



# SpringBoot Starters

# Problématique

- Le développement d'application Spring pose systématiquement la question de savoir de quelles dépendances va t on avoir besoin ?
- Quelles dépendances dois-je inscrire dans le pom.xml ?
- Ex pour une application spring Mvc + Hibernate : quelle est la liste des dépendances nécessaires ?

# Problématique

```
<dependencies>
<!-- Spring -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>5.1.2.RELEASE</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>5.1.2.RELEASE</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>5.1.2.RELEASE</version>
</dependency>
```

```
<!-- Spring Security -->
<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-web</artifactId>
  <version>5.1.1.RELEASE</version>
</dependency>
<!-- Hibernate ORM -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.2.16.Final</version>
</dependency>
<!-- Hibernate Validator -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>6.0.13.Final</version>
</dependency>
```



# La simplification

- Springboot starters sont des listes organisées de dépendances maven,
- ces collections de librairies regroupent des versions compatibles ensembles,
- Cette compatibilité est testée et vérifiée, par les équipes de développement de Spring

# Exemple : Spring MVC

```
<!-- Spring MVC -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>5.1.2.RELEASE</version>
</dependency>

<!-- Form validation :Hibernate Validator -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>6.0.13.Final</version>
</dependency>

<!-- Web template : JSP ou thymleaf etc -->
<dependency>
  <!--... etc -->
</dependency>
```

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

Contient  
Spring-web, spring-webmvc,  
hibernate-validator, json, tomcat...

# Starter dans spring initializr

- Vous sélectionnez les starter à inclure dans le pom.xml depuis l'interface web de génération du projet au moment de la génération.
- Il existe environ une trentaine de starters :  
Web, JPA, thymleaf Security...

lien:

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#using-boot-starter>

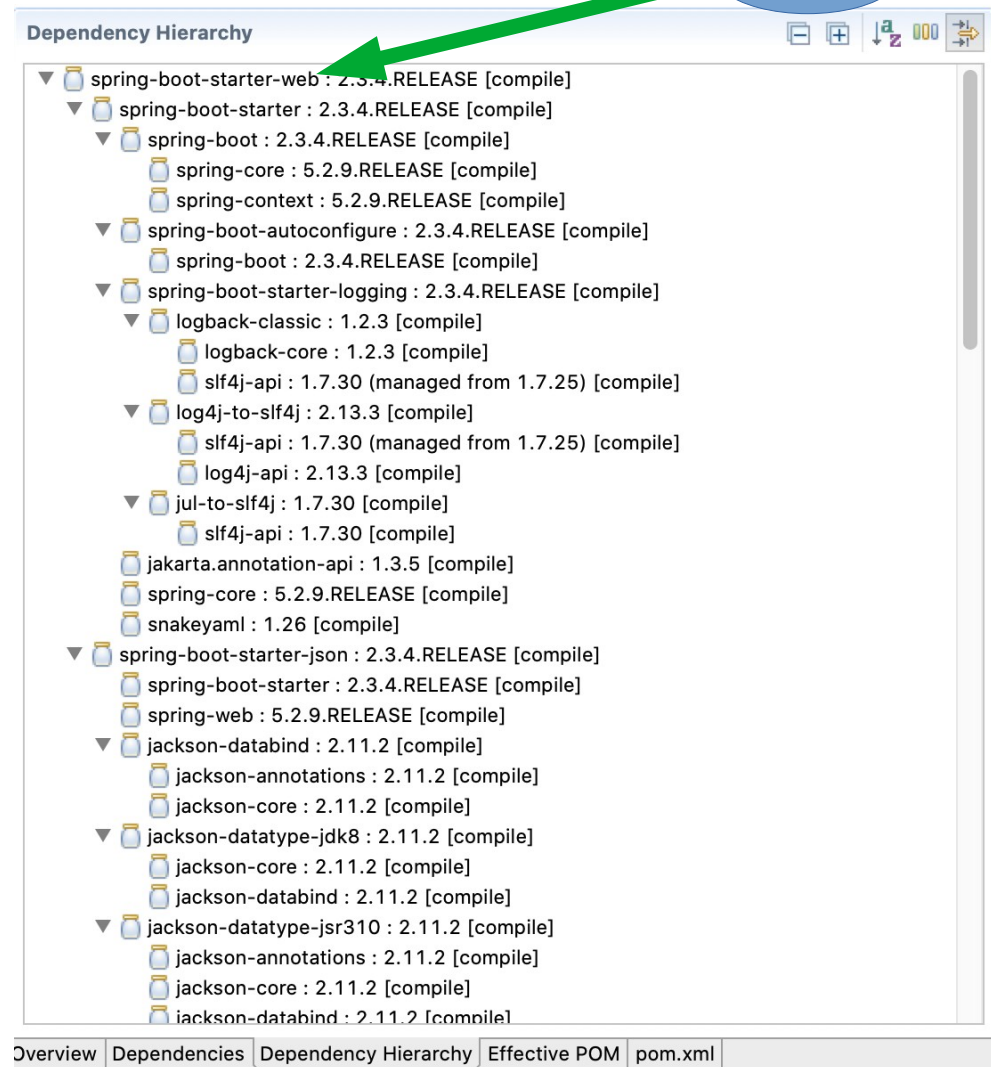
# À l'intérieur d'un Starter

- Avec Eclipse :

- Ouvrez le pom.xml
- Sélectionner l'onglet  
Dependency Hierarchy
- Déplier la dépendance à détailler

- Avec IntelliJ :

- Sélectionner View
- Puis Tool Windows
- Maven Projects
- Dependencies



# Starter in depth

## Dependency Hierarchy

- ▼  spring-boot-starter-web : 2.3.4.RELEASE [compile]
  - ▶  spring-boot-starter : 2.3.4.RELEASE [compile]
  - ▶  spring-boot-starter-json : 2.3.4.RELEASE [compile]
  - ▶  spring-boot-starter-tomcat : 2.3.4.RELEASE [compile]
  - ▶  spring-web : 5.2.9.RELEASE [compile]
  - ▶  spring-webmvc : 5.2.9.RELEASE [compile]



# SpringBoot Starter Parent

- Springboot fournit un starter particulier nommé spring-boot-starter-parent

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.3.4.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
```

On bénéficie des fonctionnalités configurées dans cette version du starter parent pour notre projet

# Surcharger les configurations du starter parent

```
<properties>  
<java.version>1.8</java.version>  
<resource.delimiter>@</resource.delimiter>  
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  
<maven.compiler.source>${java.version}</maven.compiler.source>  
<maven.compiler.target>${java.version}</maven.compiler.target>  
</properties>
```

# Des starters sans version

Les autres starters seront configurés dans les dépendances sans préciser de n° de version.

On précise la version du parent starter

```
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.3.4.RELEASE</version>
<relativePath /> <!-- lookup parent from repository -->
</parent>
```

Les starters héritent de la version du parent starter

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

Inutile de fournir des numéros de version individuellement

# Spring boot starter Parent & Plugins configuration

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

```
> mvn spring-boot:run
```

# Synthèse starter parent

- Les configuration par défaut de java version, UTF-encoding ...
- Gestion des dépendances simplifiée (starters sans n° de version)
- Configuration par défaut des plugins springboot