



semifir

Qu'est ce que spring

Qu'est ce que spring

Un framework Java open source





Qu'est ce que Spring

- Une librairie pour l'**injection de dépendance**.
- Une librairie pour l'**inversion de contrôle**.



Qu'est ce que Spring

- Un conteneur pour la gestion de cycle de vie des objets.

On le nomme aussi conteneur **léger**.

Conteneur léger

- Un conteneur léger est un conteneur qui ne nécessite pas de serveur d'application.
 - Il est autonome et peut être exécuté en tant qu'application autonome.
-

Conteneur léger

- Son infrastructure est légère, flexible et ressemble à celle d'un serveur d'application J2EE.
 - Il est facile à configurer et à exécuter.
-

Conteneur léger

- Il prend en charge la création d'objets
- La gestion des objets

- La destruction des objets
-

On ce retrouve alors avec des classes qui non pas besoin d'implémenter des interfaces spécifiques.

Historique

Quelque date

- Créé par Rod Johnson en 2003
 - Spring Framework 1.0 en 2004
 - Spring Framework 2.0 en 2006
-

Quelque date

- Spring Framework 3.0 en 2009
 - Spring Framework 4.0 en 2013
 - Spring Framework 5.0 en 2017
-

Raison de sa création

- Simplifier le développement d'applications Java
 - Rendre le développement d'applications Java plus rapide et plus facile
 - Réduire la complexité du développement d'applications Java
-

Raison de sa création

- Spring est apparue dans un contexte où le développement d'application Java était complexe.
 - Son objectif était de simplifier le développement d'applications Java.
 - Il n'est pas là pour être en concurrence avec les autres frameworks(Java EE / Jakarta EE) mais pour les compléter.
-

Aujourd'hui

- Spring est devenu un écosystème complet avec ses différents modules.
 - Il est utilisé par de nombreuses entreprises pour le développement d'applications Java.
-

Aujourd'hui

Ses modules les plus connus sont :

- Spring Boot
 - Spring AOP
 - Spring Data
 - Spring Security
-

Aujourd'hui

- Avec sa version 6 Spring intègre maintenant le support de Java 17 et de Jakarta EE 9
 - Il est complètement compatible avec les dernières versions de serveur Tomcat, Jetty et Undertow
 - Ainsi que les dernières versions de base de données et de l'ORM Hibernate
-

Cas d'utilisation

Cas d'utilisation

Les cas les plus courants de sont utilisation sont :

- Développement d'applications d'entreprise
 - Très utilisé pour faire des application scalable et robuste
 - Ainsi que pour la gestion des transactions
-

Cas d'utilisation

- Développement d'applications web
 - Spring MVC est très populaire car flexible et facile à utiliser
 - Choisi souvent dans le cas des interfaces REST
-

Cas d'utilisation

- Développement en microservices
 - Spring Boot est très utilisé pour le développement de microservices
 - Il est facile à configurer et à exécuter
-

Cas d'utilisation

- Acces aux données
 - Spring Data est très utilisé pour l'accès aux données

- Il permet d'implémenter facilement les opérations CRUD
 - Ainsi que la norme JPA
-

Cas d'utilisation

- Sécurité
 - Spring Security est très utilisé pour la sécurité
 - Il permet de sécuriser les applications web
 - Maintenue très régulièrement à jour
-

Modules les plus connus

Spring boot

Spring boot est une surcouche de Spring.

Il permet de démarrer rapidement un projet Spring en fournissant des configurations par défaut.

Spring vs Spring boot

Spring ne fournit que les composants de base.

Il faut entièrement configurer le projet ce qui peut être fastidieux.

Spring boot fournit des configurations par défaut.

Il permet de rapidement créer un projet web mais pas que.

Spring boot

Spring Boot propose un ensemble de modules pré-configurés.

Voici quelques modules :

- Spring Web : pour créer des applications web
 - Spring Data : pour accéder à des bases de données
 - Spring Security : pour sécuriser les applications
 - Spring Cloud : pour créer des applications distribuées
-

Spring initializr



Création d'un projet Spring boot

Il existe plusieurs façons de créer un projet **Spring boot**:

- A la main (long et fastidieux)
 - Avec un IDE (IntelliJ, Eclipse)
 - Avec le site [Spring initializr](#)
-

Création d'un projet Spring boot avec Spring initializr

1. Rendez-vous sur le site [Spring initializr](#)
 2. Sélectionnez les dépendances que vous souhaitez utiliser
 3. Téléchargez le projet
 4. Décompressez le projet
 5. Importez le projet dans votre IDE
-

Création d'un projet Spring boot avec IntelliJ

Il vous faut **IntelliJ IDEA Ultimate** ou **Spring Assistant** pour créer un projet **Spring boot** avec IntelliJ.

1. Ouvrez IntelliJ
2. Cliquez sur **Create New Project**

3. Sélectionnez **Spring Initializr**
 4. Sélectionnez les dépendances que vous souhaitez utiliser
 5. Cliquez sur **Next**
 6. Entrez le nom du projet
 7. Cliquez sur **Finish**
-

Création d'un projet Spring boot avec Eclipse

Il vous faut **Spring Tools 4** pour créer un projet **Spring boot** avec Eclipse.

1. Ouvrez Eclipse
 2. Cliquez sur **File** puis **New** puis **Spring Starter Project**
 3. Sélectionnez les dépendances que vous souhaitez utiliser
 4. Cliquez sur **Next**
 5. Entrez le nom du projet
 6. Cliquez sur **Finish**
-

Création d'un projet spring boot avec le CLI

Il vous faut télécharger le CLI depuis ce [lien](#)

1. Extrayez l'archive
 2. Ouvrez le fichier INSTALL.txt
 3. Suivez les instructions
 4. Une fois le CLI installé, ouvrez un shell
 5. Entrez cette commande -> `spring init --build=maven --java-version=17 --dependencies=web,data-jpa my-project`
-

Spring boot et Maven

Spring boot utilise **Maven** pour gérer les dépendances.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.5</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>kira.formation</groupId>
  <artifactId>spring.boot</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>spring.boot</name>
```

```
<description>Demo project for Spring Boot</description>
<properties>
    <java.version>17</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>
```

Maven, c'est quoi ?

Outils permettant la gestion de projet en java

Maven permet

- Décrire le projet (liste des caractéristiques, numéros de version...)
 - Lister des dépendances
 - Précise la version du jdk utilisée
-

Pour utiliser maven, il vous un fichier nommer **POM.xml**

```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.example</groupId>
    <artifactId>ProjetMaven</artifactId>
```

```
<version>1.0-SNAPSHOT</version>
<name>Archetype - ProjetMaven</name>
<url>http://maven.apache.org</url>
</project>
```

Parent POM

Spring boot permet de simplifier la configuration de Maven en utilisant un parent POM.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.5</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  ...
</project>
```

Parent POM version

Dans cette configuration, la version du parent POM est 2.7.5.

En changeant la version du parent POM, on change la version de Spring boot et de ses dépendances.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.5</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  ...
</project>
```

Dépendances

Pour ajouter une dépendance spring boot à votre projet, il vous suffit de l'ajouter dans la section **dependencies**.

```
<?xml version="-1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.5</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>kira.formation</groupId>
  <artifactId>spring.boot</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>spring.boot</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

Build

Spring boot permet de construire un jar avec toutes les dépendances facilement.

```
<?xml version="-1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.5</version>
    <relativePath/> 
  </parent>
  <groupId>kira.formation</groupId>
  <artifactId>spring.boot</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>spring.boot</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

Spring AOP

Spring AOP est un module de Spring qui permet de mettre en place l'Aspect Oriented Programming.

Spring AOP permet de séparer les préoccupations transversales de l'application.

Premier projet

Pour lancer un projet Spring boot, il suffit de créer une classe avec la méthode main et d'ajouter l'annotation @SpringBootApplication.

```
@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

}
```

Spring boot, un serveur web?

Spring boot n'est pas un serveur web. Spring boot ne sait rien faire par défaut. Il faut lui ajouter des dépendances pour lui permettre de faire des choses.

Pour faire du web vous pouvez ajouter la dépendance spring-boot-starter-web.

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Spring boot, connexion à une base de données?

Spring boot ne sait pas se connecter à une base de données.

Il y a plusieurs dépendances pour se connecter à une base de données:

- spring-boot-starter-jdbc: connecter à une base de données relationnelle avec JDBC.
- spring-boot-starter-data-jpa: connecter à une base de données relationnelle avec JPA.
- spring-boot-starter-data-mongodb: connecter à une base de données NoSQL avec MongoDB.
- spring-boot-starter-data-redis: connecter à une base de données NoSQL avec Redis.
- spring-boot-starter-data-elasticsearch: connecter à une base de données NoSQL avec Elasticsearch.
- ...

Comparaison avec les autres frameworks

Spring vs Java EE / Jakarta EE

Spring et Java EE / Jakarta EE sont deux frameworks Java pour le développement d'applications d'entreprise.

Java EE / Jakarta EE Avantages

- Standardisé
 - Complet, scalable et robuste
 - Fonctionnel sur plusieurs plateformes
 - Support la portabilité cross-plateforme
-

Java EE / Jakarta EE Inconvénients

- Lourd
 - Long pour le développement
 - Demande beaucoup de temps pour le déploiement et la maintenance
 - Beaucoup de configuration
-

Spring Avantages

- Léger
 - Facile à configurer et utilisé
 - Facile à maintenir et à déployer
 - Utilise la base des POJO
 - Implémentation rapide de JPA et de la sécurité
-

Spring Inconvénients

- Pas standardisé et encore en évolution
- Demande beaucoup de base Java pour être utilisé
- Propose beaucoup de fonctionnalités qui ne sont pas toujours utiles et peuvent être source de confusion