# 线程池

1.为什么要创建线程池？

答：创建线程和消耗线程需要消耗大量的系统资源，再加上业务处理时长，会大大的降低工作效率，使用线程池可将创建好一定数量的线程放入线程池中，在需要时从线程池中取出。当线程任务执行完毕则放回线程池。

同时方便对线程进行管理，如果当前请求数量超过已有线程执行范围，比如说程序启动时创建了100个线程，现在有101个请求进来，每个请求分配一个线程去执行，那么线程就会不够用，这时线程池可以管理成多出来的一个请求就会进入排队等候，避免无休止的创建线程导致系统崩溃。

2.线程池的种类

1.newSingleThreadExecutor

创建一个单线化的线程池，他只会用唯一的工作线程来执行任务，保证所有的任务按照指定的顺序来执行。

2，newFixedThreadEXecutor

创建指定数量级线程的线程池，可控制线程的最大并发数，超出的线程会进入排队等候。

3.newCachedThreadExecutor

创建一个可缓存的线程池

如果线程池线程数量超出处理需要，则会回收空闲线程，若无可回收的，则新创建线程池。

4.newScheduleThreadExecutor

创建定时线程池，支持定时任务周期性的执行

定时任务使用ThreadPoolExecutor创建而不是Executor创建

cacheThreadPool.execute(new Runnable

{

@override

public void run(){

}

}

)

3.扩展。。。

## Mysql和Oracle的区别

Oracle是完全支持事务的而Mysql支持innodb储存引擎行级锁情况下支持事务

# 缓存击穿 缓存穿透 缓存雪崩区别

缓存穿透：指和查询一个一定不存在的数据，缓存和数据库都没有，可能是黑户恶意发送大量请求导致DB的承受压力过大

解决：通常我们会在接口层加校验，不合法的参数进行校验，用户鉴权，不合法的参数直接return，或者在数据库查询不到的结果时，给缓存加null值

缓存雪崩：指设置缓存失效时间相同，导致大量请求直接打入DB,导致DB崩溃。

解决：在设置缓存失效时间上加上随机事件，比如1-5分钟。

缓存击穿：对于一些热点key会有大量的请求涌入，如果这时key值失效，那么请求会全部打入DB。

解决：加入分布式锁，热点key永远不会失效，或者对线程请求加锁，保证获取到锁的线程才能操作db。

# 单例模式

什么是单例模式？

单一的类只负责创建自己的对象，同时确保单个的对象被创建，该类提供了唯一访问对象的方式，可以直接访问，不需要实例化

特点：单例类只有一个实例

必须创建自己的对象

给其他对象提供这一实例

# HashMap的实现原理

hashMap的底层数据结构是一个链表加数组的一个数组结构，首先根据key值得hashCode重新计算Hash值，根据hash值得到数组中元素的下标位置，如果数组已存在元素，则将新添加的元素放入链头，存在的元素在链尾。

jdk1.8之后对HashMap的是先做出了优化，当链表中的加点数据超过8个，该链表转为红黑树来提高查询效率。

# ArrayList和LinkedList

ArryList底层的数据结构是动态数组，

LinkedList底层数据结构为双向链表

在随机访问元素的效率上，ArrayList的访问效率会更高。而LinkedList在访问数据时需要移动指针，效率比ArrayList低，而在添加删除的

#### Cookie和session的区别

session是存在于服务端的保存用户登录数据对象，在第一次请求时服务器自动生成一个session，当第二次请求时。客户端将发送cookie（文本信息）携带sessionId去请求服务，判断当前sessionId是否和保存的sessionIdz一致，找到对应用户的session

# 如何防止SQL注入

前端可能通过参数传递字符串在sql执行时满足条件完成请求，这时需要使用占位符#{}使其成为参数而不是通过字符串拼接去执行sql

# Redis集群的主从复制

当slave节点启动后，主动向master节点发送sync命令，master节点接受道命令后在后台保存快照（即rdb持久化），然后将快照文件和命令发送给从节点，从节点接受到命令，加载快照文件和缓存命令，master每次接受写的命令会同步发送给从节点，保证数据的一致性。

# Spring运行原理

在spring初始化时，将javaBean注册到IOC容器内，并被Spring去加载，根据配置内容动态加载管理对象。注入方式例如构造器创建、注解注入、setter

容器根据配置文件去创建实例和管理各个实例之间的依赖关系，对象与对象之间松散耦合，也利于功能的复用

SpringMvc

SpringMvc是一个基于Mvc模式的web开发框架，用户发送请求给前端控制器dispatcherServlet,然后经过Controller执行处理请求，返回ModelandView交给前端视图解析器去解析。

# 定时任务

1.Timer类

timer.schdule(new TimerTask(){

1. @Override
2. public void run() {
3. System.out.println("=========================");
4. }

})

重写run方法

2.@shcheduled 注解 spring提供的 注解内注入corn表达式

# Memcached和Redis的区别

Memcached储存的数据类型单一

为什么Redis单线程模型也能这么高的效率

1.纯内存操作

2.核心是基于IO非阻塞多路复用的网络模型

3.单线程避免了请求上下文的切换和竞争条件。

4.采用队列模式将并发访问变为串行访问。

# Zookeeper

什么是zookeeper？

zookeeper是一个开放源码的分布式协调服务，他是集群管理者，监视者集群各个节点的状态，根据节点提交的反馈进行下一步合理操作。

zookeeper可以实现注入数据发布订阅/ 分布式协调通知，集群管理，Master选举，分布式锁等功能。

客户端的请求可以被集群中的任何一台机器处理

# Servlet生命周期

Servlet的生命周期是什么？

Servlet运行在Servlet容器中，通过init（），service（），destory（）来表示。

包含了四个阶段。

1.加载和实例化。

2.初始化。

3.请求处理。

4.服务终止。

# Nginx转发请求

原路径： http://source.server.com/callback/test/test?username=xx

server{

listen：80；

server name：source.server.com

匹配callback

location /callback/ { 、

设置代理路径

proxy\_pass http:// http://10.1.9.1:8088

}

，默认其他路径转发

location/ {

proxy pass http://10.2.2.1:8080;

# SQL优化

1.在表中建立索引。

2.在查询语句中加入限制条件，wher，groupby

3.避免使用 in notin or like 会导致数据库放弃索引而进行全表扫描。

4.避免使用\*来查询全部字段

5.尽量不要在where条件内进行聚合函数操作，会导致索引失效。而进行全文索引

# Iterator怎么使用

是迭代器，使用Iterator（）返回一个Iterator对象，使用next()获取序列中的下一个对象，hasnext检查序列中的元素是否存在，remove删除元素

Iterator和ListIterator有什么区别

ListIterator专门用来遍历list，ListIterator实现了Iterrator接口

# 重写和重载的区别

重写是子类继承父类重写父类方法，其方法名参数类型，返回值类型均保持一致，

重载是指方法名相同，参数和返回值可以不同。

# SpringBoot和SpringMvc

SpringMvc是一个基于Servlet的Mvc框架。

Springboot为了简化Spring应用的创建所需的大量配置，实现了自动配置，是一个承载者，他集成了大量常用的第三方库配置（jsckson，redis，jdbc等），实现了开箱即用。基于spring4条件整合的一套快速开发包。

# Array和ArrayList

Array可以容纳基本数据类型和对象，而ArrayList只能容纳对象。

Array长度指定，erArrayList长度固定

# 线程有哪些状态

线程通常有五种状态，创建、就绪、运行、休眠（阻塞）、销毁（死亡）

如何保证线程的安全；

1.原子性：提供

# Docker创建容器的命令

Docker run -itd --name mysql-test -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=123456 mysql

**-p 3306:3306** ：宿主机的3306端口映射

docker run -it -d -p 127.0.0.1:5000:5000 docker.io/centos:latest /bin/bash

### ①IP:HOSTPORT:CONTAINERPORT：指定ip、指定宿主机port、指定容器port

将容器5000端口映射到宿主机5000端口，随后访问宿主http://IP:HOSTPORT

# 抽象类和普通类有什么区别

普通类不包含抽象方法，抽象类可以包含抽象方法

抽象类不能直接实例化，而普通类可以直接实例化。

# 抽象类和接口有什么区别

抽象类只能够被子类继承，而接口只能用来实现

抽象类可以有构造函数，而接口不能有

接口中的方法都是默认的public修饰，抽象类中的方法可以是任意的修饰符

# 创建线程的几种方式

1.继承Thread类创建线程

定义Thread的子类继承Thread类，并重写run方法，run方法的执行体是线程需要完成的任务。通过调用start（）方法来启动该线程。

2.通过实现Runnable接口来创建线程

实现Runnable接口，重写run方法。

3.通过Callable接口创建接口

创建callable的实现类，并实现call（）方法，call（）方法作为线程执行体，该方法有返回值。

# 线程run（）和start（）有什么区别

每个线程都是通过特定的Thread对象中的run（）方法来完成操作的，方法run（）成为方法体，通过调用Thread对象的start（）来完成线程的启动。

start()方法来启动，此时线程处于就绪状态，并没有运行。

线程状态：就绪 运行 阻塞 销毁

线程池中的submit（）和execute（）有什么区别

submit（）方法执行有返回值而execute没有

submit（）处理异常方便

# java程序内如何保证多线程的安全

1.原子性：互斥性，同一时间只能保证一个线程对数据进行操作。

2.可见性：一个线程对主内存的修改可以及时的被其他内存看到

# 什么是死锁？

只两个或两个以上的线程同时持有对方所需要的的资源，而造成线程处于等待状态，无法继续执行的状态。在线程占有资源后直到他调用wait()方法或者执行完代码块,在此期间资源无法释放则会造成死锁。

# Sleep（）和wait（）

sleep方法属于Thread类中方法，他会使线程处于休眠状态，在此期间线程不会释放资源，休眠时间到了会请求完资源并释放。

wait（）；属于object方法，一旦一个对象调用了wait（），则需要使用notify（）方法调用唤醒线程，在调用wait（）方法，该线程会释放他所持有的所有同步资源。进入对象锁池等待获取对象锁。

# ****http 响应码 301 和 302 代表的是什么****

代表请求url地址发生转移 301代表永久性转移，302代表暂时性转移

# get请求和post请求有哪些区别

get请求默认参数传递暴露在url上，post请求放在requestBody上

get'请求仅支持url编码，而post请求支持多种编码方式

# 如何实现跨域

1.什么是跨域？

只当前页面和被请求的url地址不一致时即为跨域；

当一个请求的url的协议，ip，端口号之间的任意一个与当前url不同时即为跨域。

阻止一个域的script脚本和另外一个域的内容进行交互。

如何解决跨域问题：

使用过jsonp，网页通过添加script标签，将请求到的json数据放到指定名字的回调函数的参数位置

①原生实现：

<script src="http://test.com/data.php?callback=dosomething"></script>

// 向服务器test.com发出请求，该请求的查询字符串有一个callback参数，用来指定回调函数的名字

// 处理服务器返回回调函数的数据

<script type="text/javascript">

function dosomething(res){

// 处理获得的数据

console.log(res.data)

}

</script>

② jQuery ajax：

$.ajax({

url: 'http://www.test.com:8080/login',

type: 'get',

dataType: 'jsonp', // 请求方式为jsonp

jsonpCallback: "handleCallback", // 自定义回调函数名

data: {}

});

# 创建对象的几种方式

1.new关键字

2.反射

3.clone（）；

4.反序列化