

1.常见的持久化框架:

1. hibernate

优点:

1. 屏蔽不同数据库 产品中sql语句的细微差异, 增强了程序的可移植性
2. hibernate的api没有侵入性, 业务逻辑不需要集成hibernate的任何接口, hibernate也提供了一级和二级缓存

3. 支持透明的持久化, 延迟加载, 由对象模型自动生成数据库表等.

缺点:

hibernate生成的SQL语句难以优化, spring JDBC和MyBatis直接使用原生的SQL语句, 优化空间大,

2. JPA

JPA(Java Persistence API)EJB3.0中持久化的部分规范, 发展不行

3. spring JDBC

严格来说, spring jdbc 并不能算一个orm框架, 它仅仅是使用模板方式对原生JDBC进行了一层非常薄的封装, 使用Spring JDBC可以帮助开发人员屏蔽创建数据库连接对, statement对象, 异常处理以及事务管理的重复性代码, 提高开发效率.

Spring JDBC中没有映射文件, 对象查询语言, 缓存等概念, 而是直接执行原生SQL语句.

4. mybatis

相较于hibernate, MyBatis更轻量级, 可控性也更高, 可以在映射配置文件中编写待执行的原生SQL语句, 这就给我们直接优化SQL语句的机会, 让SQL语句选择合适的索引, 能更好地提高系统的性能, 适合大数据量, 高并发等场景.

从性能角度来看, Hibernate 生成的SQL 语句难以优化, SpringJDBC 和MyBatis 直接

使用原生SQL 语句, 优化空间比较大, My Batis 和Hibernate 有设计良好的缓存机制, 三者都可以与第三方数据源配合使用: 从可移植性角度来看, H ibernate 帮助开发人员屏蔽了底层数据库方言, 而Spring JDBC 和My Batis 在该方面没有做很好的支持, 但实践中很少有项目会来回切换底层使用的数据库产品, 所以这点并不是特别重要: 从开发效率的角度来看, Hibernate 和My Batis 都提供了XML 映射配置文件和注解两种方式实现映射, Spring JDBC 则是通过ORM化的Callback 的方式进行映射。

2.MyBatis的整体架构

MyBatis的整体架构分为三层, 分别是基础支持层, 核心处理层和接口层



1.基础支持层

基础支持层包含整个MyBatis的基础模块,这些模块为核心处理层的功能提供了良好支持.

2.核心处理层

