

Java Spring Boot

Comment créer son projet

Par Rafael Da Silva Mesquita



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 créer un projet Java Spring Boot :

Pour créer leur projet, les utilisateurs doivent satisfaire à certains prérequis de base.

1. Les utilisateurs doivent avoir Java JDK installé sur leur système. Spring Boot prend en charge différentes versions de Java, mais il est recommandé d'utiliser Java 8 ou supérieur pour profiter des dernières fonctionnalités.
2. Il est nécessaire d'avoir un gestionnaire de build tel que Maven ou Gradle pour gérer les dépendances et construire votre projet. Spring Initializr génère des projets compatibles avec Maven ou Gradle, il faut donc s'assurer d'avoir l'un de ces outils installés.
3. Les utilisateurs ont la possibilité de développer des applications Spring Boot à l'aide de n'importe quel éditeur de texte, néanmoins il est fortement recommandé d'utiliser un IDE tel que Eclipse, IntelliJ, ou bien VS Code pour une meilleure productivité.
4. Il est aussi important d'avoir une certaine connaissance de base en Java pour développer des applications Spring Boot, car Spring est un framework Java.

Pour créer un projet, il existe plusieurs façons de démarrer avec Spring Boot, mais l'une des méthodes les plus simples et les plus courantes est d'utiliser Spring Initializr, un outil en ligne permettant de générer un projet de base.

Étape 1 : Utiliser Spring Initializr

Aller sur Spring Initializr (le liens : <https://start.spring.io/>) et remplissez le formulaire avec les détails de votre projet :

The screenshot shows the Spring Initializr web interface. At the top left is a hamburger menu icon. The logo "spring initializr" is in the top left. The interface is divided into several sections:

- Project:** Radio buttons for Gradle - Groovy, Gradle - Kotlin, and Maven.
- Language:** Radio buttons for Java, Kotlin, and Groovy.
- Spring Boot:** Radio buttons for 3.3.0 (SNAPSHOT), 3.3.0 (M3), 3.2.5 (SNAPSHOT), 3.2.4, 3.1.11 (SNAPSHOT), and 3.1.10.
- Project Metadata:** Text input fields for Group (com.example), Artifact (demo), Name (demo), Description (Demo project for Spring Boot), and Package name (com.example.demo).
- Packaging:** Radio buttons for Jar and War.
- Java:** Radio buttons for 22, 21, and 17.

- **Project** : choisissez le type de projet, Maven ou Gradle. Maven est plus communément utilisé et est un excellent choix si vous n'avez pas de préférence.
- **Languages** : sélectionnez le langage de programmation, Java est le plus couramment utilisé avec Spring Boot.
- **Spring Boot** : choisissez la version de Spring Boot. La version par défaut est généralement la dernière version stable.
- **Project Metadata** : complétez les champs comme Group, Artifact, Name, Description, Package name. Ces informations définissent la structure de base de votre projet et de son module.
- **Packaging** : sélectionnez Jar ou War. Jar est recommandé pour la plupart des projets Spring Boot.
- **Java** : choisissez la version de Java, et assurez-vous que cela correspond à la version de Java installée sur votre machine.

Ensuite vous pouvez aussi inclure des dépendances à votre projet. Pour un projet simple, vous pouvez commencer avec "Spring web" pour créer une application web.

Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Puis cliquez sur le bouton "Generate" pour télécharger un fichier zip contenant le squelette de votre projet.

Étape 2 : Ajout des fichiers

Extrayez le fichier zip que vous avez téléchargé, et ouvrez le sur l'IDE de votre choix (Eclipse, IntelliJ, VS Code, etc). Importez le projet :

- **Eclipse** : File > Import > Maven > Existing Maven Projects
- **IntelliJ** : Open > Sélectionnez le dossier du projet > Open
- **VS Code** : Ouvrez le dossier du projet, puis installez l'extension Spring Boot Extension Pack si ce n'est pas déjà fait.

Étape 3 : Explorer le projet

Examinez la structure du projet, les éléments clés à noter incluent :

- `'src/main/java'` : contient le code source de votre application.
- `'src/main/resources'` : ressources de l'application, comme les fichiers de configuration.
- `'src/test/java'` : contient les tests de votre application.
- `'pom.xml'` (pour Maven) ou `'build.gradle'` (pour Gradle): Fichier de configuration de votre projet qui inclut les dépendances et plugins.

Étape 4 : Création d'un "Hello World"

Créez un nouveau contrôleur dans le dossier '`src/main/java`' sous le package que vous avez spécifié lors de la création du projet. Voici un exemple simple :

```
package com.example.demo;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {

    @GetMapping("/hello")
    public String sayHello() {
        return "Hello, World!";
    }
}
```

Étape 5 : Exécutez l'application

Il existe deux moyens différents :

- **Dans l'IDE** : recherche la classe principale générée par Spring Initializr ('`DemoApplication.java`' ou similaire), faites un clic droit et choisissez "Run".
- **En ligne de commande** : naviguez dans le dossier du projet et exécutez '`mvnw spring-boot:run`' pour Maven ou '`./gradlew bootRun`' pour Gradle.

Votre application Spring Boot démarre sur le port 8080 par défaut. Ouvrez un navigateur et allez sur '<http://localhost:8080/hello>' pour voir le résultat de votre contrôleur.



Pour plus d'informations :

<https://spring.io/guides/gs/spring-boot>

Source :

<https://spring.io/quickstart>

