



# Spring Boot - Starters

Last Updated : 23 Jul, 2025

Before [Spring Boot](#) was introduced, Spring Developers used to spend a lot of time on Dependency management. Spring Boot Starters simplify dependency management by bundling commonly used libraries.

Dependency management, in general, is handled by Spring Boot's dependency management system, which uses the **Spring Boot BOM (Bill of Materials)**.

**Spring Boot Starters** are dependency descriptors that can be added under the <dependencies> section in [pom.xml](#). There are around 50+ Spring Boot Starters for different Spring and related technologies. These starters give all the dependencies under a single name.

For example, if you want to use **Spring Data JPA** for database access, you can include the **spring-boot-starter-data-jpa** dependency.

## Advantages of Spring Boot Starters

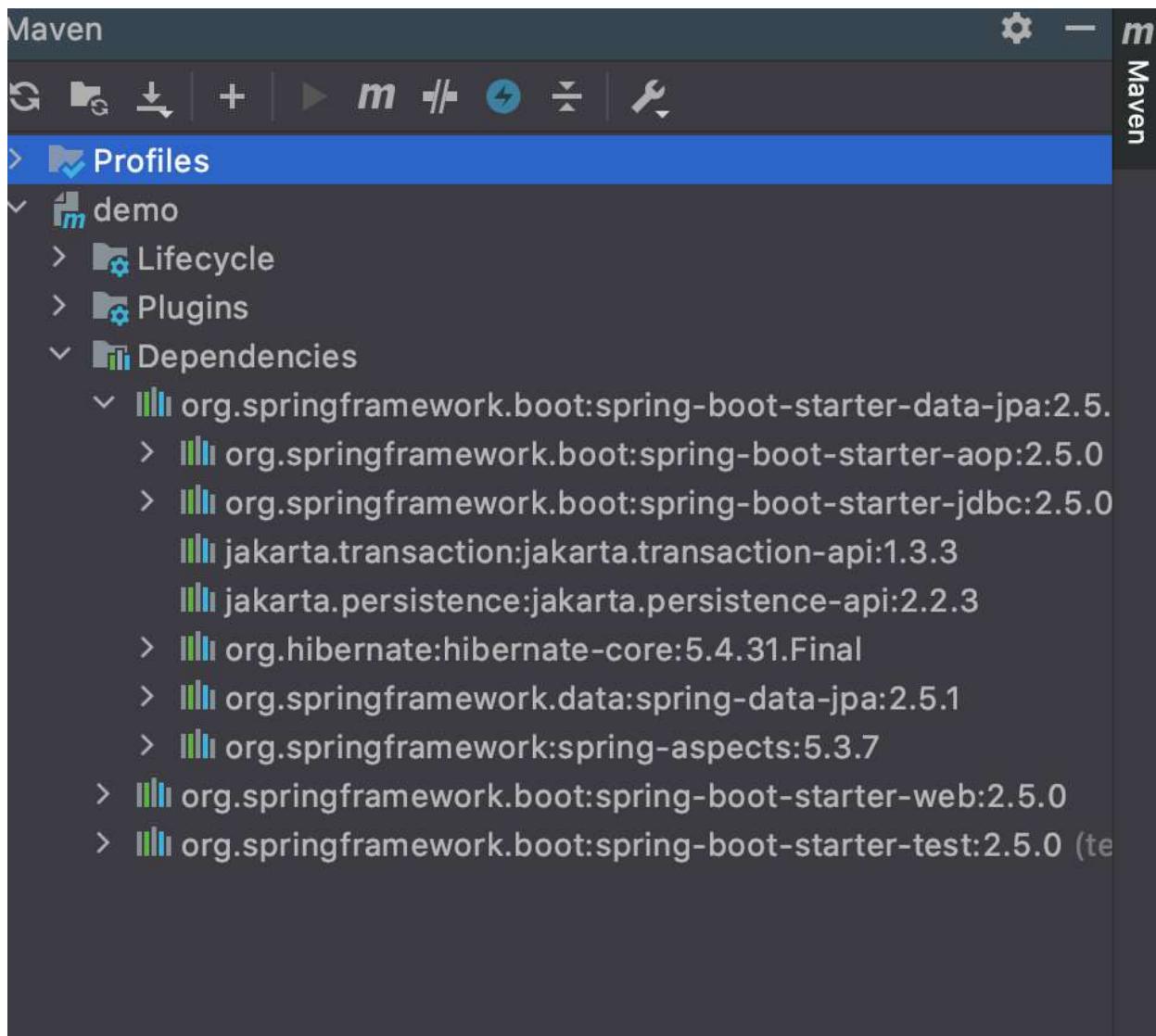
The advantages of using Starters are as mentioned below:

- Increase productivity by decreasing the Configuration time for developers.
- Managing the POM is easier since the number of dependencies to be added is decreased.
- Tested, Production-ready, and supported dependency configurations.
- No need to remember the name and version of the dependencies.

Spring Boot Starter Data JPA is illustrated below:

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

This gives all the required dependencies and can be seen under the Maven tab in IntelliJ IDEA.



In earlier days, developers used to include all those dependencies. Now Spring Boot Starters provides all those with just a single dependency. The \* in the naming convention represents the specific technology or module (e.g., spring-boot-starter-data-jpa, spring-boot-starter-webflux). It does

not always represent an application type, as some are meant for configuration, logging, or database access. For example, if we want to build web including RESTful applications using Spring MVC we have to use `spring-boot-starter-web` dependency.

## Third-Party Starters

If you want to make your own starter or other third-party starters name should not start with `spring-boot` as it reserved for official Spring Boot Starters. It can start with the name of the project. For example, if the name of the project is `gfg-code-template`, then the name of the starter can be `gfg-code-template-spring-boot-starter`.

Here out we will be discussing all 3 starters and the following starters are provided by the Spring Boot under `org.springframework.boot` group. They are namely and covered as follows:

1. Application Starters
2. Production Starters
3. Technical Starters

Let us elaborate these starters in the same sequential order which is as follows:

### 1. Spring Boot Application Starters

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

Name	Description
spring-boot-starter-aop	Starter for aspect-oriented programming with Spring AOP and AspectJ
spring-boot-starter-artemis	Starter for JMS messaging using Apache Artemis
spring-boot-starter-batch	Starter for using Spring Batch
spring-boot-starter-cache	Starter for using Spring Framework's caching support
spring-boot-starter-data-cassandra	Starter for using Cassandra distributed database and Spring Data Cassandra
spring-boot-starter-data-cassandra-reactive	Starter for using Cassandra distributed database and Spring Data Cassandra Reactive
spring-boot-starter-data-couchbase	Starter for using Couchbase document-oriented database and Spring Data Couchbase
spring-boot-starter-data-couchbase-reactive	Starter for using Couchbase document-oriented database and Spring Data Couchbase Reactive
spring-boot-starter-data-elasticsearch	Starter for using Elasticsearch search and analytics engine and Spring Data Elasticsearch
spring-boot-starter-data-jdbc	Starter for using Spring Data JDBC

Name	Description
spring-boot-starter-data-jpa	Starter for using Spring Data JPA with Hibernate
spring-boot-starter-data-ldap	Starter for using Spring Data LDAP
spring-boot-starter-data-mongodb	Starter for using MongoDB document-oriented database and Spring Data MongoDB
spring-boot-starter-data-mongodb-reactive	Starter for using MongoDB document-oriented database and Spring Data MongoDB Reactive
spring-boot-starter-data-neo4j	Starter for using Neo4j graph database and Spring Data Neo4j
spring-boot-starter-data-r2dbc	Starter for using Spring Data R2DBC
spring-boot-starter-data-redis	Starter for using Redis key-value data store with Spring Data Redis and the Lettuce client
spring-boot-starter-data-redis-reactive	Starter for using Redis key-value data store with Spring Data Redis reactive and the Lettuce client
spring-boot-starter-data-rest	Starter for exposing Spring Data repositories over REST using Spring Data REST

Name	Description
spring-boot-starter-freemarker	Starter for building MVC web applications using FreeMarker views
spring-boot-starter-groovy-templates	Starter for building MVC web applications using Groovy Templates views
spring-boot-starter-hateoas	Starter for building hypermedia-based RESTful web application with Spring MVC and Spring HATEOAS
spring-boot-starter-integration	Starter for using Spring Integration
spring-boot-starter-jdbc	Starter for using JDBC with the HikariCP connection pool
spring-boot-starter-jersey	Starter for building RESTful web applications using JAX-RS and Jersey. An alternative to spring-boot-starter-web
spring-boot-starter-jooq	Starter for using jOOQ to access SQL databases. An alternative to spring-boot-starter-data-jpa or spring-boot-starter-jdbc
spring-boot-starter-json	Starter for reading and writing json
spring-boot-starter-jta-atomikos	Starter for JTA transactions using Atomikos
spring-boot-starter-mail	Starter for using Java Mail and Spring Framework's email sending support

Name	Description
spring-boot-starter-mustache	Starter for building web applications using Mustache views
spring-boot-starter-oauth2-client	Starter for using Spring Security's OAuth2/OpenID Connect client features
spring-boot-starter-oauth2-resource-server	Starter for using Spring Security's OAuth2 resource server features
spring-boot-starter-quartz	Starter for using the Quartz scheduler
spring-boot-starter-rsocket	Starter for building RSocket clients and servers
spring-boot-starter-security	Starter for using Spring Security
spring-boot-starter-test	Starter for testing Spring Boot applications with libraries including JUnit Jupiter, Hamcrest and Mockito
spring-boot-starter-thymeleaf	Starter for building MVC web applications using Thymeleaf views
spring-boot-starter-validation	Starter for using Java Bean Validation with Hibernate Validator
spring-boot-starter-web	Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container.
spring-boot-starter-web-services	Starter for using Spring Web Services

Name	Description
spring-boot-starter-webflux	Starter for building WebFlux applications using Spring Framework's Reactive Web support
spring-boot-starter-websocket	Starter for building WebSocket applications using Spring Framework's WebSocket support

## 2. Spring Boot Production Starters

Name	Description
spring-boot-starter-actuator	Starter for using Spring Boot's Actuator which provides production-ready features to help you monitor and manage your application

## 3. Spring Boot Technical Starters

Name	Description
spring-boot-starter-jetty	Starter for using Jetty as the embedded servlet container. An alternative to spring-boot-starter-tomcat

Name	Description
spring-boot-starter-log4j2	Starter for using Log4j2 for logging. An alternative to spring-boot-starter-logging
spring-boot-starter-logging	Starter for logging using Logback. Default logging starter
spring-boot-starter-reactor-netty	Starter for using Reactor Netty as the embedded reactive HTTP server.
spring-boot-starter-tomcat	Starter for using Tomcat as the embedded servlet container. Default servlet container starter used by spring-boot-starter-web
spring-boot-starter-undertow	Starter for using Undertow as the embedded servlet container. An alternative to spring-boot-starter-tomcat

[Comment](#)[More info](#)[Advertise with us](#)

**Corporate & Communications Address:**

A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

**Registered Address:**

K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305