1-

- Apache Lucene — de la fondation Apache Software ;
- Apache Struts — de la fondation Apache Software ;
- Apache Tapestry — de la fondation Apache Software ;
- Google Guava ;
- Hadoop — de la fondation Apache Software ;
- Hibernate ;
- JavaFX ;
- JavaServer Faces ;
- JUnit — pour les tests unitaires ;
- Leonardi - de la société W4 (GPL) ;
- Modular Audio Recognition Framework - open-source du MARF Research and Development Group ;
- Nuxeo - Framework open source dédié principalement à la gestion de contenus ;
- Play framework - Framework open source (licence Apache) développé par la société Zenexity ;
- Qt Jambi - Framework permettant de faire des interfaces Qt en Java ;
- Spring ;
- WaveMaker `(en)` - Framework open source (Licence Apache) de la société VMware ;
- Zerocouplage - Framework permettant de développer des applications web, des applications mobiles et/ou ordinateurs de bureau à partir d'une unique couche métier indépendante de la couche présentation.

- Un **framework Java est** un bloc de code prérédigé **qu**'utilisent les développeurs pour créer des applications avec le langage de programmation **Java**. Les **frameworks Java** sont propres au langage de programmation **Java**. Il s'agit de plateformes **Java** destinées au développement d'applications logicielles et de programmes **Java**
- **Spring** permet de développer des applications d'entreprise en s'appuyant sur des simples POJO, il n'a pas besoin d'un conteneur d' EJB (au sein d'un serveur d'applications) mais uniquement d'un conteneur de Servlet robuste tel que Tomcat
- Spring. Spring est un **framework** Modèle-Vue-Contrôleur qui utilise **Java**, le langage le plus populaire. ... Le fait qu'il utilise **Java**, un langage fortement typé, est un avantage certain pour de nombreux développeurs Web
- **Spring** est un Framework qui, à l'origine, est créé pour permettre d'accélérer le développement d'applications Java EE; ce framework est modulaire et cela lui confère une certaine facilité pour la conception et la réalisation d'applications java E.E.
- Les modules de base de Spring (pour le développement d'application Java E.E) **sont au nombre de 20**

2

. Spring. Ce **framework est** au top du classement puisqu'il a une excellente capacité à développer des applications plus complexes qui doivent être performantes. Ainsi,

les développeurs **sont** capables de créer facilement des applications complètes pour de grosses entreprises

4-

Core. Ce module fournit les fonctionnalités de base de Spring. Il propose un conteneur IoC de base nommé BeanFactory. Les fonctionnalités de ce conteneur seront décrites au Chapitre 3.

n Context. Ce module se fonde sur le module Core. Il en étend les fonctionnalités et propose un conteneur IoC élaboré, nommé ApplicationContext, qui ajoute ses

Un aperçu des modules du framework Spring.

propres fonctionnalités, comme la prise en charge de l'internationalisation (I18N), la communication à base d'événements et le chargement de ressources. Les caractéristiques de ce conteneur IoC élaboré seront examinées au Chapitre 4.

n AOP. Ce module définit un framework de programmation orientée aspect appelé Spring AOP. Avec l'IoC, la programmation orientée aspect est un autre concept fondamental de Spring. Les Chapitres 5 et 6 présenteront les approches POA classiques et nouvelles de Spring, ainsi que l'intégration d'AspectJ avec Spring.

n JDBC. Ce module définit au-dessus de l'API JDBC native une couche abstraite qui permet d'employer JDBC avec des templates et donc d'éviter la répétition du code standard (*boilerplate code*). Il convertit également les codes d'erreur des fournisseurs de bases de données en une hiérarchie d'exceptions DataAccessException propres à Spring. Les détails de la prise en charge de JDBC par Spring seront donnés au Chapitre 7.

n TX. Ce module permet de gérer les transactions par programmation ou sous forme déclarative. Ces deux approches peuvent être employées pour ajouter des possibilités transactionnelles aux objets Java simples. La gestion des transactions sera étudiée au Chapitre 8.

ORM. Ce module intègre à Spring les frameworks classiques de correspondance objet-relationnel, comme Hibernate, JDO, TopLink, iBATIS et JPA. Tous les détails de l'intégration ORM de Spring se trouvent au Chapitre 9.

n Web MVC. Ce module définit un framework d'applications web conforme au design pattern Modèle-Vue-Contrôleur (MVC). Ce framework s'appuie sur les fonctionnalités de Spring pour qu'elles puissent servir au développement des applications web. Il sera examiné au Chapitre 10.

n Web Framework Integration. Ce module facilite l'utilisation de Spring en tant que support à d'autres frameworks web répandus, comme Struts, JSF, WebWork et Tapestry. Le Chapitre 11 s'intéressera à l'association de Spring avec plusieurs frameworks web.

n Testing. Ce module apporte la prise en charge des tests unitaires et des tests d'intégration. Il définit le framework Spring TestContext, qui rend abstraits les environnements de test sous-jacents comme JUnit 3.8, JUnit 4.4 et TestNG. Le test des applications Spring sera étudié au Chapitre 12.

n Portlet MVC. Ce module définit un framework de portlet, également conforme au design pattern MVC.

n Enterprise Integration. Ce module intègre à Spring de nombreux services d'entreprise répandus, notamment des technologies d'accès distant, les EJB, JMS, JMX, le courrier électronique et la planification, afin d'en faciliter l'usage.

Les versions de Spring

Deux ans et demi après la sortie du framework Spring 1.0 en mars 2004, la première mise à jour importante, Spring 2.0, a été publiée en octobre 2006. Voici ses principales améliorations et nouvelles fonctionnalités :

n Configuration à base de schéma XML. Dans Spring 1.x, les fichiers XML de configuration des beans n'acceptent que des DTD et toutes les définitions de caractéristiques doivent se faire au travers de l'élément <bean>. Spring 2.0 prend en charge la configuration à base de schéma XML, qui permet d'utiliser les nouvelles balises de Spring. Les fichiers de configuration des beans peuvent ainsi être plus simples et plus clairs. Dans ce livre, nous exploitons cette possibilité. Le Chapitre 3 en présentera les bases.

n Configuration à base d'annotations. En complément de la configuration basée sur XML, Spring 2.0 accepte dans certains modules une configuration à base d'annotations, comme @Required, @Transactional, @PersistenceContext et @PersistenceUnit. Les Chapitres 3, 8 et 9 expliqueront comment employer ces annotations.

Nouvelle approche de Spring AOP. L'utilisation classique de la programmation orientée aspect dans Spring 1.x se fait au travers d'un ensemble d'API propriétaires de Spring AOP. Spring 2.0 propose une toute nouvelle approche fondée sur l'écriture de POJO avec des annotations AspectJ ou une configuration à base de schéma XML. Le Chapitre 6 en donnera tous les détails.

n Déclaration plus aisée des transactions. Dans Spring 2.0, il est beaucoup plus facile de déclarer des transactions. La nouvelle approche de Spring AOP permet de déclarer des greffons (*advices*) transactionnels, mais il est également possible d'appliquer des annotations @Transactional avec la balise <tx:annotationdriven>. Le Chapitre 8 reviendra en détail sur la gestion des transactions.

n Prise en charge de JPA. Spring 2.0 ajoute la prise en charge de l'API de persistance Java dans son module ORM. Elle sera examinée au Chapitre 9.

n Bibliothèque pour la baliseform. Spring 2.0 propose une nouvelle bibliothèque pour faciliter le développement de formulaires dans Spring MVC. Son utilisation sera décrite au Chapitre 10.

n Prise en charge de JMS en mode asynchrone. Spring 1.x n'accepte que la réception synchrone de messages JMS par l'intermédiaire de JmsTemplate. Spring 2.0 ajoute la réception asynchrone de messages JMS au travers de POJO orientés message.

n Prise en charge d'un langage de script. Spring 2.0 reconnaît l'implémentation de beans à l'aide des langages de script JRuby, Groovy et BeanShell.

Novembre 2007 a vu la sortie de Spring 2.5, qui ajoute les fonctionnalités suivantes par rapport à Spring 2.0. Au moment de l'écriture de ces lignes, Spring 2.5 est la version courante du framework.

n Configuration à base d'annotations. Spring 2.0 a ajouté plusieurs annotations pour simplifier la configuration des beans. Spring 2.5 en reconnaît d'autres, notamment @Autowired et celles de la JSR 250, @Resource, @PostConstruct et @PreDestroy. Les Chapitres 3 et 4 expliqueront comment les utiliser.

n Scan de composants. Spring 2.5 peut détecter automatiquement les composants avec des annotations particulières situés dans le chemin d'accès aux classes, sans aucune configuration manuelle. Cette possibilité sera étudiée au Chapitre 3.

n Prise en charge du tissage AspectJ au chargement. Spring 2.5 prend en charge le tissage des aspects AspectJ dans le conteneur Spring IoC au moment du chargement. Il est ainsi possible d'utiliser des aspects AspectJ en dehors de Spring AOP.

*

5-

Le Front contrôleur traite les requêtes provenant de l'extérieur et de ce fait:

1. Il analyse l'URL de la requête,
2. Il appelle le contrôleur qui doit traiter la requête; ce dernier renvoie un objet de type Modèle et le nom logique de la page qui sert à construire la page à afficher,
3. Il appelle la page demandée et celle-ci construit la réponse
4. Il renvoie la réponse au client demandeur.

6-

Créer un contrôleur sous **Spring**

Pour chaque URI "/access-denied", quelque soit sa méthode d'appel (POST ou GET), on invoque la fonction accessDenied() de la classe IndexController. Cette fonction retourne le modèle de vue "accessDenied", défini dans une liste des modèles de vue gérée par Apache Tiles.

7-

Autrement dit, Spring Boot est une extension du framework Spring, qui mis à part les configurations standard requises pour la configuration d'une application Spring. Il adopte une vision de Spring, et a un éco-système de développement plus rapide et plus efficace.