

Spring

Exposés logiciels, systèmes réseaux

Sommaire

- ▶ Introduction
- ▶ Historique
- ▶ Projets
- ▶ Modules
- ▶ Concepts
 - IoC
 - AOP
 - SpEL
- ▶ Spring ROO
- ▶ Conclusion
- ▶ Références

Introduction

- ▶ Pourquoi Spring ?
- ▶ « Maman, c'est quoi Spring ? »
 - Framework Java (EE) ++
 - Libre & open-source (licence Apache 2.0)
 - Conteneur léger
- ▶ Utilisation principale
 - Facilite la programmation avec des POJO



Historique

- ▶ 2004 : Création  – Rod Johnson
Version 1 de Spring Framework
- ▶ 2006 : Version 2 de Spring Framework
- ▶ 2009 : Rachat par VMWare
Version 3 de Spring Framework
- ▶ Aujourd'hui : Version 3.2 (13/12/2012)

Projets

Spring Framework

Spring is a layered Java application platform for building enterprise solutions and is based on code published in the book *Expert One-on-One J2EE Design and Development* by Rod Johnson. [Go to...](#)

Spring Mobile

Spring Mobile is an extension to Spring MVC that aims to simplify the development of mobile web applications including intelligent device detection and progressive rendering options. [Go to...](#)

Spring Batch

Spring Batch is a lightweight, comprehensive batch framework designed to simplify and optimize the work of processing high volume batch operations. [Go to...](#)

Spring Social

Spring Social is an extension of the Spring Framework that allows you to connect your applications with Software-as-a-Service (SaaS) providers such as Facebook and Twitter. [Go to...](#)

More Spring Projects

- [Spring GemFire](#)
- [Spring .NET](#)
- [Spring for Android](#)
- [Spring Security OAuth](#)
- [Spring LDAP](#)
- [Spring Data - Apache Hadoop](#)

Spring Security

Spring Security is a powerful and highly customizable authentication and access-control framework that works efficiently in Spring applications with just a few lines of configuration. [Go to...](#)

Spring Data

Spring Data makes it easier to build Spring-powered applications that use new data access technologies like MongoDB, Neo4J, Redis, and other noSQL data stores. [Go to...](#)

Spring Integration

Spring Integration extends Spring to support the well-known *Enterprise Integration Patterns* via lightweight messaging within Spring-based applications and supports integration with external systems via declarative adapters. [Go to...](#)

Spring Web Flow

Spring Web Flow focuses on providing the infrastructure for building and running rich web applications that require multi-step interactions. [Go to...](#)

- [Spring Payment Services](#)
- [SpringSource dm Server \(Eclipse Virgo\)](#)
- [SpringSource dm Server Tools](#)
- [SpringSource Bundlor](#)
- [Spring Dynamic Modules \(Eclipse Gemini\)](#)
- [Spring Extensions](#)

Spring Roo

Spring Roo is a next-generation rapid application development tool for Java developers. With Roo you can easily build in minutes full Java applications including data access, search and rich web interactions. [Go to...](#)

Spring Web Services

Spring Web Services focuses on creating document-driven Web services using a contract-first SOAP service development approach. [Go to...](#)

Spring AMQP

Spring AMQP applies core Spring concepts to the development of AMQP-based messaging solutions including abstractions for sending, receiving and configuring AMQP messages. [Go to...](#)

Spring BlazeDS Integration

Spring BlazeDS Integration makes it easier to build Spring-powered Rich Internet Applications using Adobe Flex as the front-end client. [Go to...](#)

- [Spring Shell](#)
- [Spring Rich Client](#)
- [Spring BeanDoc](#)
- [Spring JavaConfig \(deprecated\)](#)
- [Spring Modules \(deprecated\)](#)

Modules

- ▶ Le cœur de Spring Framework 3 est composé d'un ensemble d'une vingtaine de modules qui sont regroupés en plusieurs grandes parties :
 - **Spring Core Container** : regroupe les modules de base pour mettre en œuvre le conteneur
 - **AOP and Instrumentation** : permet de mettre en œuvre l'AOP
 - **Data Acces/Integration** : regroupe les modules d'accès aux données
 - **Web** : regroupe les modules pour le développement d'applications web
 - **Test** : propose des fonctionnalités pour les tests automatisés avec Spring

Concepts

- ▶ **IoC** : Inversion de Contrôle
 - Permet l'injection de dépendance
 - Gère le cycle de vies des beans
- ▶ **AOP** : Programmation Orientée Aspect
 - Permet de créer dynamiquement des proxys s'exécutant à des événements choisis
- ▶ **SpEL** : Spring Expression Language (Spring 3)
 - Annotations
 - Fichier de configuration

Concepts – IoC

- ▶ Par fichier de configuration

Voir la démo

Concepts – IoC

- ▶ Par annotations
 - **@Scope** : singleton, prototype, session ou request
 - **@Autowired** : injection automatique par type
 - **@Required** : valide une injection de dépendances
 - **@Qualifier** : donne un nom si plusieurs instances
 - **@Resource** : injection automatique par nom
 - **@Configurable** : injecte les dépendances d'un bean bien que son conteneur ne gère pas son cycle de vie
 - **@PostConstruct** : appelle de méthode à l'instanciation
 - **@PreDestroy** : appelle de méthode à la destruction de l'instance
 - ...

Concepts – AOP

- ▶ Par fichier de configuration

Voir la démo

Concepts – AOP

- ▶ Par annotations
- ▶ **@Before** : le code est exécuté avant l'exécution de la méthode
- ▶ **@AfterReturning** : le code est exécuté après l'exécution de la méthode qui renvoie une valeur de retour (aucune exception n'est levée)
- ▶ **@AfterThrowing** : le code est exécuté lorsqu'une exception est levée suite à l'invocation de la méthode
- ▶ **@After** : le code est exécuté après l'exécution de la méthode, même si une exception est levée.
- ▶ **@Around** : le code de l'aspect permet de lancer l'exécution de la méthode et ainsi de réaliser des traitements avant et des traitements après

Concepts – SpEL

▶ Fichier de configuration

```
<bean id="monDe" class="fr.univ.mlv.expose.De">  
    <property name="valeur" value="#{T(java.lang.Math).random()}" />  
    <property name="proprietaire" value="#{moi.prenom}" />  
</bean>
```

▶ Annotations

```
@Value("#{ systemProperties['user.name'] }")  
public void setUsername(final String userName) {  
    this.userName = userName;  
}
```

Spring ROO



- ▶ Outil de SpringSource
- ▶ Création en 2009 – Version actuelle : 1.2.2
- ▶ Plusieurs particularités
 - Capable de monter rapidement une application Web Java
 - Dit « fullstack » qui s’occupe de toutes les couches d’une l’application
 - Gère le code rébarbatif

Spring ROO

- ▶ Comment créer rapidement un projet ?

Voir la démo

Conclusion

- ▶ Les plus
 - + Standard
 - + Intégration avec d'autres Frameworks
 - + Communauté importante

SPRING KEY BENEFITS

			
Modularity	Productivity	Portability	Testability
Plain old Java Objects keep your code concise, simple and modular	Over 70% of developers report productivity gains and reduction in time to deploy with Spring	Applications run on Tomcat, all Java EE servers as well as cloud platforms	Cleanly expressed dependencies make unit and integration testing easier

- ▶ Les moins
 - Application un peu lourde
 - Mise en œuvre compliquée

Références

▶ Webographie

- <http://www.springsource.org/>
- <http://static.springsource.org/spring/docs/3.0.x/javadoc-api/overview-summary.html>

▶ Bibliographie

- *Spring par la Pratique*, Arnaud Cogoluègnes
- *Expert One-on-One J2EE Design and Development*, Rod Johnson

Questions ?

