

Spring y Hibernate

Introducción

- ¿Qué es Spring?
- ¿Qué es Hibernate?
- Beneficios de utilizar Spring y Hibernate juntos

Spring Framework

Conceptos clave

- Inversión de control (IoC)
- Inyección de dependencias (DI)
- Aspectos y AOP
- Contenedor de aplicaciones

Módulos principales de Spring

- Spring Core
- Spring MVC
- Spring Data
- Spring Security

Hibernate

ORM y persistencia

- ¿Qué es Hibernate?
- Mapeo objeto-relacional (ORM)
- Ventajas de Hibernate para el acceso a datos

Componentes principales

- Sesiones y transacciones
- Configuración de Hibernate
- Mapeo de objetos y relaciones
- Consultas y criterios

Integración de Spring y Hibernate

Configuración básica

- Establecer la configuración de Hibernate en el archivo de configuración de Spring
- Establecer las propiedades de conexión a la base de datos

Integración del administrador de transacciones

- Usar el administrador de transacciones de Spring
- Beneficios de la gestión de transacciones declarativa

Inyección de dependencias en DAOs

- Configurar los DAOs de Hibernate como beans de Spring
- Inyectar dependencias de sesiones y transacciones en los DAOs

Ejemplo de código

Configuración de Spring y Hibernate

```
@Configuration
@EnableTransactionManagement
public class AppConfig {

    @Bean
    public DataSource dataSource() {
        // Configuración del origen de datos
    }

    @Bean
    public LocalSessionFactoryBean sessionFactory() {
        // Configuración de la sesión de Hibernate
    }

    @Bean
    public HibernateTransactionManager transactionManager(SessionFactory sessionFactory) {
        // Configuración del administrador de transacciones de Hibernate
    }
}
```