Thème : Spring Framework
Développé Par :
DABA DIAWARA
Contact : 781265485
Adresse : Grand Dakar
Mail : medgnouma98@gmail.com

INTRODUCTION

Le Spring Framework est un outil très important permettant aux développeurs de simplifier l'écriture et rendre plus légère l'applications J2EE. Spring est reparti en plusieurs Module

- Gestion des instances de classes (JavaBean et/ou métier).
- Programmation orientée Aspect
- Modèle MVC et outils pour les application WEB
- Outils pour DAO (JDBC)
- Outils pour ORM (Hibernate, iBatis)
- Outils pour les applications J2EE (JMX, JMA, JCA, EJB)
- Conclusion

Vue d'ensemble du Spring Framework 5

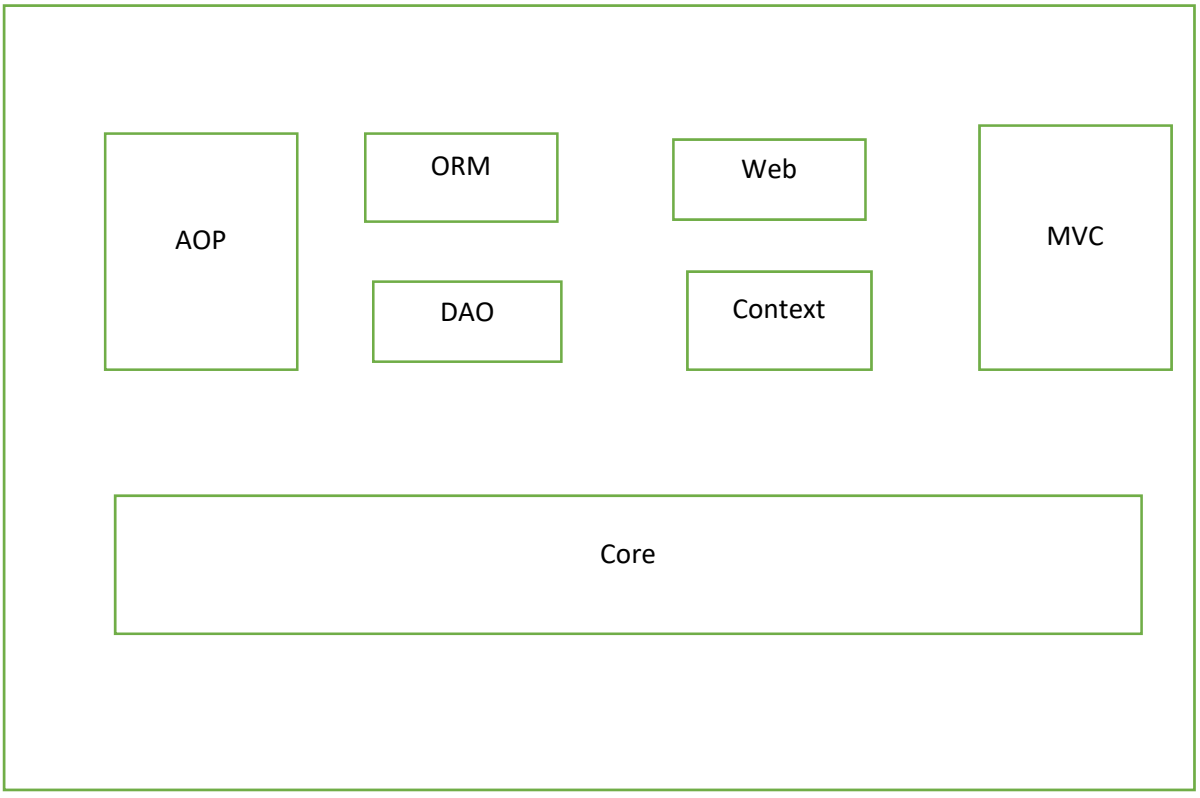
Le Spring Framework est un open source très qui permet aux développeurs d'entreprise d'être plus productif et durable basée

1) Origine et version d'architecture du Spring

- Architecture du Spring Framework

Le Spring Framework nécessite un certains nombres de fonctionnalités pour le développement d'une application d'entreprise. Cependant cela n'est pas obligatoires pour les développeurs d'intégrer leurs applications à un Framework complet.

Les développeurs peuvent choisir d'intégrer leur application à un ou plusieurs module Spring en fonction de la fonctionnalité qu'ils souhaitent utiliser dans leur application



Architecture du Framework Spring

Conteneur principal : le cœur est le conteneur le plus important du Framework Spring et tous les autres modules sont construits dessus.

AOP : ce module nous aide à implémenter les différentes préoccupations transversales dans une application comme la gestion de stock etc. Ces préoccupations sont découplées du code de l'application et sont injectées dans les différentes coupes de points à travers le fichier de configuration.

Spring Web : ce Spring permet a développé une application Web en fournissant le module Web Spring.

Ce module est construit sur le module de context d'application et fournit des fonctionnalités orientées Web

Spring MVC : il est construit sur le module Web Spring et aide à développer une application web basée sur le modèle de conception MVC

Spring DAO : toutes les application d'entreprise doivent intégrer avec la base de données le module Spring DAO facilite l'interaction avec la base de données en fournissant une abstraction sur les taches JDBC de bas niveau telles que la création d'une connexion à la base de données sa libération. Etc.

Spring ORM : il existe un certain nombre d'outils de cartographie relationnelle objet populaires tels que Hibernate, IBatis, etc. le modèle Spring ORM aide à s'intégrer à ces outils

- **ORIGINE ET VERSION DU SPRING FRAMEWORK**

Le Framework Spring a été créé par Rod Johnson et Juergen Holler en 2003.

Spring a connu plusieurs versions :

- Spring 1.0 : mars 2004
- Spring 1.1 : septembre 2004
- Spring 1.2 : mai 2005
- Spring 2.0 : octobre 2006
- Spring 2.5 : novembre 2007
- Spring 3.0 : décembre 2009
- Spring 3.1 : courant 2011

Spring 1.0 implémente les fonctionnalités de base du Framework :

- Le conteneur qui implémente le motif de conception Ioc
- Le développement orienté POJO
- l'AOP par déclaration
- Le support de JDBC, ORM et Framework Web

- la configuration XML basée sur une DTD

Spring 1.2

- Support de JMX
- support JDO 2, Hibernate 3, Top Link
- Support de JCA CCI, JDBC Rowset
- Déclaration des transactions avec @Transactional

Spring 3.1 :

- Support des conversations
- Support des caches
- Ajout de la notion de profile qui permet d'avoir des configurations du context différentes pour chaque environnement
- Ajout de nouvelles annotations pour définir certaines fonctionnalités de namespaces dans la configuration
- Support des servlets 3.0

- Spring 3.2.5 en novembre 2013.
- Spring Framework 4.0 a été publié en décembre 2013.

Les améliorations notables de Spring 4.0 comprenaient la prise en charge de Java SE (Standard Edition) 8, Groovy 2, certains aspects de Java EE 7 et WebSocket .

- Spring Framework 4.2.0 a été publié le 31 juillet 2015 et a été immédiatement mis à niveau vers la
- Version 4.2.1, qui a été publiée le 1er septembre 2015. [7] Il est « compatible avec Java 6, 7 et 8, en mettant l'accent sur les raffinements de base. et des capacités Web modernes" .
- Spring Framework 4.3 a été publié le 10 juin 2016 et sera pris en charge jusqu'en 2020. Il "sera la *dernière génération dans les exigences générales du système Spring 4 (Java 6+, Servlet 2.5+),*
- *Spring 5.0 2017*
- À partir de Spring Framework 5.1, Spring nécessite JDK 8+ (Java SE 8+) et fournit une prise en charge prête à l'emploi pour JDK 11 LTS. La mise à jour 60 de Java SE 8 est suggérée comme version de correctif minimale

pour Java 8, mais il est généralement recommandé d'utiliser une version de correctif récente.

1.1 L'INVERSION DE CONTROLE OU INJECTION DE DEPENDANCE

Une application de java entreprise se compose de plusieurs nombres de classes java(Père.java). Pour cela chaque classe java peut dépendre d'une ou plusieurs autres classes java. Ces autres classes sont appelées dépendances de la classe java Père. Généralement, chaque classe prend les références des classes dont elle dépend.

L'injection de dépendance est également connue sous le nom d'inversion de contrôle. Au lieu que la classe java obtienne ses dépendances à partir du conteneur, c'est le conteneur qui injecte les dépendances dans la classe java. Il y a donc une inversion de contrôle.

1.2 La programmation orienté Aspects (POA)

Simplement mis ensemble, le Framework Spring AOP détourne l'exécution du programme et injecte des fonctionnalités supplémentaires généralement avant, après ou autour de l'exécution de la méthode.

Prenons un exemple de l'aspect journalisation. L'exigence est de :

Consignez un message d'entrée avant d'exécuter le corps de la méthode

Consignez un message de sortie après avoir exécuté le corps du message

Enregistrez le temps nécessaire à la méthode pour terminer l'exécution.

Ici, la journalisation est connue sous le nom d'aspect. L'exécution de la méthode est connue sous le nom de *point de jointure.* Le morceau de code qui enregistre le message d'entrée, le message de sortie et le temps nécessaire pour exécuter la méthode sont appelés les trois *conseils.* La liste des méthodes pour lesquelles ce comportement est requis est connue sous le nom de *coupures de points.* Et enfin, les objets java sur lesquels cet aspect est appliqué sont connus sous le nom de *cibles.*

Il existe un certain nombre de conseils disponibles dans le cadre AOP. Ceux-ci sont :

1. *Avant conseil* - Exécuter avant le point de jointure (exécution de la méthode)
2. *Après avoir renvoyé un avis* - Exécutez si le point de jointure s'exécute normalement.
3. *Après avoir lancé un conseil* - Exécutez si le point de jointure lève une exception.
4. *Autour des conseils* - Exécuter autour du point de jointure (exécution de la méthode)

Ainsi, la journalisation du message d'entrée peut être implémentée par le conseil Avant, la journalisation du message de sortie peut être implémentée par le conseil Après et la journalisation du temps nécessaire à l'exécution de la méthode peut être implémentée à l'aide du conseil Around.

1.3 BILAN DES SOLUTIONS APPORTEES PAR SPRING

Le but de Spring est de faciliter et de rendre productif le développement d'applications, particulièrement les applications d'entreprises.

Spring propose de nombreuses fonctionnalités de base pour le développement d'applications :

Un conteneur léger implémentant le design pattern IoC pour la gestion des objets et de leurs dépendances en offrant des fonctionnalités avancées concernant la configuration et l'injection automatique. Un de ses points forts est d'être non intrusif dans le code de l'application tout en permettant l'assemblage d'objets faiblement couplés.

Une gestion des transactions par déclaration offrant une abstraction du gestionnaire de transactions sous-jacent

Faciliter le développement des DAO de la couche de persistance en utilisant JDBC, JPA, JDO ou une solution open source comme Hibernate, iBatis, ... et une hiérarchie d'exceptions

un support pour un usage interne à Spring (notamment dans les transactions) ou personnalisé de l'AOP qui peut être mis en oeuvre avec Spring AOP pour les objets gérés par le conteneur et/ou avec AspectJ

Faciliter la testabilité de l'application

Spring favorise l'intégration avec de nombreux autres Framework notamment ceux de type ORM ou web.

Une application typique utilisant Spring est généralement structurée en trois couches :

La couche présentation : interface homme machine

La couche service : interface métier avec mise en œuvre de certaines fonctionnalités (transactions, sécurité, ...)

La couche accès aux données : recherche et persistance des objets du domaine

1.4 Evolution de Spring

* L'évolution du printemps

Après avoir planté le décor et décrit les circonstances de la genèse de Spring Framework, nous passons à Spring Boot.

* Monolithe

Après être arrivés à l'état actuel de l'écosystème Spring, nous examinons pourquoi les équipes – disposant d'outils aussi puissants que Java, Spring (et d'autres Framework) – ont fini par créer de très grandes applications monolithiques.

* Micro services

Nous discutons brièvement de ce qu'est l'architecture des micro services et pourquoi la messagerie en tant que forme abstraite de communication entre des services individuels amène les développeurs et les architectes à réfléchir à la nature même de cette communication.

* Intégration de printemps

Les modèles d'intégration d'entreprise sont brièvement présentés comme la base de l'intégration Spring. Après cela, Spring Intégration est utilisé pour découpler trois composants d'application d'une manière qu'ils ne connaissent pas les uns des autres, mais ils échangent des messages à la place.

* Spring Cloud Stream

Dans la dernière partie, Spring Cloud Stream est présenté au public en tant que combinaison de Spring Intégration et Spring Boot. Nous faisons évoluer notre exemple d'application vers un ensemble de micro services axés sur les messages.

1.5 Les nouveautés de Spring Framework 5

Spring a sorti la première release candidate de la version 5 de son Framework il y a un peu plus d'un mois.

A l'heure où nous écrivons cet article, la version 5 de Spring est disponible en RC2. Une RC3 est prévue pour courant juillet, peu de temps avant la version finale.

Nous vous proposons par conséquent de découvrir ensemble les nouveautés et améliorations apportées à Spring Framework 5.

Passage à Java 8

Dans sa version 4, Spring était compatible avec Java 6. Avec la version 5, le code du Framework a été réécrit en Java 8.

Les développeurs en ont donc profité pour :

Améliorer l'accès aux paramètres d'une méthode en utilisant les améliorations de la réflexivité de Java 8

Utiliser les méthodes par défaut des interfaces de Spring

Utiliser les classes `Charset` et `StandardCharsets` (disponible depuis le JDK 7)

Et dans une optique de compatibilité avec Java 9, l'instanciation des classes se fera désormais par Constructor#newInstance au lieu de Class#newInstance qui va devenir obsolète.

Spring WebFlux

Spring se met à la programmation réactive avec WebFlux dans sa version 5.

Concrètement, il s'agit de pouvoir implémenter une application web (contrôleur REST) ou des clients HTTP de manière réactive. Pour ce faire, Spring 5 intègre désormais Reactor et permet de manipuler les objets Mono (qui contient un objet) et Flux (qui peut contenir N objet(s)).

Plutôt qu'un long discours, un peu de code :

Implémentation d'un contrôleur réactive