**<https://spring.io/projects>**

**Spring Boot - Is an opinionated Framework also** a module of Spring Framework. It allows us to build a stand-alone application with minimal or zero configurations. It is better to use if we want to develop a simple Spring-based application or RESTful services

**Tells how to build an app by providing most of the features and functionalities.**

**if you include the spring boot starter pom for jpa, you'll get autoconfigured for you an in-memory database, a hibernate entity manager, and a simple data source.**

**This is an example of an opinionated (Spring's opinion that it's a good starting point) default configuration that you can override.**

Spring Boot makes it easy to quickly bootstrap and start developing a Spring-based application. It avoids a lot of boilerplate code. It hides a lot of complexity behind the scene so that the developer can quickly get started and develop Spring-based applications easily.

Ref

https://www.baeldung.com/spring-boot-h2-database

Java Persistence API (JPA) is a collection of classes and methods to persistently store the vast amounts of data into a database.

[**https://spring.io/projects/spring-boot**](https://spring.io/projects/spring-boot)

**comparison between Spring and Spring Boot**

|  |  |
| --- | --- |
| **Spring** | **Spring Boot** |
| **Spring Framework** is a widely used Java EE framework for building applications. | **Spring Boot Framework is a module in spring framework** which iswidely used to develop **REST APIs**. |
| It aims to simplify Java EE development that makes developers more productive. | It aims to shorten the code length and provide the easiest way to develop **Web Applications**. |
| The primary feature of the Spring Framework is **dependency injection**. | The primary feature of Spring Boot is **Autoconfiguration**. It automatically configures the classes based on the requirement. |
| It helps to make things simpler by allowing us to develop **loosely coupled** applications. | It helps to create a **stand-alone** application with minimum configuration. |
| The developer writes a lot of code (**boilerplate code**) to achieve a basic functionality. | It **reduces** boilerplate code. |
| To test the Spring project, we need to set up the sever explicitly. | Spring Boot offers **embedded server** such as **Jetty** and **Tomcat**, H2 Database etc. |
| Developers manually define dependencies for the Spring project in **pom.xml**. | Spring Boot comes with the concept of **starter** in pom.xml or build.gradle file that internally takes care of downloading the dependencies **JARs** based on Spring Boot Requirement. |

**Spring Boot vs. Spring MVC Comparison**

|  |  |
| --- | --- |
| **Spring Boot** | **Spring MVC** |
| **Spring Boot** is a module of Spring for packaging the Spring-based application with sensible defaults. | **Spring MVC** is a model view controller-based web framework under the Spring framework. |
| It provides default configurations to build **Spring-powered** framework. | It provides **ready to use** features for building a web application. |
| There is no need to build configuration manually. | It requires build configuration manually. |
| There is **no requirement** for a deployment descriptor. | A Deployment descriptor is **required**. |
| It avoids boilerplate code and wraps dependencies together in a single unit. | It specifies each dependency separately. |
| It **reduces** development time and increases productivity. | Comparatively It takes **more** time to achieve the same. |