

Spring, Spring Boot, Spring MVC y Spring Data

Spring

Spring es un marco de desarrollo de aplicaciones empresariales para Java que proporciona un enfoque integral y modular para simplificar el desarrollo y la creación de aplicaciones robustas y escalables.

Su arquitectura se basa en conceptos como Inversión de Control (IoC) e Inyección de Dependencias (DI), lo que facilita la gestión de objetos y la creación de aplicaciones empresariales flexibles. Por un lado, la Inversión de Control permite delegar el control de la creación de objetos al contenedor de Spring. Por el otro lado, la Inyección de Dependencias, permite que los objetos dependientes sean proporcionados por el contenedor de Spring.

Spring ofrece diversos módulos, como Spring Core Container, Spring AOP (Aspect-Oriented Programming), Spring DAO (Data Access Object), entre otros, que abordan distintos aspectos del desarrollo.

Spring Boot

Spring Boot es un proyecto dentro del ecosistema de Spring que se centra en simplificar y acelerar el proceso de desarrollo de aplicaciones Spring. Se destaca por su configuración automática basada en convenciones, lo que significa que muchas decisiones de configuración pueden ser manejadas automáticamente por el marco, sin requerir configuración manual extensa.

Spring Boot también permite la embebibilidad de servidores web, lo que facilita la ejecución de aplicaciones como aplicaciones independientes sin necesidad de servidores externos. Su objetivo principal es proporcionar una experiencia de desarrollo rápida y sencilla para proyectos basados en Spring.

Spring MVC

Spring MVC es un módulo dentro del proyecto Spring que proporciona una arquitectura de patrón de diseño MVC (Modelo-Vista-Controlador) para el desarrollo de aplicaciones web y servicios RESTful de forma óptima y fácil. Implementa todas las características básicas de Spring como Inversión de Control e Inyección de Dependencia.

Spring MVC divide la aplicación en tres componentes: el Modelo (lógica de negocios y datos), la Vista (lógica de la interfaz de usuario) y el Controlador (lógica detrás de las solicitudes, procesamiento de entrada y coordinación de respuestas).

Utiliza una configuración declarativa, es decir, asocia URLs a controladores y vistas para simplificar el desarrollo. Así mismo es altamente extensible lo cual permite la integración con otros componentes de Spring y tecnologías.

Spring MVC funciona a través del DispatcherServlet, el cual, al recibir una solicitud HTTP, busca en el HandlerMapping para llamar al controlador correspondiente. El controlador procesa la solicitud al invocar los métodos de servicio según el método HTTP especificado. Después, el DispatcherServlet utiliza un ViewResolver para gestionar la vista y presenta los datos del modelo en el navegador.

Spring Data

Spring Data es una herramienta que simplifica significativamente el acceso a datos, bases de datos relacionales y no relacionales, servicios de datos en la nube a través de diversos proyectos que son específicos de una base de datos determinada.

En **bases de datos relacionales**, Spring Data simplifica la interacción con JDBC y ORM permitiendo a los desarrolladores trabajar con representaciones orientadas a objetos de los datos sin escribir consultas SQL tediosas. Ofrece soporte para frameworks ORM como Hibernate, JPA y MyBatis, mejorando la legibilidad y mantenibilidad del código al facilitar la creación de consultas y manipulación de datos.

Por otro lado, con **bases de datos no relacionales** como MongoDB, Cassandra y Neo4j, Spring Data facilita el trabajo ofrece adaptadores y abstracciones. Esto permite a los desarrolladores cambiar entre bases de datos sin modificar considerablemente el código, brindando flexibilidad y adaptabilidad en entornos de desarrollo dinámicos.