

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence de l'ouvrage **EP4JASP** dans la zone de recherche
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

1. Préambule	23
2. Introduction	26
3. Organisation du livre	28
4. Public visé	28
5. Pourquoi Spring	29
6. Prérequis pour aborder Spring et Jakarta EE	29
7. Objectifs du livre	31

Chapitre 1

Éléments constitutifs du framework

1. Introduction	33
2. Historique	33
3. Utilité de Spring dans un projet	37
4. Vue d'ensemble et sujets abordés dans l'ouvrage	42
4.1 Les modules Spring	42
4.2 Sujets abordés	44
4.3 Version des briques Spring utilisées dans l'ouvrage	44
4.4 Version des ressources externes	45
4.5 Gestion des dépendances Maven	45
4.6 Utilisation d'un BOM (Bill Of Materials) Maven	47
5. Compléments	50
6. Points clés	51

2 _____ Java Spring

Le socle technique des applications Jakarta EE

Chapitre 2 Spring et les design patterns

1. Introduction	53
2. Le singleton en Java	54
2.1 Préoccupation	55
2.2 Solution Spring	57
2.3 Exemple	57
3. L'inversion de contrôle	61
4. Façade	62
5. Fabrique.	63
6. Décorateur.	64
7. Proxy	65
8. Modèle Vue Contrôleur (MVC)	66
9. Les templates	68
10. Stratégie	69
11. Points clés	70

Chapitre 3 Rappels sur les éléments externes à Spring

1. Codage equals et hashCode	71
1.1 Description de la problématique	74
1.2 Mise en œuvre	75
2. Projet Lombok.	77
3. Les systèmes de log.	79
4. Bases de données H2.	81
4.1 Description de la problématique	81
4.2 Mise en œuvre	82
4.2.1 Installation	82
4.2.2 Configuration du POM	82

- 4.2.3 Connexion à la base de données en utilisant JDBC 83
- 4.2.4 Utiliser un listener de servlet pour démarrer
et arrêter la base 84
- 5. Projets Maven 85
 - 5.1 Description de la problématique. 85
 - 5.2 Mise en œuvre 85
 - 5.3 Installation de Maven sous Windows 86
 - 5.4 Utilisation d'un archétype Maven 87
 - 5.5 Contenu du fichier pom.xml
dans les cas simples utilisés dans ce livre 88
- 6. Spring et les versions de Java 90
- 7. Points clés 94

Chapitre 4
Le conteneur Spring

- 1. Introduction 95
- 2. Les origines 95
- 3. Les modules fondamentaux 97
 - 3.1 Composition d'un bean. 97
 - 3.2 Le singleton et le prototype 98
 - 3.3 Les objets fondamentaux du package core. 98
 - 3.3.1 Le PropertyEditor 99
 - 3.4 Les PropertyValue 101
 - 3.5 Le BranWrapper 102
 - 3.6 La BeanFactory. 102
 - 3.7 L'interface BeanDefinition 103
 - 3.8 Le PropertyPlaceholderConfigurer 103
 - 3.9 Les objets fondamentaux du package context. 106
 - 3.10 Relations entre le bean, sa BeanFactory ou son contexte 107

4.	Configuration des beans	108
4.1	Configuration par un fichier XML	110
4.2	Configuration avec des annotations	111
4.3	Configurer les beans applicatifs implicitement	113
4.4	Configuration par lambdas	114
5.	Utilisation des beans : injection par setters et constructeurs	114
5.1	Mapping via le constructeur en XML	115
5.2	Comparaison des méthodes d'instanciation	117
5.3	Autres annotations de configuration	117
5.3.1	L'annotation @Primary	117
5.3.2	Les annotations @Profile et @Conditional	119
6.	Contrôle du cycle de vie : construction et destruction	129
7.	Exemple illustrant les mappings standards	130
7.1	Le projet Maven	130
7.2	Fichier de configuration de Spring	131
7.3	Dépendance Spring Core et système de log	131
7.4	Dépendance vers les bibliothèques de tests unitaires	132
7.5	Exemple illustrant l'utilisation des logs	133
7.6	Fichier de configuration spécifique pour les tests	134
8.	Le Spring Expression Language	135
8.1	Utilisation de l'ExpressionParser	135
8.2	L'EvaluationContext	136
8.3	Utilisation avec @Value	138
9.	Serveurs J2EE, Java EE et Jakarta EE	139
9.1	Généralités	139
9.2	Web services	145
9.2.1	Via une servlet	145
9.2.2	Serveur intégré	145
9.3	Les EJB	148
9.3.1	Fonctionnement	148
9.3.2	Création de l'EJB distant	151
9.3.3	Configuration de Maven côté client	152

- 9.3.4 Création du client 153
- 9.3.5 Intégration plus poussée 154
- 9.3.6 Le design pattern Business Delegate 156
- 10. Points clés 160

Chapitre 5
Configuration avancée

- 1. Introduction 161
- 2. Les fichiers de ressources 161
 - 2.1 Les fichiers de ressources standards 162
 - 2.2 Les fichiers de ressources chargés via le contexte Spring 163
 - 2.3 Les fichiers de ressources chargés via un service ResourceLoaderAware 164
 - 2.3.1 Le programme principal 164
 - 2.3.2 Le service ResourceLoaderAware 165
- 3. Convertisseurs et formateurs 166
 - 3.1 Built-in converters 166
 - 3.1.1 Pour les types standards 166
 - 3.1.2 Pour les tableaux et les listes 167
 - 3.1.3 Pour les énumérations 168
 - 3.1.4 Pour les objets en général 168
 - 3.2 Convertir un Array en Collection et String 170
 - 3.3 Converter Factory 171
 - 3.3.1 Classe StringToUtilisateurConverterFactory 171
 - 3.3.2 Classe UtilisateurToStringConverterFactory 171
 - 3.3.3 Classe LesConversionsAvecFactories 172
 - 3.4 Les formateurs par défaut 172
 - 3.4.1 Classe LesConversionsFormateurs 172
 - 3.5 Les formateurs personnalisés 173
 - 3.5.1 Classe CarteDeCredit 173
 - 3.5.2 Classe CarteDeCreditParser 174
 - 3.5.3 Classe CarteDeCreditPrinter 174

6 --- Java Spring

Le socle technique des applications Jakarta EE

3.5.4	Classe CarteDeCreditFormatter	175
3.5.5	Classe LesConversionsFormateursEx2	175
4.	Les BeanWrappers, le binding et les validateurs	176
4.1	Classe LesBeanWrappers	176
4.2	Classe UtilisateurValidator	177
5.	Points clés	178

Chapitre 6

Programmation orientée aspect avec Spring

1.	Introduction	179
2.	Pourquoi l’AOP	185
3.	Les concepts d’AOP	186
4.	Limites de Spring AOP et utilisation d’AspectJ	188
5.	Le support @AspectJ dans Spring	188
5.1	Activation du support	189
5.2	Activation de @AspectJ avec configuration XML	189
5.3	Activation de @AspectJ avec configuration Java	190
5.4	Déclaration d’un aspect	190
5.5	Déclaration d’un pointcut	191
5.6	Déclaration des greffons simples	193
5.7	Les types génériques	195
5.8	Détermination de noms d’arguments	196
5.9	Ordre d’appel des greffons interceptant le point de jonction	196
5.10	Injection	197
5.11	Le mécanisme des proxy	197
5.12	La face cachée des proxy en AOP	198
6.	Les API bas niveau pour Spring AOP	201
6.1	L’interface PointCut	201
6.2	L’interface ClassFilter	201
6.3	L’interface MethodMatcher	202

7. Points clés 202

Chapitre 7
Les tests et Spring

1. Introduction 203

2. Les mock objects 204

2.1 Mocks spécialisés pour "environnement" 205

2.2 Support 205

2.2.1 Utilités générales 205

2.2.2 Spring MVC 207

2.3 Tests d'intégration 207

2.3.1 Vue d'ensemble 207

2.3.2 Mise en cache du contexte de test 208

2.3.3 Tests back et front 208

2.4 Annotations 209

2.4.1 @ContextConfiguration 209

2.4.2 @WebAppConfiguration 210

2.4.3 @ContextHierarchy 210

2.4.4 @ActiveProfiles 211

2.4.5 @TestPropertySource 212

2.4.6 @DirtiesContext 212

2.4.7 Interface TestExecutionListener 213

2.4.8 @TransactionConfiguration 214

2.4.9 @Transactional 214

2.4.10 @Rollback 215

2.4.11 @BeforeTransaction 215

2.4.12 @AfterTransaction 215

2.4.13 @Sql, @SqlConfig et @SqlGroup 215

2.5 Annotations standards 216

3.	Les frameworks de test JUnit et TestNG	217
3.1	Utilisation avec JUnit 4	217
3.1.1	Spring JUnit Runner	217
3.1.2	@IfProfileValue	218
3.1.3	@ProfileValueSourceConfiguration	218
3.1.4	@Timed	218
3.1.5	@Repeat	219
3.1.6	Méta-annotations de support pour les tests	219
3.2	Les arcanes du framework Spring TestContext	223
3.2.1	Classes et interfaces du framework de tests	223
3.2.2	Configuration du TestExecutionListener par les annotations	225
3.2.3	Les TestExecutionListeners	226
3.2.4	Mélange XML et classes annotées	230
3.3	Configuration des contextes de tests	231
3.3.1	Configuration de contexte avec des initialiseurs de contexte	231
3.3.2	L'héritage dans la configuration des contextes	231
3.3.3	Prise en compte des profils d'environnement	232
3.3.4	Configuration de contexte avec des fichiers de propriétés de test	237
3.3.5	Déclarer un fichier de propriétés pour les tests	238
3.3.6	Détection du fichier de propriétés par défaut	239
3.4	Hiéarchie des contextes de tests	239
3.4.1	Héritage et surcharge des propriétés de test	239
3.4.2	Chargement d'un WebApplicationContext	240
3.4.3	Conventions	241
3.4.4	La sémantique de ressources explicites	241
3.4.5	Injecter des objets mockés	242
3.4.6	Cachabilité du contexte de test	243
3.4.7	Les hiérarchies des contextes	244
3.4.8	L'injection de dépendance dans les tests	247

- 3.5 Le scope session lors d'un test de requête 249
 - 3.5.1 Le scope session lors d'un test de requête 249
 - 3.5.2 Test de beans de scope request 250
 - 3.5.3 Configuration d'un bean de scope session 251
- 3.6 Les transactions 252
 - 3.6.1 La gestion des transactions 252
 - 3.6.2 Les transactions managées par le test 252
 - 3.6.3 Activation et désactivation des transactions 253
 - 3.6.4 Comportement du commit et du rollback
d'une transaction 254
 - 3.6.5 Exécution de code en dehors d'une transaction 255
 - 3.6.6 Configuration d'un gestionnaire de transactions 256
 - 3.6.7 Démonstration de toutes les annotations liées
à la transaction 256
- 3.7 Les scripts SQL. 257
 - 3.7.1 Exécuter des scripts SQL 258
 - 3.7.2 La sémantique du path des ressources 260
 - 3.7.3 La détection de script par défaut 261
 - 3.7.4 Déclaration de plusieurs @Sql. 261
 - 3.7.5 Phases d'exécution pour les scripts 262
 - 3.7.6 Script de configuration avec SqlConfig. 262
 - 3.7.7 La gestion des transactions pour @Sql 263
- 3.8 Classes de support TestContext. 265
 - 3.8.1 Classes de support de JUnit 265
 - 3.8.2 Spring JUnit Runner. 266
- 3.9 Framework Spring MVC Test 267
 - 3.9.1 Projet autonome 267
 - 3.9.2 Tests côté serveur 268
 - 3.9.3 Définir les attendus 273
 - 3.9.4 Ajouts de filtres. 275
 - 3.9.5 Tests REST côté client 276
- 3.10 Autres ressources 276
- 4. Points clés 277

Chapitre 8**Partie back de l'application**

1. Description de la problématique	279
2. Mise en œuvre	282
2.1 Configuration d'une entité de la couche domaine	282
2.2 Accès à l'objet du domaine	284
2.3 Suppression en cascade	285
2.4 Requêtes typées	285
2.5 Cache de niveaux 1 et 2	286
2.6 Bean Validation (JSR-303)	286
2.7 L'API Criteria	289
2.8 Accès à la base de données	289
2.9 Le fichier persistence.xml	290
2.10 Tests JPA	291
3. Pour aller plus loin	293
3.1 Librairie Java jcabi-aspects	293
3.2 Métriques AspectJ	293
4. Utilisation de MapStruct	293
4.1 Approche API-First	294
5. Les outils	295
5.1 Swagger	295
5.2 OpenAPITools	295
5.3 Autres	296
6. Les générateurs de code	296
7. Utilisation du plugin	297
7.1 Pour le générateur Swagger	297
7.2 Pour le générateur OpenAPITools	297
8. Personnalisation	298
8.1 Swagger	298
8.2 OpenAPITools	298
9. Conception d'une description API	299

- 10. Outillage pour le design de l'API 299
- 11. Spring Actuator 300
- 12. Points clés 302

Chapitre 9
Spring dans un contexte web JSP

- 1. Spring MVC 303
 - 1.1 Fonctionnement global 305
 - 1.1.1 Configuration simple XML 305
 - 1.1.2 Configuration par annotation 308
 - 1.2 Éléments complexes du contrôleur. 309
 - 1.2.1 Fonctionnement global du contrôleur. 309
 - 1.2.2 Exemple de la classe Simple 311
 - 1.2.3 Simple revisité. 311
 - 1.2.4 Par le path 312
 - 1.2.5 Par un motif sur le path 313
 - 1.2.6 Par le path et une méthode 313
 - 1.2.7 Mappé par path + méthode
+ présence de paramètres de query. 314
 - 1.2.8 Mappé par path + méthode + présence d'un header . 314
 - 1.2.9 Par l'absence d'un header 315
 - 1.2.10 Par consommation 316
 - 1.2.11 Par production via Accept=application/json 316
 - 1.2.12 Par production via Accept=application/xml 317
 - 1.2.13 Paramètres de query 318
 - 1.2.14 Groupes de paramètres de query 319
 - 1.2.15 Variable du path 319
 - 1.2.16 Corps de la requête 320
 - 1.2.17 En-tête et corps de la requête. 321
 - 1.2.18 Arguments dans la requête 321
 - 1.2.19 Arguments de la réponse 322
 - 1.2.20 Session. 323

1.2.21	Handler personnalisé	323
1.2.22	Lire les données encodées dans l'URL	324
1.2.23	Lire une structure XML	325
1.2.24	Écrire dans une structure XML via Accept=application/xml	325
1.2.25	Lire une structure JSON.	326
1.2.26	Écrire dans une structure JSON via Accept=application/json	327
1.2.27	HTML généré par un template JSP	328
1.2.28	Mapping à partir d'un modèle	328
1.2.29	Utiliser des variables dans un template de vue.	329
1.2.30	Data binding avec des variables d'URI	329
1.2.31	Types primitifs	330
1.2.32	Date	333
1.2.33	Conversion des collections.	333
1.2.34	Utiliser des collections formatées	334
1.2.35	Utiliser des objets personnalisés	334
1.2.36	Utiliser un convertisseur personnalisé	335
1.2.37	Validation	339
1.2.38	@ExceptionHandler dans un contrôleur	340
1.2.39	@ExceptionHandler global	340
1.2.40	Templates de String pour les URI	340
1.2.41	UriComponentsBuilder	341
2.	Client REST	342
2.1	Utilisation du RestTemplate	342
2.2	Le bean de domaine avec l'annotation REST pour les champs manquants	343
3.	Spring Security	343
3.1	Introduction à Spring Security	343
3.2	Spring Security dans un environnement web	346
3.2.1	Authentification par Spring	346
3.2.2	Authentification par page de login personnalisée	347
3.2.3	Authentification par base de données.	347

4. Points clés 349

Chapitre 10
Intégration JSF2

1. Introduction 351

2. Mojarra 353

3. Architecture 354

4. Cycle de vie 356

 4.1 Requête. 356

 4.2 Restore View ou Reconstruct Component Tree 356

 4.3 Apply Request Value. 357

 4.4 Perform Validation 357

 4.5 Synchronize Model ou Update Model Values 357

 4.6 Invoke Application Logic 358

 4.7 Render Response 358

 4.8 Réponse 358

 4.9 Fichier web.xml 358

 4.10 Dépendances 359

 4.11 Fichier faces-config.xml. 359

 4.12 Bean managé sans Spring 361

 4.13 Exemple de vue JSF 361

 4.14 Aperçu d'un composant JSF 362

5. Intégration Spring 363

 5.1 Architecture 364

 5.2 Dépendances 364

 5.3 Fichier web.xml 364

 5.4 Fichier faces-config.xml. 365

 5.5 Couches basses (back) 366

6. Points clés 366

Chapitre 11**Application Spring Angular**

1. Introduction	367
2. La partie backend	368
2.1 Génération d'un backend	368
3. La partie frontend	371
3.1 Angular CLI	372
3.2 Création du projet initial	372
3.3 Démarrage de l'application	372
3.4 Créer un service Compte	373
4. Points clés	375

Chapitre 12**Spring-HATEOAS**

1. Introduction	377
1.1 Exemple de liens hypermédias codés manuellement	380
1.2 Aller plus loin avec l'hypermédia	382
1.3 Autoconfiguration par annotation	383
1.4 Fournisseurs de relations	383
1.5 Fournisseur d'URI compacte	384
1.6 Support du côté client	385
1.7 Découverte de liens coté client	386
1.8 Utilisation du <code>@RepositoryRestResource</code>	386
2. Points clés	394

Chapitre 13
Documentation Spring REST Docs

- 1. Introduction 395
 - 1.1 Dépendance sur la bibliothèque de tests 398
 - 1.2 Dépendance sur les plugins Maven 398
 - 1.3 Les extraits (snippets) 400
- 2. Exemple JUnit 5 (Jupiter) 401
- 3. Requête et réponse 402
- 4. Réponse avec un JSON imbriqué 403
 - 4.1 Les paramètres de requête 404
 - 4.2 Les paramètres inclus dans le path 404
 - 4.3 Les Request parts 405
 - 4.4 Les Request parts payloads 405
 - 4.5 Les champs 405
 - 4.6 Liens hypermédias dans la réponse 406
 - 4.7 Les en-têtes 407
- 5. Personnalisation de la documentation 407
- 6. Couplage Swagger 2 408
 - 6.1 Utilisation Springfox 408
 - 6.2 En dehors de Spring Boot 409
- 7. Utilisation avec Spring Data Rest 411
- 8. Récapitulatif sur la documentation générée 411
- 9. Points clés 412

Chapitre 14

Spring Boot

1. Introduction	413
2. Configuration des exemples.	415
2.1 Configuration Maven pour la version 1.5 de Spring Boot . . .	415
2.2 Configuration Maven pour la version 2 de Spring Boot	415
2.3 Utilisation du hot swapping.	416
2.4 Packaging et lancement de l'application	416
2.5 Application Spring MVC minimum	417
3. L'auto-configuration Spring Boot	418
4. Les starters	420
4.1 Les starters courants	420
4.2 Les starters orientés messages.	423
4.3 Les bases de données	423
4.4 Les web services.	424
4.5 Les moteurs de rendus.	424
4.6 Les starters moins courants	425
5. Spring MVC	426
6. Personnalisation de la bannière	426
7. Événements d'applications	427
8. La récupération des arguments de la ligne de commande	428
8.1 CommandLineRunner.	428
8.2 ApplicationRunner	429
8.3 La configuration yaml et profiles	429
9. L'annotation EnableConfigurationProperties.	431
10. La configuration des logs	431
10.1 Les logs en couleur.	431
10.2 Choix du type de logs	436
11. L'auto-configuration pour Spring MVC	436
12. La gestion des sessions	437

- 13. Le guide de migration de la version 1.5 à la version 2.x 439
 - 13.1 Fichiers de configuration 439
 - 13.2 Comportements différents 439
 - 13.3 Démarrage 440
 - 13.4 Utilisation de ApplicationRunner ou CommandLineRunner . 440
 - 13.5 Configuration externalisée 440
 - 13.6 Développement d'applications web 441
 - 13.7 Sécurité 442
- 14. Description avancée de Spring Boot 443
- 15. Le chargement du BootstrapContext 444
- 16. Mise en place des listeners 445
 - 16.1 Le starting : ApplicationStartingEvent 445
 - 16.2 L'environnement : l'ApplicationEnvironmentPreparedEvent . 446
 - 16.3 L'EnvironmentPostProcessorApplicationListener 448
- 17. La configuration des logs 448
- 18. Création de l'ApplicationContext 449
- 19. Préparation et rafraîchissement du contexte 449
- 20. L'EnableAutoConfiguration 450
- 21. Lancement des runners 451
- 22. Points clés 451

Chapitre 15
Spring et le NoSQL

- 1. Introduction 453
- 2. Les modèles de données 455
 - 2.1 Modèle Clé-valeur 455
 - 2.2 Modèle Documents 455
 - 2.3 Modèle Orienté colonnes 456
 - 2.4 Les bases orientées Graph 457
- 3. Principes des bases de données 457

4. Pourquoi et quand utiliser une base NoSQL	460
5. Problèmes liés à l'utilisation des bases NoSQL	462
6. Limitations des bases de données NoSQL	463
7. Spring et le NoSQL	464
8. Cache de données	465
8.1 Cache simple	465
9. Cacher des données avec GemFire	468
10. GemFire en tant que base de données NoSQL	469
11. Redis en autonome	471
11.1 Utilisation de Redis pour le cache de données	471
11.2 Utilisation de Redis pour gérer des messages	474
12. MongoDB	476
12.1 MongoDB avec Spring Boot	477
12.2 MongoDB avec une API REST	479
13. Points clés	483

Chapitre 16 Spring Batch

1. Introduction	485
2. Architecture d'un batch	489
3. Exemple Spring Batch version 4	490
4. Chargeur H2 depuis un CSV	492
5. Dépendances Spring Batch 3 et 4	496
6. Autres nouveautés de la version 4	496
7. Évolutions des batchs	498
8. Points clés	499

Chapitre 17
Les intergiciels à messages (MOM)

- 1. Introduction 501
- 2. Implémentations open source 503
- 3. Implémentations propriétaires 504
- 4. Les cas d'utilisation. 504
- 5. JMS et ActiveMQ. 507
- 6. RabbitMQ. 512
 - 6.1 Spring AMQP et RabbitMQ. 512
 - 6.2 Exemple RabbitMQ. 513
- 7. Points clés 516

Chapitre 18
Spring et Kotlin

- 1. Introduction 517
- 2. Caractéristiques principales du langage Kotlin. 519
 - 2.1 Les méthodes et les fonctions. 519
 - 2.2 L'immutabilité des objets 519
 - 2.3 Les types. 520
 - 2.4 La gestion des valeurs nulles. 524
 - 2.5 Les appels chaînés sécurisés 524
 - 2.6 Les lambdas 525
- 3. Contrôleur Spring MVC, Spring Boot en Kotlin 526
 - 3.1 Fonction principale 527
 - 3.2 Test associé à la fonction principale. 527
- 4. Les plugins. 530
- 5. Points clés 532

Chapitre 19**Introduction à Spring Reactor et Spring Webflux**

1. Introduction	533
2. Spring Reactor	534
2.1 Présentation	534
2.2 Utilisation de Reactor Core	538
2.2.1 Les threads	538
2.2.2 Les FutureTask	538
2.2.3 Le CompletableFuture	539
2.2.4 Flux et Mono	540
2.3 Les tests	544
2.3.1 Vérifications avec StepVerifier	544
2.3.2 Emissions manuelles avec TestPublisher	547
3. WebFlux	549
3.1 Définition du terme réactif	550
3.2 Les couches réactives	551
3.2.1 la couche contrôleur	551
3.2.2 La couche des services	555
3.2.3 La couche repository	556
3.2.4 Repository reactif R2DBC	556
4. Client réactif	561
5. Les tests avec WebFlux	562
5.1 Tests unitaires	562
5.1.1 Tests unitaires avec des applications réactives	562
5.1.2 Utilisation de MockWebServer	563
5.2 Tests d'intégration	565
6. Server Site Event avec Spring	568
7. Pour aller plus loin	570
8. Points clés	570

Chapitre 20
Spring et JHipster

- 1. Introduction 571
- 2. Généralités 572
 - 2.1 JHipster web 573
 - 2.2 Personnalisation de l’outil JHispter 578
 - 2.3 Niveaux de personnalisations 579
- 3. Créer un blueprint 584
 - 3.1 Blueprint pour utiliser lombok dans le domain 586
 - 3.2 Exécution du blueprint local 589
- 4. JHipster comme outil multi-technologies 590
 - 4.1 Côté client 590
 - 4.2 Côté serveur 592
 - 4.3 Côté déploiement 594
 - 4.4 Côté Spring 595
- 5. Structure du projet 596
 - 5.1 La partie front 596
 - 5.2 Les grandes lignes de la partie back 596
 - 5.2.1 Spring Boot 597
 - 5.2.2 La classe serveur HTTP/servlet 603
- 6. JHipster et WebFlux 615
 - 6.1 Configuration DatabaseConfiguration 616
 - 6.2 Configuration DateTimeFormatConfiguration 618
 - 6.3 Configuration LocaleConfiguration 618
 - 6.4 Configuration ReactorConfiguration 619
 - 6.5 Configuration SecurityConfiguration 619
 - 6.6 Configuration WebConfigurer 619
 - 6.7 Les tests 619
- 7. Points clés 620

Chapitre 21**Utilisation de GraphQL**

1. Introduction	621
2. Le schéma	622
2.1 La section Query	624
2.2 La section Mutation	625
3. Intégration de GraphQL dans Spring	626
4. L'extension Spring Data Querydsl	627
5. Les contrôleurs GraphQL	629
6. Autoconfiguration	632
6.1 Version HTTP	632
6.2 Le Service GraphQLService	632
6.3 Le RuntimeWiringConfigurer	633
6.4 Gestion des erreurs	634
7. Conclusion	634
8. Points clés	635

Chapitre 22**Nouveautés de Spring 6**

1. Le futur de Spring	637
2. Prise en charge des modules Java	638
3. Spring Native	638
4. Project Leyden	639
5. Spring Observability	639
6. Obsolescence programmée	639
Index	641

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence de l'ouvrage **EP2JEE** dans la zone de recherche
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1 Introduction à Jakarta EE

1. Introduction	17
2. L'écosystème Java	18
3. La plateforme Java SE	19
3.1 Introduction	19
3.2 Les différentes implémentations de la plateforme	21
3.3 Installation du JDK sur Windows	22
3.3.1 Installation de la plateforme OpenJDK	22
3.3.2 Installation de la plateforme Oracle JDK	22
3.3.3 Configuration des variables d'environnement	23
3.3.4 Tester le bon fonctionnement	24
3.4 Installation du JDK sur Linux	25
3.4.1 Installation de la plateforme OpenJDK	25
3.4.2 Installation de la plateforme Oracle JDK	25
3.4.3 Configuration de la variable JAVA_HOME	27
3.4.4 Tester le bon fonctionnement	28
4. La plateforme Jakarta EE	28
4.1 Introduction	28
4.2 Le processus d'évolution de la plateforme	30
4.3 Les serveurs d'applications	30
4.3.1 La notion de conteneur	30
4.3.2 Différents serveurs d'applications	31
4.4 Les spécifications Jakarta EE	32
4.5 La répartition des spécifications	33

5.	Le protocole HTTP	34
5.1	Introduction	34
5.2	Les URI	37
5.3	La requête	38
5.3.1	La structure.	38
5.3.2	Les types de requêtes	39
5.3.3	Les attributs de requêtes	40
5.4	La réponse	42
5.4.1	La structure.	42
5.4.2	Les codes de statut	43
5.4.3	Les attributs de réponses	45
5.5	La gestion du cache	45
5.6	La gestion de l'authentification	47
5.7	Les types de médias.	48
6.	L'environnement de développement.	49
6.1	Tomcat	49
6.1.1	Téléchargement	49
6.1.2	Installation sur Windows	50
6.1.3	Installation sur Linux	51
6.2	MySQL	53
6.2.1	Téléchargement	53
6.2.2	Installation sur Windows	53
6.2.3	Installation sur Linux	62
6.3	Eclipse	63
6.3.1	Téléchargement et installation	63
6.3.2	Premier démarrage	64
6.3.3	Configuration.	65
6.3.4	Création d'un projet d'application web	70
6.3.5	Déploiement et exécution d'une application web	75
6.4	Gradle.	77
6.4.1	Présentation	77
6.4.2	La gestion des dépendances	78
7.	Les exemples du livre	81
7.1	Introduction	81
7.2	Description fonctionnelle	81
7.3	Le diagramme de classes d'analyses	82
7.4	La base de données	83

8. Conclusion 84

Chapitre 2
Les servlets

1. Introduction 85

2. La structure d'une application web 86

 2.1 Présentation 86

 2.2 La structure logique d'une application web 86

 2.3 La structure physique d'une application web 87

 2.3.1 Présentation 87

 2.3.2 Le paramétrage 90

3. Le projet 92

4. Les servlets 93

 4.1 Qu'est-ce qu'une servlet ? 93

 4.2 Le cycle de vie d'une servlet 94

 4.3 La première servlet 95

 4.3.1 La création de la servlet 95

 4.3.2 La déclaration d'une servlet 99

 4.3.3 Les paramètres d'initialisation d'une servlet 101

 4.3.4 La mise en évidence du cycle de vie d'une servlet 102

 4.4 La classe HttpServlet 104

 4.5 Les différents contextes 105

 4.5.1 Le contexte d'application 105

 4.5.2 Le contexte de session 106

 4.5.3 Le contexte de requête 106

 4.6 La lecture de la requête 106

 4.6.1 Lecture des informations de l'URL 106

 4.6.2 Lecture de l'en-tête de la requête 107

 4.6.3 Lecture des paramètres 109

 4.7 La création de la réponse 114

 4.7.1 Écriture de l'en-tête de la réponse 115

 4.7.2 Écriture du corps de la réponse 117

 4.7.3 La gestion de l'envoi de la réponse 117

 4.7.4 Premier exemple de mise en œuvre 118

 4.7.5 Déléguer la création de la réponse 120

4.7.6	Second exemple de mise en œuvre	123
4.7.7	La redirection	126
4.7.8	La gestion des erreurs	128
4.8	Les servlets asynchrones	129
5.	Le suivi de l'activité de l'utilisateur	133
5.1	Présentation	133
5.2	Les cookies	134
5.2.1	Présentation	134
5.2.2	La création d'un cookie	135
5.2.3	La lecture d'un cookie	137
5.3	La session	138
5.3.1	Présentation	138
5.3.2	Le suivi de session	139
5.3.3	La création d'une session	141
5.3.4	L'utilisation d'une session	142
5.3.5	La liaison d'attributs	145
6.	Les filtres	145
6.1	Présentation	145
6.2	Mise en œuvre	146
6.2.1	La création d'un filtre	146
6.2.2	La déclaration du filtre	151
6.2.3	L'utilisation d'un filtre	153
7.	Les événements	155
7.1	Présentation	155
7.2	Le paramétrage des classes	155
7.3	Les événements liés au contexte d'application	156
7.4	Les événements liés au contexte de session	157
7.5	Les événements liés au contexte de requête	158
8.	La sécurité	160
9.	Conclusion	160

Chapitre 3
La présentation avec les JSP

- 1. Introduction 161
- 2. Le projet 162
 - 2.1 La création du projet 162
 - 2.2 La création d'une JSP 163
- 3. Le principe d'exécution 166
- 4. Le paramétrage d'une JSP 170
- 5. Les directives 172
 - 5.1 Présentation 172
 - 5.2 La directive page 173
 - 5.2.1 Présentation 173
 - 5.2.2 Les attributs de la directive 173
 - 5.2.3 Un exemple de mise en œuvre 174
 - 5.3 La directive taglib 175
 - 5.3.1 Présentation 175
 - 5.3.2 Les attributs de la directive 175
 - 5.3.3 Un exemple de mise en œuvre 175
 - 5.4 La directive include 176
 - 5.4.1 Présentation 176
 - 5.4.2 Les attributs de la directive 176
 - 5.4.3 Un exemple de mise en œuvre 176
- 6. Les éléments de script 178
 - 6.1 Présentation 178
 - 6.2 Les déclarations 179
 - 6.2.1 Présentation 179
 - 6.2.2 Un exemple de mise en œuvre 179
 - 6.3 Les scriptlets 180
 - 6.3.1 Présentation 180
 - 6.3.2 Un exemple de mise en œuvre 181
 - 6.4 Les expressions 183
 - 6.4.1 Présentation 183
 - 6.4.2 Un exemple de mise en œuvre 183
 - 6.5 Les commentaires 183
 - 6.6 L'EL (Expression Language) 184

7.	Les objets disponibles dans une JSP	184
7.1	Présentation	184
7.2	Un exemple de mise en œuvre	186
8.	La gestion des erreurs	188
8.1	Présentation	188
8.2	Les erreurs à la transformation	188
8.3	Les erreurs à l'exécution	190
8.4	L'utilisation de l'instruction try/catch	191
8.5	L'utilisation des pages d'erreur JSP	191
8.6	L'utilisation des pages d'erreur globales	193
9.	L'utilisation de fragments	194
9.1	Présentation	194
9.2	L'inclusion statique	194
9.3	L'inclusion dynamique	194
9.4	Un exemple d'inclusion	195
9.5	Une inclusion variable	197
10.	Les actions standards	197
10.1	Présentation	197
10.2	Qu'est-ce qu'un JavaBean ?	198
10.3	<code><jsp:useBean></code>	199
10.3.1	Présentation	199
10.3.2	Exemple	200
10.4	<code><jsp:setProperty></code>	201
10.4.1	Présentation	201
10.4.2	Exemple	202
10.5	<code><jsp:getProperty></code>	203
10.5.1	Présentation	203
10.5.2	Exemple	203
10.6	<code><jsp:include></code>	203
10.7	<code><jsp:forward></code>	203
10.8	<code><jsp:param></code>	204
11.	L'EL (Expression Language)	205
11.1	Présentation	205
11.2	La syntaxe	205
11.3	Les objets disponibles	206
11.4	La recherche dans les différents contextes	209

11.5	Les opérateurs	209
11.5.1	Les opérateurs arithmétiques	209
11.5.2	L'opérateur de concaténation des chaînes de caractères	210
11.5.3	Les opérateurs de comparaison	210
11.5.4	Les opérateurs logiques	210
11.5.5	L'opérateur empty	211
11.5.6	L'opérateur conditionnel	211
11.5.7	L'opérateur d'assignation	211
11.5.8	L'opérateur point-virgule	211
11.5.9	Les parenthèses	212
11.5.10	La précedence des opérateurs	212
11.6	La manipulation des énumérés	212
11.7	La manipulation des variables et méthodes statiques	213
11.8	La manipulation des collections	213
11.8.1	La création	213
11.8.2	L'exploitation	214
11.9	La gestion des erreurs	218
12.	Les balises JSTL	220
12.1	Présentation	220
12.2	Les modifications du projet	222
12.3	L'utilisation d'une librairie dans une JSP	222
12.4	La librairie de base (core)	222
12.4.1	<c:out>	223
12.4.2	<c:set>	223
12.4.3	<c:remove>	224
12.4.4	<c:catch>	224
12.4.5	<c:if>	224
12.4.6	<c:choose>, <c:when>, <c:otherwise>	224
12.4.7	<c:forEach>	225
12.4.8	<c:forTokens>	226
12.4.9	<c:import>	227
12.4.10	<c:url>	227
12.4.11	<c:redirect>	228
12.4.12	<c:param>	228

12.5	La librairie d'internationalisation et de formatage (fmt)	228
12.5.1	Les principes	228
12.5.2	La définition de la Locale	231
12.5.3	Le chargement des ressources	231
12.5.4	L'affichage des messages	233
12.5.5	Un exemple de mise en œuvre	234
12.5.6	<fmt:timeZone> et <fmt:setTimeZone>	236
12.5.7	<fmt:formatDate> et <fmt:parseDate>	237
12.5.8	<fmt:formatNumber> et <fmt:parseNumber>	239
12.6	La librairie de fonctions (fn)	241
12.6.1	fn:contains	242
12.6.2	fn:containsIgnoreCase	242
12.6.3	fn:endsWith	242
12.6.4	fn:escapeXml	242
12.6.5	fn:indexOf	242
12.6.6	fn:join	243
12.6.7	fn:length	243
12.6.8	fn:replace	243
12.6.9	fn:split	243
12.6.10	fn:startsWith	243
12.6.11	fn:substring	244
12.6.12	fn:substringAfter	244
12.6.13	fn:substringBefore	244
12.6.14	fn:toLowerCase	244
12.6.15	fn:toUpperCase	245
12.6.16	fn:trim	245
13.	Les balises personnalisées	245
13.1	Présentation	245
13.2	La création d'une balise personnalisée	246
13.3	L'utilisation d'une balise personnalisée	248
13.4	L'envoi d'informations à une balise personnalisée	248
13.5	La récupération d'informations d'une balise personnalisée	250
14.	Conclusion	253

Chapitre 4**La persistance des données avec JDBC et JPA**

1. Introduction	255
2. La spécification JDBC	256
2.1 Description	256
2.2 Les différents types de pilotes	256
2.3 L'API JDBC	257
2.4 Le projet	259
2.4.1 La création du projet	259
2.4.2 Le référencement du pilote JDBC	260
2.5 La connexion	260
2.5.1 Introduction	260
2.5.2 La première connexion	260
2.5.3 La mise en place d'un pool de connexion	266
2.5.4 Compléments sur l'interface Connection	270
2.6 L'exploitation des données	272
2.6.1 L'interface Statement	272
2.6.2 L'interface PreparedStatement	278
2.6.3 L'interface CallableStatement	283
2.6.4 L'utilisation du ResultSet	284
2.6.5 La gestion des exceptions	290
2.7 La gestion des transactions	293
2.8 Conclusion	295
3. La spécification JPA	295
3.1 Description	295
3.2 Les principales implémentations	295
3.3 L'API JPA	296
3.3.1 Les entités	296
3.3.2 Le gestionnaire d'entités	297
3.3.3 Les transactions	299
3.4 Le projet	299
3.5 La connexion	302
3.5.1 Le fichier persistence.xml	302
3.5.2 L'EntityManager et l'EntityManagerFactory	306

3.6	Le mappage	309
3.6.1	Le contexte	309
3.6.2	Les annotations indispensables : <code>@Entity</code> et <code>@Id</code>	310
3.6.3	La gestion des tables : <code>@Table</code> , <code>@SecondaryTable</code>	311
3.6.4	La gestion des variables membres : <code>@Column</code> , <code>@Basic</code> , <code>@Transient</code>	314
3.6.5	La gestion avancée des variables membres	316
3.6.6	La gestion de l'héritage	318
3.6.7	La gestion des clés primaires simples : <code>@Id</code> , <code>@GeneratedValue</code>	320
3.6.8	La gestion des clés primaires composites avec <code>@IdClass</code>	321
3.6.9	La gestion des clés primaires composites avec <code>@EmbeddedId</code> , <code>@Embeddable</code> et <code>@MapsId</code>	326
3.6.10	La gestion des relations 1:1	328
3.6.11	La gestion des relations 1:n	333
3.6.12	La gestion des relations n:m	338
3.6.13	Les annotations <code>@JoinColumn</code> et <code>@JoinTable</code>	340
3.6.14	La gestion des collections	341
3.6.15	La gestion des Map	342
3.7	L'exploitation des données	342
3.7.1	Les méthodes de l'EntityManager	342
3.7.2	L'utilisation des transactions	344
3.7.3	L'enregistrement d'une entité	345
3.7.4	La recherche d'une entité par son identité	347
3.7.5	La modification d'une entité	348
3.7.6	La suppression d'une entité	350
3.7.7	Les traitements complexes	351
3.7.8	Le chargement différé	356
3.7.9	Le tri	360
3.7.10	La manipulation d'une entité détachée	360
3.7.11	La validation des données	363
3.8	Le JPQL	363
3.8.1	Présentation	363
3.8.2	L'exécution d'une requête JPQL	364
3.8.3	Les requêtes simples	365
3.8.4	Les requêtes avec restrictions	366
3.8.5	Les requêtes imbriquées	367

- 3.8.6 Les requêtes paramétrées 367
- 3.8.7 Les requêtes avec jointure 368
- 3.8.8 Les calculs d'agrégats 369
- 3.8.9 Les requêtes nommées 370
- 3.8.10 Les fonctions disponibles 370
- 3.8.11 L'exploitation du résultat 372
- 3.9 L'API Criteria 374
- 3.10 JPA et l'injection de dépendances 376
 - 3.10.1 Présentation 376
 - 3.10.2 L'implémentation 376
 - 3.10.3 La mise en place 376
- 4. Conclusion 383

Chapitre 5
Le framework de présentation JSF

- 1. Présentation générale 385
- 2. Présentation de JSF 386
 - 2.1 Généralités 386
 - 2.2 Principes de fonctionnement 386
 - 2.3 Le projet 388
- 3. Le paramétrage général 390
 - 3.1 Le fichier faces-config.xml 390
 - 3.2 Le fichier beans.xml 390
 - 3.3 Le fichier web.xml 391
- 4. L'exemple 392
- 5. Les managedBeans 393
 - 5.1 Présentation 393
 - 5.2 Mise en œuvre 393
- 6. Les facelets 396
 - 6.1 Les bibliothèques 396
 - 6.2 L'EL (Expression Language) 400
 - 6.3 Mise en œuvre 400
- 7. Les conversions 406

8.	Les validations	410
8.1	La validation dans la facelet	411
8.2	La validation dans un validateur	413
8.3	La validation avec la spécification Bean Validation	415
9.	L'internationalisation	418
9.1	La création des fichiers .properties	418
9.2	La déclaration dans le fichier faces-config.xml	419
9.3	La modification des facelets	419
9.4	L'internationalisation des validateurs	423
9.5	Bean Validation et l'internationalisation	424
9.6	L'utilisation des messages paramétrés	425
10.	Conclusion	426

Chapitre 6

Les services web REST

1.	Introduction	427
2.	Présentation des services web REST	428
2.1	Les principes	428
2.2	Les caractéristiques	429
2.3	Vers les services web REST	430
2.4	Les limitations	430
3.	La mise en place de l'environnement de développement	431
3.1	Le choix d'une implémentation	431
3.2	La mise en place du projet et des dépendances	431
4.	La spécification JAX-RS	433
4.1	Introduction	433
4.2	Le fonctionnement général	433
5.	La configuration de l'application	436
6.	L'exposition des ressources	437
6.1	Présentation	437
6.2	La classe ressource et ses méthodes	437
6.3	L'extraction et l'injection d'informations	438
6.4	Les types de retours des méthodes ressources	447

- 7. La validation des données 451
 - 7.1 Les annotations 451
 - 7.2 Le paramétrage des implémentations JAX-RS 452
 - 7.3 La validation des informations élémentaires 452
 - 7.4 La validation des entités 454
- 8. Les providers 455
 - 8.1 Présentation 455
 - 8.2 Les entity providers 455
 - 8.3 Les exceptions providers 470
- 9. Les filtres et les intercepteurs 472
 - 9.1 Les filtres 472
 - 9.2 Les intercepteurs 476
- 10. Complément sur le déploiement 479
- 11. L'API cliente 480
 - 11.1 Introduction 480
 - 11.2 Le fonctionnement général 481
 - 11.3 La mise en place du projet et des dépendances 482
 - 11.4 La mise en œuvre 482
- 12. Le côté client en JavaScript 490
 - 12.1 Introduction 490
 - 12.2 La mise en place du projet 491
 - 12.3 Une requête GET retournant un contenu JSON 493
 - 12.4 Une requête POST envoyant un contenu JSON 495
- 13. Conclusion 497

Chapitre 7

Introduction aux WebSockets

- 1. Introduction 499
- 2. Le fonctionnement 500
 - 2.1 Les étapes de la communication 500
 - 2.2 L'établissement de la communication 500
- 3. La spécification WebSocket 502
- 4. L'exemple 502

5.	Le côté serveur	503
5.1	La mise en place de l'environnement de développement	503
5.2	Le développement par annotations	504
5.3	La session	506
5.4	Les encodeurs et les décodeurs	507
6.	Le côté serveur de l'exemple	510
7.	Le côté client en Java	512
7.1	Introduction	512
7.2	La mise en place de l'environnement de développement	512
7.3	Le développement par annotations	513
7.4	La mise en place de la communication	514
8.	Le côté client en JavaScript	515
8.1	Introduction	515
8.2	L'interface WebSocket	516
8.3	Le côté client de l'exemple	516
9.	Conclusion	518

Chapitre 8

Déploiement d'applications sur Tomcat

1.	Introduction	519
2.	L'installation de l'environnement	519
2.1	L'installation de Java	519
2.2	L'installation de Tomcat	520
2.2.1	La vérification préalable	520
2.2.2	Le téléchargement de Tomcat	521
2.2.3	L'installation à partir d'une archive	522
2.2.4	L'installation à partir d'un installeur sur un environnement Windows	524
2.2.5	La vérification du bon fonctionnement	531
3.	L'architecture	531
3.1	L'organisation physique	531
3.2	L'organisation logique	533

4.	L'administration générale	534
4.1	L'application à déployer	534
4.1.1	Le projet	534
4.1.2	La livraison	536
4.1.3	La gestion du nom de domaine	536
4.2	Vue d'ensemble	537
4.3	Le fichier server.xml	537
4.3.1	La structure générale	537
4.3.2	La balise <Server>	538
4.3.3	La balise <Service>	539
4.3.4	La balise <Connector>	539
4.3.5	La balise <Executor>	542
4.3.6	La balise <Engine>	543
4.3.7	La balise <Host>	544
4.3.8	La balise <Alias>	547
4.3.9	La balise <Context>	547
4.3.10	La balise <Valve>	550
4.3.11	Les balises <Resource> et <ResourceLink>	552
5.	La mise en place de la sécurité	553
5.1	La restriction d'accès	554
5.1.1	La restriction sur l'adresse IP de l'émetteur	554
5.1.2	La restriction sur l'adresse IP du serveur	556
5.2	L'authentification et l'autorisation	556
5.2.1	Présentation	556
5.2.2	Les ressources à protéger	557
5.2.3	Les autorisations	559
5.2.4	Le schéma d'authentification	560
5.2.5	La base de comptes	563
5.3	Le chiffrement	568
5.3.1	La théorie	568
5.3.2	La mise en œuvre	569
5.4	L'utilisation du HTTP/2	576
5.4.1	Introduction	576
5.4.2	Mise en œuvre	576
5.4.3	Focus sur le push	577

6. La gestion des logs	581
6.1 Les logs d'accès	581
6.2 Les logs de l'activité logicielle	584
6.2.1 Le mécanisme	585
6.2.2 Le paramétrage	585
7. Conclusion	592
Index	593