

Java Spring Boot

Les dépendances et les plugins

Par Rafael Da Silva Mesquita



+

Maven™



Qu'est ce que Java Spring Boot ?

Java Spring est un framework de développement pour Java, assez populaire, conçu pour simplifier la création d'applications d'entreprise. Il fournit une infrastructure complète de support pour développer des applications Java. Spring permet aussi aux développeurs de créer des applications de haute performance, facilement testables, réutilisables et bien structurées.

Prérequis pour mettre en place les dépendances et les plugins :


Pour utiliser les dépendances et les plugins, les utilisateurs doivent satisfaire à certains prérequis de base.

1. Les utilisateurs doivent avoir les outils nécessaires à ceux pour la création du projet Spring, c'est-à-dire avoir JDK (Java Development Kit) installé sur sa machine, avoir un IDE adapté à Spring Boot ainsi qu'un gestionnaire de build tel que Maven ou Gradle.
2. Il est aussi essentiel d'avoir au préalable, une compréhension solide du langage et une certaine affinité avec le framework. Ceci permet le discernement de son fonctionnement interne, ainsi que de comprendre les principes de gestion des dépendances.
3. Une compréhension de base de la structure et de la syntaxe des fichiers pom.xml (pour Maven) et build.gradle (pour Gradle) est nécessaire pour ajouter, exclure ou mettre à jour des dépendances ainsi que des plugins.

Les dépendances :

Qu'est ce que c'est ?

Les **dépendances** sont des **bibliothèques** externes nécessaires à votre projet pour fonctionner correctement. Dans un projet Spring Boot, ces **dépendances** incluent souvent des **frameworks**, des **bibliothèques de support**, des **outils**, etc...



À quoi servent-elles ?

Les **dépendances** fournissent des fonctionnalités supplémentaires et des fonctionnalités prêtes à l'emploi à votre application. Par exemple, la **dépendance** Spring Web fournit des fonctionnalités pour créer des applications web avec Spring Boot.

Comment les mettre en place ?

Les **dépendances** sont spécifiées dans le fichier de configuration de votre projet : '**pom.xml**' ou '**build.gradle**'. Vous pouvez les ajouter manuellement ou les sélectionner lors de la création du projet en utilisant des outils comme Spring Initializr.

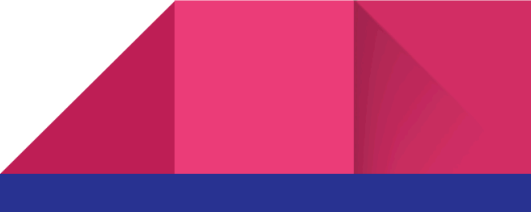
Les plugins :

Qu'est-ce que c'est ?

Les **plugins** sont des outils qui étendent les fonctionnalités de votre gestionnaire de build (Maven ou Gradle). Ils peuvent être utilisés pour exécuter des tâches spécifiques, comme compiler le code, générer des rapports, exécuter des tests, etc...

À quoi servent-elles ?

Les **plugins** simplifient le processus de construction, de test et de déploiement de votre application. Par exemple, le **plugin** Spring Boot Maven facilite la création d'un exécutable JAR pour votre application Spring Boot.



Comment les mettre en place ?

Les **plugins** sont également spécifiés dans votre fichier de configuration de build ('**pom.xml**' ou '**build.gradle**'). Vous pouvez les ajouter en déclarant leurs identifiants et configurations dans le fichier de build correspondant.

Mettre en place les dépendances et les plugins :

1. Identifiez les **dépendances** requises : décidez quelles **dépendances** sont nécessaires pour votre projet Spring Boot. Par exemple, **Spring Web**, **Spring Data JPA**, **Spring Security**, etc...
2. Ajoutez les **dépendances** dans le fichier de configuration de build :
 - Pour Maven ('**pom.xml**') : ajoutez les dépendances sous la balise '**<dependencies>**'.
 - Pour Gradle ('**build.gradle**') : ajoutez les dépendances sous la section '**dependencies**'.

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

Exemple des dépendances dans la balise **<dependencies>** dans le fichier '**pom.xml**'

3. Configurez les **Plugins** :

- Pour Maven : configurez les **plugins** nécessaires sous la balise '`<build><plugins>`'.
- Pour Gradle : configurez les **plugins** sous la section '`plugins`'

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

Exemple des plugins dans la balise `<build><plugins>` dans le fichier 'pom.xml'

4. Synchronisez et téléchargez les **dépendances** :

- Dans Maven : exécutez '`mvn clean install`' ou utilisez votre IDE pour synchroniser les **dépendances**.
- Dans Gradle : exécutez '`./gradlew build`' pour télécharger les **dépendances**.

5. Utilisez les **dépendances** et les **plugins** dans votre projet : Maintenant que les **dépendances** et les **plugins** sont configurés, vous pouvez commencer à utiliser les fonctionnalités qu'ils fournissent dans votre application Spring Boot.

Source :

<https://docs.spring.io/spring-boot/docs/current/maven-plugin/reference/htmlsingle/>

<https://docs.spring.io/spring-boot/docs/2.6.1/reference/htmlsingle/>

