benefits of hibernate framework

Hibernate eliminates all the boiler-plate code that comes with JDBC and takes care of managing resources, so we can focus on business logic.

Hibernate framework provides support for XML as well as JPA annotations, that makes our code implementation independent.

Hibernate provides a powerful query language (HQL) that is similar to SQL. However, HQL is fully object-oriented and understands concepts like inheritance, polymorphism and association.

Hibernate is an open source project from Red Hat Community and used worldwide. This makes it a better choice than others because learning curve is small and there are tons of online documentations and help is easily available in forums.

Hibernate is easy to integrate with other Java EE frameworks, it's so popular that Spring Framework provides built-in support for integrating hibernate with Spring applications.

Hibernate supports lazy initialization using proxy objects and perform actual database queries only when it's required.

Hibernate cache helps us in getting better performance.

For database vendor specific feature, hibernate is suitable because we can also execute native sql queries.

Following steps are required to integrate Spring and Hibernate frameworks together.

Add hibernate-entitymanager, hibernate-core and spring-orm dependencies.

Create Model classes and corresponding DAO implementations for database operations. Note that DAO classes will use SessionFactory that will be injected by Spring Bean configuration.

If you are using Hibernate 3, you need to configure org.springframework.orm.hibernate3.LocalSessionFactoryBean or org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean in Spring Bean configuration file. For Hibernate 4, there is single class org.springframework.orm.hibernate4.LocalSessionFactoryBean that should be configured.

Note that we don't need to use Hibernate Transaction Management, we can leave it to Spring declarative transaction management using @Transactional annotation.