What is Java Spring Boot?

Java Spring Framework (Spring Framework) is a popular, open source, enterprise-level framework for creating standalone, production-grade applications that run on the Java Virtual Machine (JVM).

Java Spring Boot (Spring Boot) is a tool that makes developing web application and microservices with Spring Framework faster and easier through three core capabilities:

1. Autoconfiguration
2. An opinionated approach to configuration
3. The ability to create standalone applications

Why is Spring Framework so popular?

Spring Framework offers a *dependency injection* feature that lets objects define their own dependencies that the Spring container later injects into them. This enables developers to create modular applications consisting of loosely coupled components that are ideal for [microservices](https://www.ibm.com/topics/microservices" \o "microservices" \t "_blank) and distributed network applications.

Spring Framework also offers built-in support for typical tasks that an application needs to perform, such as data binding, type conversion, validation, exception handling, resource and event management, internationalization, and more. It integrates with various Java EE technologies such as RMI (Remote Method Invocation), AMQP (Advanced Message Queuing Protocol), Java Web Services, and others. In sum, Spring Framework provides developers with all the tools and features the need to create loosely coupled, cross-platform Java EE applications that run in any environment.

the Spring Framework is a comprehensive and widely-used Java framework for building enterprise-grade applications. It is modular in nature and offers various modules to address different concerns in application development. Here are some of the main modules of the Spring Framework:

1. **Spring Core Container**: This module provides the fundamental functionality of the Spring Framework, including **the Inversion of Control (IoC**) **and Dependency Injection (DI)** features. It allows you to **manage and configure** Java objects (beans) and their dependencies.

2. Spring AOP (Aspect-Oriented Programming): This module supports **Aspect-Oriented Programming**, enabling you to **separate cross-cutting concerns** from the core business logic. It allows you to define aspects (cross-cutting behaviors) and apply them to the application's components.

3. Spring Data Access / Spring JDBC: This module simplifies data access from relational databases using

the JDBC (Java Database Connectivity) APIIt **provides utilities and abstractions** to perform database operations and handle exceptions.

4. Spring ORM (**Object-Relational Mapping**): This module integrates popular Object-Relational Mapping frameworks like Hibernate, JPA (Java Persistence API), and others, providing seamless interaction with relational databases **using plain old Java objects (POJOs**).

5. Spring Web MVC: This module offers a **Model-View-Controller (MVC) architecture** for building web applications. It provides components for **handling HTTP requests**, **managing views, and mapping URLs to controller methods.**

6. Spring Web: This module provides various utilities and integration features for web-based applications, including multipart file uploading, WebSockets, and support for various web-related protocols.

7. Spring Security: This module offers powerful security features for Spring-based applications. It provides authentication, authorization, and various security mechanisms to protect web and non-web applications.

8. Spring Test: This module supports testing Spring applications, providing utilities for unit testing, integration testing, and mocking components.

9. Spring Messaging: This module facilitates messaging within Spring applications, supporting different messaging protocols and technologies like JMS (Java Message Service) and STOMP (Simple Text Oriented Messaging Protocol).

10. Spring Batch: This module is used for **batch processing i**n Spring applications. It supports **the bulk processing of large volumes of data** with transaction management, monitoring, and retry capabilities.

11. Spring Integration: This module provides support for integrating different systems and applications in a **message-driven manner.** It supports **Enterprise Integration Patterns (EIP**) and facilitates the integration **of various technologies and protocols.**

12. Spring Web Services: **This module enables the development of SOAP and RESTful web services**. It provides support for **creating web services, handling XML payloads**, and **integrating with other web service technologies.**

15. Spring Session: This module **facilitates session management** in Spring applications. It allows you to manage sessions across multiple nodes in a distributed environment and provides additional features **for session data storage**.

16. Spring WebSocket: This module offers WebSocket support for real-time communication between clients and servers. It enables bidirectional, full-duplex communication channels over a single TCP connection.

19. **Spring Social**: This module integrates popular social media platforms (e.g., Facebook, Twitter, LinkedIn) into Spring applications, allowing users to authenticate and interact with these platforms.

20. **Spring Security OAuth:** This module provides support for implementing OAuth (Open Authorization) protocols for secure authorization and authentication in Spring applications.

21. Spring Data Redis: This module offers integration with Redis, an in-memory data store. It provides support for using Redis as a caching solution and for data storage in Spring applications.

22. Spring LDAP: This module facilitates communication with LDAP (Lightweight Directory Access Protocol) servers, enabling directory authentication and user management in Spring applications.