



# QU'EST CE QUE SPRING 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.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven..
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.7.5</version>
9     <relativePath/> <!-- lookup parent from repository -->
10  </parent>
11  <groupId>kira.formation</groupId>
12  <artifactId>spring.boot</artifactId>
13  <version>0.0.1-SNAPSHOT</version>
14  <name>spring.boot</name>
15  <description>Demo project for Spring Boot</description>
16  <properties>
```

# 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**

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apa
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>org.example</groupId>
6   <artifactId>ProjetMaven</artifactId>
7   <version>1.0-SNAPSHOT</version>
8   <name>Archetype - ProjetMaven</name>
9   <url>http://maven.apache.org</url>
10 </project>
```

# PARENT POM

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

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven..
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.7.5</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    ...
12
13 </project>
```

# PARENT POM

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

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven..
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.7.5</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    ...
12
13 </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.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven..
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.7.5</version>
9     <relativePath/> <!-- lookup parent from repository -->
10  </parent>
11  ...
12
13 </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.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven..
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.7.5</version>
9     <relativePath/> <!-- lookup parent from repository -->
10  </parent>
11  ...
12
13 </project>
```

# DÉPENDANCES

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

```
1 <?xml version="-1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven..
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.7.5</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>kira.formation</groupId>
12    <artifactId>spring.boot</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>spring.boot</name>
15    <description>Demo project for Spring Boot</description>
16    <properties>
```

# BUILD

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

```
1 <?xml version="-1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.7.5</version>
9     <relativePath/> <!-- lookup parent from repository -->
10  </parent>
11  <groupId>kira.formation</groupId>
12  <artifactId>spring.boot</artifactId>
13  <version>0.0.1-SNAPSHOT</version>
14  <name>spring.boot</name>
15  <description>Demo project for Spring Boot</description>
16  <properties>
```

# PREMIER PROJET

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

```
1
2 @SpringBootApplication
3 public class Application {
4
5     public static void main(String[] args) {
6         SpringApplication.run(Application.class, args);
7     }
8
9 }
```

# PREMIER PROJET

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

```
1
2 @SpringBootApplication
3 public class Application {
4
5     public static void main(String[] args) {
6         SpringApplication.run(Application.class, args);
7     }
8
9 }
```

# 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.
-



# LA SUITE

03-Architecture-backend