

Spring Boot

- Si vous êtes amenés à travailler sur une application développée autour de l'architecture Microservices en Java, vous aurez tôt ou tard affaire à Spring Boot.
- Spring Boot est un framework qui facilite le développement d'applications fondées sur Spring en offrant des outils permettant d'obtenir une application packagée en jar, totalement autonome

Ce que propose Spring Boot

- L'auto-configuration

Cette fonctionnalité est la plus importante de Spring Boot. Elle permet de **configurer automatiquement** votre application à partir des *jar* trouvés dans votre Classpath.

Exemple :

Prenons l'exemple d'une application web dans laquelle vous avez les dépendances : Hibernate et Spring MVC. Normalement, vous devez créer les fichiers de configuration

suivants :

- appconfig-mvc.xml
- web.xml
- persistance.xml

Ce que propose Spring Boot

- Comme vous ne connaissez pas nécessairement la syntaxe de ces fichiers par cœur, il vous faut consulter la documentation ou vous inspirer d'un ancien projet. Vous devez ensuite écrire le code Java qui permet de lier ces fichiers XML à *ApplicationContext* de Spring

Voici l'équivalent de l'ensemble de ces étapes avec Spring MVC :

```
1 @EnableAutoConfiguration
```

Les Starters

- Les starters viennent compléter l'auto-configuration et font gagner énormément de temps, notamment lorsqu'on commence le développement d'un Microservice.

Prenons l'exemple où vous souhaitez créer un Microservice. En temps normal, vous aurez besoin des dépendances suivantes :

- Spring ;
- Spring MVC ;
- Jackson (pour json) ;
- Tomcat ;
- ...

Avec Spring Boot, vous allez tout simplement avoir une seule dépendance dans votre **pom.xml** :

html

```
1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-web</artifactId>
4 </dependency>
```



Tous les starters de Spring Boot sont au format *spring-boot-starter-*

NOM_DU_STARTER

Ce starter va charger les dépendances présentes dans le pom suivant :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <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/xsd/maven-
   4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <parent>
5     <groupId>org.springframework.boot</groupId>
6     <artifactId>spring-boot-starters</artifactId>
7     <version>1.5.9.RELEASE</version>
8   </parent>
9   <artifactId>spring-boot-starter-web</artifactId>
10  <name>Spring Boot Web Starter</name>
11  <description>Starter for building web, including RESTful, applications using Spring
12    MVC. Uses Tomcat as the default embedded container</description>
13  <url>http://projects.spring.io/spring-boot/</url>
14  <organization>
15    <name>Pivotal Software, Inc.</name>
16    <url>http://www.spring.io</url>
17  </organization>
18  <properties>
19    <main.basedir>${basedir}/...</main.basedir>
20  </properties>
21  <dependencies>
22    <dependency>
23      <groupId>org.springframework.boot</groupId>
24      <artifactId>spring-boot-starter</artifactId>
25    </dependency>
26    <dependency>
27      <groupId>org.springframework.boot</groupId>
```

```
28 <artifactId>spring-boot-starter-tomcat</artifactId>
29 </dependency>
30 <dependency>
31 <groupId>org.hibernate</groupId>
32 <artifactId>hibernate-validator</artifactId>
33 </dependency>
34 <dependency>
35 <groupId>com.fasterxml.jackson.core</groupId>
36 <artifactId>jackson-databind</artifactId>
37 </dependency>
38 <dependency>
39 <groupId>org.springframework</groupId>
40 <artifactId>spring-web</artifactId>
41 </dependency>
42 <dependency>
43 <groupId>org.springframework</groupId>
44 <artifactId>spring-webmvc</artifactId>
45 </dependency>
46 </dependencies>
47 </project>
```

Name	Description
<code>spring-boot-starter</code>	Core starter, including auto-configuration support, logging and YAML
<code>spring-boot-starter-activemq</code>	Starter for JMS messaging using Apache ActiveMQ
<code>spring-boot-starter-amqp</code>	Starter for using Spring AMQP and Rabbit MQ
<code>spring-boot-starter-aop</code>	Starter for aspect-oriented programming with Spring AOP and AspectJ
<code>spring-boot-starter-artemis</code>	Starter for JMS messaging using Apache Artemis
<code>spring-boot-starter-batch</code>	Starter for using Spring Batch
<code>spring-boot-starter-cache</code>	Starter for using Spring Framework's caching support
<code>spring-boot-starter-cloud-connectors</code>	Starter for using Spring Cloud Connectors which simplifies connecting to services in cloud platforms like Cloud Foundry and Heroku
<code>spring-boot-starter-data-cassandra</code>	Starter for using Cassandra distributed database and Spring Data Cassandra
<code>spring-boot-starter-data-cassandra-reactive</code>	Starter for using Cassandra distributed database and Spring Data Cassandra Reactive
<code>spring-boot-starter-data-couchbase</code>	Starter for using Couchbase document-oriented database and Spring Data Couchbase
<code>spring-boot-starter-data-couchbase-reactive</code>	Starter for using Couchbase document-oriented database and Spring Data Couchbase Reactive