# 延迟加载

多表操作的时候需要延迟加载

<setting name=*"lazyLoadingEnabled"* value=*"true"*/>

<setting name=*"aggressiveLazyLoading"* value=*"false"*/>

全部配置，所有的映射文件都生效

单独的配置，通过属性

fetchType=*"eager"*

*配置积极加载还是懒加载 lazy*

如果说，你要查询的数据，确实都需要，那就最好用一条sql语句关联查询。

他要比两条sql语句性能要高

但是如果只要查询一个表的数据，或者两个表，不确定的情况下。

最好用延迟加载，分步查询 lazy

这个时候虽然发出多条sql语句，内存得到合理利用。

利大于弊

# Spring

是一个业务处理层的框架，相对于mybatis是数据库层的框架

UserService user=new UserService()

CourseServuice course=new CourseService()

关键就是 如果需要改变页面实现

UserService2 CourseService2

在处理业务的时候加重处理

在业务层比较中的一个东西

事务处理

业务是比较复杂的，不止一个sql处理

开始事务

执行sql

执行sql

执行sql

提交事务

关闭连接

使用框架自动给我们开启事务，提交事务，关闭连接等操作

1.管理对象的 ioc 原理工厂模式 对象反射机制

2.在调用之前做什么，之后做什么 aop 动态代理模式

## Ioc

管理对象的

A a=new A()

B b=new B();

控制权在我们这边。

控制权反转，交给spring容器了

Spring容器 在加载的时候就已经实例化对象了。

A a;

getA(){

}

就自动从容器中获得对象了。

Spring 工厂 ，生产对象的一个工厂

IOC/DI

控制反转/依赖注入

原理都是从工厂里面拿对象

依赖注入

不仅从工厂中获取，而且还给你装配对象

什么是spring ?

基本配置 提高项目的开发效率

作用？降低项目的维护成本

Mybatis 算是一个比较简单的

Jdbc 操作

容器中的对象只要一个就可以，单例模式，也是spring默认的一种模式

在spring 中一般是管理无状态的对象，而且是单例模式

只有方法的类 ，单例

Struts和spring 整合的时候，需要多例

Struts中的action对象是多例，action对象也是让spring 管理的。也是多利的

开发模式 23 种

获取bean 两种

通过名字（可以是多次，不同的名字）

通过类型（只能定义一次）

## Aop

管理事务重复操作

## Springmvc

就是mvc的框架

之前我们来一个请求，就要写一个映射配置

而且来一个请求，都要写一个servlert类来处理

配置文件映射太多，servlet类也太多，不便于管理