# 一、基 础 面 试 题

## String类中常用的方法?

split()：把字符串分割成字符串数组

indexOf()：从指定字符提取索引位置

replace()：替换

subString()：截取字符串

equals()：比较

concat()：将指定字符串连接到此字符串的结尾

用string类里的哪一个方法替换掉第五个元素?

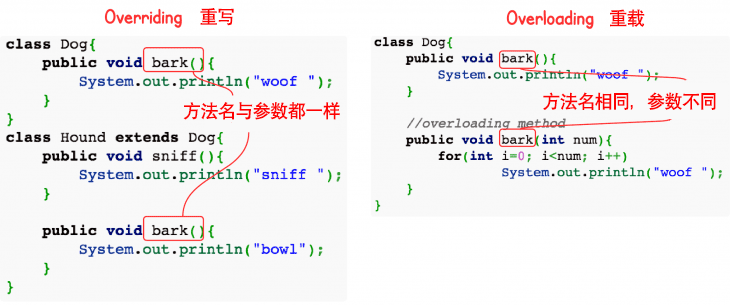
String str = new String("helloWorld");

System.out.println(str.replace(5, 'T'));

## 什么是重载和重写？

我理解得Java里方法的重写(Overriding)和重载(Overloading)就是指的java里多态，重写就是父类与子类之间的多态，重载是在一个类中多态的体现(1)方法重载是一个类中定义了多个方法名相同,而他们的参数的数量不同或数量相同而类型和次序不同,则称为方法的重载(Overloading)。

(2)方法重写是在子类存在方法与父类的方法的名字相同,而且参数的个数与类型一样,返回值也一样的方法,就称为重写(Overriding)。



## String中的==和equals的区别?

==比较的是内存地址, equals比较的是内容String a=”abc” String b=”abc” System.out.println(a==b) 输出的是true.String a= new String(“abc”),String b=new String(“abc”) System.out.println(a==b) 输入的是false

## **[Java中的String，StringBuilder，StringBuffer三者的区别](https://www.cnblogs.com/su-feng/p/6659064.html)?**

在开发中String我们主要用于少量的字符串操作,因为他是fina修饰的，所以它修饰的都是不可变字符串.StringBuffer :我们主要用于可变的字符串中,我们在拼接Sql语句的时候用的,还有就是他是线程安全的，在字符串后面拼接的方法是append.拼接内容，Stringbuilder也是拼接字符串的，初始化的时候创建一个对象，使用append再进行拼接的时候，还是在原有的这个对象中拼接新的字符串，而不是创建新对象。

## 5.Final关键字?final和finally的区别?

其实Final是一个安全修饰符,就是用final修饰的类不能被继承,用final声明的方法不能被重写,使用final声明的变量就相当于常量,不能被修改。

Finally是在异常里经常用到的, 就是try和cach里的代码执行完以后,必须要执行的方法,我们经常在finally里写一些关闭资源的方法,比如说关闭数据库连接,或者关闭IO流什么的.

## 6.Java里可不可以有多继承？

不可以，想多继承的话怎么办？ 用接口

## 7.int和Integer的区别？

Int是基本数据类型呀Integer是java为int提供的封装类，是引用数据类型；

Int的默认值为0，而integer的默认值为null

## 8.Java的三大特性

这个java的三大特性无非就是封装，继承，多态嘛

封装体现到我们把方法和变量都封装到一个类里,直接用对象调用就可以了

继承的体现是我们可以复用父类里的代码,我们在项目中可以把公有的方法写到base的一个类中,让其他类继承就可以了.

多态就是Java里的继承，重写，重载,都是多态，父类引用指向子类对象的时候，那是对象向上转型，,比如加号如果两边是数字的化,那就是一个运算符,如果两边是字符串的话,那他就是个连接符.

## 9.java arrayList的存储结构,初始化的时候创建多大的数组?

ArrayList是基于数组实现的，是一个动态数组,容量能自动增长，初始化长度是10, 扩容规则: 扩容后的大小= 原始大小\*1.5。

ArrayList是线程不安全的，只能用在单线程环境下，多线程环境下可以考虑用Collections.synchronizedList(List l)函数返回一个线程安全的ArrayList类，也可以使用concurrent并发包下的CopyOnWriteArrayList类。

## 10.ArrayList与LinkedList区别

ArrayList使用数组方式存储数据，所以根据索引查询数据速度快，而新增或者删除元素时需要设计到位移操作，所以比较慢。

LinkedList使用双向链接方式存储数据，每个元素都记录前后元素的指针，所以插入、删除数据时只是更改前后元素的指针指向即可，速度非常快，然后通过下标查询元素时需要从头开始索引，所以比较慢，但是如果查询前几个元素或后几个元素速度比较快。开发中什么时候到ArrayList?,我们在做查询的时候把查询出来的数据经常存到arraylist里.

## 11.抽象类和接口的区别?

抽象类里可以有普通方法也可以有抽象方法,接口里只能有抽象方法,但是在jdk1.8以后也可以在接口中添加普通方法和静态方法,1.9以后还可以定义私有方法，定义默认方法就是用default修饰一下这个方法,那接口和抽象类就没什么区别了,为什么还要有抽象类? 接口的好处是解决java里的面不能多继承的问题,可以降低程序的耦合度,如果不需要多继承的话那就可以用抽象类就行.

## 12.你了解过哪些数据结构

我差不多了解过五种吧，比如说栈：这种数据结构的特点是，先进后出，就像我们给弹夹里填子弹一样，最后添进去的最先出来。

还有队列：先进先出，后进的后出。

还有数组：数组的特点是查询快，增删慢，因为他查询的时候首地址是连续的，也可以通过索引快速定位到某一个元素，但是删除的话就慢，因为删除得重新创建一个数组，大小得-1，把原来数组中的剩余的元素再复制过来，比较慢一些

链表：查询慢，增删快；列表分为单向列表和双向列表，单向列表呢是无序的，双向列表呢是专门有一条链记录顺序的，列表中的每一个节点都存三个元素，自己的内存地址，元素，下一个节点的内存地址。

树：了解过二叉树，排序树,平衡树，红黑树；

二叉树：每个节点下都有两个子节点

排序树：根据排列顺序进行折半查找节点中的元素

平衡树：每个节点和每个节点所拥有的子节点都是相同的

红黑树：红黑树其实近似于平衡树，就是节点是红色或者黑色的，根节点必须是黑色的，叶子节点也是黑色的，查询的时候会很快。

## 13.jdk1.8 和1.7区别

我了解JDK8的新特性有:就是接口里也可添加普通的方法了,不用非得是抽象方法了,但是必须得用default进行修饰,调用的时候也还是得实现这个接口以后才能调用,还有就是添加了一个Lambda 表达式,让我们遍历集合数据的时候速度更快,还有就是JDK1.8以后的map存储方式是1.7的也有些区别.如果不相等则形成链表结构，jdk1.7后加的在前面，先加的移下，这种情况叫碰撞。这种碰撞的情况应尽量避免，否存一个索引中链表的数据大量时，该索引当再次插入一个对象时equals比较全部影响效率。

因此jdk1.8改善这种碰撞情况的出现，jdk1.8中的HashMap存储结构是由数组、链表、红黑树这三种数据结构形成，红黑树查询删除快新增慢。参考文章https://blog.csdn.net/changhangshi/article/details/82114727

## 14.HashMap和Hashtable的区别

这两个都是是Map接口下的实现类，我们开发的时候经常用的是HashMap虽然HashMap不是线程安全的但是他的存储效率比较高,Hashtable是线程安全我们看过他底层的put方法前面加的synchronized关键字,但是他的效率太低,我们开发很少用.**为什么说hashTable是线程安全的?** 因为看源码的时候里面的put方法前面加了synchronized关键字,**什么情况下用hashMap?** 我们有时候在自定义返回json字符串的时候,封装到List中进行返回.**你知道Hashmap的底层实现吗?** 参考后面的答案

•

通过Collections.synchronizedMap()返回一个新的Map,这个新的map就是线程安全的。 看java.util.concurrent.ConcurrentHashMap.

## 15.解决hashmap线程不安全问题?

可以通过collections集合工具类对不安全的线程进行包装，使其变成线程安全的，也可以在使用其时加synchronized关键字进行同步,但是查询效率会降低

## 16.Hashmap的底层实现原理?扩容机制?

这个我在论坛上看过,HashMap底层是数组+链表+红黑树实现,hashMap默认初始化容量为16，也就是说数组索引值为0-15，每个数组中存储一个链表当我们创建hashmap时会先创建一个数组，当我们用put方法存数据时，先根据key的hashcode值计算出hash值，然后用这个哈希值确定在数组中的位置，再把value值放进去，如果这个位置本来没放东西，就会直接放进去，如果之前就有，就会生成一个链表，把新放入的值放在尾部，当用get方法取值时，会先根据key的hashcode方法获取到hash值，确定位置，再根据equals方法从该位置上的链表中取出该value值。

## 17.hash碰撞怎么产生，怎么解决？

这道题我正好在上家面试的时候问到过,当时我也不会,我回去专门上网查了一下,JDK解决方法是链地址法（拉链法）：创建一个链表数组，数组中每一格就是一个链表。若遇到哈希冲突，则将冲突的值加到链表中即可,链表中的节点大于8个后转换成红黑树。（java 里的 hashMap 就是使用的这种方法）

## 18.如何遍历Map集合?

有两种遍历方式,一种是使用keyset把map中的key中全部都存储到set集合中,还有一种是使用Map的那个内部类Set<Map.Entry<K,V>> entrySet() 返回Entry对象,里面存的key和value

## 19.说五个开发中常见的异常？

我们开发经常会遇到比如说IOException, NullpointException, FileNotException,SQLException,ClassNotFoundException,ClassCastException, pathException,还有公司里有时候会有自己的异常框架,直接继承

## 20.什么是接口?有没有做过接口开发?

你是说的前后端交互的接口是吧,我主要就是做Java后台开发的,我们开发的时候有一个接口文档,我们按照接口文档里规定的返回类型参数请求的URL进行开发就行,我们一般用postman做的测试,接口文档里主要包括什么?请求的URL，参数，返回值类型，请求方式get，post,put,delete,**以下接口文档不用背,只是让看下知道什么是接口文档,有印象就行**

|  |  |
| --- | --- |
| **请求方法** | **GET** |
| **URL** | http://sso.taotao.com/user/check/{param}/{type} |
| **参数说明** | 格式如：zhangsan/ 1，其中zhangsan是校验的数据，type为类型，可选参数1、2、3分别代表username、phone、email  可选参数callback：如果有此参数表示此方法为jsonp请求，需要支持jsonp。 |
| **示例** | http://sso.taotao.com/user/check/zhangsan/1 |
| **返回值** | {  status: 200 //200 成功  msg: "OK" // 返回信息消息  data: false // 返回数据，true：数据可用，false：数据不可用  } |

## 21.IO和NIO的区别?

这个NIO是JDK1.7以后有的,它们俩的主要区别是:io 是面向流是阻塞io,nio是面向缓冲,非阻塞的io;

io的话每次从流中读取一个或多个字节,直到读取完所有的字节,没有缓存到任何地方.nio读取的是数据是有缓存,就是说他读取的数据是在缓冲里读的. 另外的话,java中的各种io是阻塞的.就是说一个线程调用read或者write()时,这个线程就已经被阻塞了,直到读取到一些数据为止,或者是完全写入.在这个过程中不能干其他的事情. nio的非阻塞模式,当发送一个读取数据的请求的时候,如果没有读取到可用的数据,就什么也不会获取,且不会让线程阻塞写也是这样.非阻塞的IO的空闲时间可用用来做其他的操作所以,一个单独的非阻塞线程可以管理多个输入和输出通道,另外NIO还有一个selector(选择器),它是可以管理多个输入输出的通道.我大概了解的也就是这样.

## 22.在Java中要想实现多线程代码有三种手段?

一种是继承Thread类

另一种就是实现Runnable接口,这样的好处是解决的Java类里不能多继承的问题

**推荐实现多线程的方法--实现Runnable接口**

原因:

(1).Thread类中定义了多种方法可以被派生类使用或重写，但是只有run()方法必须被重写的，在run()方法中实现啊这个线程的主要功能,这就是Runnable接口所需实现的方法

(2).通过继承Thread的实现方法与实现Runnable接口的效果相同，并且Java只能是单继承、多实现，如果一个类中已经继承其他所需的类，那实现一个接口是必须的。

第三种是实现Callable接口

实现Callable接口，重写call()方法

Callable接口是属于Executor，对比与Runnable接口功能的区别是:

(1).Callable可以在任务结束后提供一个返回值，Runnable没有这个功能

(2).Callable中的call()方法可以抛出异常，而Runnable的run()方法不能抛出异常

(3).运行Callable可以拿到一个Future对象，Future独享表示异步计算的结果，它提供了

检查计算是否完成的方法。由于线程属于异步计算模型，因此无法从别的线程中得到函数

的返回值，在这种情况下，就可以使用Future来监视目标线程调用call()方法的情况，

当调用Future的get()方法以获取结果时，当前线程就会阻塞，直到call()方法结束返回结果。

## 23.线程的状态

其实线程一般具有六种状态，即新建、正在运行、计时等待、永久等待,阻塞、终止。

新建状态

线程创建好

正在运行状态

调用了start方法状态

阻塞状态

加了同步锁的代码,多线程进行访问的时候,会有阻塞状态

无限等待状态

调用了wait方法,无限等待了,等待notify或者notifyAll方法进行唤醒

计时等待状态

自动唤醒,相当于使用了sleep(1000) 或者wait(传递时间);

死亡(结束)

不用结束线程,执行完run方法里的代码后,会自动结束线程

## 24.线程同步,什么是锁?

实现线程同步有三种方式：

1. **同步代码块**：在代码块上加上“synchronized”关键字的话，则此代码块就称为同步代码块。

**同步代码块格式：**

synchronized(监视对象){

需要同步的代码 ;

}

解释：监视对象有三种：对象、String、.class文件（只要是不变的对象都可以做监视对象）

1. **同步方法**

**同步方法定义格式：**

synchronized 方法返回值 方法名称(参数列表){}

在方法上加synchronized，是把当前对象做为监视器

1. **同步锁**

Lock lock = new ReentrantLock();(可以在类中直接new)

有可能出现线程不安全的代码前面加lock.lock();执行完以后释放锁使用lock.unlock();方法;

## 25.多线程怎么解决高并发？Java中怎么使用的多线程

使用多线程解决高并发的话,我们使用的是线程池的方式,我们使用的是Spring给提供的那个线程池的技术,里面配置最小连接数,最大连接数,还有队列数,还有使用默认的拒绝策略,队列数的意思是超过最大连接数以后会进入到队里里进行排队,如果在使用多线程有可能产生数据不一致的情况,我们还会使用ThreadLocal来解决,ThreadLocal和Synchonized都用于解决多线程并发访问但是ThreadLocal与synchronized有本质的区别:  
    synchronized是利用锁的机制，使变量或代码块在某一时该只能被一个线程访问。而ThreadLocal是为每一个线程都提供了变量的副本，使得每个线程在某一时间访问到的并不是同一个对象，这样就隔离了多个线程对数据的数据共享。而Synchronized却正好相反，它用于在多个线程间通信时能够获得数据共享。

## 26.什么是多线程?在哪里应用过?

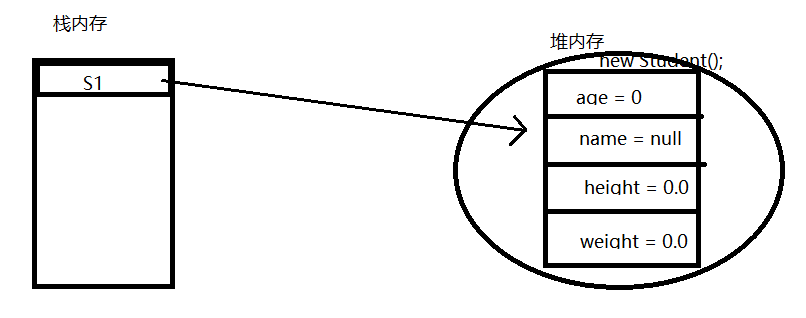
线程其实就是合理使用CPU资源,多几个通道来执行程序，其实main方法就是主线程，用继承了Thread类的里run方法里写代码逻辑,再用start启动就是多线程了，我们项目中的多线程是用的spring给提供的线程池技术,线程池就是把准备好的线程放到线程池里，如果处理请求需要调用线程的时候，就从线程池里去调用，用完以后再放回到线程池里，这样就能提高程序的执行效率,还有我们现在解决这些高并发除了多线程以外也配合中间件来解决,redis啦,activeMQ啦,还有solr等等, 然后把这几个中间件的原理说一下....

## 27.jvm内存分析? 解释下堆和栈? 解释下虚拟机?

我原来学java的时候知道JVM内存结构主要有三大块：堆内存、方法区和栈。堆内存是JVM中最大的一块内存地址,它主要由年轻代和老年代还有持久代组成,所有new出来的对象都存储在该区域. 栈就是暂存数据的地方,每个线程包含一个栈区，栈存放在一级缓存中，存取速度较快,栈中只保存基础数据类型的对象和自定义对象的引用.每个栈中的数据都是私有的，其他栈不能访问。方法区存放了要加载的类的信息（如类名、修饰符等）、静态变量、构造函数、final定义的常量、类中的字段和方法等信息。

声明并创建对象

1. 访问属性



## 28.自定义Annotation,自定义注解

以前写过,用@interface定义就行, 我记得里面定义属性返回类型必须都是基本数据类型,或者String,或者枚举,什么的,我们一般都是用的第三方带的注解,很少自己写.

## 29.有没有遇到过内存溢出，内存泄漏?有没有遇到遇到过OOM?

开发的时候遇到过,其实内存溢出指的是jvm内存溢出，通过调整堆，栈的大小来解决。代码中出现死循环或递归调用也会造成内存溢出,一般我们去调整一下他的内存大小就行,配置一下tomcat里的一个参数就能调整,一般造成内存溢出的原因是什么? 一方面有可能是代码中有长时间没有关闭的连接,或者是用IO读取完文件以后没有及时关闭,原来再使用hibernate的时候就遇到过内存溢出,就是因为没有及时的去关闭缓存造成的,还有就是代码问题,比如说写了死循环,程序出问题,递归自己调用自己的时候代码写的有问题都有可能造成内存溢出.我感觉内存泄漏和内存溢出差不多.

## 30.说一下GC垃圾回收过程

我们Java中的垃圾回收都是自动的,我们很少几乎不去手动的干预,我就是先说一下这个JVM中的的堆内存结构,他主要分为新生代和老年代，新生代就是用来存放刚被new出来 的对象，一般情况下占堆的1/3空间,还有就是新生代中又分为3个区具体我也记不清是哪三个区了 反正老年代里存放咱们整个应用程序中生命周期长的内存对象,我原来再网上看过,我们的JVM垃圾回收算法我记得差不多有4种,标记清除法,复制算法,标记整理算法,分代收集算法,我就了解其中的两种,比如说 说清楚其中的两个就行,都说出来就有点太假了,根本不可能记不住那么多

第一种就是标记-清除法

分为：标记，请除

标记阶段：直接在内存里标记无用的对象，然后清除阶段直接回收被标记的对象；

缺点：形成内存碎片,一些大的对象无法找到足够的空间而触发新的垃圾收集动作。

第二种：复制算法：

将内存划分为大小相等的两块，当一块的内存用完了，就把还存活的对象复制到另外一块上面，然后将之前的那块清理掉

缺点：浪费内存太多（对老年代的使用，效率低）

第三种：标记-整理算法

将存活的对象都向一端移动，然后直接清理掉这端边界以外的内存

第四种：分代收集算法：（当前商业虚拟机都采用这个）

根据对象的存活中期的不同将内存划分为几块，一般Java堆分为新生代和老年代

新生代：用复制算法 老年代用标记整理算法进行回收

他们回收的时间段是在确定了有哪些对象需要被回收后，在CPU空闲的时候自动进行回收，或者是在堆内存满了以后进行回收，还有就是主动调用System.gc()方法来进行回收。

## 31.什么是反射，反射能干嘛？

反射： 就是将类的各个组成部分封装成其他对象,比如说wd对象。在java 中，只要给定类的名字，通过Class.forName方法可以或者到类对象,通过Class类对象就可以获得类的所有信息,比如说类里的所有成员变量或者所有方法,我们可以对类里的这些方法进行执行或者用动态代理模式进行增强.我们常用的Spring和MyBatis框架就是利用Java反射+动态代理的技术编写的.

## 32.get和post请求的区别?

Get是用url传递数据的不安全,而且传递的数据量比较小,post是可以传递大数据量,其实我们还有在接口开发里还有put请求修改, delete请求方式啥的.

## 33.session和cookie 区别

**Session是保存到服务器端,cookie是保存到客户端的,session相对于cookie来说更安全一些,如何从cookie中取数据? request.getCookie,session数据的共享问题如何解决,参考后面答案.**

## 34.HTTP和HTTPS的区别

我们公司用的是HTTPS协议 他俩一个是加密 一个不加密 HTTPS是基于http开发的 是http的安全版 HTTPS协议需要到ca申请证书 一般免费证书很少 需要交费 他们两个链接的端口引入不一样 http是80 https是443http是超文本传输协议 信息是明文传输的 HTTPS则是具有安全性的ssl加密传输协议

## 35.Jvm优化?

其实优化这一块我也说不太好,我就照我理解的方式说一下,比如说我们得遵守咱们Java的编码规范,比如说尽量少使用静态变量,循环最好是不要超过三层以上的嵌套,因为后期维护起来比较麻烦,也比较耗费jvm的内存,去掉项目中不必要的jar包,如果感觉jvm内存不够用,如果你的应用是跑到tomcat服务器上的话可以去修改下catalina.bat文件设置下Xms初始化堆的大小,和XXm最大允许分配堆内存,其实这个也是不设置的越大越好,也得经常更改,在我个人理解上如果设置的越大,有可能垃圾回收机制不会及时回收不长用的对象.x

## 36.java中的集合框架用过哪些?

java中的集合分为单列集合和双列集合，单列集合顶级接口为Collection,双列集合顶级接口为Map。

Collection的子接口有两个：List和Set。

List接口的特点：元素可重复，有序(存取顺序)。

list接口的实现类：

ArrayList:底层实现是数组，查询快，增删慢，线程不安全，效率高；

Vector:底层实现是数组，查询快，增删慢，线程安全，效率低；【废弃】

LinkedList:底层实现是链表，增删快，查询慢，线程不安全，效率高；

Set接口的特点：元素唯一，不可重复，无序。

Set接口实现类：

HashSet:底层实现hashMap，数组+链表实现，不允许元素重复，无序。

TreeSet:底层实现红黑二叉树，实现元素排序

Map接口的特点：key-value键值对形式存储数据

Map接口实现类：

HashMap：底层数组+链表实现，线程不安全效率高；

TreeMap：底层红黑二叉树实现，可实现元素的排序；

LinkedHashMap：底层hashmap+linkedList 实现，通过hashmap 实现key-value 键值对存储，通过链表实现元素有 序。

## 37.类的初始化过程

Student s = new Student();在内存中做了哪些事情?

加载Student.class文件进内存

在栈内存为s开辟空间

在堆内存为学生对象开辟空间

对学生对象的成员变量进行默认初始化

对学生对象的成员变量进行显示初始化

通过构造方法对学生对象的成员变量赋值

学生对象初始化完毕，把对象地址赋值给s变量

## 38.为什么要同时重写Object类里的hashcode和Equals方法

重写equals方法是为了比较两个不同对象的值是否相等

重写hashCode是为了让同一个Class对象的两个具有相同值的对象的Hash值相等。

同时重写hashCode()与equals()是为了满足HashSet、HashMap等此类集合的相同对象的不重复存储。

## 说几个Object类里的常用方法?

Equals, hashcode, toString, wait和notitfy 这两个Object类里的方法

# 后 端 框 架 部 分

## 1.Spring

### 1.说一下你对Spring的理解?

关于Spring的话，我们平时做项目一直都在用，不管是使用ssh还是使用ssm，都可以整合。Spring里面主要的就三点，也就是核心思想，IOC，DI,AOP。

其实spring这个框架也用到了Java里的反射机制.

DI就是依赖注入，把我们需要的类啊，接口啥的注入到spring中去。

IOC控制反转，像我们之前开发，如果想创建一个对象，就new一个，如果想这个对象中定义其他的变量或者对象，就在对象内部创建一个成员变量。但是现在的话，如果想用这个类的对象，咱们可以在spring的配置文件中配置一个bean，指定对应的全路径名称。spring通过配置文件用反射的方式，就可以直接帮我们获取到这个类的对象。还有AOP，就是面向切面编程，它的原理的话，我看过它的底层代码，它实际上就是实现了JDK的动态代理，以前的话用这个做过事务的控制，现在的话我们都用注解来控制事务。

AOP执行过程是一个纵向的过程，把每个方法当作一个点.基于这些点可以进行增强处理.形成了横向的切面,包含了原有方法和增强方法.不改变原有代码结构,还添加了额外的功能.

你了解的AOP的使用场景有哪些?**事务管理，日志打印,还有就是在老项目中也有可能用它来做权限管理.**

整体来说的话，Spring在使用的时候非常方便，在配置文件中配置要依赖的对象，或者在配置文件中将对象及属性进行注入,当然现在基本都用注解的方式，更方便。

除了这些，我们之前的项目也用过spring的其他产品，像spring boot（简化新Spring应用的初始搭建以及开发过程，用我的话来理解，就是spring boot其实不是什么新的框架，它默认配置了很多框架的使用方式，就像maven整合了所有的jar包，spring boot整合了所有的框架），spring cloud微服务框架。比spring更简单，快速，方便。

**(然后就可以扯微服务，和spring boot、cloud)。。。**

### 2.Spring常用的注解

**我们开发的时候常用的注解也就@service业务逻辑,@Transactionnal事务的注解,有时候需要注入DAO的话还会用到@Repository还有就是springMVC里的注解啦,比如说@Controller对应表现层的注解在spring未来的版本中,@RestController这个控制器返回的都是json格式,还有@RequestMapping进入的URL,@responsebody也是定义该方法返回JSON格式,还有就是@RequestParam获取参数,@RequestBody获取前台的数据是JSON格式的数据,@PathVariable从URL请求路径中获取数据,大概常用的也就这些.**

### 3.Spring中bean的生命周期?

1.Spring启动，查找并加载需要被Spring管理的bean，进行Bean的实例化放到Spring容器中

2.代码中可以通过Autowrited或者resources注解从容器中拿到实例化后的bean

3.调用完以后,如果没有其他类进行调用后,Spring将调用它的destory()接口方法进行销毁

### 4.Spring中的bean是单例的还是多例的?

**Bean里有个scope属性,默认配置是singleton单例的,**单例是不安全的，会导致属性重复使用。**如果变成多例的话配置成prototype,我们在项目中的有时候在controller中有大量成员变量的话,就可以用这个多例模式,还要结合的ThreadLocal类修饰一下**

### 5.Spring中Bean的作用范围?

Spring中默认的是单例的,变多例的话,得在bean里配置一个scope属性prototype,就会变成多例的,还有三个scope的配置就是和web服务器相关了的,比如说request,session,global-session这三个范围,就跟那个jsp作用差不多,后面三个也不常用

### 6.Spring中Bean是单例模式的那怎么保证线程安全?

我知道当Bean对象对应的类里存在可变的成员变量的时候并且其中存在改变这个变量的线程时，多线程操作该Bean对象时会出现线程安全。

当多线程中存在线程改变了bean对象的可变成员变量时，其他线程无法访问该bean对象的初始状态，从而造成数据错乱,我们解决方法是:

1.在Bean对象中尽量避免定义可变的成员变量；

2.在bean对象中定义一个ThreadLocal成员变量，将需要的可变成员变量保存在ThreadLocal中

### 7.Spring中bean的三个缓存?

Spring中的三个缓存是什么

一级缓存：singletonObjects，存放初始化后的单例对象

二级缓存：earlySingletonObjects，存放实例化，未完成初始化的单例对象（未完成属性注入的对象）

三级缓存：singletonFactories，存放ObjectFactory对象都是Map集合

单例对象先实例化存在于singletonFactories中，后存在于earlySingletonObjects中，最后初始化完成后放入singletonObjects中。

### 8.Spring中@Autowired和@Resource的区别?

@Autowired默认的是按照类型进行注入, 如果没有类型会按照名称(红色字体)进行注入.

如果想直接按照名称注入需要加入@Qualifier(**"gatheringDao"**)

@Autowired

@Qualifier(**"gatheringDao"**)  
**private** GatheringDao **gatheringDao**;

@Resource默认的会按照名称注入,名称找不着会按照类型来找,如果这里写了名称,就直接按照名称找了不会按类型找@Resource(name = **"aaa"**)

@Resource  
**private** GatheringDao **gatheringDao**;

### 9.单元测试的时候怎么启动spring容器的

这个可以通过读取配置文件的方式也行,用applicationContext读取xml配置文件,其实我更喜欢使用Spring配合junit用注解的方式去启动,下面代码注解要背出来

|  |
| --- |
| **@RunWith(SpringJUnit4ClassRunner.class)**  **@ContextConfiguration(locations={"classpath\*:spring-config-test.xml"})**  public class TestProjectDao {  **@Autowired**  **ProjectDao projectDao;**  @Test  public void testCreateProjectCode(){  long applyTime = System.currentTimeMillis();  Timestamp ts = new Timestamp(applyTime);  Map codeMap = projectDao.generateCode("5", "8",ts,"院内");  String projectCode = (String)codeMap.get("\_project\_code"); |

### 10.事务的传播特性？

知道有7种,常用就那两种,一种是\_REQUIRED：支持当前事务，如果当前没有事务，就新建一个事务。这是最常见的选择,增删改的方法可以用这个还有一个是SUPPORTS：支持当前事务，如果当前没有事务，就以非事务方式执行。查询的时候可以用这个.

### 11.Spring的三种注入方式?

我们项目中是配置扫描包,通过注解方式注入的,还有通过seter和构造方法的方式注入.

### 12.Spring是如何管理事务的?

我们用的是Spring给提供的一个管理事务的类DataSourceTransactionManager,用的时候是直接在需要使用事务的service中用@Transactional注解的方式来管理事务的,他底层实现也是AOP的方式来管理的事务.

### 13.SpringMvc执行流程?

springmvc是一个前端框架用来做控制转发的，它的核心流程有3个，就是用户发送请求到dispatchservlet,dispatchservlet调用相对应的handler，也就是我们通常所说的controller,controller返回modelAndview给dispatchservlet,dispatchservlet将得到的model And view传给viewReslover进行渲染，viewReslover将解析后的model And view传给dispatchservlet，最后dispatchservlet响应给用户。

[扩展]

springmvc的入口是servlet,这个servlet是单例的，如果有多个请求同时发出是dispatchservlet自动启动线程。

### 14.SpringMVC的Controller里可不可以写成员变量

可以写啊，注入的service不就是成员变量么，你是不想问struts2里的获取参数的方式啊？Struts2早就不用了，他那种用成员变量获取参数的方式，在高并发下会造成线程不安全，SpringMVC是使用的形参接收前台数据的，线程比较安全。

### 15.简单说一下SpringMVC与Spring是如何整合的?

**简单的说 springMVC在ssm中整合 就是 在 web.xml 里边配置springMVC的核心控制器:DispatcherServlet; 它就是对指定后缀进行拦截;然后在springMVC.xml里边配置扫描器,可以扫描到带@controller注解的这些类,现在用springMVC都是基与注解式开发, 像@service,@Repository @Requestmapping,@responsebody 啦这些注解标签 等等 都是开发时用的,每个注解标签都有自己的作用;它还配置一个视图解析器,主要就是对处理之后的跳转进行统一配置 ,大致就是这些,如何使用springMVC获取表单里的数据? 通过形参和表单里的name值保持一致就能获取到**

### 16.你给我说一下SpringBoot吧?

SpringBoot :约定大于配置：同样通过自动配置，引入依赖后，加载默认配置文件，如果要自定义的话可以通过 java 类实现，

也可以在全局配置文件(application.properties,application.yml)中定义；

SpringBoot优点

1. 快速创建独立运行的Spring项目以及与主流框架集成
2. 使用嵌入式的Servlet容器应用无需打成WAR包 (jar)
3. Starters(场景启动器)自动依赖与版本控制
4. 大量的自动配置，简化开发，也可修改默认值
5. 5.无需配置大量的XML，无代码生成，开箱即用

然后把springcloud里的每一个组件说一下,后面有答案

### 17.SpringCloud的常用组件挨个介绍一下?

**Eureka组件**(服务发现)音标(ju`ri:kə):  相当于我们使用的dubbox的时候zookeeper注册中心.  
**Feign组件**: 调用服务的时候用的组件  
**Hystrix**(hist`riks)**熔断器**: 我们在调用服务的时候,有可能涉及到服务的连锁调用,比如说A服务调用B服务,B服务里还调用了C服务, 使用A服务的时候,B服务和C服务都得正常运行才可以使用,B调用C没有调通的时候, B直接给A返回内容, 不至于像以前报错.

**服务网关 zull**: 前后端进行调用的时候可以,可以走同一个IP地址,因为项目端口号太多,配置这个以后就可以直接走一个端口号,他自动会给你分配具体调用的哪一个端口  
**分布式配置Spring Cloud Config:**我们把多个项目的配置文件归置为一个,修改配置文件以后,不用再重新部署某一个项目啦.  
**消息总线:Spring Cloud Bus:** 修改完配置文件以后不用重启项目.

### 18.SpringBoot和SpringMVC与springCloud关系? 1.springboot是springmvc的升级版,其实就把springmvc里的web.xml文件去掉了,改为全注解的开发2.SpringCloud通过Springboot把其他的通信组件等等进行了封装,你如果使用SpringCloud的,那就必须得使用SpringBoot,使用SpringBoot的话不一定非得使用SpringCloud.

### 19.Springboot如果要直接获取form数据的话要怎么做？

也是通过形参里添加参数就行啊,和form表单里的name值对应上就行了啊,在form表单的action中配置上请求改方法的路径不就行了么.

### 20.SpringCloud的用什么版本

我们用的是2.0的版本,我那会学SpringCloud的时候也了解过,他们的版本是根据伦敦的地铁站进行命名的,我们用的是比较稳定版本就是2.0这个版本

### 21.你用了SpringCloud微服务，分布式的项目，微服务之间是如何通信？

我们把所有的微服务都发布到Eureka注册中心上了，服务之间互相调用的时候我们用的是Feign进行通信的，而且他自带负载均衡，自带实现那个Ribbon那个接口

### 22.SpringCloud里的负载均衡的策略有哪些？是怎么负载均衡的？

这个知道，用的Ribbon那个组件实现的负载均衡，底层代码也看过，有两种负载均衡的方式，一种是轮询，还有一种随机方式，Feign默认支持负载均衡，我们一般都不会手动配置。

### 23使用的SpringCloud有什么好处？

用了他的的好处就是可以用他里面的那些组件实现微服务之间的通信比较简单，比如说。。。。。把组件挨个说一下。 我们在用的时候是结合springboot还有mybatis一起用的，比如说使用了springboot的话就简化了好多的代码量，加快开发速度。

### 24.SpringCloud里的熔断方式有几种，hystrix？

使用过服务降级的方式，在代码端做过业务处理，我还知道那个熔断器里的三种状态，开启状态，不是一直熔断，而是默认休眠5秒钟（这个数据也可以配置），然后会进入到一个半开状态，当有服务请求过来的时候，还试图去访问下要调用服务是否能调用上，如果能调用上，关闭close正常访问服务去。

## MyBatis框架

### 1.Mybatis框架简介

**Mybatis 框架也是一个持久层框架，我们目前做的这个项目就是用的这个框架，我觉他相对于以前的hibernate来说比较简单,把sql语句写在配置文件里,解除了代码和sql语句的耦合度,写一些复杂的查询比较灵活.**

### 2.MyBatis中# 和 $ 的区别?

**这两个符号一般是在使用Mybatis编写底层SQL语句时使用，#就是一个占位符，具体的使用是#{id}，而$是一个原样输出的标识，是${value},我在项目里一直是使用#，因为这样可以防止Sql注入，安全性高, 使用#号相当于我们在使用JDBC的时候使用使用了预加载SQL语句的方法PreparStatement（SQL），里面有问号占位符，使用PreparStatement往占位符里设置内容就可以了，但是用$符号的话，就是一个字符串的拼接，相当于使用我们JDBC里的Statement（sql）方法，这样前端能拼接字符，所以不安全。**

**但是现在我们都是前后端分离，前端都是html页面，不会再有sql注入这个问题了，因为他们通过js传递数据，做过验证，以前出现过的sql注入问题都是在JSP页面中发生的。**

### 3.Mybatis 和 Hibernate 的区别

**Hibernate一个是全封装，mybatis是半封装，使用hibernate做单表查询操作的时候比较简单(因为hibernate是针对对象进行操作的),但是多表查询起来就比较繁琐了,比如说5张表10张表做关联查询,就算是有SQLquery那后续的维护工作也比较麻烦,还有就是Hibernate在Sql优化上执行效率上会远低于MyBatis(因为hibernate会把表中所有的字段查询出来,比较消耗性能)我们以前在做传统项目方面用过hibernate，但是现在基本上都在用mybatis.**

### 4.Mybatis缓存？

我知道的Mybatis缓存分为一级缓存和二级缓存,一级缓存是SQLSession的缓存,默认的是开启的(什么时候失效?关闭SqlSession或者是执行增删改的时候缓存就会失效).

二级缓存是SqlSessionFactory,他需要在主配置文件中配置,在需要缓存的DAO配置文件中打开就行,我们项目中一般不用这个.都是用redis存热点数据的.

### 5.Mybatis里多对多如何处理? 举个多对多的例子?

比如说学生表和课程表,一个学生可以选择多门课程, 一门课程都能被多个学生选择,这样两张表的关系就是多对多的关系, 怎么处理多对多的情况? 遇到这种情况我们得创建一张中间的桥表,关联后就是 课程表对桥表就是一对多,学生表对桥表也是一对多,就可以了.

### 6.MyBatis执行流程

我们项目中都是结合Spring用的，我们都是把MyBatis的sqlSessionFactory交给Spring来管理的，在启动服务器的时候会通过Spring监听，加载配置文件里的所有的文件，Spring会通过反射的方式创建Mapper的实现类，我们调用Service来调用dao层的。

我们在使用springboot的时候也有结合MyBatista使用过，用的时候是他通过集成mybatis官网提供的继承tk.包下的那个mapper，会实现单表的全部增删改查

### 7.Mybatis中的resultMap和ResultType有什么区别?

ResultMap返回的是自定义的ResultMap标签中的ID,resultmap里配置是一些mybatis里的一对多,多对一相关配置, resultType返回的是类.

### 8.Mybatis中的传入参数用的哪个属性?

在配置文件中可以parameterType这个属性,如果注解开发的话可以使用@Param定义

### 9.Mybatis中的动态SQL语句用过哪些?

if, where foreach 这些都用过.

# 前 端 框 架 和Linux 面 试 题

## 1.JQuery中常用的选择器有哪些?

**常用的比较基础的 id class 标签 这三个,其实开发的时候还有用到 attr这些属性选择器也常用,还有一些其他的,用的话,会去查API**

## 2.通过JQuery如何获取表单中的一个元素

|  |
| --- |
| 比如说用id选择器,$(“#username”).val() 就能获取 表单中 <input type=”text” id=”username”> |

## 3.说一下Vue中常用的指令？

比如说：循环：v-for 判断 v-if 双向绑定 v-model 点击事件什么的@click什么的，定义组件components什么的，我们ajax通信用的是axios工具

## 4.Linux查询日志的指令，查询进程、杀死进程

我们查看tomcat的日志 是tail -f logs/catalina.out 或者是 cat 或者是 vim都行,或者我们用工具连上去查看,EditPlus连接上查看, 查询进程 ps aux|grep 进程名 杀死进程 kill -999 进程id 复制命令：cp 剪切命令： mv

# 互联网架构思想和中间件

## redis非关系型数据库

### 1.给我介绍一下redis,以及redis为什么快?

Redis是一个非关系数据库,我们项目中主要用它来存储热点数据的,减轻数据库的压力.我们项目中很多地方用到了redis,比如说APP端首页热点数据的展示,还有在以前的项目中我们也用过redis来实现单点登录,至于redis为什么快?主要有这么几点,第一呢就redis的数据都是存储到内存中的加载比较快,第二点呢是redis中数据结构比较简单,还有第三个原因是他是单线程的,避免了上下文进行切换和竞争条件,第四点他使用的是多路I/O复用模型,非阻塞IO(红色字体作为理解).第五来说redis使用的底层模型不同,Redis直接自己构建的VM机制

（1）多路 I/O 复用模型

多路I/O复用模型是利用 select、poll、epoll 可以同时监察多个流的 I/O 事件的能力，在空闲的时候，会把当前线程阻塞掉，当有一个或多个流有 I/O 事件时，就从阻塞态中唤醒，于是程序就会轮询一遍所有的流（epoll 是只轮询那些真正发出了事件的流），并且只依次顺序的处理就绪的流，这种做法就避免了大量的无用操作。

\*\*这里“多路”指的是多个网络连接，“复用”指的是复用同一个线程。\*\*采用多路 I/O 复用技术可以让单个线程高效的处理多个连接请求（尽量减少网络 IO 的时间消耗），且 Redis 在内存中操作数据的速度非常快，也就是说内存内的操作不会成为影响Redis性能的瓶颈，主要由以上几点造就了 Redis 具有很高的吞吐量。

### 2.你们在项目中是怎么操作redis的?有过封装么?

项目中，我们一般都通过springData操作的redis,他底层也是整合redis官方提供的jedis工具包来操作redis,我们开发的时候用的是单机版。

### 3.redis持久化方式有几种?

有RDB和AOF这两种，RDB是一种快照的方式来存储的，这也是redis的默认的持久化方式，每隔一段对数据进行一次存储，

save 900 1 #在900秒(15分钟)之后，如果至少有1个key发生变化，则dump内存快照。

save 300 10 #在300秒(5分钟)之后，如果至少有10个key发生变化，则dump内存快照。

save 60 10000#在60秒(1分钟)之后，如果至少有10000个key发生变化，则dump内存快照。

还有一种是AOF是，通过配置appendfsync的三种状态，always: 每次操作都会立即写入aof文件中 everysec: 每秒持久化一次(默认配置)no: 不主动进行同步操作，默认30s一次只要数据发生改变都会保存到硬盘一份,这种方式对数据的保存完整性比较高，但是性能比较差。而RDB存在的问题主要是服务器宕机或者断电，会造成数据丢失

### 4.使用redis缓存有没有遇到什么问题 ?

你是说的那个缓存雪崩和缓存穿透,,我知道缓存雪崩是因为redis中多个key同时失效后,又遇到高并发后就会造成大量的请求直接请求数据库,导致数据库服务器宕机的情况,一般这种情况我们会给redis中的key设置不同的生命周期就能解决,或者是我们给使用分布式锁来解决;还有缓存穿透的话就是大并发请求过来,查询一个连数据库都没有的数据,频繁的请求数据库导致数据库宕机,这种解决方案是我们会存到缓存一个key值value为空,给个失效时间,也可以避免这种问题.

### 5.说一下redis集群吗?

Redis本身就支持集群操作redis\_cluster，另外redis还支持主从复制，以前的老版本中有一个哨兵模式，在主服务器宕机时，从服务器可以自动转换为主服务器。我们公司搭建的redis集群是用的ruby脚本配合搭建的,我们一共搭建了6台服务器,3主3备,他们之间通信的原理是有一个乒乓协议进行通信的,他们判断一个节点的状态是用投票选举机制判断的,半数以上判断一个接口是宕机了的话,备用节点就会启动,对,我再给你说下一他们往里存储数据的机制吧,其实这个redis搭建好集群以后每个节点都存放着一个hash槽,每次往里存储数据的时候,redis都会根据存储进来的key值算出一个hash值,通过这个hash值可以判断到底应该存储到哪一个哈希槽中,取的时候也是这么取的,这就是我了解的redis集群.

### 6.在项目那一块用过,使用场景?

我们在项目的活动模块用过,APP端的首页展示的数据都是使用redis存储的热点数据,因为用户进入APP的时候首先进入是首页,使用的hash数据类型存储首页信息,特卖模块中也用过redis,用它的list数据类型.

### 7.redis的数据类型?

我了解的有

string,list、set、hash、sortedSet

list，我们可以向list的两端添加数据

集合set，存放的数据是无序的，集合中的数据是不重复的，由于它是无序的，所以不能通过下标来获取制定元素、

order set有序集合、

hash

实际上我们常用的也就是string 和hash

redis是通过key-value存储的 set key value string

hset key value hget key value

### 8.Redis中的事务如何实现? redis中的分布式锁如何实现?

我知道你说的那个redis的事务,就那个命令multi开启事务exec执行事务中的所有操作什么的命令,我们开发的时候没有用到过,我们用的是Redis的分布式锁,用过里面的setnx结合的lua脚本去使用的,但是这种锁单节点的Redis服务器使用没有问题,我们测过差不多能抗住3000的并发,如果是Redis集群就不行了,因为他那个锁有可能在主服务还没有同步到从服务器的时候,主服务器宕机了,锁就找不到了,我们就得用Redisson这个框架去实现了,他里面提供的一个监控锁的机制,这个框架提供了很多种分布式锁的解决方案,比如可重入锁,还有读写锁,还有什么异步锁,直接使用就行.里面我的记的有trylock和unlock这些个方法,

### 9.如何使用redis去实现一个消息队列

我知道的redis实现消息队列的方式有两种，第一种是用他本身的list数据类型，他有一个lpush和rpuop命令，可以使用lpush从集合左边添加进去数据,使用rpop命令从右边拿取数据。

第二种实现方式：redis还提供了一组命令可以让开发者实现"发布/订阅"(publish/subscribe)模式。"发布/订阅"模式也可以实现进程间的消息传递，但是我们没有用过这种，我们项目中的消息队列都是用RabbitMQ实现的。

### 10.Redis和mongoDB的区别?

mongodb是非关系型数据库最像关系型数据库的一种数据库,有数据库，集合和文档的关联，集合就相当于咱们关系型数据库里的表，文档就是一条数据，他存储数据的方式是以bson方式进行存储的。但是Redis是key和value方式存储的,在项目中我们可以把一些频繁访问的数据存到redis中,不频繁的访问数据存到mysql数据库中,可以给mysql数据库减轻压力.

### 11.Redis在你们项目中起到的主要作用是什么?

主要就是缓存热点数据的,减轻服务器压力.

### 12.Redis是如何同步的？

其实redis里的数据来源于数据库，比如说我们前台的广告位展示的时候，数据是存在redis中的，数据库里的数据更新以后，我们会把redis里的缓存清空一下。让他第一次查的时候从数据库里查，第二用户再进来的时候就是从redis中取了。

### 13.Redis的自动清理机制?Redis 的内存回收机制?

你是说的是redis服务器内存满了如何清理是吧,可以在配置文件中设置：  
redis最大内存不足"时,数据清除策略,默认为"volatile-lru"。volatile-lru  ->对"过期集合"中的数据采取LRU(近期最少使用)算法.还有一个策略是volatile-random ->对"过期集合"中的数据采取"随即选取"算法,并移除选中的K-V,直到"内存足够"为止.

## ElasticSearch

### 2.1你们用的ElasticSearch是哪个版本的?

当时我们用的是6.8.3那个版本

### 2.2 ElasticSearch中分词是怎么做的?

我们用的是IK分词器,我知道如果还需要自定义一些分词的话得修改里面有一个dic文件,自定义比如说小儿氨酚黄那敏颗粒这个词在字典里没有,得需要自定义一下.

### 2.3.ElasticSearch的开发端口号是多少

我们开发的时候端口号是9300, 还有一个9200的端口号,是用kibana客户端工具访问的一个端口号

### 2.4.ElasticSearch和Solr的区别?

我知道他俩的底层都是基于lucene实现的,都是使用的lucene的[倒排索引](https://blog.csdn.net/u010558660/article/details/53407455)实现的,solr在实时建立索引的时候会产生IO阻塞查询性能会比ElasticSearch差一些,还有就是因为Solr自身不支持分布式,ElasticSearch是实时处理数据,而且默认的支持分布式的,可以组成一个网络,如果其中一台服务器宕机,会分配其他节点工作,可以扩展多台服务器,所以查询效率会更快,据说可以处理PB以上级别的数据.

### 2.5.你们操作ElasticSearch是怎么操作的？你用过ElasticSearch里的哪些方法？

我们在用的时候是公司在Linux服务器上搭建的ElasticSearch服务，我们操作的时候，是用Springboot结合的ElasticSearch用的，因为SpringBoot自带的有ElasticSearch的依赖，只需要配置连接就行，我还用过ElasticSearch里的聚合查询，里面可以根据分类进行聚合查询，聚合里还有一个桶的概念，先找到聚合名称再遍历里面的桶。

### 2.6.ElasticSearch中的match和term的区别?如何实现精确查询

match是对搜索的内容进行分词查询的,term是代表完全匹配,也就是精确查询

### 2.7.ElasticSearch中的字符串类型有哪两种?

ElasticSearch 5.0以后，字符串类型有重大变更，移除了string类型，string字段被拆分成两种新的数据类型： text和keyword。

text： 会分词，然后进行索引，用于全文搜索。

          支持模糊、精确查询

          不支持聚合

keyword：不进行分词，直接索引，keyword用于关键词搜索

                  支持模糊、精确查询

                  支持聚合

### 2.8.你们是如何实现的ElasticSearch的?

你是说的如何调用ES的吧, 我们使用的是springboot调用的ES,我们自定义一个接口,让这个接口继承ElasticsearchRepository这个接口,这个接口里封装的有一些save和saveall一些方法,还有delete方法,我们从数据库查询出来的内容使用提前定义好的索引的pojo类,直接就能存到索引库里.

### 2.9.ES如何与数据库数据保持一致的

我们是结合的RabbitMQ来做的,因为我们是微服务架构,项目之间进行通信就可以了,我们把审核通过的景点,发送到rabbitmq消息队列中一个景点id,我们通过监听到这个id就可以查询出该景点信息,然后调用ElasticsearchRepository中的save方法存到索引库去,如果修改了信息我们也是发送消息来同步的.

### 2.10.ES中的Filter和query有什么区别?

Query是查询的时候用的吧, 那个filter是结果过滤,比如我们在京东搜索完手机以后你可以过滤一下5G手机还有内存8G什么的,这些就属于结果过滤.

### 2.11.有没有做过ES调优呢?

了解过一些,在网上看着过,主要是优化一下他的线程池,

## MQ消息队列

### 3.1.使用ActiveMQ的好处?

减轻服务器压力,降低项目之间的耦合度(解耦),是做异步的.

### 3.2.ActiveMQ消息发送失败解决方案?在使用activeMQueue的时候有没有遇到过什么问题?

第一种用数据库配合着解决：

怎么配合呢咱们这边不是发送的商品的ID么，在发送之前把ID记录在数据库里面去，然后设置一个状态字段，0代表这个消息发送中,然后监听端进行消费,消费成功后把这个字段在改为1.然后就是几种突发情况,第一种情况比如说突然断电了然后我的消息首先是记录在数据库里面了，然后他的那个状态是不是为0啊，然后我们有一个用quartz写的定时器,每隔5分钟会去数据库跑个批然后把状态为0的数据,再重新发送一遍消息,直到消费成功为止.

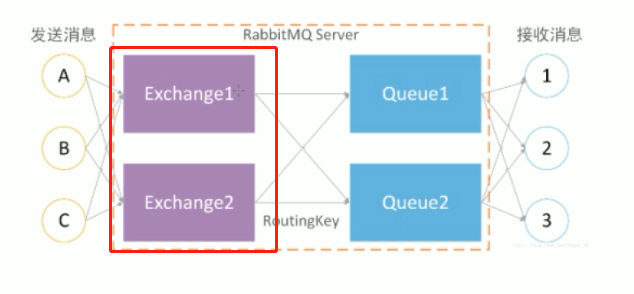
第二种解决：

在发送消息的时候设置提交的方式，改成手动提交的方式，在后台改成commit状态改成手动方式，如果发送成功的话，然后commit手动提交方式。

### 3.3.MQ用过哪些？了解哪些？

我们用过的消息队列有 ActiveMQ和RabbitMQ,我还知道有个kafka大数据里用的,这三种消息队列处理速度上来说 Kafka 大于 RabbitMQ 大于 ActiveMQ, 从安全上来讲: ActiveMQ 大于 RabbitMQ 大于 Kafaka, 我知道银行也用的RabbitMQ挺安全的 我给你说下ActiveMQueue和RabbitMQ他俩的区别吧, ActiveMQ他发消息的方式有两种,一种是推拉式的Queue,还有一种是发布式的(Topic), 区别在于推拉式的发送,只允许有一个消费端进行消费,如果不消费的话就一直存在队列中,订阅式的是发送一个消息,可以有多个消费端,如果没有消费的话,他也不会一直保留到消息队列中,这两种我开发的时候都用过.

### 3.4.说一下RabbitMQ？

RabbitMQ和ActiveMQueue不一样的地方是ActiveMQ发送消息是直接发送到队列中了，但是RabbitMQ发送消息的时候是先发送到交换器中,发送消息的方式我记得有五种,一种是简单模式,和activeMQ差不多,还有一种是work消息模式,还有三种订阅模式,一种是**fanout模式**,可以有多个消费者消费到相同的消息,还有Direct模式,可以根据routingKey来定义接收消息,还有一种是Topic模式,可以根据\*号或者#号来模糊匹配Routeingkey来发送和消费消息, 

### 3.5.RabbitMQ中如何确定消息是否消费成功？

Rabbit中有一个ACK机制机制。当消费者获取消息后，会向RabbitMQ发送回执ACK，告知消息已经被接收。不过这种回执ACK分两种情况：自动ACK：消息一旦被接收，消费者自动发送ACK，手动ACK：消息接收后，不会发送ACK，需要手动调用，我们如果发送的消息比较重要的话都会使用手动的方式去控制，在消息的监听中，设置为false，手动进行ACK。

### 3.6.如何避免消息堆积？

RabbitMQ中有一个workqueue机制，可以有多个消费者同时监听消费同一个队列，接收到消息以后，通过线程池，异步消费。而且这种消费模式不会出现重复消费的情况，一条消息只会被消费一次，更像是银行排队，特别多人排队的时候需要有多个窗口进行办理业务，只要有一个人在一个窗口办理完业务，其他窗口就不会再叫到这个人。

### 3.7.如何保证发送消息的服务器宕机依旧能保存消息？

我们可以设置消息的持久化方式，把交换机里的持久化参数设置为true，把消息队列里的消息持久化也设置为true，即使消费者服务器宕机了，重启服务器，消息依旧能保存到交换机里，重新消费掉

**3.8.RabbitMQ怎么实现的?**

其实面试官想知道你是怎么用的, 我们是结合springboot去使用的,有一个对象AMQPTemplate去发送消息,还有使用些注解@RabbitListener去接收一下消息,我们项目中用的是topic发送消息的方式,其实还有其他的几种发送的方式.然后把其他四种给说一下就行

## 4.Nginx负载均衡服务器

### 4.1.介绍一下Nginx反向代理

他就是一个反向代理服务器,其实它的优点就是内存占用少,并发访问能力强,我知道的新浪,斗鱼,好这些大公司都使用这个技术,它底层实现是c语言实现的.我们主要用到nginx两大主要的功能吧,一个是反向代理,一个是负载均衡,先说一下这个反向代理,咱们还得先说一下正向代理,其实咱们平时调试开发都是正向代理,只不过我们不说这个词.比如吧,我们访问一台tomcat,默认端口号是8080,那我们访问的时候可能就是localhost:8080,这样顺着来呢，就可以理解成一个正向代理,就这样理解哈,这个时候，如果我们想要再来一台服务器呢,我们可以配一下,把端口号改成8081,通过访问不同的端口号来访问, 但是当我们项目要上线的时候，如果需要把一个项目如果部署到两台服务器上，比如淘宝，这么大，它的主界面不可能是在一台服务器上放着，就不能是访问8080或者8081这些端口了，这个时候，需要有一个代理的服务器，能够给这两台服务器做一个代理，直接不需要进行标明，就可以访问到任意一台服务器，找到这个主界面。这里呢，这个代理就可以代理这些服务器了，这个时候这个代理，我们可以理解成反向代理。反向代理严格的概念是通过代理服务器来接收网路上的请求，然后将请求转发给内部网路的服务器。而nginx可以干这个活，做这个代理，我们可以在nginx中配置端口，ip或者域名指向这些不同端口，甚至不同ip的服务器。这就是反向代理这个概念。

### 4.2.说下nginx负载均衡

Nginx还有一个重要的功能叫做负载均衡，我们做服务器的集群，怎样保证集群中服务器被均等的进行访问呢，不能说我们认为搭建好了服务器的集群它就会均衡的去访问，这个时候我们可以统一的去访问nginx这个服务器，在nginx的配置信息中，去配置好这些服务器，它配置文件是这样的，只要你配上，默认访问的比率就是一样的，这个就是负载均衡，当然nginx更厉害的是可以配置权重，比如说哈，我两台服务器，其中一台性能比另外一个性能好2倍，那我是不是应该访问性能好的服务器频率更高一些，咱们就可以在nginx的配置文件中配置一个weight属性，指定权重。当然还有其他一些配置的，比如有些服务器需要整修，那咱们就可以配置某台服务器暂时down掉，这样用户访问的时候，就不会访问到这台服务器，当修好之后，我们在把这个配置信息干掉就行了。

### 4.3.如何配置Nginx?

我们配置nginx是在nginx安装目录下的conf下有一个nginx.conf文件，主要是修改这个配置文件，比如咱们配置反向代理和负载均衡服务器，配置一个proxy\_pass 指向代理服务器，配一下upstream server指向要访问的ip和端口，这个可以配置多个ip,可以设置weight权重什么的?

### 4.4.Nginx使用方法? **其实nginx服务器搭建这一块是由公司的运维去做的,我原来在自己的虚拟机里也配置过,可以搭建两台nginx服务器,主备关系,用那个keepalived管理主备服务器,keepalived其实就是让主备服务器之间发送心跳协议,保证nginx永不宕机.**

## 5.其他中间件

### 5.1.说下thymeleaf里的一些基本语法标签？

比如循环 th:each="user : ${users}" 变量 th:text="${user.name}" 有超链接的数据的话用的 th:utext=”${image}”

### 5.2.图片是怎么存储的?

我们用的是FastDFS文件服务器，FastDFS架构包括一个Tracker server 和一个 Storage server。

客户端请求Tracker server 进行文件上传、下载，通过Tracker server 调用Storage server完成文件上传和下载。Tracker server可以完成负载均衡，它其实是这样的，这个Storage server 会定时的向tracker server 发送自己的状态，tracker server就能知道那个storage server是空闲还是忙碌，当上传下载的时候，就会调用空闲的storage server ，当上传或下载的时候，这个storage server会生成filed id ，并将文件写到磁盘，返回一个路径信息和文件名，拿到路径信息我们就可以存到数据库或调用了。

# 项目部分

## 1、你们公司日志这一块是怎么记录的？

日志处理我们使用的是log4j,有一个log4j的配置文件,可以配置log输出的位置和log的输出形式,我们对于整个项目，设置了一个全局异常，当出现异常信息的时候，将异常信息记录到log中

Logger logger = LoggerFactory.getLogger(GloableException.class);

logger.error("-------------出错了------------------");

当有些需要记录内容的信息，也可以通过日志文件进行记录。

对于用户登陆日志记录,我们需要自己封装一个日志记录的工具类,可以将用户登陆的信息记录到数据库中。(具体操作步骤看如下链接)

## 2、库存警告是怎么使用的？

我们后台每隔5分钟会跑批查看库存小于100的商品，然后说就是会有相应的短信提醒，告诉库管或者商户具体哪个商品少了，然后商户或者库管登录后台进行补充库存

## 3、权限管理你们是单独的一个呀，还是怎么弄的？

我们之前做过权限管理，我们做的时候使用的五张表+SpringSecurity去做的权限管理，五张表分别是一张用户表，一张角色表，一张权限表，还有两张用户角色的桥表和角色权限的桥表。

我们是通过用户的id查询它所对应的角色，通过角色查询他所对应的权限，通过权限找到相对应的url,通过SpringSecurity的注解和表达式去控制相应的角色或者页面展示内容的

## 4、项目中你主要负责那块？

这就集思广益了，按照自己简历上的项目，然后针对性的找出自己要说哪几点，首先介绍这个模块是干什么的，然后这个是怎么用的，然后在哪里用的。

## 5、项目中的管理工具用的什么

你是说的项目版本管理,还是项目进度管理,我们项目版本管理用的是SVN,我们项目进度管理项目经理用的是禅道,还有那个project软件管理的进度,还有我们项目里jar包管理项目之间的依赖用的是maven做的.

## 6、项目中用的什么做的压力测试

我们用的是Jmeter做的,模拟多个请求同时去请求接口,然后能导出测试报告.

## 7、项目当中的多线程，线程池是怎么回事

我知道多线程就是为了解决多任务同时执行的需求，合理使用CPU资源。多线程的运行是根据CPU切换完成，如何切换由CPU决定，因此多线程运行具有不确定性。

线程池：现在服务器端的应用程序几乎都采用了“线程池”技术，其实和数据库连接池差不多,这主要是为了提高线程使用效率。因为如果服务器对应每一个请求就创建一个线程的话，在很短的一段时间内就会产生很多创建和销毁线程动作，导致服务器在创建和销毁线程上花费的时间和消耗的系统资源要比花在处理实际的用户请求的时间和资源更多。线程池就是为了尽量减少这种情况的发生。（适用于短时间内多任务的情况，如果线程执行时间较长不适用线程池）

## 8、用户注册怎么知道谁有URL权限

给不同类型的用户设置不同的角色，通过角色匹配权限,其实我们权限控制无非就是控制他的程序的入口URL,用springSecurity控制的角色和权限

## 9、项目中遇到的难点你们是怎么克服的，有什么固定的标准流程吗？

比方说公司要用redis，买一些关于redis实战的书进行学习，网上也找找关于redis方面的资料进行学习，遇到一些问题问问我们项目组长，技术总监，他们有什么相应的解决方案，然后自己去解决去。

## 10、项目迭代是怎么实现？

我们做项目迭代的时候,新增完功能以后会有一个系统集成测试（SIT测试），测试完了之后再进行上线。

## 11、第三方支付

我们常用的第三方支付有：支付宝、微信。他们的原理都差不多。都是在点击支付时，直接调用第三方支付接口，传入appid、appsecret、订单编号、订单金额、回调url，直接跳转到第三方支付页面，接下来的支付过程，我们都不需要管，支付成功以后，第三方支付平台会直接回调我们的url。给我们返回：状态码、订单编号、支付流水号三个参数。我们首先根据订单编号，找到我们的订单，把支付流水号和状态码更新到我们的订单里边。

## 12、第三方登录

我做过微信小程序端的登录,提前申请微信小程序的开发者选项,可以获取到appid和密钥可以根据这个appid还有密钥,就可以实现微信的登录了,我也是看的demo开发的,我还记得,如果是别的应用中如果用到这个微信登录的话,就得是用一个auth2协议就行,用的时候去网上查就行,有现成的demo.

## 13、开发过程中你是如何确保开发的进度和质量的？

一般都是项目经理定的，规定一下项目工作日，根据功能点来估计一下工作日

## 14、前后端通过接口调用的时候,接口安全如何做的?

我们使用JWT做的前后端接口的安全控制,访问我们接口的时候必须通过我们约定好的在header中存放token信息,判断这个token信息是否是我们这个后台给提供的token信息,token信息在什么产生的?是在登录的时候产生的token码.还有一种情况,用户初次进入到我们的系统,用户是没有登录的,这时候我们跟前台确定一个token码,前台要给我我们Token码+时间戳请求我们后台,我们后台会根据约定好了,进行相应的截取时间戳和token码,进行相应的判断.

## 15、你们项目经理主要是做什么的？

(米纳)项目经理是这个产品的第一负责人，管理整个项目的进度和质量，从需求阶段就带着我们和产品经理讨论、分析产品原型的需求，然后指定开发计划并分配任务，在开发过程中经常组织我们开会讨论分析问题，了解任务进展；还要经常与其它部门人员沟通、协调各种问题。

## 16、那你们前端是和你们一起做的，还是并行的？

我们是并行开发的,我都有接口文档,我们后台根据接口文档进行开发就行,前台的同事也是通过接口文档来对接的,有问题的话我们都在一起的,调试就行了。

## 17.那数据和页面的融合是谁做的？

前台和我们一块儿配合的做,他们调试前台,我们负责调试以后台。

## 18.你们做前后端交互的时候都用的什么？

我们交互的时候是异步的ajax,我记得前端好像用的 vue 或者racet框架做的.

## 19.那你一般都将什么数据进行一些安全处理？你们怎么保证数据的安全性？

涉及到钱的都需要处理，支付宝不是有扫码嘛，先用沙箱环境测试，测试完以后在配置（公钥、私钥、商户 ID 还有其他的一些），配置完再用公司自己写的安全框架,我们用的是JWT验证用户的信息,有些功能需要验证用户信息,比如说下单,加入购物车,支付等这些都得验证用户信息.

## 20.产品经理直接给你原型，那后期有些功能不明确，你怎么处理？

我先自己琢磨琢磨，琢磨完在跟产品经理和项目经理确认一下再进行开发。

## 21.单点登录跟谁单点？

其实单点登录无非是在分布式的项目中应用,多个系统中都要用到用户的信息,以前用过CAS技术实现的单点登录,现在我们就直接用jwt技术,实现用户信息的共享,登录后,会把token信息响应给前台,前台请求的时候会携带token访问后台接口,后台可以根据token获取到用户的相关信息.

## 22.你们在做这个项目最复杂的查询关联了几个表？说下你有印象的，还有应用场景？

五张表查询,原来做权限的时候关联的查询过.用户表,角色表,权限表,用户角色表,角色权限表桥表.

## 23.你怎么理解高可用的?

就是互联网项目里的7\*24小时不允许服务器宕机，比如说我们的redis,会搭建集群,还有eureka也会搭建集群.

## 24.如何解决高并发?

我在公司采用的，都是通过redis缓存，mycat分库分表,把上面中间件的使用场景全部都说一遍.

## 25.那你们的项目经理是你们的还是其他的？

我们的项目经理，这项目组就是我们公司全部都做。

## 26.你们公司的接口是怎么管理的？

那个接口开发，我们就是这么做的，就是用的那个springMVC结合swgger来做的那个接口开发，swgger提供一些注解，可以把接口文档通过网页的形式可以展示出来，上面能标清楚他的url,参数，返回类型什么的，比较方便，原来我们没有用swgger的时候，用的都是那个word文档，但是用容易丢，不好维护，后来才用的swgger，现在同类型的产品也挺多的，我们公司反正用的都是swgger

## 27、pc端和app是不是共享一个接口,为什么

一般得看需求因为controller不是一个的有可能pc端的是返回页面移动端的是返回json数据格式根据实际的需求而定

## 28.秒杀这一块你们是如何控制超卖的?

我们把库存信息也是存放到了redis中,用到redis中increment特性定义库存,减库存用到increment -1 特性减库存.还会用到redis的list队列,有100件库存,队列里会存100件商品,卖一件,rpush出来一件.防止超卖.

## 29.你们这个框架怎么搭建的？

我们后台是用最近做的项目是后台用的是SpringBoot+springCloud+MyBatis去实现的,中间件用的是redis,ElasticSearch,RabbitMQ,这些技术,还有开发工具用的是Idea

## 30.高并发的接口调用

我们项目中解决高并发都是用的中间件解决的,然后把redis,RabbitMQ,ElasticSeach这些个中间件使用场景都说一遍

## 31.你们订单号是怎么组成的?

**我们订单号是用时间戳+推特的那个分布式id生成器实现的.**

## 32、fastDFS下载

文件上传后返回一个路径信息和文件名，拿到路径信息我们就可以存到数据库或调用了。

## 33、跨域是怎么解决的

跨域这块呢，之前我们用过JSONP来解决跨域问题，现在项目用到springMVC了，它里边有个@CrossOrigin(origins="http://localhost:9105",allowCredentials="true")

开发时侯跨域调用这块还用到过HttpClient这个技术，把它放到工具类里变用的时候直接调用就行了，项目开发中也用到过一些其他这样的技术，像遇到C语言向java语言这种跨语言的通信，就用到WebService技术，它的底层通信用的是soap协议，用的是cxf方式来发送的。

## 34.你们项目是分布式的,那你有了解过分布式事务么?

当然有了,因为我们项目比较大访问用户也比较多,我们在支付的时候,和下单的时候都用到了分布式事务.比如实时支付吧，一笔支付，是对买家账户进行扣款，同时对卖家账户进行加钱，这些操作必须在一个事务里执行，要么全部成功，要么全部失败。而对于买家账户属于买家中心，对应的是买家数据库，而卖家账户属于卖家中心，对应的是卖家数据库，对不同数据库的操作必然需要引入分布式事务。还有就是用户下单买家在电商平台下单，往往会涉及到两个动作，一个是扣库存，第二个是更新订单状态，库存和订单一般属于不同的数据库，需要使用分布式事务保证数据一致性。我们使用的解决方案是使用支付宝用得那个TCC补偿性分布式事务解决方案.问到什么是TCC,按照下面TCC的原理去说.

(TCC是什么?)

TCC是三个英文单词的首字母缩写,分别对应Try、Confirm和Cancel三种操作，这三种操作的业务含义如下：

Try：预留业务资源

Confirm：确认执行业务操作

Cancel：取消执行业务操作

1、Try：尝试执行业务。

完成所有业务检查(一致性)

预留必须业务资源(准隔离性)

2、Confirm：确认执行业务。

真正执行业务

不做任何业务检查

只使用Try阶段预留的业务资源

3、Cancel：取消执行业务

释放Try阶段预留的业务资源

TCC的原理

我给你用这个账务拆分为说一下TCC吧，比如说我们账务拆分的业务场景是，分别位于三个不同分库的帐户A、B、C，A账户和B账户一起向C账户转帐共80元：

1、Try：尝试执行业务。

完成所有业务检查(一致性)：检查A、B、C的帐户状态是否正常，帐户A的余额是否不少于30元，帐户B的余额是否不少于50元。

预留必须业务资源(准隔离性)：帐户A的冻结金额增加30元，帐户B的冻结金额增加50元，这样就保证不会出现其他并发进程扣减了这两个帐户的余额而导致在后续的真正转帐操作过程中，帐户A和B的可用余额不够的情况。

2、Confirm：确认执行业务。

真正执行业务：如果Try阶段帐户A、B、C状态正常，且帐户A、B余额够用，则执行帐户A给账户C转账30元、帐户B给账户C转账50元的转帐操作。

不做任何业务检查：这时已经不需要做业务检查，Try阶段已经完成了业务检查。

只使用Try阶段预留的业务资源：只需要使用Try阶段帐户A和帐户B冻结的金额即可。

3、Cancel：取消执行业务

释放Try阶段预留的业务资源：如果Try阶段部分成功，比如帐户A的余额够用，且冻结相应金额成功，帐户B的余额不够而冻结失败，则需要对帐户A做Cancel操作，将帐户A被冻结的金额解冻掉。

(TCC 怎么用的)

Github上有他们的源码,我们直接把源码挡下来,安装到我们本地的仓库里,用的时候我们把需要使用分布式事务的代码,上加上@Compensable注解,里面还有一些其他的属性配置上就可以了

## 35、项目中有没有涉及到分库分表,怎么拆分的?(MyCat)

我们用的MyCat中间件拆分的,他支持mysql集群，或者mariadb，提供高可用性数据分片集群。我知道拆分数据库的方式一般分为垂直拆分和水平拆分,我们在项目中用的是水平拆分。后端用户可以把MyCat它看作是一个数据库代理，用MySQL客户端工具和命令行访问。

## 36、你们项目的边界是什么?

我们公司的项目边界都是架构师和项目经理定的,这个我没参与过.

项目边界其实就是针对整个项目要完成到什么程度的一个定义,就是至少需要哪些个功能点啦,达到什么样的要求,都可以称之为项目边界。红色字体用于理解

理论:在执行项目的过程中，有两次机会定义范围。高端范围在预定义的项目过程中加以定义。这些范围声明有助于建立项目的边界。收集商业需求时，范围定义得越详细越好。如果把范围看作是一个箱子，那么高端范围用来定义箱子的大小和形状；而需求则定义箱子的内容。

# 数据库面试题

## 1.解释一下什么是索引?解释一下单列索引和联合索引?索引什么时候失效?，怎么添加索引？

单列索引是指在表的某一列上创建索引，联合索引是在多个列上联合创建索引。单列索引可以出现在where条件的任何位置，而联合索引需要按照一定的顺序来写。在多条件查询的时候，联合索引的效率更高，我们联合索引也最多创建两列。

我们创建索引的时候也得考虑到我们这张表的更新频率，如果表里索引比较多的话是比较影响更新速度的，为什么会影响更新速度？因为创建索引的过程其实就是构建一个二叉树，而每次更新完数据都得重新计算二叉树，所以就影响更新速度。我们表中索引我大概就了解这么多了。

索引并不是时时都会生效的，比如以下几种情况就能导致索引失效效：

1. 如果条件中有or，即使其中有条件带索引也不会使用(这也是为什么尽量少用or的原因)

注意：要想使用or，又想让索引生效，只能将or条件中的每个列都加上索引

1. like查询是以%开头，会导致索引失效
2. 如果列类型是字符串，那一定要在条件中将数据使用引号引用起来,否则索引失效
3. 如果mysql估计使用全表扫描要比使用索引快,则不使用索引

此外，查看索引的使用情况

索引其实可以理解为数据的查询目录，建索引的目的就是为了提高对表的查询速度；没有索引时，查询时全表检索，有了索引就可以根据索引快速查找需要的数据；但是索引也不能乱建，因为索引需要维护，会导致增删改的效率降低。会使数据的维护变的复杂，影响开发的效率，索引也会占用数据库的物理空间；所以我们一般在项目的开发阶段、测试阶段、试运行阶段都很少去创建索引，因为有了索引，系统bug造成垃圾数据特别不好删除。只有在项目正式上线后才去增加索引，以提高项目运行速度。索引我们一般都创建在经常作为查询条件的字段、排序的字段和作为关联关系的字段上边。尽量避免在大文本字段、数据量比较小的字段（比如性别），增删改性能大于检索性能的字段上边；另外，有些情况，即使增加了索引，索引也不会生效，比如：索引字段使用了不等于（！=或者<>）符合，用了函数、进行了运算，使用了is null或is not null，

和不匹配的数据类型进行比较、like查询时两边都用了%等；还有一个要注意的地方是，如果在多个字段上建立联合索引，那么组合索引的第一个列被where子句引用时，索引才会起作用。因为想要使用索引增加查询效率，必然要牺牲增删改效率，为了解决这个问题，我们经常对数据库做主从复制，读写分离。同时创建两个数据库，一主一从，两个数据库数据完全一致，主的数据库用来进行写的操作，操作后数据库会自动把数据同步到从的数据库，从的数据库用来执行读的操作。这样我们建立索引时，就可以只在读的数据库创建就可以了。这样就索引即能增加查询效率，有不影响增删改效率。这样做了之后，我们还可以对他进一步优化，比如数据库引擎的优化，主数据库因为执行增删改操作，所以用事务型引擎Innodb，

读的数据库，不需要事务，就可以用效率更高的MyIASM引擎。同时根据实际情况，也可以配置一主多从或者多主多从。索引的创建常用的有2中方式：CREATE 【UNIQUE】INDEX index\_name ON table\_name (column\_list)；或者ALTER TABLE table\_name ADD INDEX index\_name (id,name)；修改用：ALTER TABLE table\_name REBUILD INDEX index\_name (column\_list)；删除用：DROP INDEX index\_name ON talbe\_name

或者：ALTER TABLE table\_name DROP INDEX index\_name

查看用：select \* from all\_indexes where table\_name='student';

Conlum\_list中多个字段用”，”号分割。

## 2.关系型数据库和非关系数据库的区别?

关系型数据库是表与表之间有关系比如一对多, 多对一,等值连接什么的,非关系型数据库就是里面没有多张表,没有什么关联

## 3.你有没有做过数据库建模,自己设计表和模块

一个有三年工作经验的人,一定要说设计过,(实际工作经验的程序员：在系统设计、需求文档，数据建模都应该有所涉及)数据库建模就是使用PowerDsiger工具，先分析项目需求，前端先出相应的原型,根据原型,我开始做相应的表,设计初期的时候表会有些小浮动修改等,再根据需求设计详细字段。如果后期客户需求改变时，表结构后期跟着调整，就是这样使用工具不断完善过程就是建模。

## 说一下左连接和右连接的区别?

我给你举个例子吧,比如说有两张表,一张员工表,一张部门表,我们想通过两表等值连接的方式查询出员工信息和员工所在的部门,但是想把没有分配员工的部门也想展示出来,那就得用到外连接了,如果用左连接的话就把部门表放到左边就可以了.

## 5.什么是读写分离?

这个我们项目中使用MyCat来做的,在mycat里配置好主库和从库,做增删改的时候是对主库进行操作,查询的时候是对从库进行操作,其实mysql本身从5.6以后的版本就带主从复制的功能了,他们是利用mysql里的log文件进行的数据同步.

## 6.where和having的区别?

他俩都是条件语句,where是在添加查询条件的使用的,having是在分组查询的时候添加分组条件用的.

## 7.去除表里重复的数据?

用那个distic就可以去除.

## 8.关于统计的sql语句,聚合函数？

count max min sum avg

## 9.数据库视图？

原来我们公司做过一个项目的时候，用的是5张表的联查，然后用sql语句来写的话，比较慢，比较麻烦，然后我们把这5张表的联查创建了了视图，然后就直接查找的是视图，查询速度快，这个视图就是只能做查询，而不能做增删改操作。

## 10.你们项目中事务是怎么配置的?

我们用的是Spring给提供的一个管理事务的类DataSourceTransactionManager,用的时候是直接在需要使用事务的service中用@Transactional注解的方式来管理事务的,他底层实现也是AOP的方式来管理的事务.

## 11.说一下什么事务?代码中事务是如何控制的?

在开发的时候我们同时对两条以上的sql语句进行操作,操作过程中中的如果有一条操作失败,整个事务回滚，只有全部正确才完成提交,比如说银行转账,从A账户向B账户转账,必须得两条SQL语句同时执行成功.事务一般有ACID四个特性,原子性(Atomicity),一致性(Consistency),隔离性(Isolation),持久性(Durability),

原子性:就是每个事务都是一个整体,不可拆分,事务中所有sql语句要么都执行成功要么都失败

一致性:事务在执行前后的的数据状态保证一致,比如转账前,两个金额总数是2000,那转账后的总数也是2000.

持久性:事务一旦执行成功,对数据的修改是持久的,就算关机了也是保存到硬盘上的.

**我们在项目中直接配置到Spring里就用啦,用的注解式的事务.**

**隔离级别:**

我们在开发的时候如果不注意事务的隔离级别话就会出现,脏读,不可重复读,幻读这些情况,比如说,把下面的概念挨个说一下

事务隔离级别:

* **脏读**: 数据库设置隔离方式(read uncommitted) 读未提交

指当一个事务正在访问数据，并且对数据进行了修改，而这种修改还没有提交到数据库中，这时，另外一个事务也访问这个数据，然后使用了这个数据。因为这个数据是还没有提交的数据， 那么另外一个事务读到的这个数据是脏数据，依据脏数据所做的操作可能是不正确的。

* **不可重复读:** 数据库设置隔离方式(read committed) 读已提交

指在一个事务内，多次读同一数据。在这个事务还没有结束时，另外一个事务也访问该同一数据。那么，在第一个事务中的两次读数据之间，由于第二个事务的修改，那么第一个事务两次读到的数据可能是不一样的。这样就发生了在一个事务内两次读到的数据是不一样的，因此称为是不可重复读。 修改状态为(repeatable read)可重复读,就在一个事务中两次读取到的数据是一样的.mysql默认也是这个隔离级别的.

* **幻读:** 数据库设置隔离方式serializable

指当事务不是独立执行时发生的一种现象，例如第一个事务对一个表中的数据进行了修改，这种修改涉及到表中的全部数据行。同时，第二个事务也修改这个表中的数据，这种修改是向表中插入一行新数据。那么，以后就会发生操作第一个事务的用户

## **12.MySQL和Oracle区别**

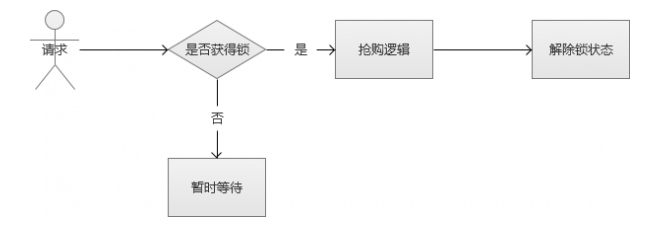
Mysql相比Oracle更简单更轻量,在开发中mysql用的更多一些。但是Oracle的一些基本点也都知道，Oracle的分页和mysql不一样，用的是伪列，mysql用的是limit。SELECT \* FROM ( SELECT A.\*, ROWNUM RN FROM (SELECT \* FROM TABLE\_NAME) A WHERE ROWNUM <= 40 ) WHERE RN >= 21 [控制分页查询的每页的范围]。Mysql主键可以设置自增，Oracle中并没有主键自增，我们可以用sequence方法来实现主键自增。

## 13.关于悲观锁和乐观锁

悲观锁思路

解决线程安全的思路很多，可以从“悲观锁”的方向开始讨论。

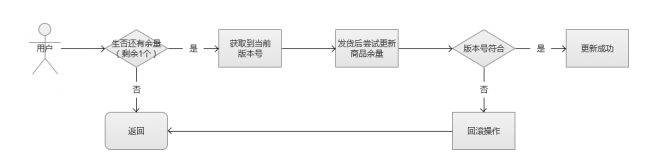
悲观锁，也就是在修改数据的时候，采用锁定状态，排斥外部请求的修改。遇到加锁的状态，就必须等待。



虽然上述的方案的确解决了线程安全的问题，但是，别忘记，我们的场景是“高并发”。也就是说，会很多这样的修改请求，每个请求都需要等待“锁”，某些线程可能永远都没有机会抢到这个“锁”，这种请求就会死在那里。同时，这种请求会很多，瞬间增大系统的平均响应时间，结果是可用连接数被耗尽，系统陷入异常。

乐观锁思路

这个时候，我们就可以讨论一下“乐观锁”的思路了。乐观锁，是相对于“悲观锁”采用更为宽松的加锁机制，大都是采用带版本号（Version）更新,我原来在hibernate中用过。实现就是，这个数据所有请求都有资格去修改，但会获得一个该数据的版本号，只有版本号符合的才能更新成功，其他的返回抢购失败。这样的话，我们就不需要考虑队列的问题，不过，它会增大CPU的计算开销。但是，综合来说，这是一个比较好的解决方案。



有很多软件和服务都“乐观锁”功能的支持，例如Redis中的watch就是其中之一。通过这个实现，我们保证了数据的安全。

## 14.简单说一下sql语句是怎么优化的

我们一般在开发的时候需要注意SQL规范,比如说最基本的不要用\*查询所有字段,还要就是经常用表别名,经常commit提交事务尽量让及时释放回滚点,如果使用函数的话尽量使用内部的函数,还有一个优化的地方是项目上线后,产生大量数据后,由测试人员发现某些的功能点响应比较慢,反馈给我们开发人员,我们开发人员找到这个功能点对应的sql语句,如果这个sql语句写的比较复杂,我们就得用explain去分析一下SQL语句的执行计划,看下是不是索引失效了,或者是执行到某一个子查询的语句执行比较慢,然后我们去优化这些个sql, 优化方式是什么?比如说加索引,或者创建视图.

## 15.什么是死锁

**打个比方，假设有P1和P2两个进程，都需要A和B两个资源，现在P1持有A等待B资源，而P2持有B等待A资源，两个都等待另一个资源而不肯释放资源，就这样无限等待中，这就形成死锁，这也是死锁的一种情况。给死锁下个定义，如果一组进程中每一个进程都在等待仅由该组进程中的其他进程才能引发的事件，那么该组进程是死锁的。竞争不可抢占资源引起死锁,也就是我们说的第一种情况，而这都在等待对方占有的不可抢占的资源。**

## 16.死锁的发生必须具备4个条件

**1.互斥条件:** 其实就是进程对所分配到的资源进行排它性使用，是指在一段时间内某个资源只能由一个进程占用。如果此时还有其它进程请求资源，那么其它线程只能等待，直到占有资源的进程用完被释放掉。**2.请求和保持条件:** 指进程已经保持至少一个资源，但又提出了新的资源请求，而该资源已被其它进程占有，此时请求进程阻塞，但又对自己已获得的其它资源保持不放。3.**不剥夺条件:** 指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。4.**环路等待条件:** 指在发生死锁时候，一定存在一个进程相当于资源的环形链，也就是进程的集合像{P0，P1，P2，···，Pn}中的P0正在等待一个P1占用的资源；P1正在等待P2占用的资源，……，Pn正在等待已被P0占用的资源

## 17.MySQL里加索引的原则？

主键索引(主键中的每个值都唯一),唯一索引(不允许其中任何两行具有相同索引值的索引),普通索引,全文索引,组合索引。

优缺点：大大提高了查询速度(原理：通过不断的缩小想要获得数据的范围来筛选出最终想要的结果)，但是降低了更新表的速度（增加和修改)。

## 18.Mysql优化方案?

我们一般从以下几点入手:

1. 使用支持mysql主从复制的版本

2. 在使用MySQL时使用索引

3. 优化sql语句

4. 优化MySQL服务，修改MY.INI文件,根据服务器配置缓存的大小

5. 根据服务器配置索引的缓存

6.任何地方都不要使用 select \* from table ，用具体的字段列表代替“\*”

7.尽量避免向客户端返回大数据量，若数据量过大，考虑是否可以分页返回7. 使用第三方技术mycat，对数据库拆分，水平拆分和垂直拆分，我们用的水平拆分

8. 设置表内存放数据上限，同时通过mycat配置读写分离

## 19.Mysql的批量导入

我们用mybatis的时候做过批量导入,有一个forearch动态sql语句的方式去实现过批量导入,还有我们在执行批量导入的时候注意sql语句的执行,inser into(id,name)values(1,’张三’),(2,’李四’) ......这样的方式导入,执行效率高.

## 20.mysql内部函数多了解一下

concat,trim,replace,substring,curdate() #返回当前日期,curtime() #返回当前时间,now() #返回当前日期+时间if(value，true，false) #如果value值为真，则返回true，否则，返回false

selectif (salary > 3000，'Hight'，'Low') from salary;

selectid，salary， if (salary <=> NULL，'NULL'，'NOT NULL') from salary;

2、ifnull(value1，value2)#如果value1不为空，则返回value1，不然返回value2

#可以用来进行空值替换

selectifnull(salary，0.00) from salary; 等 等..

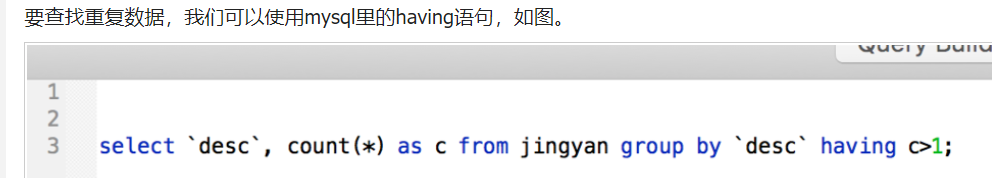
## 21.mysql的存储引擎用过哪些？

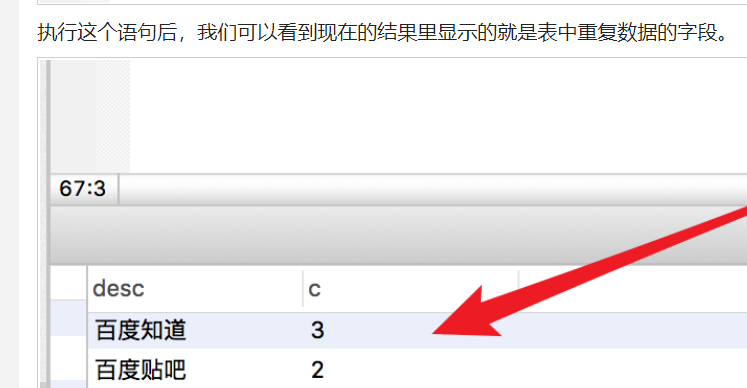
我们默认用的是innodb支持行级别事务的， 还有一种是ISAM的，这种是支持表级别事务的。

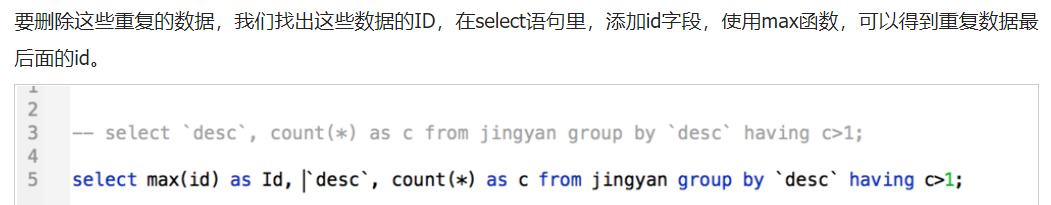
## 22.常考SQL语句？

## 23.如何删除表里的重复数据？













## 24.Mysql数据库配置文件调优

如果是Linux上的话调试的是/etc/my.cnf文件,里面可以设置一些增加一些内存,增加连接数,调整下缓存大小,我自己有一个文档,一般都是看着文档去调试一下

# 面试现场出的场景题

## 你项目中大概有多少微服务?怎么搭建的,会涉及到哪些服务?

大概得有个七八个微服务,让我现在搭建的话,我会用springBoot和Springcloud那一套全家桶进行搭建,持久层使用JPA,第三方的中间件会用到rabbitMQ,freemarker,ElasticSearch等这些中间件技术.

## 2.ERP系统里主要是干什么？

ERP是一种企业资源管理与业务流程管理的计算机系统，对企业内部所有的经营活动，包括采购，财务，人力资源，物料等等的整合，最终达成高效化经营的目的。

## 3.你们做的这个项目大概有多少接口？

大概有几百个，涉及到接口三要素的就是接口

## 4.什么是策略？

就是解决项目中的问题，solr，redis，等等。

## 5.你平常和哪几个部门进行沟通？协调哪些内容？

产品经理：沟通一些业务逻辑、功能点。

测试人员：把一些bug提交到BugFree的平台，有哪些bug然后改一下

## 6.需求文档设计,接口文档设计,系统设计文档都写过哪些,怎么写的?

接口文档也叫接口api，三要素：url,参数，返回值。

需求文档写过吗？当然写过啊。客户在做需求调研的时候 ，我们也写过的。需求文档大概是这样写的。比如客户的那些需求呢理出来，。

## 7.开发环境是否统一

我们在开发项目时主要用到IDEA 、JDK1.8、Maven、 ElasticSearch、 nginx、 Tomcat8.0、 MySql、 redis等开发工具或环境搭建，那么我们在开发同一项目的时候 开发环境必须统一，公司项目的开发主要还是基于maven管理的聚合工程，分布式架构项目，我们每个人负责的功能模块都不相同，如果开发环境不统一，可能导致我们项目合并时候出现版本冲突，或者jar包等不兼容之类的事

## 8.专业书籍你都看过那些?

我原来看过Java编程思想,看了几十页看不下去了,后来看过一些高并发编程,分布式服务框架,Redis实战啥的,家里有一堆,不想打游戏的时候就看看.

## 9.平常有自己搭服务器嘛?

搭建过,比如说dubbox的注册中心 zookeeper,solr Redis 都是我们自己搭建的 这些都是最基本的

## 10.有一个新的页面(或者模块)让你开发，需要多长时间：

您能给我说一说这页面或者模块要实现的功能，它要实现什么样的效果，还有它的需求文档等等一些资料，这样我才能根据这些资料，来大致算一算要多长时间能完成。

## 11.在开发的时候设计过哪些文档,有什么规范吗?

传统方式： 接口文档三要素：URL 、参数、返回值

需求文档： 客户做需求调研的时候写过，根据客户的需求我们来写文档。画流程图、写开发步骤、主要功能点的介绍等

系统设计：有哪些功能点、规定了具体方法、类、字段，涉及数据库表、字段，后期编码

互联网项目：采用敏捷式开发，利用swagger设计文档，生成文档的URL、参数、返回类型。

## 12.谈谈你上家公司的薪酬结构

以总工资10000为例子,我们上家公司签合同的时候基本工资4000多、岗位工资3000多、绩效工资3000,再扣点税和社保,差不多每个月到手也就9500左右

## 13.你们公司有没有专门的测试环境,专门有一个地址 发布项目进行内部测试的?

**有 我们有专门测试的服务 专门测试的服务 有一个测试的版本 有一个环境 比如说连接数据库 测试环境无非就是把配置变一下，一般都是有生成环境 开发环境 还有测试环境 一般都是这么分的 咱们公司是这样的吗 测试是怎么做的呢？**

## 14.上家公司的地址?

**上家公司是在上地那块的嘉华大厦里呢， HR还会聊你在哪住,每天坐几号地铁过去,这个一定要说清楚,我在沙河高教园住的呢,我每天直接坐昌平线到西二旗,换13号线人太多了我就直接骑个小黄车就过去.**

## 15.公司有技术分享吗?

**当然有啦,我们公司每周五的下午都会有一个小组做技术分享，分享一个技术,或者我们项目当中相关的新的业务，如遇到了什么问题，是怎么解决的。最新技术的分享 ，像springboot 微服务,分布式事务啦等等,**他会问分布式事务,照着后面的说

## 16.说一下你们公司的离职流程

**如果你以在职的身份找工作的化,可以这么说,现在我们项目也正好做完了,这几天也没有新的项目,就是配合测试的人员测一下,大概一个星期左右就可以交接完到岗.**

## 17.感觉在开发中比较有成就的项目

**说你简历上最近的这个项目就行,用的中间件比较多,技术也比较多**

## 18.与前台的接口联调怎么做？

**我们都是一个局域网,一开始进行简单测试的时候,我这边直接启动服务器就可以他们就可以访问到我的项目.后期我们会把项目打包发布到测试环境进行UAT测试.**

## 19.技术选**型这块是你自己**选的吗

**嗯,有一部分是我自己选的,比如说redis的使用,在搜索的时候,上面能显示的那些品牌,规格什么的,当时准备直接从索引库中查,但是评估了一下比较麻烦,我就把他存到redis中了.**

## 20.有没有自己设计的模块？数据库？

**当然有,商品模块就是我自己设计的,当时我也是到处参考,问以前做过这一块的朋友然后设计出来的.**

## 21.写完之后有测试吗？秒杀这块怎么模拟？怎么测试？

**我开发完的模块自己都有测试,秒杀这一块我主要测试的业务逻辑,我知道你想问怎么模拟多用户秒杀,测试的那边他们用jemeter模拟的线程组去测试,这个我以前也用过**

## 22.拿到一个需求怎么去设计？技术选型什么的？

**我得结合咱们以前的项目中已有的技术去设计,还得根据业务逻辑和场景去考虑架构,比如说,把咱们电商那一套说一下.**

## 23.你负责的模块你测试过吗?

**有啊,我测完了才提交测试那边呢**

## 24.通过什么方式学新的技术呢?

**平常关注一些公众号什么的,有时候也买一些书看.**

## 25.你们的搭建好的服务器是放到机房还是哪个地方?

**我们公司用的是阿里云的服务器,直接发布到阿里云上就行了,我觉得现在应该是除了银行或者那些电信运营商,应该很少还有自己组建机房的吧,都应该用的阿里云的吧.**

## 26.服务器搭建是你搭建的吗?

**搭建过,在linux上搭建redis啦,zookeeper,solr,还有mysql啥的都没问题,我都有现成的文档.**

## 27.你这个项目的用户量大概有多大?

**以电商项目为例,我们这个项目上线以后,用户量大概也就四五万, 每天访问量也就一万左右PV,并发量也就1000多**

## 28.上线的项目中app端客户如果出现错误操作，提示温和的温馨提示信息后，如何根据这个提示信息找到项目中的错误

我们会根据日志记录，到时候查日志就行了。

# 项目架构和项目部署问题

## 1.项目怎么部署到服务器上

## 2.什么是微服务?为什么使用微服务

**微服务也是一种分布式的架构,把项目中的每一个模块采用单独的业务逻辑封装，每个项目都可以独立部署和扩展,服务之间可以通过springCloud的Eureka组件作为注册中心使用Feign组件进行服务间的互相通信,我个人认为使用了微服务架构可以降低我们代码的耦合度,在开发的时候,比较好分配项目,还有就是后期再进行项目功能扩展的时候也比较灵活.**

## 3.BS架构和CS架构的区别?

**BS架构就是浏览器和服务器交互 CS就是客户端和服务器交互**

## 4.简单介绍下分布式

**分部式项目就是项目中的模块,把每一个模块拆分成为一个个的工程，这样我们各个模块之间的耦合度不就降低了嘛，然后我们各个模块之间通过dubbo框架来相互通讯，我们公司用到的是一个第三方的插件dubbo来实现项目之间通信，这样我们的代码的拓展性不就更好了嘛，最后我们在部署项目的时候，可以根据用户的访问量来给各个模块的服务器增大配置，我们还可以采用nginx负载均衡来实现一个反向代理，当用户请求后台服务器的时候，有nginx来决定哪台服务器空闲或者压力小，来让用户访问。我们这样做的目的不都是为了防止高并发嘛**

## 5.说一下Docker

用过,Docker就是为了缩短代码从开发、测试到部署、上线运行的周期，能让项目具备可移植性，易于构建，并易于协作。（通俗一点说，Docker就像一个盒子，里面可以装很多物件，如果需要这些物件的可以直接将该大盒子拿走，而不需要从该盒子中一件件的取。比如说我们可以在这个容器里装好zookeeper,redis,mysql,tomcat等软件,用的时候直接用就可以,项目部署的时候,直接把当前的Docker给测试组就可以,或者是运维项目组就行.）

## 6.用过Junit的断言吗?

我们公司有要求做单元测试,但是都是后期补的,有的补全了,有的没补全,主要是我们公司里没有QA,所以这一块要求的不严, 其实断言就是断定结果的,就是符合我们预期输出的结果就返回true 不符合返回false 给出错误信息

## 7.什么是RestFull风格架构?

就是一中架构风格,可以直接通过URL访问controller,我们前后端分离的项目不都是采用这样的方式架构的么.

## 8.docker容器常用的命令

记住常用的创建容器,删除容器,查询所有容器,启动容器命令就行,其他的看看就行.

# 镜像和容器的区别?  
镜像可以理解为java类,  容器理解成为java中的对象.  
  
  - 查看镜像文件命令:  docker images  
  - 搜索镜像:  docker  search  搜索名字  
  - 创建交互式容器: docker run  -it --name=mycentos centos:7 /bin/bash  
  - 查询所有的容器:  docker  ps  -a    
  - 查询正在运行的容器: docker ps

## 9.项目中共部署了几个项目

我们的项目采用分布式的架构，每一个模块都可以作为一个项目，每一个项目都可以分为服务端和消费端，按照其功能分可以分为7个，分别是e3-web-manager,cart,item,portal,Search,sso,order.这几个项目组成

## 10.你们项目有几个war包啊？

就以我们最近这个电商项目来说吧,这个项目最终上线的时候差不多有8个war包商品管理,搜索,用户,平台后台,商家后台,门户网站,购物车,还有一个秒杀

# 算法相关面试题

## 1.JAVA中常用的加密算法

对于不可逆的加密算法有

MD5和SHA, 通过散列算法进行加密SHA加密比MD5安全性更高,常用SHA-256加密算法。

DES加密算法，对称加密，客户端和服务端公用一个key,该key最好是随机生成,对于这种加密算法加密效率高,但是据说24小时以内可以破解。

AES加密算法，不对称加密算法，通过公钥加密，私钥解密，私钥加密，公钥验证签名。(支付宝) 是目前比较安全的加密算法，但效率偏低。安全性是DES加密算法的1000多倍数。

## 2.排序算法? 说一下时间复杂度和空间复杂度

冒泡排序：

每次拿两个数进行比较，总是把小的放在最前面，大的放在最后面，第二次是第二个数和第三个数比较，如果第三个小于第二个，那么第三个和第二个替换位置，一次这样比较替换，第一轮循环完了呢，放在最后面的那个数就是这 一串数字中最大的数。第二轮循环和第一轮一样，第一个和第二个比较，最小的放在最前面，依次类推。如果有10个数字，那么就进行9次循环，最小的在最前面，最大的放在最后面。

好处呢，就是查询速度快

二分查找：

二分查找就是从一串数字中找到某个数字。

原理就是必须有一串数字是从小到大排序，把这串数字进行划分，分为三段，前半段，中止段，和中止后半段。查找数字或者字符先从中止段查找，查到的数字或者字符和中止正好相等，那么久直接取出来，如果比它大就从后半段查找，后半段在进行二分法，进行递归的调用查找，如果比中止小，走前面的中止段，在进行拆分，进行二分查找法。

## 3.什么是时间复杂度和空间复杂度?

# 十、23种设计模式

## 1.单例和工厂是如何实现的?

单例：其实单例本质的就是控制对象的产出，要想产生一个唯一的实例，你就要私有化构造方法，然后私有化一个的静态的全局变量，并且提供一个公共的静态的方法供外界访问，单例大概就是这样的。

工厂：它是 java设计模式的一种，工厂模式的优势还是很多的，我举一个简单的共厂的例子吧，就是假如我们有好多接口，好多的接口的实现类，我们如果使用工厂模式的话，当我们有需求需要改变接口的实现类的话，直接在工厂里面改变返回的类型就好。这样会应对不同的需求变化。

## 2.你在项目哪些地方用过设计模式

Spring框架用的就是工厂模式

# 十一、开发工具使用

## 1.你说一下你们项目中的maven工具是怎么用的?

Maven 简介及优点：maven 是一个项目管理工具，其核心特点就是通过 maven 可以

进行包的依赖管理，保证 jar 包版本的一致性，以及可以使多个项目共享 jar 包，从

而能够在开发大型 j2ee 应用的时候，减小项目的大小，并且和 ant 比起来，maven

根据“约定优于配置”的特性，可以对其项目的编译打包部署进行了更为抽象的封

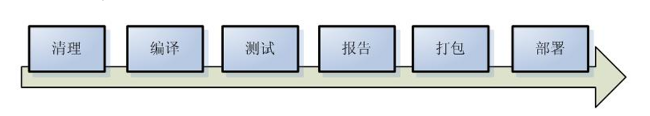
装，使得自己不需要像 ant 那样进行详细配置文件的编写

Maven 常用命令：maven –v 查看版本号

Maven package 项目打包工具 Maven test 测试 Maven install 模板安装，将打包的 jar/war 复制到本地仓库中 Maven deploy 发布命令 Maven clean 删除

## 2.Maven生命周期？

以上，mvn clean、mvn compile、mvn test、mvn package 等命令的执行就是Maven的一个生命周期



## 3.maven的jar包冲突怎么解决

可以使用mvn分析包冲突命令：mvn dependency:tree

手动排除，在pom.xml中使用<exclusion>标签去排除冲突的jar包：代码不用背，知道就行

|  |
| --- |
| <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-netflix-hystrix</artifactId>  <version>1.4.4.RELEASE</version>  <exclusions>  <exclusion>  <groupId>com.google.guava</groupId>  <artifactId>guava</artifactId>  </exclusion>  </exclusions>  </dependency> |

## 4.平常用什么开发工具

IDEA,Eclipse,MyEclipse,数据库操作用nvacat或者sqlyong,还有数据库建模用powerDesner

## 5.SVN使用流程

我每天早上过去先下载下最新的代码,我写完了要提交的话,就是先同步,再提交,有时候也会遇到冲突的时候. 冲突怎么解决的? 我是先把自己的代码备份一份,然后把服务器上最新的版本代码下载下来,然后把我新增的代码方进行,再提交就行.

## Git使用流程和常用命令

从远程服务器拉取项目就是git clone 往远程服务器上传是git push 平常都是用工具,不写命令.遇到冲突,直接拉取就行, 拉取下来把里面的多余的符号去掉再提交就行.

拉取项目:git clone 创建分支, new branch 合并分支meger

## 7.你们git服务器是用的gitlab还是github还是本地的?

我们公司搭建的是Gitlab做的远程服务器端的.

**8.项目中有多少台服务器?**

我们项目大概十多台服务器吧

# 十二、人资常见问题

## 1、为什么要离职?

1.我们公司的项目准备南迁，而我个人不喜欢去南方发展,我也想换个工作环境.

2.我们公司转型做别的了,老板做微商了,开发这一块就不需要了,所以就离职了

3.合同正好到期了,我也想换个环境了.

4.老板转行做实体经济去了,做出口贸易了.

## 2、团队管理?

首先，要让自己的组员处理好之间的关系，如果有问题的话，在一块进行解决，有新技术的话一块分享。就像我上一家项目经理，根据功能点和自己团队中人水平的高低进行分工，然后我就整个管理一下项目的进度，每天在干什么，每天在做什么，做到哪里，然后用禅道做一下评估和安排，大概就是这么做。

## 3、职业发展规划?

像自己干开发这么长时间了，今后是要往项目经理这方法发展了。我觉得做项目经理肯定要处理好与组员的关系。关系处理好了，这样大家有问题了共同解决，有新技术了共同分享。公司要是给我们分配下任务了。根据组员水平的高低去分配任务，分完之后我就每天去管理项目的进度，每天应该该做什么，进行到哪一步。用project做一下工作量的评估和安排。就像我们上个项目经理他那种人格魅力和人员分工上做的特么好，我们周末经常去撸串。我觉得我像他那样做就能做到项目经理。让下面的组员觉的跟着我干舒服。这是最主要的。

# 十三、面试失败的原因

## 面试官问的问题,我们在回答的时候感觉太像背

## 问到项目的细节不会说

## 不要总是说这个模块不是我负责的?

## 瞎说

# 十四、面试心得

## 面试终极秘籍：笑

|  |
| --- |
| 笑。大家在面试过程中一定要学会笑。笑可以减少紧张情绪，笑可以缓解尴尬，微可以拉近和面试官的距离。大家记住一点，面试官都讨厌总是面试，他非常希望你能成功。所以不要因为回答不上问题而苦着脸，不要尴尬，尴尬时就傻笑。 |

## 给予面试官信心

|  |
| --- |
| 面试过程中，要对自己有信心，还要让面试官对你有信心。所有的答案都要给面试官肯定的回答。坚决不能说，可能、或许、好像这样的词语。 |

## 技术不行业务补

|  |
| --- |
| **技术的终点是业务**，脱离业务的技术，只能是纸上谈兵。技术就是为了实现业务的。技术上总有解决不了的问题，这时候从业务考虑，完全可以用其他方法实现。我们java涉及到的技术点太多了，没有谁能把不同面试官的所有技术问题都打上来的，现在大家都面临面试了，没有那么多时间去学习了，如果你还纠结于技术的学习，那么你会发现，技术是永远学不完的，学的越多，不懂的越多，越学越没有自信。那么怎么办呢？  一句话：“技术不行，业务补”，针对你的第一个项目，多想想，仔细想想，打好草稿，你的项目是做什么的，主要业务流程是什么，你的模块的独特性在哪里，项目主要有哪些用户，项目的亮点在哪里，赢利点在哪里，同类竞争网站有哪些。项目的主要架构是什么样的，开发过程中遇到了哪些问题，有哪些问题是遗留没有解决的。等等。首先给自己定一个目标，打一份草稿，这份草稿，可以保证你，面试官一句话不说，你自己介绍项目就能介绍30分钟以上。可能面试的时候，很多面试官不给你说业务的机会，但是你一定要有所准备，当技术说不上来的时候，就说业务，用业务来描述技术。当然在描述业务的时候，也不能光说业务，在业务中引出技术，只有说出为什么在这块要使用这个技术，使用这个技术做什么，用它有什么好处就可以了。 |

## 机会不等人-海量面试-一气呵成

|  |
| --- |
| 银行创想，长期招聘不招人；  公司不会一直招人，招够了就不招了 |

## 项目介绍半小时

|  |
| --- |
| 1. 项目架构 2. 项目框架 |

## 传递正能量

|  |
| --- |
| 面试中，成功的最佳办法，就是不断给面试官传递正能量，告诉面试官，   1. 我能吃苦，我能加班，我能出差。 2. 给我分配了任务，我就会积极去解决，不会逃避，不找理由，不找借口，方法总比问题多。只要用心，就没有解决不了的问题。 3. 我有很强的学习能力，经常喜欢去网络上自己学习。 4. 我有很多同行业的朋友，我们经常交流，互相帮助。 5. 我认为技术没有难不难，只有会不会，我们这一行业新技术层出不穷，关键在于学习力，我对我的学习力还是非常自信的； 6. 这个技术点，如果让我做，我肯定能做，就是不知道该如何表达。 |

## 每个知识点要讲透彻，把知识点串起来，不给面试官提问的机会

## 准备3个技术点，每次面试都要想办法讲出来，逐渐掌握主动权

## 毕业生分享心得

|  |
| --- |
| 1、那么为什么面试官总是比你懂呢？胆大、心细、脸皮厚；  面试时，一定要自信，你掌握的知识点是最新的，是很高大上的，你的知识面是非常广的，已经站在了市场上开发程序员的80%以上。那么为什么面试官总是比你懂呢？不是面试官比你懂，而是他在问，你在答，他可以不懂装懂。所以只要你自信点，把你的话术说出来就OK了。面试官听你说的自信，他就认为你懂了，看你说的不自信，就会认为你也就懂点皮毛。  2、总是有几个问题，被追问，一问就不知道怎么回答了怎么办？  这个时候可以尝试引申回答，或者说自己的思路。甚至可以直接告诉面试官，这个问题不太懂。不懂的原因是之前没有接触过，但是多年的工作经验造就了我很强的自学能力，我相信给我1-2天时间，我肯定能掌握这个技术。  3、技术在精，不在多。  如果你的能力有限，记忆力不行，就可以在项目里少写几个技术，但是功能模块不能少，有很多人，在企业里边3-5年，只做了1-2个项目，用的还知识SSH框架，从来没有用过springmvc，redis、solr、工作流等，但是他再找工作时，也非常自信，因为他在企业里边真正工作过了几年，他觉得自己的能力就值这个钱。所以如果你的学习能力、记忆能力差，那么大部分技术，掌握个基础就可以了，知道他是做什么的，用它有什么好处就OK了，只要找一个技术点，可以掌握的很扎实就可以了。  4、总时被问到底层怎么办？  不要想着每个底层问题都说过面试官，这个没有几个人能做到，这么多技术，你学都学不过来，怎么可能还对底层研究的特别清楚呢。不懂的，就大胆告诉面试官，这个我没有研究过，但是工作这么多年，您交给我的任务，我肯定能在合理的时间，甚至比很多人更短的时间内完成。  5、吹的这么厉害，到了公司做不了怎么办？  实际的经验告诉我们，面试时的问题，和你在工作中遇到的技术，差别大部分时候是差别很大的。之前毕业的学长都告诉我们，面试时问了一大堆，结果到了公司，做的项目中没有用到几个高大上的技术，即使使用了，也都是别人已经封装配置好了，你只需要直接使用就可以了，并且还有使用的例子。  6、总是被查学历怎么办？  培训的学生，没有几个是有学历的，即使有学历，也很少能排上用场。能排上用场的也大多数都是找的非专业或者是朋友推荐、实习生等工作。有能力的人，很少有因为学历原因出问题的。之前有个学生是这样的。面试邀请上，明确硬性要求必须是“本科计算机相关专业”；这个学生办的是假学历，面试过程中，面试官当场百度后，  问；“你这个学历是14年的本科，但是我网上查到，你这个学校是15年才从大专转成本科的啊”？  学生答；“您是百度查的吧”；  面试官：“是”；  学生答；“那就对了”；  面试官：“嗯，怎么就对了？”；  学生官：“之前也有人问过我这个问题，我也不知道怎么回事，学校申请升本很多年了，有些专业是本科，有些是专科的，我也不知道怎么回事，我读大学时，学校告诉我本科比专科会多次2000元学费，我肯定就学本科了啊”；  面试官:"还能这样，你怎么上的大学?";  学生答；“考上的啊，我也不知道怎么回事”；  之后这个学生就收到了offer，入职时，学生办理入职手续。hr给了学生一张表，让学生填写，hr直接上网去查学生的学历，问学生：“你的学历怎么差不到呢”？  学生答：“您是不是查错了，再查查”；  hr："还是查不到，你自己查过吗？";  学生答：“没有啊，都是国家给的，我也不太清楚”；  hr：“那你明天把原件带过来吧？”；  学生答：“好”；  结果几个月过去了，学生一直没有带原件过来，hr也没有再问过。  7、面试了很多家，还是没有成功？我是不是该降低标准呢？  降低标准可以，但是不能太离谱了，简历相应要进行修改。工作是自己争取来的，绝对不是请求别人施舍过来的，做软件的公司都不缺钱，但是绝不会施舍。你简历包装1年，企业不给你面试机会，很正常，企业不需要初级的开发人员，企业不缺钱，是希望能招到能干活的人，而不是不能干活，还会拖慢进度的人。但是你包装了3年，有面试机会了，你本来说的还可以，问你要多少钱，你说我要8k，那么你作为一个面试官你会怎么想，10K以下的程序员都是初级的，工作3年了，还是一个初级，你敢相信吗？你会怀疑吗？。第二个问题，问你期望多少，你说15k，面试官问，能谈吗，10K接受吗？你说可以。这时候面试官怎么想？你会相信这是真的吗？所以，万事一定要把握一个度。  8、白天面试了一天很累，面试回来就回去休息了，或者打几把游戏。背话术时总是走神，心里总是惦记着游戏、小说或者电视剧，这样能被会吗？所以面试时，一定不能分心。白天去面试，面试必录音，面试结束后就回学校，回教室，找同学、找老师、百度、听录音，把自己面试中遇到的问题，总结一下，弥补自己的不足。这样，你第二天再去面试时，肯定会更有信心。如果能坚持这样，面试一次学到至少1个技术点，面试10家以后，还有什么理由不成功呢。10个技术点不够，面试20家，是不是就能学到20个知识点，还不够30、40、50、呢？ 只要我们坚持，不放弃，认证对待。那么世上无难事，只怕有心人。  9、有些人有说了，老师我非要找工作吗？找不到这个工作，我就不行了吗？  请你想想你来培训花了父母多少钱，父母挣钱容易吗？人活一辈子，总是要有一段时间吃苦的，与其到老了后悔，不如年轻时经历了，到以后回忆起来，年轻的一切都是那么幸福。人必须学会坚持，没有坚持，有再好的条件，你也很难成功。一但有了第一次放弃，那么你就可能习惯放弃。当你第一次享受到坚持的果实以后，你就成长了，你就会发现：世上无难事，只怕有心人。    最后送大家一句话：学校就是你们的家，我就是你们的后盾，面试过程中，一定要经常回学校来做复盘，工作中遇到问题，也要首先想到学校，想到  穆总。没有任何一所学校能向培训这么对每一个学生负责，希望大家学会感恩， 穆志军 |

## 面试切忌问题

|  |
| --- |
| 1. **面试中，切忌怀疑自己**：即使说错了，也要坚持。只能说，可能是我理解的不够深，不太理解您的意思，不过这个使用肯定是没有问题。 2. **面试中，切忌靠大牛**：千万不要说，这个问题是我和另一个大牛一起做的，他主要我辅助这种话。如果不太懂，可以直接说，这个不太理解，之前项目中只是根据网上的一些方法或者api直接copy做的，或者直接网上找的一个工具类，直接调用。还没有来的及去深入了解。 3. **面试中，切忌太多的可能、好像、不肯定语句**： 面试过程中，你一定要给面试官肯定的答案，会就是会，不会就是不会。不要说太多可能、好像等话。除非你是在说思路。 4. **面试中，不要因为怕自己理解的太浅，而不说：**面试过程中，一定要懂多少，就要尽可能的说出来，不要怕说错，更不要怕追问。被质疑了或者追问了，再说可能理解的不够深，最好能反过来向面试官请教； 5. **面试中，切忌一问一答方式：**把面试一定要当成一场聊天，每个问题就是一个话题，不是面试官在问，你在答，你也可以反问，可以转移话题，最好的方式是把问题展开来聊，不要局限于问题本身，加入一些业务，加入一些开发中间遇到问题、解决问题的过程。你只要把面试官聊开心了，你的面试也就成功了。哪怕面试过程中，都是你在请教面试官在答。 6. **面试中，切忌和面试官较真：**当然如果你能在面试中，和面试官因为一个问题争吵起来，那么说明你离面试成功已经很近了，只要稍微注意谦虚点，让一让面试官就OK了。 7. **面试中，切忌怀疑自己说的太low，而不好意思说：**面试过程中，要把面试官当成小孩，耐心的给他讲解你这个知识点的开发过程，不要把面试官想的太高大上，等面试官的思想和你同步。一定要多说，不厌其烦的说，等面试官不想听了，去打断你。 8. **面试中，切忌太长时间的沉默：**面试过程中，切完不要和面试官长时间的沉默，当面试官暂停提问，想着接下来问你什么问题时，一定要主动引起话题。有以下几个个方法：1、说自己的擅长技术点；2、说自己的项目特色业务；3、说自己最近关注的新技术；4、说工作中遇到的一些问题和解决过程的故事。5、主动想面试官提问（贵公司主要做什么项目、用什么框架、主要用到什么技术、公司的文化、团队的组成、请假面试官贵姓、什么职位、招聘人的标准等等） 9. **面试结束后，必问问题：**如果自己感觉回到的不太好，可以说：您好，今天的面试我感觉表现的非常差，请问我是不是没有机会和您共事了，都是做开发的，能不能交个朋友，加个微信呢。   **如果自己感觉回答的还可以：**就说，您好，如果我有幸能进入咱们公司，那么您觉得我哪方面需要加强或补充呢。 咱们最快能什么时间通知我面试结果呢。如果面试官说1-2天，就说：“好，那我就回去等通知了，今天和您聊的很开心，学到了不少东西，很期望和您能成为同事，无论成功或者失败，您是否都能给我一个通知呢”。  如果面试官说：3天内或者一周内，那么就说：“这个能不能稍微快点呢，我之前面试了一家，让我明（后）天就去入职，就是离我家太远了，我也不想再多面试了”； |

## 面试不能等

|  |
| --- |
| 1. 现场要offer，如果面试时间在30分钟以上，自己感觉不错，一定要现场要offer，让面试官尽快给答复，要不当天，要不第二天。告诉面试官我已经有了一个offer准备入职；因为好事多磨，面试官可能开始觉得你还不错，但很可能一会就会有个比你更好的。 2. 复试一定要先拒绝一次，直接问面试官，可不可以马上进行复试，我可能没有时间再来参加复试了。 |

## 1.12面试指导方针

在面试官问到技术的时候

是什么？有什么特点？ what

为什么用它? why

在项目的哪一个地方用过? 使用的具体步骤?使用过程中问题总结？ how

## 1.13面试自我介绍之大套路

