

Spring

Java Spring Framework (Spring Framework) es una popular estructura empresarial de código abierto para crear aplicaciones independientes de nivel de producción que se ejecutan en la máquina virtual Java (JVM). Ofrece una función de inyección de dependencias que permite que los objetos definan sus propias dependencias que el contenedor de Spring luego inyecta en ellos. Esto permite a los desarrolladores crear aplicaciones modulares que constan de componentes poco acoplados que son ideales para microservicios y aplicaciones de red distribuida. De igual forma ofrece como elemento clave el soporte de infraestructura a nivel de aplicación, brindando un completo modelo tanto para la configuración como para la programación de aplicaciones empresariales desarrolladas bajo Java, sin discriminación en cuanto al despliegue de la plataforma.

Todo esto trae consigo una gran ventaja, ya que permite que los equipos de desarrollo puedan enfocarse directamente en la lógica empresarial que requiere la aplicación, haciendo el proceso más corto, rápido y eficaz, ahorrando líneas de código evitando tareas repetitivas.

Beneficios

- Spring está organizado de forma modular. A pesar de la cantidad de paquetes y clases que tiene, solo debemos ocuparnos de aquellos que necesitemos para nuestro desarrollo e ignorar el resto.
- Utiliza algunas de las tecnologías existentes, como varios frameworks ORM, JEE, temporizadores Quartz y JDK, frameworks de registro y otras tecnologías de visualización.
- Probar una aplicación escrita con Spring es un proceso simple, porque el código dependiente del entorno se traslada a este framework. Además, mediante el uso de JavaBeanstyle, se vuelve más fácil utilizar la inyección de dependencia para hacer pruebas, para ello podemos hacer uso de datos dummies o mocks, para ver las respuestas.
- El framework web de Spring es un framework MVC web bien diseñado, que proporciona una excelente alternativa a los frameworks web como Struts u otros frameworks web sobre diseñados o menos populares.
- Spring proporciona una API para traducir excepciones específicas de la tecnología (como por ejemplo las generadas por JDBC, Hibernate o JDO) en excepciones consistentes y no verificadas.
- Los contenedores de IoC (Inversion of Control) tienden a ser livianos, especialmente cuando se comparan con los Enterprise JavaBeans (EJB). Esto es ideal para desarrollar y desplegar aplicaciones en máquinas con memoria y recursos limitados.
- Spring proporciona una interfaz de gestión de transacciones coherente que puede reducirse a una transacción local (utilizando una única base de datos) y ampliarse a transacciones globales.
- Permite separar el registro, la auditoría, las transacciones declarativas, la seguridad, el almacenamiento en caché, de la lógica comercial a través de la AOP (Programación Orientada a Aspectos).

- Cuenta con plantillas para diversas tecnologías entre la cuales podemos destacar las siguientes: JDBC, Hibernate y JPA, de forma tal que no hay necesidad de escribir un código extenso, ya que con estas plantillas simplifica el trabajo en cuanto a los pasos básicos a implementar de estas tecnologías.

Spring MVC

Spring MVC es un subproyecto Spring, un marco de Java para crear aplicaciones web y servicios Restful de forma óptima y fácil. Sigue el patrón de diseño Modelo-Vista-Controlador e implementa todas las características básicas de un marco de resorte central como Inversión de Control e Inyección de Dependencia.

El patrón MVC da como resultado la separación de los diferentes aspectos de la aplicación como lógica de entrada, lógica comercial y lógica de la interfaz de usuario, al mismo tiempo que proporciona un acoplamiento entre los elementos.

Aspectos importantes de MVC

- La Vista: Éste es responsable de hacer una representación de los modelos y, en general una salida HTML, una vista representa la información proporcionada en un formato particular. Normalmente, JSP+JSTL se utilizan para crear una página de vista. Aunque Spring también es compatible con otras tecnologías de visualización como Apache Velocity, Thymeleaf y FreeMarker.
- El Controlador: Procesa las solicitudes de los usuarios, construye un modelo que sea apropiado y luego, lo pasa a la vista de su representación, Es responsable de administrar el flujo de la aplicación Spring MVC.
- El Modelo: los datos de la aplicación son encapsulados, y en general consistirán en POJO.

¿Cómo funciona?

Después de recibir una solicitud HTTP, el DispatcherServlet hace la consulta a HandlerMapping para llamar al controlador apropiado, éste toma la solicitud y llama a los métodos de servicio según el método GET o POST, y el DispatcherServlet tomará la solicitud con la ayuda de ViewResolver. Después de todo ese trayecto, DispatcherServlet pasa los datos del modelo a la vista y finalmente éste se representa en el navegador.

Ventajas

Aquí podemos ver las ventajas más importantes de usar Spring MVC:

- Mapeo flexible: proporciona las anotaciones específicas que redirigen fácilmente la página.
- Desarrollo rápido: Spring MVC facilita el desarrollo rápido y paralelo.
- Ligero: utiliza un contenedor de servlet liviano para desarrollar e implementar su aplicación.
- Código comercial reutilizable: en lugar de crear nuevos objetos, nos permite reutilizar los objetos existentes.
- Fácil de probar: en Spring, generalmente creamos clases de JavaBeans que le permiten inyectar datos de prueba utilizando los métodos setter.

- Funciones separadas: Spring MVC separa cada función, donde el objeto modelo, el controlador, el objeto de comando, la resolución de vista, DispatcherServlet, el validador, etc. pueden ser cumplidos por un objeto especializado.

Spring Boot

Java Spring Boot (Spring Boot) es una herramienta que hace que el desarrollo de aplicaciones web y microservicios con Spring Framework sea más rápido y fácil a través de tres funcionalidades principales:

- Configuración automática
- Un enfoque obstinado de la configuración
- La capacidad de crear aplicaciones independientes

A pesar de lo completo que es Spring Framework y de las funcionalidades que presenta, hacen falta un tiempo considerable y conocimientos suficientes para configurar, instalar e implementar las aplicaciones Spring. Spring Boot alivia este esfuerzo con tres funcionalidades importantes.

Configuración automática

La configuración automática significa que las aplicaciones se inicializan con dependencias preestablecidas que no tienen que configurarse manualmente. Como Java Spring Boot viene con funciones de configuración automática integradas, configura automáticamente tanto el Spring Framework subyacente como los paquetes de terceros según su configuración (y según las mejores prácticas, lo que ayuda a evitar errores).

Aunque puede anular estos valores predeterminados una vez que se completa la inicialización, la función de configuración automática de Java Spring Boot le permite comenzar a desarrollar sus aplicaciones basadas en Spring rápidamente y reduce la posibilidad de errores humanos.

Aplicaciones independientes

Spring Boot ayuda a los desarrolladores a crear aplicaciones que simplemente se ejecutan. Específicamente, le permite crear aplicaciones independientes que se ejecutan por sí mismas, sin depender de un servidor web externo, al integrar un servidor web como Tomcat o Netty en su aplicación durante el proceso de inicialización. Como resultado, puede iniciar su aplicación en cualquier plataforma con el comando Ejecutar (puede renunciar esta función para crear aplicaciones sin un servidor web integrado).