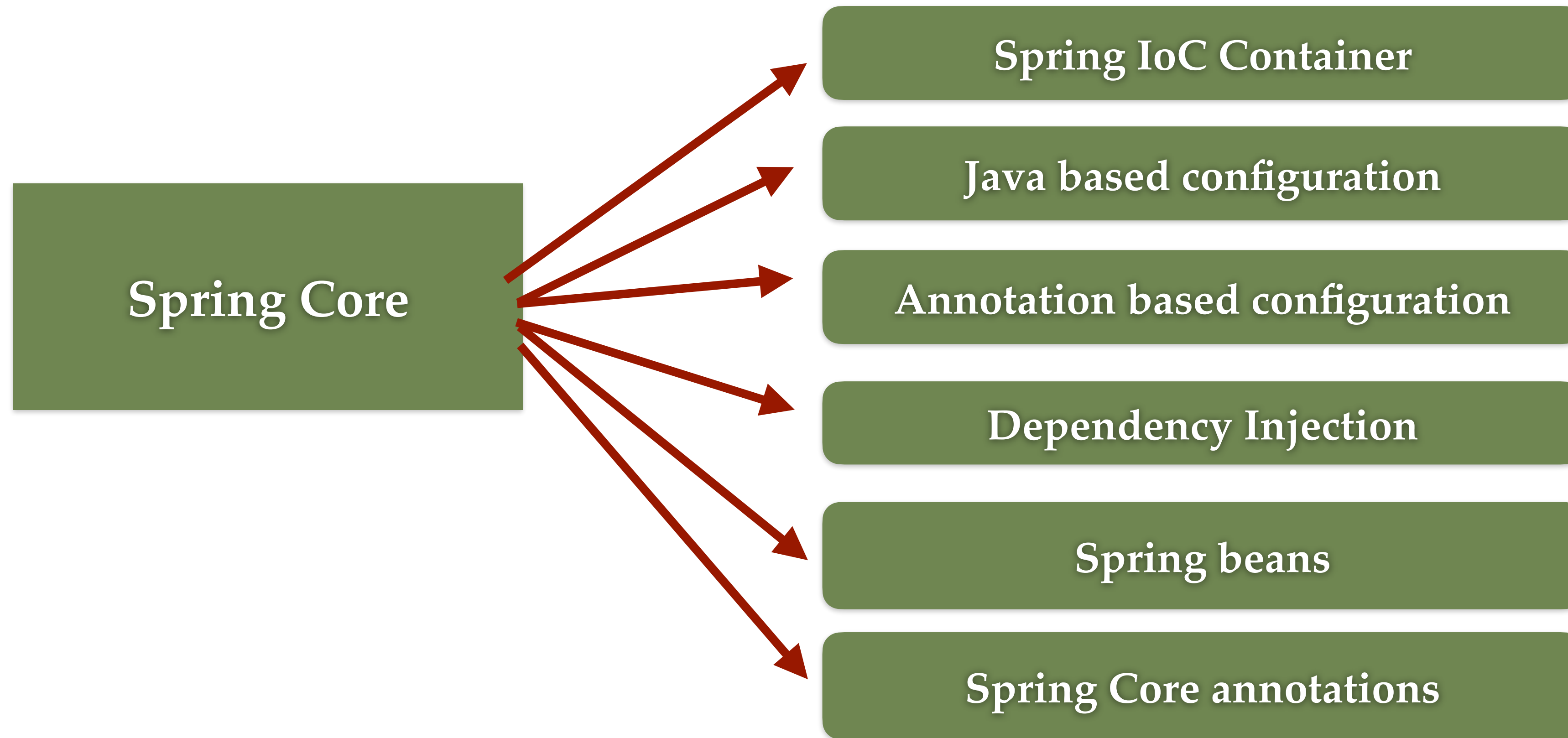


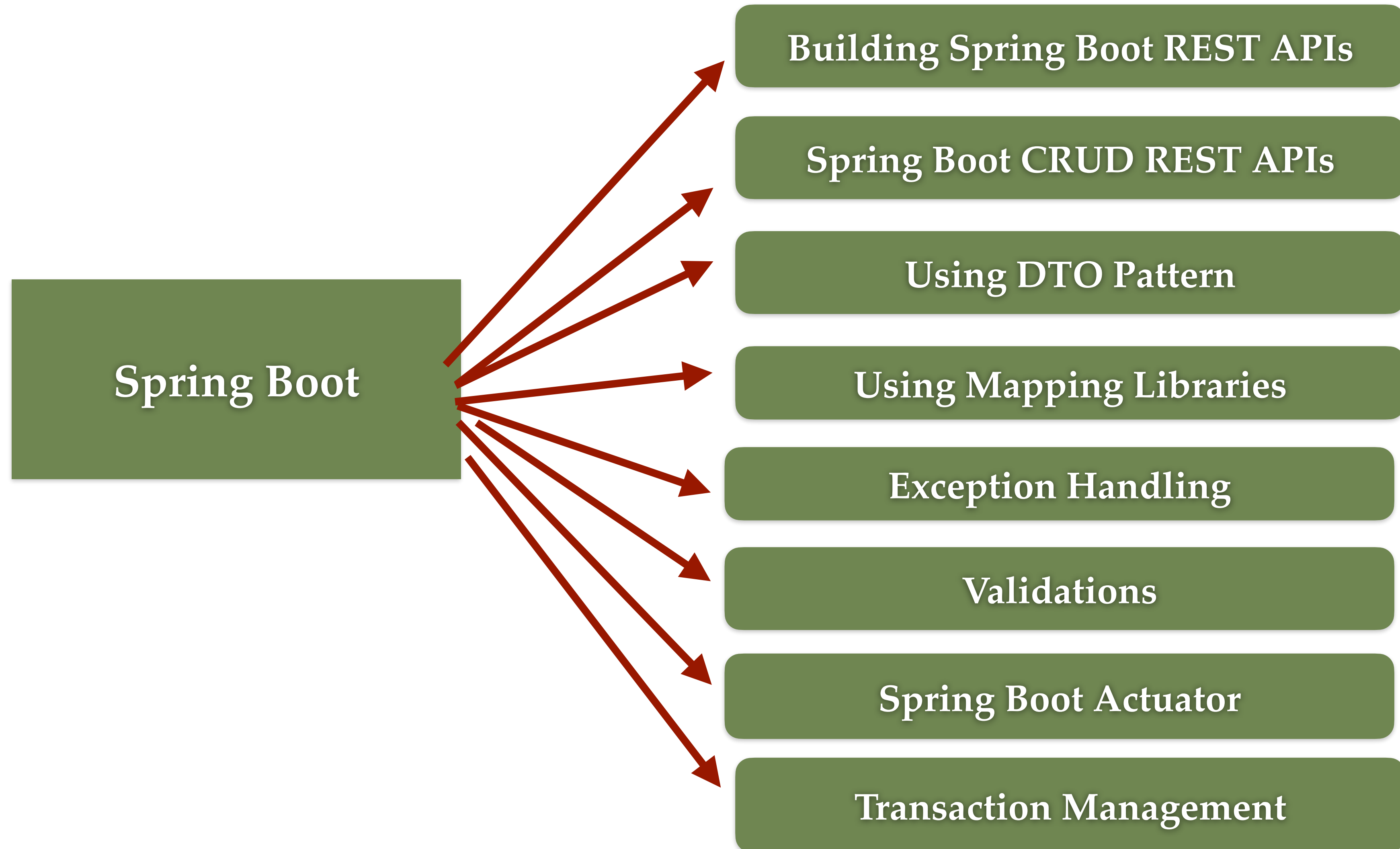
Spring Framework 6 and Spring Boot 3 for Beginners

(Includes Simple Projects)

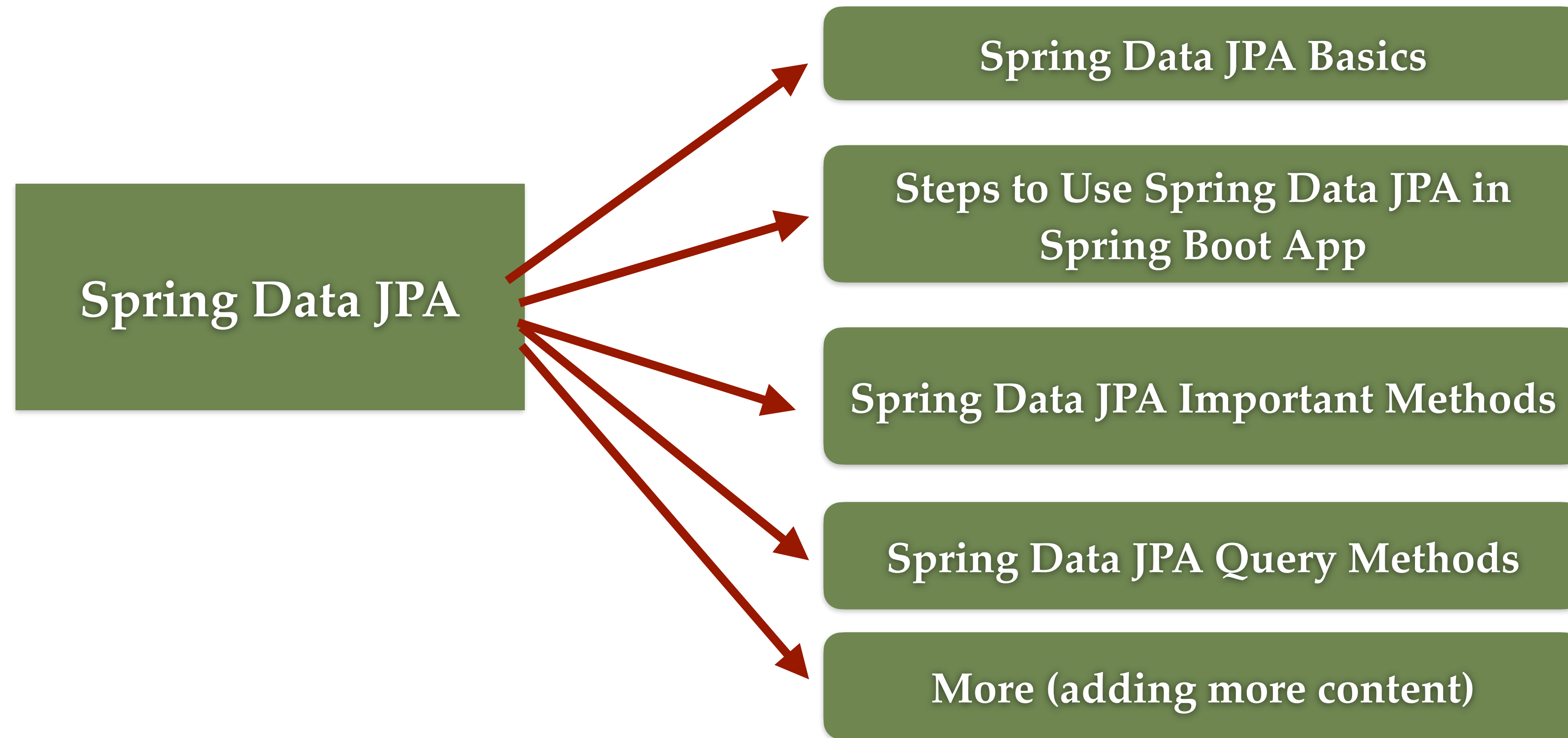
Learn Spring Core



Learn Spring Boot

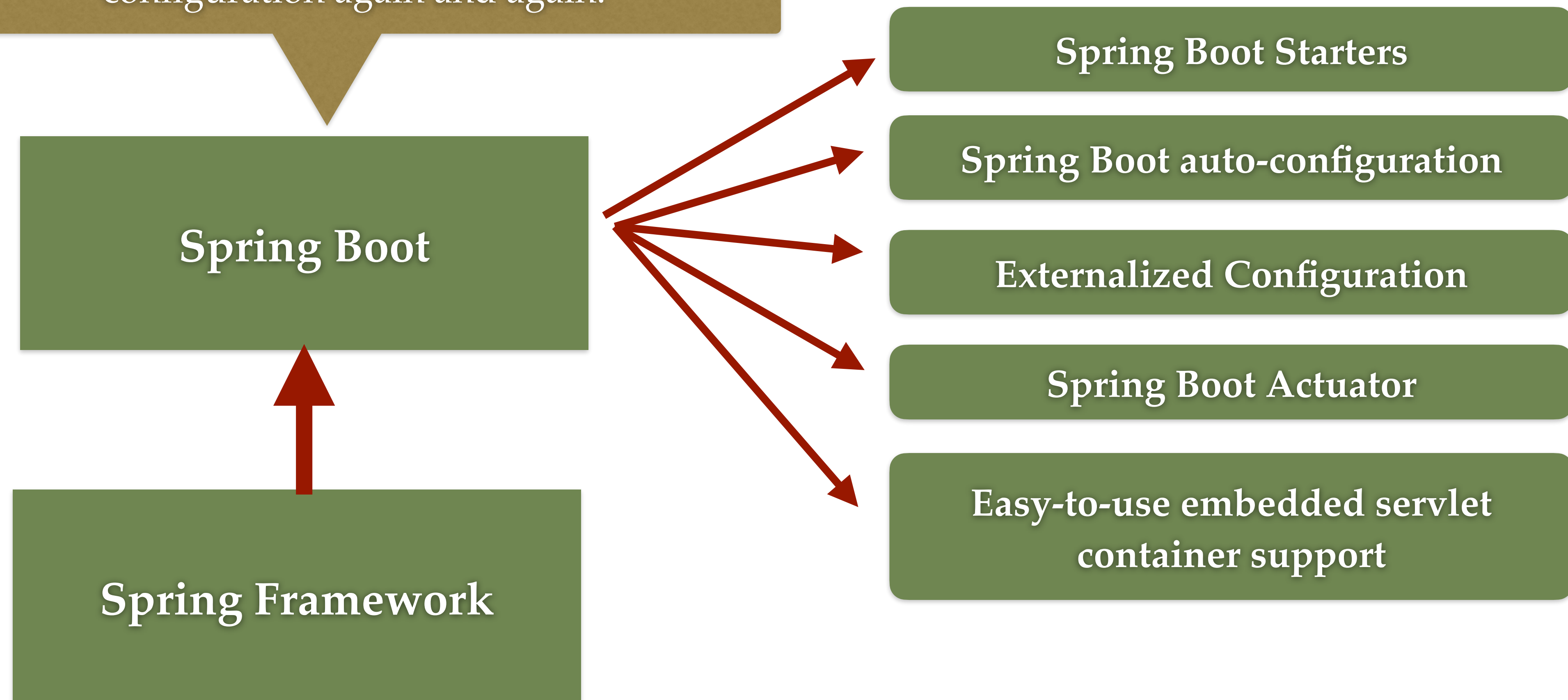


Learn Spring Data JPA



What is Spring Boot?

The main goal of Spring Boot is to quickly create Spring-based applications without requiring developers to write the same boilerplate configuration again and again.



Spring Framework

1. Spring is a very popular Java framework for building web and enterprise applications.
2. Spring is very popular for several reasons:
 - dependency injection
 - Easy to use but powerful database transaction management capabilities
 - Good Integration with other Java frameworks like JPA / Hibernate ORM, Struts / JSF / etc. web frameworks
 - Web MVC framework for building web applications

What is a Problem?

Basically, Spring-based applications have a lot of configurations.

For example:

When we develop Spring MVC web application using Spring MVC then we need to configure:

Component scan, Dispatcher Servlet, View resolver, Web jars(for delivering static content) among other things.

When we use Hibernate/JPA in the same Spring MVC application then we would need to configure a **Data source, Entity manager factory/session factory, Transaction manager among other things.**

When you use cache, message queue, NoSQL in the same Spring MVC application then we need to configure:

Cache configuration

Message queue configuration

NoSQL database configuration

One more major problem - We need to maintain all integration of different Jar dependencies and it's compatible versions.

What is a Solution

- > Spring Boot is the solution**
- > Spring Boot automatically configures the configurations based on the jar dependencies that we add to our project.**

Create Your First Spring Boot App

By Ramesh Fadatare (Java Guides)

Understanding Spring Boot App Structure

(Important files and folders)

By Ramesh Fadatare (Java Guides)

Spring Boot **Starters** Feature

By Ramesh Fadatare (Java Guides)

Problem

Spring MVC

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-json</artifactId>  
  <version>3.0.0</version>  
  <scope>compile</scope>  
</dependency>  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-tomcat</artifactId>  
  <version>3.0.0</version>  
  <scope>compile</scope>  
</dependency>  
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-web</artifactId>  
  <version>6.0.2</version>  
  <scope>compile</scope>  
</dependency>  
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-webmvc</artifactId>  
  <version>6.0.2</version>  
  <scope>compile</scope>  
</dependency>
```

```
<dependency>  
  <groupId>org.hibernate.orm</groupId>  
  <artifactId>hibernate-core</artifactId>  
  <version>6.1.5.Final</version>  
  <scope>compile</scope>  
</dependency>  
<dependency>  
  <groupId>org.springframework.data</groupId>  
  <artifactId>spring-data-jpa</artifactId>  
  <version>3.0.0</version>  
  <scope>compile</scope>  
</dependency>  
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-aspects</artifactId>  
  <version>6.0.2</version>  
  <scope>compile</scope>  
</dependency>
```

Hibernate

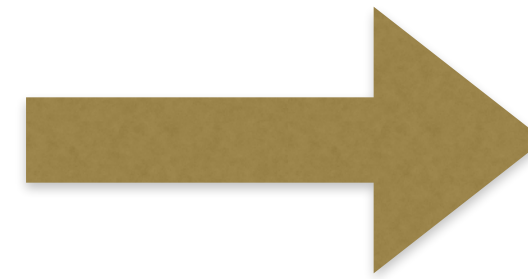
Spring Security

```
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-aop</artifactId>  
  <version>6.0.2</version>  
  <scope>compile</scope>  
</dependency>  
<dependency>  
  <groupId>org.springframework.security</groupId>  
  <artifactId>spring-security-config</artifactId>  
  <version>6.0.0</version>  
  <scope>compile</scope>  
</dependency>  
<dependency>  
  <groupId>org.springframework.security</groupId>  
  <artifactId>spring-security-web</artifactId>  
  <version>6.0.0</version>  
  <scope>compile</scope>  
</dependency>
```


Solution - spring-boot-starter-web

Spring MVC

```
<!-- ... -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-json</artifactId>
  <version>3.0.0</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-tomcat</artifactId>
  <version>3.0.0</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>6.0.2</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>6.0.2</version>
  <scope>compile</scope>
</dependency>
```



Spring Boot Web Starter

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

Solution - spring-boot-starter-data-jpa

Spring Data JPA with
Hibernate

```
<dependency>
  <groupId>org.hibernate.orm</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>6.1.5.Final</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.springframework.data</groupId>
  <artifactId>spring-data-jpa</artifactId>
  <version>3.0.0</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-aspects</artifactId>
  <version>6.0.2</version>
  <scope>compile</scope>
</dependency>
```



Spring Boot Starter JPA

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

Solution - spring-boot-starter-security

Spring Security

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-aop</artifactId>
  <version>6.0.2</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-config</artifactId>
  <version>6.0.0</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-web</artifactId>
  <version>6.0.0</version>
  <scope>compile</scope>
</dependency>
```



Spring Boot Starter
Spring Security

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


Spring Boot Starters

1. These starters are pre-configured with the most commonly used library dependencies so you don't have to search for the compatible library versions and configure them manually.
2. For example, when we add the **spring-boot-starter-web** dependency, it will by default pull all the commonly used libraries while developing Spring MVC applications, such as spring-webmvc, jackson-json, validation-api, and tomcat.
3. One more example, the **spring-boot-starter-data-jpa** starter module includes all the dependencies required to use Spring Data JPA, along with Hibernate library dependencies, as Hibernate is the most commonly used JPA implementation.

Spring Boot Starter Parent

By Ramesh Fadatare (Java Guides)

Spring Boot Starter Parent

The **spring-boot-starter-parent** project is a special starter project that provides:

- Default configurations
- Manage the versions of dependencies
- Default configurations for Maven plugins

Spring Boot **Auto** **Configuration** Feature

By Ramesh Fadatare (Java Guides)

Problem

Basically, Spring-based applications have a lot of configurations.

For example:

1. When we develop Spring MVC web application using Spring MVC then we need to configure:

Component scan, Dispatcher Servlet, View resolver, Web jars(for delivering static content) among other things.

2. When we use Hibernate/JPA in the same Spring MVC application then we would need to configure a

Data source, Entity manager factory/session factory, Transaction manager among a host of other things.

Solution

When we add Spring MVC Jar dependency to our application then can we automatically configure some Spring beans related to Spring MVC.

Ex: Dispatcher Servlet, View resolver

When we add Hibernate Jar dependency to our application then can we automatically configure Spring beans related to Hibernate.

Ex: Data source, Entity manager factory/session factory, Transaction manager

Spring Boot Auto Configuration

Spring Boot auto-configuration attempts to automatically configure Spring application based on the jar dependencies that you have added to project.

Example 1: if you add **spring-boot-starter-web** Jar dependency to your Spring boot application, Spring Boot assumes you are trying to build a SpringMVC-based web application and automatically tries to register Spring beans such as **DispatcherServlet**, **ViewResolver** if it is not already registered.

Example 3: If you add **spring-boot-starter-data-jpa** starter dependency then it assume that you are trying to use Hibernate to develop DAO layer so Spring boot automatically register the Spring beans such as **Data source**, **Entity manager factory**/ **session factory**, **Transaction manager**.

Spring Boot auto-configuration attempts to automatically configure Spring application based on the jar dependencies that you have added to project.

Spring Boot Auto Configuration in an Action

-> Spring boot auto configuration is implemented in **spring-boot-autoconfigure.jar**

-> Example of Spring MVC:

- Configure DispatcherServlet (**DispatcherServletAutoConfiguration**)
- Configure Embedded Tomcat Server (**EmbeddedWebServerFactoryCustomizerAutoConfiguration**)
- Configure Default Error Page (**ErrorMvcAutoConfiguration**)
- Configure JSON to Java Conversion (**JacksonAutoConfiguration**)