52
7. Proxy	53
8. Modèle Vue Contrôleur (MVC)	55
9. Le contrôleur de partie frontale (Front Controller)	56
10. Les assistants de vue (View Helper)	57
11. Les templates	58
12. Stratégie	59
13. Points clés	59
Rappels sur les éléments externes à Spring	
1. Codage equals et hashCode	61
1.1 Description de la problématique	64
1.2 Mise en œuvre	65
2. Projet Lombok	66
3. Les systèmes de log	

	69
4. Bases de données H2	71
4.1 Description de la problématique	71
4.2 Mise en œuvre	71
4.2.1 Installation	71
4.2.2 Configuration du POM	72
4.2.3 Connexion à la base de données en utilisant JDBC	72
4.2.4 Utiliser un listener de servlet pour démarrer et arrêter la base	73
5. Projets Maven	74
5.1 Description de la problématique	74
5.2 Mise en œuvre	75
5.3 Installation de Maven sous Windows	76
5.4 Utilisation d'un archétype Maven	76
5.5 Contenu du fichier pom.xml dans les cas simples utilisés dans ce livre	77
6. Rappels Java 8 et 9	79
6.1 Nouveauté Java 8+	79
6.2 Le langage de programmation	80
6.3 Collections	81
6.4 Sécurité	81

6.5 Les dates	81
6.6 Concurrency	81
6.7 Optional	82
6.8 Parallel arrays	83
6.9 Date et Time	83
6.10 Lambda	84
6.11 L'API Stream	85
7. Points clés	86
Le conteneur Spring	
1. Introduction	87
2. Les contextes de Spring	88
3. Configuration des beans	89
3.1 Fichier XML	90
3.2 Annotations pour configurer l'application	91
3.3 Configurer les beans applicatifs implicitement	92
3.4 Configuration par lambdas	93
4. Utilisation des beans : injection par setters et constructeurs	

	94
4.1 Mapping via le constructeur en XML	95
4.2 Comparaison des méthodes d'instanciation	96
5. Contrôle du cycle de vie : construction et destruction	97
6. Exemple illustrant les mappings standards	98
6.1 Le projet Maven	98
6.2 Fichier de configuration de Spring	98
6.3 Dépendance Spring Core et système de log	99
6.4 Dépendance vers les bibliothèques de tests unitaires	100
6.5 Exemple illustrant l'utilisation des logs	101
6.6 Fichier de configuration spécifique pour les tests	101
7. Points clés	103
Configuration avancée	
1. Introduction	105
2. Fichier XML et fichiers properties	105
3. Les fichiers de ressources	106

3.1 Les fichiers de ressources standards	106
3.2 Les fichiers de ressources chargés via le contexte	108
3.3 Les fichiers de ressources chargés via un service ResourceLoaderAware	109
3.3.1 Le programme principal	109
3.3.2 Le service ResourceLoaderAware	110
4. Convertisseurs et formateurs	110
4.1 Built-in converters	111
4.1.1 Pour les types standards	111
4.1.2 Pour les tableaux et les listes	112
4.1.3 Pour les énumérations	112
4.1.4 Pour les objets en général	113
4.2 Convertir un Array en Collection et String	114
4.3 Converter Factory	115
4.3.1 Classe StringToUtilisateurConverterFactory	115
4.3.2 Classe UtilisateurToStringConverterFactory	116
4.3.3 Classe LesConversions5	116
4.4 Les convertisseurs et les formateurs par défaut	117
4.4.1 Classe LesConversions6	117
4.5 Les formateurs personnalisés	117
4.5.1 Classe CarteDeCredit	118

4.5.2 Classe CarteDeCreditParser	118
4.5.3 Classe CarteDeCreditPrinter	119
4.5.4 Classe CarteDeCreditFormatter	119
4.5.5 Classe LesConversions7	120
5. Les BeanWrappers, le binding et les validateurs	121
5.1 Classe LesBeanWrappers	121
5.2 Classe UtilisateurValidator	122
6. Points clés	122
 Programmation orientée aspect avec Spring	
1. Introduction	123
2. Les concepts d'AOP	124
3. Limites de Spring AOP et utilisation d'AspectJ	125
4. Le support @AspectJ dans Spring	126
4.1 Activation du support	126
4.2 Activation de @AspectJ avec configuration XML	126
4.3 Déclaration d'un aspect	127

4.4 Déclaration d'un pointcut	128
4.5 Déclaration des greffons simples	130
4.6 Les types génériques	132
4.7 Détermination de noms d'arguments	133
4.8 Ordre d'appel des greffons interceptant le point de jonction	133
4.9 Injection	134
4.10 Le mécanisme des proxies	134
4.11 La face cachée des proxies en AOP	135
5. Les API bas niveau pour Spring AOP	138
5.1 L'interface PointCut	138
5.2 L'interface ClassFilter	138
5.3 L'interface MethodMatcher	139
6. Points clés	139
Les tests et Spring	
1. Introduction	141
2. Les Mock objects	142
2.1 Mocks spécialisés pour "environnement"	143

2.2 Support	143
2.2.1 Utilités générales	143
2.2.2 Spring MVC	144
2.3 Tests d'intégration	145
2.3.1 Vue d'ensemble	145
2.3.2 Mise en cache du contexte de test	145
2.3.3 Tests back et front	146
2.4 Annotations	147
2.4.1 @ContextConfiguration	147
2.4.2 @WebAppConfiguration	147
2.4.3 @ContextHierarchy	148
2.4.4 @ActiveProfiles	149
2.4.5 @TestPropertySource	149
2.4.6 @DirtiesContext	150
2.4.7 Interface TestExecutionListener	150
2.4.8 @TransactionConfiguration	151
2.4.9 @Transactional	152
2.4.10 @Rollback	152
2.4.11 @BeforeTransaction	153
2.4.12 @AfterTransaction	153
2.4.13 @Sql, @SqlConfig et @SqlGroup	153

2.5 Annotations standards	154
3. Les frameworks de test JUnit et testNG	155
3.1 Utilisation avec JUnit	155
3.1.1 Spring JUnit Runner	155
3.1.2 @IfProfileValue	155
3.1.3 @ProfileValueSourceConfiguration	156
3.1.4 @Timed	156
3.1.5 @Repeat	156
3.1.6 Méta-annotations de support pour les tests	157
3.2 Les arcanes du framework Spring TestContext	159
3.2.1 Classes et interfaces du framework de tests	159
3.2.2 Configuration du TestExecutionListener par les annotations	160
3.2.3 Les TestExecutionListeners	161
3.2.4 Mélange XML et classes annotées	166
3.3 Configuration des contextes de tests	166
3.3.1 Configuration de contexte avec des initialiseurs de contexte	166
3.3.2 L'héritage dans la configuration des contextes	167
3.3.3 Prise en compte des profils d'environnement	168
3.3.4 Configuration de contexte avec des fichiers de propriétés de test	173
3.3.5 Déclarer un fichier de propriétés pour les tests	173

3.3.6	Détection du fichier de propriétés par défaut	174
3.4	Hierarchie des contextes de tests	175
3.4.1	Héritage et surcharge des propriétés de test	175
3.4.2	Chargement d'un WebApplicationContext	175
3.4.3	Conventions	176
3.4.4	La sémantique de ressources explicites	177
3.4.5	Injecter des objets mockés	178
3.4.6	Cachabilité du contexte de test	178
3.4.7	Les hiérarchies des contextes	180
3.4.8	L'injection de dépendance dans les tests	183
3.5	Le scope session lors d'un test de requête	185
3.5.1	Le scope session lors d'un test de requête	185
3.5.2	Test de beans de scope request	186
3.5.3	Configuration d'un bean de scope session	186
3.6	Les transactions	187
3.6.1	La gestion des transactions	187
3.6.2	Les transactions managées par le test	188
3.6.3	Activation et désactivation des transactions	188
3.6.4	Comportement du commit et du rollback d'une transaction	190
3.6.5	Exécution de code en dehors d'une transaction	191
3.6.6	Configuration d'un gestionnaire de transactions	191

3.6.7 Démonstration de toutes les annotations liées à la transaction	192
3.7 Les scripts SQL	193
3.7.1 Exécuter des scripts SQL	193
3.7.2 La sémantique du path des ressources	196
3.7.3 La détection de script par défaut	196
3.7.4 Déclaration de plusieurs @Sql	197
3.7.5 Phases d'exécution pour les scripts	198
3.7.6 Script de configuration avec SqlConfig	198
3.7.7 La gestion des transactions pour @Sql	199
3.8 Classes de support TestContext	200
3.8.1 Classes de support de JUnit	200
3.8.2 Spring JUnit Runner	201
3.8.3 Classes de support TestNG	202
3.9 Framework Spring MVC Test	203
3.9.1 Projet autonome	203
3.9.2 Tests côté serveur	204
3.9.3 Définir les attendus	209
3.9.4 Ajouts de filtres	211
3.9.5 Tests REST côté client	212
3.10 Autres ressources	212

4. Points clés

	213
Partie back de l'application	
1. Description de la problématique	215
2. Mise en œuvre	217
2.1 Configuration d'une entité de la couche domaine	217
2.2 Accès à l'objet du domaine	219
2.3 Requêtes typées	221
2.4 Cache de niveaux 1 et 2	221
2.5 Bean Validation (JSR-303)	222
2.6 L'API Criteria	224
2.7 Accès à la base de données	225
2.8 Le fichier persistence.xml	225
2.9 Tests JPA	226
3. Points clés	228
Spring dans un contexte web JSP	
1. Spring MVC	229

1.1 Fonctionnement global	230
1.1.1 Configuration simple XML	230
1.1.2 Configuration par annotation	234
1.2 Éléments complexes du contrôleur	235
1.2.1 Fonctionnement global du contrôleur	235
1.2.2 Exemple de la classe Simple	236
1.2.3 Simple revisité	237
1.2.4 Par le path	237
1.2.5 Par un motif sur le path	238
1.2.6 Par le path et une méthode	238
1.2.7 Mappé par path + méthode + présence de paramètres de query	239
1.2.8 Mappé par path + méthode + présence d'un header	240
1.2.9 Par l'absence d'un header	240
1.2.10 Par consommation	241
1.2.11 Par production via Accept=application/json	241
1.2.12 Par production via Accept=application/xml	242
1.2.13 Paramètres de query	243
1.2.14 Groupes de paramètres de query	244
1.2.15 Variable du path	244
1.2.16 Variable Matrix	245
1.2.17 Variables Matrix multiples	246

1.2.18 Header	246
1.2.19 Corps de la requête	247
1.2.20 En-tête et corps de la requête	248
1.2.21 Arguments dans la requête	248
1.2.22 Reader sur la requête	249
1.2.23 InputStream de la requête	250
1.2.24 Arguments de la réponse	251
1.2.25 Writer de la réponse	251
1.2.26 OutputStream de la réponse	252
1.2.27 Session	252
1.2.28 Handler personnalisé	253
1.2.29 Utiliser l'annotation @ResponseBody seule	254
1.2.30 Utiliser l'annotation @ResponseBody avec UTF-8 requis	255
1.2.31 Utiliser l'annotation @ResponseBody avec UTF-8 produit	256
1.2.32 Réponse avec un statut personnalisé	256
1.2.33 Réponse avec un en-tête personnalisé	257
1.2.34 Obtention d'un cookie	258
1.2.35 Lire une chaîne de caractères	258
1.2.36 Écrire une chaîne de caractères	259
1.2.37 Lire les données encodées dans l'URL	259
1.2.38 Retourner une MultiValueMap	260

1.2.39 Lire une structure XML	261
1.2.40 Écrire dans une structure XML via Accept=application/xml	261
1.2.41 Lire une structure JSON	262
1.2.42 Écrire dans une structure JSON via Accept=application/json	263
1.2.43 Lire un Atom	264
1.2.44 Écrire un Atom	265
1.2.45 Lire un flux RSS	265
1.2.46 Écrire un flux RSS	266
1.2.47 HTML généré par un template JSP	267
1.2.48 Mapping à partir d'un modèle	267
1.2.49 Utiliser des variables dans un template de vue	268
1.2.50 Data binding avec des variables d'URI	269
1.2.51 Types primitifs	269
1.2.52 Date	272
1.2.53 Conversion des collections	273
1.2.54 Utiliser des collections formatées	273
1.2.55 Utiliser des objets personnalisés	274
1.2.56 Utiliser un convertisseur personnalisé	274
1.2.57 Validation	278
1.2.58 @ExceptionHandler dans un contrôleur	279
1.2.59 @ExceptionHandler global	280

1.2.60	Templates de String pour les URI	280
1.2.61	UriComponentsBuilder	281
1.2.62	GET /async/callable/response-body	282
2.	Client REST	282
2.1	Utilisation du RestTemplate	282
2.2	Le bean de domaine avec l'annotation REST pour les champs manquants	282
3.	Spring Security	283
3.1	Introduction à Spring Security	283
3.2	Spring Security dans un environnement web	285
3.2.1	Authentification par Spring	285
3.2.2	Authentification par page de login personnalisée	286
3.2.3	Authentification par base de données	286
4.	Points clés	288
Intégration JSF2		
1.	Introduction	289
2.	Mojarra	290

3. Architecture	291
4. Cycle de vie	293
4.1 Requête	294
4.2 Restore View ou Reconstruct Component Tree	294
4.3 Apply Request Value	294
4.4 Perform Validation	294
4.5 Synchronize Model ou Update Model Values	295
4.6 Invoke Application Logic	295
4.7 Render Response	295
4.8 Réponse	295
4.9 Fichier web.xml	296
4.10 Dépendances	296
4.11 Fichier faces-config.xml	297
4.12 Bean managé sans Spring	298
4.13 Exemple de vue JSF	299
4.14 Aperçu d'un composant JSF	300
5. Intégration Spring	300
5.1 Architecture	301
5.2 Dépendances	302
5.3 Fichier web.xml	

5.4 Fichier faces-config.xml	302
5.5 Couches basses (back)	303
6. Points clés	304
Application Spring Angular	
1. Introduction	305
2. La partie backend	306
2.1 Génération d'un backend	306
3. La partie frontend	309
3.1 Angular CLI	310
3.2 Création du projet initial	310
3.3 Démarrage de l'application	311
3.4 Créer un service Compte	311
4. Points clés	314

Spring-HATEOAS

1. Introduction	315
1.1 Exemple de liens hypermédias codés manuellement	317
1.2 Aller plus loin avec l'hypermédia	320
1.3 Autoconfiguration par annotation	321
1.4 Fournisseurs de relations	321
1.5 Fournisseur d'URI compacte	322
1.6 Support du côté client	323
1.7 Découverte de liens coté client	323
1.8 Utilisation du @RepositoryRestResource	324

2. Points clés	334
-----------------------	------------

Documentation Spring REST Docs

1. Introduction	335
2. Exemples de génération de la documentation	336
2.1 Dépendance sur la bibliothèque de tests	337
2.2 Dépendance sur les plug-ins Maven	337
2.3 Les extraits (snippets)	339
3. Exemple JUnit 5	339

4. Requête et réponse	341
5. Réponse avec un JSON imbriqué	342
5.1 Les paramètres de requête	343
5.2 Les paramètres inclus dans le path	343
5.3 Les Request parts	343
5.4 Les Request parts payloads	344
5.5 Les champs	344
5.6 Liens hypermédias dans la réponse	344
5.7 Les en-têtes	345
6. Personnalisation de la documentation	346
6.1 Utilisation de @AutoConfigureRestDocs	346
7. Couplage Swagger 2	346
7.1 Utilisation Springfox	346
7.2 En dehors de Spring Boot	347
8. Utilisation avec Spring Data Rest	349
9. Récapitulatif sur la documentation générée	349
10. Points clés	

350

Spring Boot

1. Introduction

351

2. Configuration des exemples

352

2.1 Configuration Maven pour la version 1.5 de Spring Boot

352

2.2 Configuration Maven pour la version 2 de Spring Boot

353

2.3 Utilisation du hot swapping

353

2.4 Packaging et lancement de l'application

354

2.5 Application Spring MVC minimum

354

3. L'autoconfiguration Spring Boot

355

3.1 La configuration automatique des Beans Spring

355

4. Les starters

357

4.1 Les starters courants

357

4.2 Les starters orientés messages

359

4.3 Les bases de données

360

4.4 Les web services

361

4.5 Les moteurs de rendus

361

4.6 Les starters moins courants	362
5. Spring MVC	363
6. Personnalisation de la bannière	363
7. Événements d'applications	364
8. La récupération des arguments de la ligne de commande	365
8.1 CommandLineRunner	365
8.2 ApplicationRunner	365
8.3 La configuration yaml et profiles	365
9. La configuration des logs	367
9.1 Les logs en couleur	367
9.2 Choix du type de logs	371
10. L'autoconfiguration pour Spring MVC	372
11. La gestion des sessions	373
12. Le guide de migration de la version 1.5 à la version 2.0	374
12.1 Fichiers de configuration	374
12.2 Comportements différents	375

12.3 Démarrage	375
12.4 Utilisation de ApplicationRunner ou CommandLineRunner	375
12.5 Configuration externalisée	376
12.6 Développement d'applications web	377
12.6.1 Web Starter en tant que dépendance transitive	377
12.7 Sécurité	377
13. Points clés	378
Spring et le NoSQL	
1. Introduction	379
2. Les modèles de données	380
2.1 Clé-valeur	380
2.2 Documents	380
2.3 Orienté colonne	381
2.4 Les bases orientées Graph	381
3. Principes des bases de données	381
3.1 Consistance des données	381
4. Pourquoi et quand utiliser une base NoSQL	

	383
5. Problèmes liés à l'utilisation des bases NoSQL	383
6. Limitations des bases de données NoSQL	384
7. Spring et le NoSQL	385
8. Cache de données	386
8.1 Cache simple	386
9. Cacher des données avec GemFire	389
10. GemFire en tant que base de données NoSQL	391
11. Redis en autonome	393
11.1 Utilisation de Redis pour le cache de données	393
11.2 Utilisation de Redis pour gérer des messages	395
12. MongoDB	398
12.1 MongoDB avec Spring Boot	398
12.2 MongoDB avec une API REST	400
13. Points clés	405

Spring Batch

1. Introduction	407
2. Architecture de l'application	410
3. Exemple Spring Batch 4	410
4. Chargeur H2 depuis un CSV	411
5. Dépendances Spring Batch 3 et 4	416
6. Autres nouveautés de la version 4	416
7. Points clés	417

Les intergiciels à messages (MOM)

1. Introduction	419
2. Implémentations open source	421
3. Implémentations propriétaires	422
4. Les cas d'utilisation	422

5. JMS et ActiveMQ	425
5.1 Utilisation simplifiée	426
6. RabbitMQ	429
6.1 Spring AMQP et RabbitMQ	429
6.2 Exemple RabbitMQ	430
7. Points clés	433
Spring et Kotlin	
1. Introduction	435
2. Caractéristiques principales du langage Kotlin	439
2.1 Les méthodes et les fonctions	439
2.2 L'immutabilité des objets	439
2.3 Les types	440
2.4 La gestion des valeurs nulles	443
2.5 Les appels chaînés sécurisés	444
2.6 Les lambdas	445
3. Contrôleur Spring MVC, Spring Boot en Kotlin	445

3.1 Fonction principale	446
3.2 Test associé à la fonction principale	446
4. Les plug-ins	452
5. Points clés	453
Spring et JHipster	
1. Introduction	455
2. JHipster comme outils multitechnologies	457
2.1 Côté client	457
2.2 Côté serveur	458
2.3 Côté déploiement	461
2.4 Côté Spring	461
3. Exemple de projet	462
4. Structure du projet	464
4.1 La partie front	464
4.2 Les grandes lignes de la partie back	464
4.2.1 Spring Boot	464

4.2.2 La classe serveur HTTP/servlet	469
5. Points clés	488
Index	489