1. [RESTful](https://www.oschina.net/question/2258299_2192938" \t "http://www.oschina.net/question/tag/_blank)**技术 2.sprintBoot 3. Redis**

**<http://www.banzg.com/archives/639.html>**

# springmvc与struts的区别

1、Struts2是类级别的拦截， 一个类对应一个request上下文，SpringMVC是方法级别的拦截， 一 个方法对应一个request上下文，而方法同时又跟一个url对应,所以说从架构本身上SpringMVC 就容易实现restful url,而struts2的架构实现起来要费劲，因为Struts2中Action的一个方法可以对 应一个url，而其类属性却被所有方法共享，这也就无法用注解或其他方式标识其所属方法了。

2、由上边原因，SpringMVC的方法之间基本上独立的，独享request response数据，请求数据通 过 参数获取，处理结果通过ModelMap交回给框架，方法之间不共享变量，而Struts2搞的就比较 乱，虽然方法之间也是独立的，但其所有Action变量是共享的，这不会影响程序运行，却给我们编 码 读程序时带来麻烦，每次来了请求就创建一个Action，一个Action对象对应一个request上下 文。

下面的说的很对。struts2每次请求的时候都会创建一个action的实例，这样会保证线程的安全。S truts1只是在第一次请求的时候创建一个action的实例，以后每次相同的请求都直接从内存中去读 取，它是[单例模式](https://www.baidu.com/s?wd=%E5%8D%95%E4%BE%8B%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBujI-nvPhuWbzmhuhmvD30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWR3n1f3PWbz" \t "https://zhidao.baidu.com/question/_blank)，安全性不行。

strust2的action里面很多情况下都是需要定义成员变量用来接收表单上的值，如果只是一个对象的 话，那么很可能会出现多个线程共用一个成员变量 出现[线程安全](https://www.baidu.com/s?wd=%E7%BA%BF%E7%A8%8B%E5%AE%89%E5%85%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBujI-nvPhuWbzmhuhmvD30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWR3n1f3PWbz" \t "https://zhidao.baidu.com/question/_blank)问题，因此struts2每次请求都要创 建一个Action对象  
3、由于Struts2需要针对每个request进行封装，把request，session等servlet生命周期的变量 封 装成一个一个Map，供给每个Action使用，并保证线程安全，所以在原则上，是比较耗费内存的。

4、 拦截器实现机制上，Struts2有以自己的拦截器机制，SpringMVC用的是独立的AOP方式，这 样 导致Struts2的配置文件量还是比SpringMVC大。

5、SpringMVC的入口是servlet，而Struts2是filter（这里要指出，filter（过滤器）和servlet是 不 同的。以前认为filter是servlet的一种特殊），这就导致了二者的机制不同，这里就牵涉到servlet 和filter的区别了。

6、SpringMVC集成了Ajax，使用非常方便，只需一个注解@ResponseBody就可以实现，然后直 接 返回响应文本即可，而Struts2拦截器集成了Ajax，在Action中处理时一般必须安装插件或者自 己写 代码集成进去，使用起来也相对不方便。

7、SpringMVC验证支持JSR303，处理起来相对更加灵活方便，而Struts2验证比较繁琐，感觉太烦乱。

8、Spring MVC和Spring是无缝的。从这个项目的管理和安全上也比Struts2高（当然Struts2也 可 以通过不同的目录结构和相关配置做到SpringMVC一样的效果，但是需要xml配置的地方不少）。

9、 设计思想上，Struts2更加符合OOP的编程思想， SpringMVC就比较谨慎，在servlet上扩展。

10、SpringMVC开发效率和性能高于Struts2。  
11、SpringMVC可以认为已经100%零配置

1. **mybatis与hibernate的区别。**

## 两者相同点

* Hibernate与MyBatis都可以是通过SessionFactoryBuider由XML配置文件生成SessionFactory，然后由SessionFactory 生成Session，最后由Session来开启执行事务和SQL语句。其中SessionFactoryBuider，SessionFactory，Session的生命周期都是差不多的。
* Hibernate和MyBatis都支持JDBC和JTA事务处理。

## Mybatis优势

* MyBatis可以进行更为细致的SQL优化，可以减少查询字段。
* MyBatis容易掌握，而Hibernate门槛较高。

## Hibernate优势

* Hibernate的DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。
* Hibernate对对象的维护和缓存要比MyBatis好，对增删改查的对象的维护要方便。
* Hibernate数据库移植性很好，MyBatis的数据库移植性不好，不同的数据库需要写不同SQL。
* Hibernate有更好的二级缓存机制，可以使用第三方缓存。MyBatis本身提供的缓存机制不佳。

## 他人总结

* Hibernate功能强大，数据库无关性好，O/R映射能力强，如果你对Hibernate相当精通，而且对Hibernate进行了适当的封装，那么你的项目整个持久层代码会相当简单，需要写的代码很少，开发速度很快，非常爽。
* Hibernate的缺点就是学习门槛不低，要精通门槛更高，而且怎么设计O/R映射，在性能和对象模型之间如何权衡取得平衡，以及怎样用好Hibernate方面需要你的经验和能力都很强才行。
* iBATIS入门简单，即学即用，提供了数据库查询的自动对象绑定功能，而且延续了很好的SQL使用经验，对于没有那么高的对象模型要求的项目来说，相当完美。
* iBATIS的缺点就是框架还是比较简陋，功能尚有缺失，虽然简化了数据绑定代码，但是整个底层数据库查询实际还是要自己写的，工作量也比较大，而且不太容易适应快速数据库修改。

1. 怎么配置hibernate/mybatis.

1添加jar包。 2编写hibernate.cfg.xml

**[html]** [view plain](http://blog.csdn.net/gwblue/article/details/22691335" \o "view plain) [copy](http://blog.csdn.net/gwblue/article/details/22691335" \o "copy) [IMG_256](https://code.csdn.net/snippets/267224" \o "在CODE上查看代码片" \t "http://blog.csdn.net/gwblue/article/details/_blank)[IMG_257](https://code.csdn.net/snippets/267224/fork" \o "派生到我的代码片" \t "http://blog.csdn.net/gwblue/article/details/_blank)

1. <!DOCTYPE hibernate-configuration PUBLIC
2. "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
3. "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd"**>**
5. **<hibernate-configuration>**
6. **<session-factory** **>**
7. **<property** name="hibernate.connection.driver\_class"**>**com.mysql.jdbc.Driver**</property>**
8. **<property** name="hibernate.connection.url"**>**jdbc:mysql://localhost:3306/hibernate\_frist**</property>**
9. **<property** name="hibernate.connection.username"**>**root**</property>**
10. **<property** name="hibernate.connection.password"**>**root**</property>**
11. **<property** name="hibernate.dialect"**>**org.hibernate.dialect.MySQLDialect**</property>**
12. **<property** name="hibernate.show\_sql"**>**true**</property>**
13. **<property** name="hibernate.format\_sql"**>**true**</property>**
14. **<mapping** resource="com/bjpowernode/hibernate/User.hbm.xml"**/>**
15. **</session-factory>**
16. **</hibernate-configuration>**

1.mybatis-config.xml。

Hibernate主要用来实现Java对象和表之间的映射，除此之外还提供还提供数据查询和

获取数据的方法，可以大幅度减少开发时人工使用SQL和JDBC处理数据的时间

1. jdbc与hibernate的区别，

<http://www.cnblogs.com/JemBai/archive/2011/04/13/2014940.html>

**相同点：**

◆两者都是JAVA的数据库操作中间件。

◆两者对于数据库进行直接操作的对象都不是线程安全的，都需要及时关闭。

◆两者都可以对数据库的更新操作进行显式的事务处理。

**不同点**：

◆使用的SQL语言不同：JDBC使用的是基于关系型数据库的标准SQL语言，Hibernate使用的是HQL(Hibernate query language)语言

◆操作的对象不同：JDBC操作的是数据，将数据通过SQL语句直接传送到数据库中执行，Hibernate操作的是持久化对象，由底层持久化对象的数据更新到数据库中。

◆数据状态不同：JDBC操作的数据是“瞬时”的，变量的值无法与数据库中的值保持一致，而Hibernate操作的数据是可持久的，即持久化对象的数据属性的值是可以跟数据库中的值保持一致的。

解释spring中的IOC与AOP

<http://blog.csdn.net/eson_15/article/details/51090040> //通俗的举例子 很容易看懂 也比较深刻

<http://www.tuicool.com/articles/bquQZf> //这个术语更专业 难懂些

5.什么时候使用hibernate延迟加载

因为内存容量有限 ，为了减少并发量，减少系统资源的消耗，

我们让数据在需要的时候才进行加载，这时我们就用到了懒加载。

比如部门ENTITY和员工ENTITY,部门与员工1对多，如果lazy设置为false，那么只要加载了一个部门的po,就会根据一对多配置的关系把所有员工的po也加载出来。但是实际上有时候只是需要用到部门的信息，不需要用到员工的信息，这时员工po的加载就等于浪费资源。如果lazy设置为true,那么只有当你访问部门po的员工信息时候才回去加载员工的po的信息。

1. **线程与进程的区别**

一个程序至少有一个进程,一个进程至少有一个线程. 进程和线程的主要差别在于它们是不同的操作系统资源管理方式。进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程

1. ajax的用途，工作原理，组成

<http://www.cnblogs.com/SanMaoSpace/archive/2013/06/15/3137180.html>

1. 怎么使用同步请求

Ajax请求默认的都是异步的  
如果想同步 async设置为false就可以（默认是true）  
  
var html = $.ajax({  
  url: "some.php",  
  async: false  
}).responseText;

1. 面向对象的特征

<https://zhidao.baidu.com/question/555117886.html>

10.怎么实现多态

方法的重写Overriding和重载Overloading是Java多态性的不同表现.

接口

11.怎么循环map集合

 遍历方法一 hashmap entrySet() 遍历

 遍历方法二foreach遍历

 遍历方法三 迭代器遍历iterator

getkey()

getvalue()

1. 冒泡排序

<http://blog.csdn.net/cbs612537/article/details/8294960/>

13.九大内置对象

request、response、session、application、out、pagecontext、config、page、exception

<http://www.admin10000.com/Document/152.html>

1. 抽象类与接口的区别

1、抽象类和接口都不能直接实例化，如果要实例化，抽象类变量必须指向实现所有抽象方法的子类对象，接口变量必须指向实现所有接口方法的类对象。

2、抽象类要被子类继承，接口要被类实现。

3、接口只能做方法申明，抽象类中可以做方法申明，也可以做方法实现

4、接口里定义的变量只能是公共的静态的常量，抽象类中的变量是普通变量。

5、抽象类里的抽象方法必须全部被子类所实现，如果子类不能全部实现父类抽象方法，那么该子类只能是抽象类。同样，一个实现接口的时候，如不能全部实现接口方法，那么该类也只能为抽象类。

6、抽象方法只能申明，不能实现。abstract void abc();不能写成abstract void abc(){}。

7、抽象类里可以没有抽象方法

8、如果一个类里有抽象方法，那么这个类只能是抽象类

9、抽象方法要被实现，所以不能是静态的，也不能是私有的。

10、接口可继承接口，并可多继承接口，但类只能单根继承

<http://www.cnblogs.com/azai/archive/2009/11/10/1599584.html> //我全部认真地看完了 还是有蛮大收获的

1. session的销毁

<http://blog.csdn.net/u012554102/article/details/49454797> //这个写得还不错

1.session的持有者(即客户端浏览器)在最大无活动等待时间(MaxInactiveInterval)内无任何响应或请求   
2.session被调用invalidate()方法强制销毁了   
3.jsp服务器重起或发生中断(此时应该是所有在服务器上的session全都OVER了)

16.get与post的区别

1. get是从服务器上获取数据，post是向服务器传送数据。  
2. get是把参数数据队列加到提交表单的ACTION属性所指的URL中，值和表单内各个字段一一对应，在URL中可以看到。post是通过HTTP post机制，将表单内各个字段与其内容放置在HTML HEADER内一起传送到ACTION属性所指的URL地址。用户看不到这个过程。  
3. 对于get方式，[服务器端](https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4nyN-mWbsmHuWry7-nW6s0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErjTdn1RLrH6" \t "https://zhidao.baidu.com/question/_blank)用Request.QueryString获取变量的值，对于post方式，[服务器端](https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4nyN-mWbsmHuWry7-nW6s0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErjTdn1RLrH6" \t "https://zhidao.baidu.com/question/_blank)用Request.Form获取提交的数据。  
4. get传送的数据量较小，不能大于2KB。post传送的数据量较大，一般被默认为不受限制。但理论上，IIS4中最大量为80KB，IIS5中为100KB。  
5. get安全性非常低，post安全性较高。但是执行效率却比Post方法好。   
  
建议：  
1、get方式的安全性较Post方式要差些，包含机密信息的话，建议用Post数据提交方式；  
2、在做数据查询时，建议用Get方式；而在做数据添加、修改或删除时，建议用Post方式；

1. oracle数据库的导入导出

imp

Exp

<http://www.cnblogs.com/fjfzhkb/archive/2007/09/03/879807.html>

1. **string与stringbuffer的区别**

简单地说，就是一个变量和常量的关系。StringBuffer对象的内容可以修改；而String对象一旦产生后就不可以被修改，重新赋值其实是两个对象。  
StringBuffer的内部实现方式和String不同，StringBuffer在进行字符串处理时，不生成新的对象，在内存使用上要优于String类。所以在实际使用时，如果经常需要对一个字符串进行修改，例如插入、删除等操作，使用StringBuffer要更加适合一些。  
String:在String类中没有用来改变已有字符串中的某个字符的方法，由于不能改变一个java字符串中的某个单独字符，所以在JDK文档中称String类的对象是不可改变的。然而，不可改变的字符串具有一个很大的优点:编译器可以把字符串设为共享的。   
StringBuffer:StringBuffer类属于一种辅助类，可预先分配指定长度的内存块建立一个字符串缓冲区。这样使用StringBuffer类的append方法追加字符 比 String使用 + 操作符添加字符 到 一个已经存在的字符串后面有效率得多。因为使用 + 操作符每一次将字符添加到一个字符串中去时，字符串对象都需要寻找一个新的内存空间来容纳更大的字符串，这无凝是一个非常消耗时间的操作。添加多个字符也就意味着要一次又一次的对字符串重新分配内存。使用StringBuffer类就避免了这个问题。  
StringBuffer是线程安全的，在多线程程序中也可以很方便的进行使用，但是程序的执行效率相对来说就要稍微慢一些。  
StringBuffer的常用方法  
StringBuffer类中的方法要偏重于对字符串的变化例如追加、插入和删除等，这个也是StringBuffer和String类的主要区别。

<http://blog.csdn.net/lclai/article/details/6141548>

1. 一级缓存与二级缓存的区别

1.一级缓存是针对session级别的，当这个session关闭后这个缓存就不存在了。  
2.二级缓存是SessionFactory级别的，二级缓存我们通常使用其他的一些开源组件，比如hibernate经常使用的就是ECache，这个缓存在整个应用服务器中都会有效的。

<http://bbs.csdn.net/topics/360040271>

20，oracle与mysql的分页机制

select \* from(select rownum r,p.from pet p where rownum＜＝(当前页码\*每页显示条数))ps where ps.r＞(( 当前页码-1 )\*每页条数)

SELECT \* FROM

(

SELECT A.\*, ROWNUM RN

FROM (SELECT \* FROM TABLE\_NAME) A

WHERE ROWNUM <= 40

)

WHERE RN >= 21

oracle：select \* from (select rownum rn,a.\* from fnd\_user a where rownum < 21) where rn > 9

limit

rownum

21. ArryList,LinkedList/vector区别

·   
ArrayList是最常用的List实现类，内部是通过数组实现的，它允许对元素进行快速随机访问。数组的缺点是每个元素之间不能有间隔，当数组大小不满足时需要增加存储能力。就要将已经有数组的数据复制到新的存储空间中，增长的大小，在参考多篇文章后，应该是增加一半。当从ArrayList的中间位置插入或者删除元素时，需要对数组进行复制、移动、代价比较高。因此，它适合随机查找和遍历，不适合插入和删除。

· Vector与ArrayList一样，也是通过数组实现的，不同的是它支持线程的同步，即某一时刻只有一个线程能够写Vector，避免多线程同时写而引起的不一致性，但实现同步需要很高的花费，因此，访问它比访问ArrayList慢。Vector允许用户设置capacityIncrement，这样在每次需要扩充数组的size的时候, Vector会尝试按照预先设置的capacityIncrement作为增量来增加空间。如果capacityimcrement并未设定，默认是增加一倍。

· LinkedList是用链表结构存储数据的，很适合数据的动态插入和删除，随机访问和遍历速度比较慢。另外，他还提供了List接口中没有定义的方法，专门用于操作表头和表尾元素，可以当作堆栈、队列和双向队列使用。在详细了解过后，可以发现，其实LinkedList与ArrayList相比，末尾增加元素，随机访问元素，末尾删除元素均没有性能优势，只有在开始处增删元素时，才有性能优势。

22.怎么判断一个整型数据是否是2的n次幂

1. struts2的工作原理

<http://duanxx.iteye.com/blog/1662918>

<http://www.cnblogs.com/langtianya/archive/2013/04/09/3011090.html> //感觉这个好像清晰点

24.请求转发与重定向的区别

1，请求重定向：客户端行为，response.sendRedirect(),从本质上讲等同于两次请求，前一次的请求对象不会保持，地址栏的URL地址会改变。

2，请求转发：服务器行为，request.getRequsetDispatcher().forward(requset,response);是一次请求，转发后请求对象会保存，地址栏的URL地址不会改变。（服务器内部转发，所有客户端看不到地址栏的改变）

25.js的继承

prototype的方式

apply的方式：

call+prototype的方式：

<http://blog.csdn.net/wyyfwm/article/details/46349071>

26，拦截器与过滤器的区别

a,拦截器是基于java反射机制的，而过滤器是基于函数回调的.

b,拦截器只对Action起作用，而过滤器几乎可以对所有请求起作用。

c,拦截器可以访问action上下文，值栈里的对象，而过滤器不能。

d,在action的生命周期里，拦截器可以多次调用，而过滤器只能在容器初始化时调用一次

e,过滤器依赖于servlet容器，而拦截器不需要

27.oracle与mysql怎么取当前时间

o:sysdate

m:now() `'YYYY-MM-DD HH:MM:SS

curdate YYYY-MM-DD’

curtime HH:MM:SS

28，索引的作用

为什么要创建索引呢？这是因为，创建索引可以大大提高系统的性能。   
第一，通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。   
第二，可以大大加快 数据的检索速度，这也是创建索引的最主要的原因。   
第三，可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。   
第四，在使用分组和排序 子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。   
第五，通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。

也许会有人要问：增加索引有如此多的优点，为什么不对表中的每一个列创建一个索引呢？这种想法固然有其合理性，然而也有其片面性。虽然，索引有许多优点， 但是，为表中的每一个列都增加索引，是非常不明智的。这是因为，增加索引也有许多不利的一个方面。

第一，创建索引和维护索引要耗费时间，这种时间随着数据 量的增加而增加。   
第二，索引需要占物理空间，除了数据表占数据空间之外，每一个索引还要占一定的物理空间，如果要建立聚簇索引，那么需要的空间就会更大。   
第三，当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，这样就降低了数据的维护速度

29，数组与集合的区别

 一、数组声明了它容纳的元素的类型，而集合不声明。

       二、数组是静态的，一个数组实例具有固定的大小，一旦创建了就无法改变容量了。而集合是可以动态扩展容量，可以根据需要动态改变大小，集合提供更多的成员方法，能满足更多的需求。

       三、数组不论是效率还是类型检查都是最好的。

1.数组是大小固定的,一旦创建无法扩容;集合大小不固定,

2.数组的存放的类型只能是一种,集合存放的类型可以不是一种(不加泛型时添加的类型是Object);

3.数组是java语言中内置的数据类型,是线性排列的,执行效率或者类型检查(不懂),都是最快的.

  ArrayList就是基于数组创建的容器类.

30，一个jsp引用另一个jsp有哪些方法

(1)include指令 include指令告诉容器：复制被包含文件汇总的所有内容，再把它粘贴到这个文件中。 <%@ include file="Header.jsp"%>

(2)include标准动作

<jsp:include page=“Header.jsp”/>

(3)采用JSTL

<c:import url="http://www.sina.com/index.html">

31，有个student表，重复的名字

select name from student group by name  having count(name)>1

32，集合，剔除重复

List<string> li1 = new List<string> { "8", "8", "9", "9" ,"0","9"};

HashSet<string> hs = new HashSet<string>(li1); //此时已经去掉重复的数据保存在hashset中

第二种 自己写逻辑 for循环比较

for (int i = 0; i < li2.Count; i++) //外循环是循环的次数 {

for (int j = li2.Count - 1 ; j > i; j--) //内循环是 外循环一次比较的次数 {

if (li2[i] == li2[j])

{

li2.RemoveAt(j);

}

}

}

33，什么设计模式实现了spring中的AOP

代理模式

34.对事务的理解

http://jingyan.baidu.com/article/414eccf6763aa56b431f0acf.html

查询某天最大

查询前5条

34.mybatis的set注入

35，jQuery中移除类与添加类

用addClass()与removeClass()来动态的添加或移出一个css类

例如：

1.$(“#para1”).addClass('highlight');添加一个“highlight”css 类给id为para1的元素。

2.$(‘#para1’).removeClass(‘'highlight');从id为para1的元素中移出一个‘highlight’css类。

36，jQuery中去除空格

jQuery中使用trim去掉字符串两端的空格及制表符

jQuery.trim(str)

描述: 去掉字符串起始和结尾的空格。

$.trim()函数会移除字符串开始和结尾处的所有换行符，空格(包括连续的空格)和制表符（tab）。如果这些空白字符在字符串中间时，它们将被保留，不会被移除。

如果要去掉其他位置的所有空格，可以使用原生的replace方法替换空格

37.文件上传，下载

38，导成Excel表格

39，将a表中的数据导入到b表中去

1》insert into B select \* from A where ID not in (select ID from B)

ID为可以标识一行的列，换成你自己的

2》insert into 表B(列名1) select 列名2 from 表A

列名1：指定复制到B表中的哪一列

列名2：指复制A表中的哪一列

在Oracle中从表B复制数据到表A

有两张表，A表中有a，b，c字段，B表也有a，b，c字段但是比A表还多了d，e两个字段，现在要把B表中的数据复制到A表中，还要把d，e两个字段也加到A表中？这个SQL语句怎么写啊？A表和B表已经存在了(Oracle)

--先在A中添加两个字段

alter table A add d varchar2(100);

alter table A add e varchar2(100);

insert into A (a,b,c,d,e) select a,b,c,d,e from B;

commit;

40，a:(id.name.bid) (1.a.1) (2.b.3)

b(id.address) (1.长沙)（2，上海）（3，株洲）

select\*from a，b

41，spring的常用注解

使用注解来构造IoC容器

用注解来向Spring容器注册Bean。需要在applicationContext.xml中注册<context:component-scan base-package=”pagkage1[,pagkage2,…,pagkageN]”/>。

如：在base-package指明一个包

1<context:component-scan base-package="cn.gacl.java"/>

表明cn.gacl.java包及其子包中，如果某个类的头上带有特定的注解【@Component/@Repository/@Service/@Controller】，就会将这个对象作为Bean注册进Spring容器。也可以在<context:component-scan base-package=” ”/>中指定多个包，如：

1<context:component-scan base-package="cn.gacl.dao.impl,cn.gacl.service.impl,cn.gacl.action"/>

多个包逗号隔开。

1、@Component

@Component

是所有受Spring 管理组件的通用形式，@Component注解可以放在类的头上，@Component不推荐使用。

2、@Controller

@Controller对应表现层的Bean，也就是Action，例如：

1@Controller

2@Scope("prototype")

3publicclassUserAction extendsBaseAction<User>{

4……

5}

使用@Controller注解标识UserAction之后，就表示要把UserAction交给Spring容器管理，在Spring容器中会存在一个名字为"userAction"的action，这个名字是根据UserAction类名来取的。注意：如果@Controller不指定其value【@Controller】，则默认的bean名字为这个类的类名首字母小写，如果指定value【@Controller(value="UserAction")】或者【@Controller("UserAction")】，则使用value作为bean的名字。

这里的UserAction还使用了@Scope注解，@Scope("prototype")表示将Action的范围声明为原型，可以利用容器的scope="prototype"来保证每一个请求有一个单独的Action来处理，避免struts中Action的线程安全问题。spring 默认scope 是单例模式(scope="singleton")，这样只会创建一个Action对象，每次访问都是同一Action对象，数据不安全，struts2 是要求每次次访问都对应不同的Action，scope="prototype" 可以保证当有请求的时候都创建一个Action对象

3、@ Service

@Service对应的是业务层Bean，例如：

1@Service("userService")

2publicclassUserServiceImpl implementsUserService {

3………

4}

@Service("userService")注解是告诉Spring，当Spring要创建UserServiceImpl的的实例时，bean的名字必须叫做"userService"，这样当Action需要使用UserServiceImpl的的实例时,就可以由Spring创建好的"userService"，然后注入给Action：在Action只需要声明一个名字叫“userService”的变量来接收由Spring注入的"userService"即可，具体代码如下：

1//注入userService

2@Resource(name = "userService")

3privateUserService userService;

注意：在Action声明的“userService”变量的类型必须是“UserServiceImpl”或者是其父类“UserService”，否则由于类型不一致而无法注入，由于Action中的声明的“userService”变量使用了@Resource注解去标注，并且指明了其name = "userService"，这就等于告诉Spring，说我Action要实例化一个“userService”，你Spring快点帮我实例化好，然后给我，当Spring看到userService变量上的@Resource的注解时，根据其指明的name属性可以知道，Action中需要用到一个UserServiceImpl的实例，此时Spring就会把自己创建好的名字叫做"userService"的UserServiceImpl的实例注入给Action中的“userService”变量，帮助Action完成userService的实例化，这样在Action中就不用通过“UserService userService = new UserServiceImpl();”这种最原始的方式去实例化userService了。如果没有Spring，那么当Action需要使用UserServiceImpl时，必须通过“UserService userService = new UserServiceImpl();”主动去创建实例对象，但使用了Spring之后，Action要使用UserServiceImpl时，就不用主动去创建UserServiceImpl的实例了，创建UserServiceImpl实例已经交给Spring来做了，Spring把创建好的UserServiceImpl实例给Action，Action拿到就可以直接用了。Action由原来的主动创建UserServiceImpl实例后就可以马上使用，变成了被动等待由Spring创建好UserServiceImpl实例之后再注入给Action，Action才能够使用。这说明Action对“UserServiceImpl”类的“控制权”已经被“反转”了，原来主动权在自己手上，自己要使用“UserServiceImpl”类的实例，自己主动去new一个出来马上就可以使用了，但现在自己不能主动去new“UserServiceImpl”类的实例，new“UserServiceImpl”类的实例的权力已经被Spring拿走了，只有Spring才能够new“UserServiceImpl”类的实例，而Action只能等Spring创建好“UserServiceImpl”类的实例后，再“恳求”Spring把创建好的“UserServiceImpl”类的实例给他，这样他才能够使用“UserServiceImpl”，这就是Spring核心思想“控制反转”，也叫“依赖注入”，“依赖注入”也很好理解，Action需要使用UserServiceImpl干活，那么就是对UserServiceImpl产生了依赖，Spring把Acion需要依赖的UserServiceImpl注入(也就是“给”)给Action，这就是所谓的“依赖注入”。对Action而言，Action依赖什么东西，就请求Spring注入给他，对Spring而言，Action需要什么，Spring就主动注入给他。

4、@ Repository

@Repository对应数据访问层Bean ，例如：

1@Repository(value="userDao")

2publicclassUserDaoImpl extendsBaseDaoImpl<User>{

3………

4}

@Repository(value="userDao")注解是告诉Spring，让Spring创建一个名字叫“userDao”的UserDaoImpl实例。

当Service需要使用Spring创建的名字叫“userDao”的UserDaoImpl实例时，就可以使用@Resource(name = "userDao")注解告诉Spring，Spring把创建好的userDao注入给Service即可。

1//注入userDao，从数据库中根据用户Id取出指定用户时需要用到

2@Resource(name = "userDao")

3privateBaseDao<User> userDao;

a＝1 b＝2

a＝a+b

b＝a-b

a＝a-b

1+1+2+3+5+8+13+……求n项和用java程序实现

有一桶糖果有y颗，每天吃一半加一个，第

x天只剩下一个，求y＝f（x）f的关系用java程序实现

public static void main(String[] args) {

 int lastnum =1;

 int days = 10;//第

 for(int i = 1;i<=x;i++){

 lastnum=2\*(lastday+1);

 }

 //结果是3070个桃子

 System.out.println("一共"+lastday+"个桃子。");

}

public static void main(String[] args) {

 int lastday =1;

 int days = 10;//第十天

 for(int i = 1;i<=days;i++){

 lastday=2\*(lastday+1);

 }

 //结果是3070个桃子

 System.out.println("一共"+lastday+"个桃子。");

}

￼

◦