# 目录

[目录 1](#_Toc396494577)

[版本修订 2](#_Toc396494578)

[1 纯JDBC 3](#_Toc396494579)

[2 Apache DbUtil 3](#_Toc396494580)

[3 Spring JDBC 3](#_Toc396494581)

[4 Mybatis 3](#_Toc396494582)

[5 Hibernate 3](#_Toc396494583)

[6 JPA 4](#_Toc396494584)

[7 spring data jpa 4](#_Toc396494585)

[8 废话一下事务处理 4](#_Toc396494586)

[9 还有其它的吗？ 4](#_Toc396494587)

# 版本修订

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 描述 | 修订人 |
| V1.0 | 2014-8-19 | 创建 | 陈绪绍 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 纯JDBC

辅以数据源。特点是烦琐，一堆sql点位符参数设置，一堆结果集获取再设置至daomin，一堆事务模板代码，commit，roolback。得考虑线程安全问题。

# Apache DbUtil

相对纯JDBC方便多了，部分用法和mybatis类似，只是呼声没mybatis那么高，也没有提供mybatis中的sql重用功能。当然，自己扩展一下注解，写写工具方法，加上ThreadLocal，也能规避线程安全问题。

# Spring JDBC

Spring JDBC Template等。不赘述了。不用考虑线程安全问题。

# Mybatis

很多互联网公司采用的持久化方案，特点是灵活，自己写sql，表间想怎么关联就怎么关联；上手容易，会sql就行了；也方便以后诸如数据库主从切分这种拓展。不过使用了mybatis，就别想享受hibernate那种全能orm诸如级联插入，级联更新等功能了，当然级联查询还是可以的。官方较全的文档，学习2,3天够了。

# Hibernate

设计好domain的关联关系就可以了。基本不用怎么写sql，操纵的都是save，remove等面向对象api，优点是方便，写一个BaseDao，开发中不用怎么写持久层代码。缺点是优化较难，研究不到一定层次，优化不好。上手易，用好难。

# JPA

比Hibernate更高一层的API。相应的实现有Hibernate，OpenJPA，TopLink等。和Hibernate类似的理念，类似的API，中不过SesssionFactory换成了EntityManagerFactory，Session换成了EntityManager。同样，写一个BaseDao，开发中不用怎么写持久层代码。

# spring data jpa

对JPA原始API进行了再封装。用好Repository，也不用怎么写持久层代码。

# 废话一下事务处理

如果应用中引入了Spring。没有理由不使用Spring的声明式事务处理。使用Spring的声明式事务，大大减少了重复的commit，rollback事务代码，最重要的是一个线程中，能确保你单例的Dao、Service是线程安全的。

# 还有其它的吗？

也许有。