面试笔记整理

基础JAVAEE:

java有哪些集合,他们的特点是什么?

List (有序可重复)

arrayList : 基于动态数组的数据结构(有序,可重复,关注的时索引,查询速度快,默认长度10 ,扩容的长度为当前长度的1.5倍)

linkedList : 基于链表的数据结构(增删操作较多)

Set（无序,不可重复)

HashSet(不能存储重复的元素,存储顺序和取出的顺序不一定一致Set集合没有索引,默认长度16 ,当长度的负载因子达到0.75 会自动扩容,扩容的长度为当前长度*2)

Map (无序,不可重复 ,查询快, 非同步的线程是不安全的 可以存null值)

HashMap : 底层数组加链表 存储的格式是键值对 它根据键的HashCode值存储数据 (JDK1.8 以后当链表长度大于8时,会转成红黑数,默认长度为16 ,当负载因子达到0.75,就会自动扩容,扩容的长度为当前长度*2 ,构造一个空的 HashMap 在把原来的HashMap 存入新的HashMap当中, 所以他存储的位置可能会有变化)

能否让HashMap同步?

HashMap可以通过下面的语句进行同步:

Map m = Collections.synchronizeMap(hashMap);

Hashtable (不允许存null值,同步的,所以线程是安全的)

HashSet和HashMap的区别

HashMap	*HashSet*
HashMap实现了Map接口	HashSet实现了Set接口
HashMap储存键值对	HashSet仅仅存储对象
使用put()方法将元素放入map中	使用add()方法将元素放入set中
HashMap中使用键对象来计算ha	HashSet使用成员对象来计算hashcode值，对于两个对象来说hash
HashMap比较快，因为是使用唯	HashSet较HashMap来说比较慢

String、StringBuffer 、StringBuilder 的理解

String : 底层是 char 数组, 数组被final修饰过的,所以是不可变的, 长度是不可变的.

StringBuilder : 底层是char数组, 没有被final修饰过, 所以长度是可以改变的,默认长度为16 (线程不安全的)

StringBuffer: 底层是char数组,初始长度为16 , 方法被synchronized 修饰过, (线程安全)

抽象类(abstract class)和接口(interface)有什么异同?

成员区别: 抽象类成员可以是变量或常亮,拥有构造方法,可以有抽象方法和非抽象方法； 接口比抽象类更抽象接口的成员变量只能是常亮,没有构造方法,都是抽象方法.

关系区别: 抽象类子类使用extends关键字来继承,只能单继承或者多重继承;接口使用implements来实现,可以多实现

设计理念: 抽象类的方法可以由public,protected和defalut修饰符; 而接口默认修饰符只有public;

---

final、finally、finalize的区别?

final : 用于声明属性,方法和类， 分别表示属性不可被更改 , 方法不可覆盖, 被修饰的类不可被继承

finally : 异常处理语句结构的一部分,表示总是执行. (一般用来释放资源)

finalize : Object 类的一个方法, 在垃圾收集器执行的时候回调用被回收对象的此方法,可以覆盖此方法提供垃圾收集时的其他资源回收,例如关闭文件等.

---

object中的方法

clone() :创建并返回此对象的一个副本(克隆)

equals():判断是否相等
finalize():垃圾回收器方法
getClass():返回类对象
hashCode():返回哈希码值
notify():唤醒在此对象等待的单个线程
notifyAll():唤醒在此对象等待的所有线程
toString():返回的对象用字符串来表示
wait():是线程等待

请叙述一下您对线程池的理解?

降低资源损耗;通过重复利用已创建的线程降低线程创建和销毁造成的消耗;
提高响应速度;当任务到达时,任务可以不需要等到线程的创建就能立即执行;
提高线程的可管理性;线程是稀缺资源,如果无限制的创建,不仅会消耗系统资源,还会降低系统的稳定性,使用线程池可以进行统一的分配,调优和监控.

线程的可用状态及生命周期

1.就绪(Runnable):线程准备运行,不一定马上开始运行;
2.运行中(Running):进程正在执行线程的代码;
3.等待中(Waiting):线程处于阻塞的状态,等待外部的处理结束;
4.睡眠中(Sleep):线程被强制睡眠;
5.I/O阻塞(BlockedonI/O)：等待I/O操作完成。
6.同步阻塞(BlockedonSynchronization)：等待获取锁。
7.死亡(Dead)：线程完成了执行。

什么是锁?锁的作用是什么?

锁就是对象,锁的作用是保证线程的同步,解决线程安全问题.
持有锁的现在可以在同步中执行,没有锁的线程即使获得CPU的执行权也进不去.

什么情况下导致线程死锁，遇到线程死锁该怎么解决?

指多个线程因为争抢同一个资源而互相等待,若无外力作用,这些进程都将无法向前推进.

如何避免死锁:
加锁顺序(线程按照一定的顺序加锁)
加锁限时(线程尝试获取锁的时候加上一定的时限,超过时限则放弃对该锁的请求,并释放自己占有的锁)

说说你对Java中反射的理解

反射就是在运行状态当中,对于任意的一个类,能够获取这个类的所有属性和方法,对于任何一个对象,都能调用它的任意一个方法,这种动态获取的信息及动态调用对象方法的功能称为反射;
反射机制的优点就是可以实现动态的创建对象和编译,体现出很大的灵活性,缺点是对性能有影响.
获取字节码有三种方法: 1. 根据类名: 类名.class ; 2.根据对象: 对象.class , 3.根据全限定名 : Class.forName(全限定类名)

动态代理的区别和,什么场景使用?

静态代理通常只是代理一个类, 而动态代理是代理一个接口下的多个实现类;(注意代理的接口,也就是你的业务类必须要实现接口)
静态代理实现知道自己代理的是什么,而动态代理不知道要代理的是什么东西,只有运行之后知道;

Java中的设计模式?

创建型模式（5种）：工厂方法模式，抽象工厂模式，单例模式，建造者模式，原型模式。
结构型模式（7种）：适配器模式，装饰器模式，代理模式，外观模式，桥接模式，组合模式，享元模式。
行为型模式（11种）：策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式。

单例设计模式的一般定义: 一个类中只允许有一个实例;
实现思路： 让类的构造方法私有化, 同时提供一个静态方法去实例化这个类;
懒汉式: 当你是使用这个对象的时候它才会创建这个对象,时间换空间(线程不安全的)
饿汉式: 当这个类初始化的时候就会创建好这个对象,空间换时间 (线程是安全的,推荐使用)

简单工厂设计模式的一般定义: 简单工厂又称为静态工厂,由工厂对象决定创建哪一个产品对象.
实现思路: 写一个,让他制造出我们想要的对象.

适配器设计模式的一般定义: 某类继承这个适配器,从而实现我们需要实现的方法;
实现思路:通过写一个适配器类,里面写了所有的抽象方法,但这些方法是空的,并且适配器类要定义成抽象的,如果适配器类可以自己实现就没有意义了,适配器的作用,继承适配器,简化操作.

模板设计模式的一般定义:定义一个算法骨架将具体实现交给子类去实现,
实现思路:在类中定义一个抽象方法,距离实现交给子类去实现;

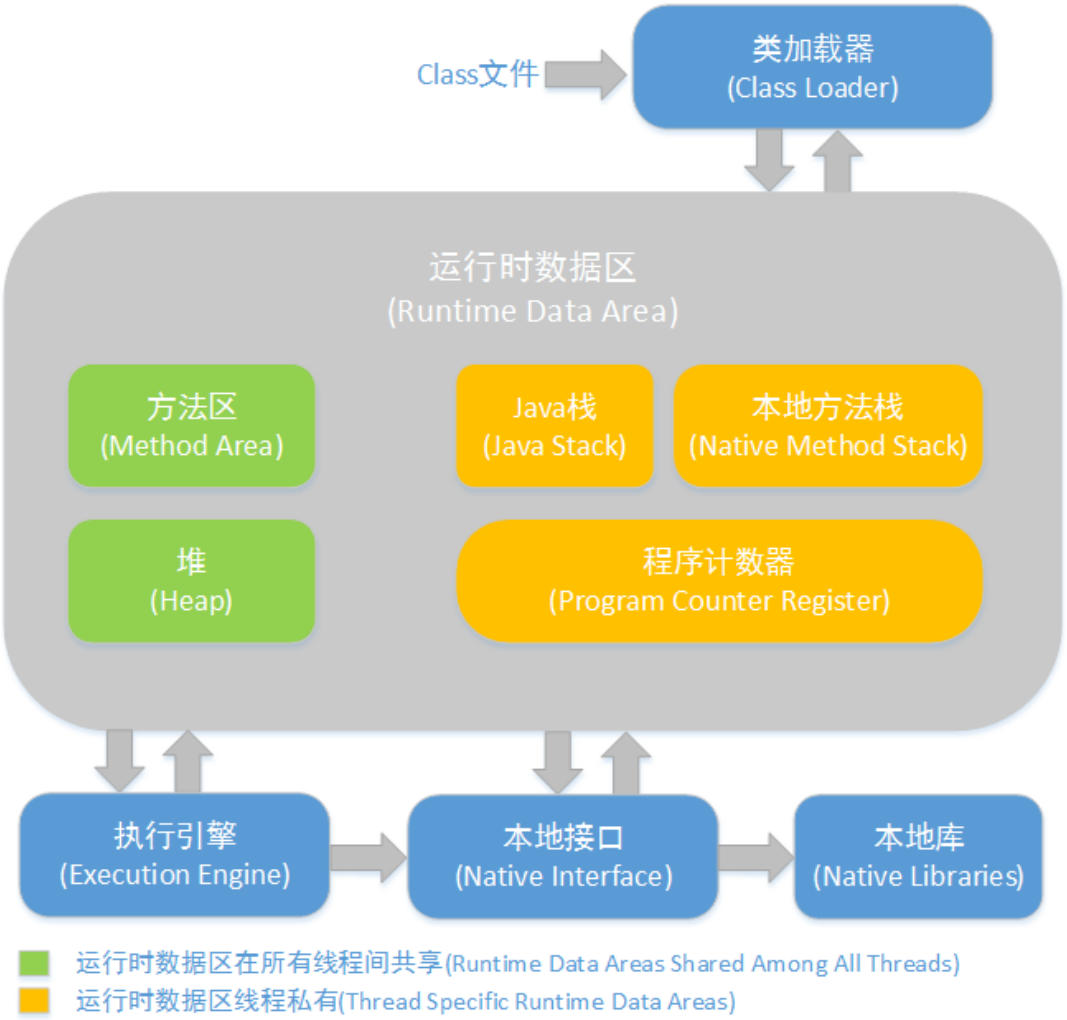
装饰者设计模式的一般定义:就是给一个对象增加一些新功能,而且是动态的;
实现思路: 要求装饰对象和被装饰对象实现同一个接口，装饰对象持有被装饰对象的实例;

JDK1.8 十大新特性详解

① 允许给接口添加一个非抽象的方法实现，只需要使用default关键字即可，这个特征又叫做扩展方法.
② Lambda 表达式: Collections.sort(names, (a, b) -> b.compareTo(a));
③ 函数式接口: 指仅仅只包含一个抽象方法的接口，每一个该类型的lambda表达式都会被匹配到这个抽象方法
④方法与构造函数引用: 允许你使用 :: 关键字来传递方法或者构造函数引用
⑤Lambda 作用域: 可以直接访问标记了final的外层局部变量，或者实例的字段以

及静态变量。
⑥访问局部变量, 访问对象字段与静态变量
⑦访问接口的默认方法: 如Stream 接口表示能应用在一组元素上一次执行的操作序列.
⑧Date API: Clock类提供了访问当前日期和时间的方法，Clock是时区敏感的，可以用来取代 System.currentTimeMillis() 来获取当前的微秒数。
⑨Annotation 注解: 支持多重注解, 只需要给该注解标注一下@Repeatable即可。
⑩Java 8引入了一个新的Optional类: 这是一个可以为 null 的容器对象。如果值存在则 isPresent() 方法会返回 true ， 调用 get() 方法会返回该对象。

JVM内存区域划分?



1. 本地方法栈 : 执行虚拟机所使用到的本地方法服务
2.java堆 : 存放对象实例和数组的区域
3.方法区: 存储已经被虚拟机加载的类信息,常亮,静态变量等区域

一个类的加载过程:

1.加载:启动类加载,负责从启动类路径中加载类;
2.链接: 2.1校验: 校验生成的字节码是否正确, 2.2 准备:分配内存并初始化默认值给所有的静态变量, 2.3 解析: 所有符号内存引用被方法区的原始引用所替代.
3.初始化:静态代码块被执行.

对象创建的过程:

说下原生jdbc操作数据库流程?

第一步：Class.forName()加载数据库连接驱动；

第二步：DriverManager.getConnection()获取数据连接对象;

第三步：根据SQL获取sql会话对象，有2种方式 Statement、PreparedStatement;

第四步：执行SQL处理结果集，执行SQL前如果有参数值就设置参数值setXXX();

第五步：关闭结果集、关闭会话、关闭连接

GET和POST的区别?

① GET请求的数据会附在URL之后（就是把数据放置在HTTP协议头中），以?分割URL和传输数据，参数之间以&相连.

② GET方式提交的数据最多只能是1024字节，理论上POST没有限制，可传较大量的数据。

③POST的安全性要比GET的安全性高。通过GET提交数据，用户名和密码将明文出现在 URL 上，因为(1)登录页面有可能被浏览器缓存，(2)其他人查看浏览器的历史纪录，那么别人就可以拿到你的账号和密码了.

---

session共享怎么做的（分布式如何实现session共享）?

利用Redis做session共享；

方案是重写服务器中的HttpSession和HttpservletRequest, 首先实现HttpSession接口,重写session所有的方法,将session以hash值得方式存在Redis中,

一个session的key就是sessionID, setAtrribute重写之后就是更新Redis中的数据,getAttribute重写之后就是Redis中的数据,等等需要将 HttpSession 的接口一一实现。

什么是jsp，什么是Servlet? jsp和Servlet有什么区别?

jsp本质就是事业servlet,每个jsp页面都是一个servlet 的实例

Servlet是由java提供用于开发web服务器应用程序的一个组件,用来生成动态内容

jsp和Servlet有什么区别： 1.jsp是html页面中内嵌的java代码,侧重页面显示　2. Servlet是html代码和java代码分离,侧重逻辑控制,MVC设计思想中JSP位于视图层,servlet位于控制层

---

jsp有哪些域对象?

（1）pageContext,在当前jsp页面有效，跳到其它页面失效

（2）request,指一次请求范围内有效，从http请求到服务器处理结束，返回响应的整个过程。在这个过程中使用forward（请求转发）方式跳转多个jsp，在这些页

面里你都可以使用这个变量

（3）session,指当前会话有效范围，浏览器从打开到关闭过程中，转发、重定向均可以使用

（4）application context域-指只能在同一个web中使用，服务器未关闭或者重启，数据就有效

谈谈你对ajax的认识?

Ajax 是一种创建交互式网页应用的网页开发技术,通过异步模式,提升了用户体验,优化了浏览器和服务器之间传输,减少不必要的数据往返,减少了宽带的占用.

最大的特点是可以实现局部刷新,在不更新整个页面的前提下维护数据.

常用的Linux命令?

列出文件列表：ls 【参数 -a -l】

创建目录和移除目录：mkdir rmdir

创建文件: touch 文件名称

打包并压缩：tar -zcvf

解压压缩包: tar -xvf

查找字符串：grep

显示当前所在目录：pwd

创建空文件：touch

编辑器：vim vi

删除:rm -rf

修改: mv 目录名称 新目录名称

动态打印日志信息：tail –f 日志文件

Mysql性能优化?

1.当只要一行数据的时候实例 limit 1

2.选择正确的数据库引擎, MyISAM 适用于一些大量查询的应用, InnoDB的写操作比较优秀

3.用not exists代替not in
4.充分使用索引, B-TREE 仍然是最高效的索引之中的一个
5.用 NOSQL 的方式使用 MYSQL

内连接与外连接的区别?

内连接 : 也被称为自然连接,只有两个表相同的数据才会出现在结果集中.
外链接不仅包含符合连接条件的行，1.左外链接(左边的表不加限制） 2.右外链接(右边的表不加限制）3. 全外链接(左右两表都不加限制)

mysql行转列?

mysql中行转列是通过group_concat()函数来实现的。默认是使用逗号隔开.

事务的四大特性是什么?

1.原子性: 整个事务当中所有的操作,要么全部成功,要么全部失败.
2.一致性:在事务开始之前和事务结束之后,数据库的信息一定是正确的.
3.隔离性: 一个事务的成功或失败对于其他事务是没有任何影响的,2个事务之间是互相独立的.
4持久性: 在事务完成以后,这个事务所对数据库的操作会永久保存在数据库当中,不会被回滚.

四种隔离级别?

1.读未提交（read uncommitted）:也就是脏读,事务可以读取其他事务未提交的数据
2.读已提交（read committed）:一个事务读取到另一个事务已提交的数据(解决了脏读问题.oracle默认)
3.可重复读(repeatable read）:在一个事务中读取到的数据始终保持一致,无论另一个事务是否提交(解决脏读、不可重复读,mysql默认)
4.可串行化（serializable）:同时只能执行一个事务,相当于事务中的单线程

在千万级的数据库查询中，如何提高效率?

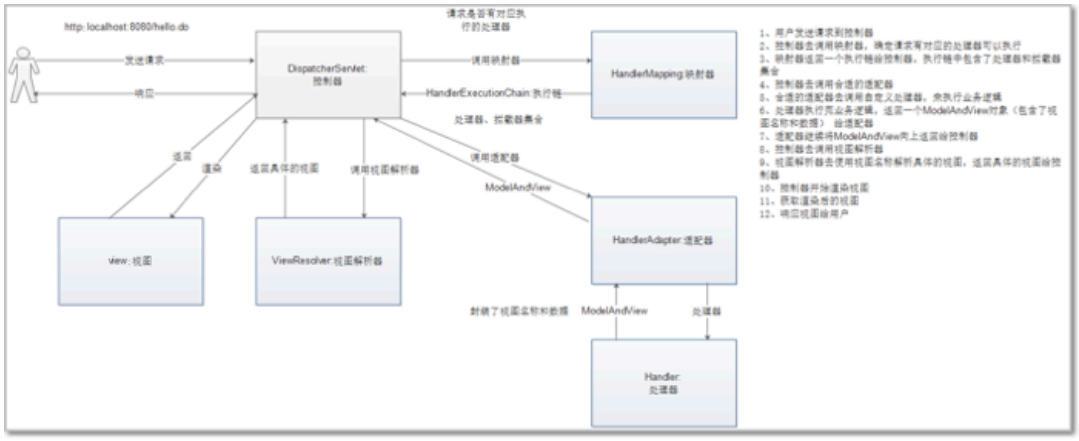
1.数据库设计方面
a.对查询进行优化,尽量避免全表扫描
b.应尽量避免在where子句中对字段尽量null判断
c.索引并不是越多越好,索引固然可以提高select的效率,但同时也降低了insert及update操作
d.尽量使用数字型字段,是因为引擎在处理查询和连接时会逐个比较字符串中每个字符,而对于数字一次就可以了
e.避免频繁创建和删除临时表,以减少系统表资源的消耗

2.语句方面
f.应尽量避免在where子句中使用!= 或<.> 操作符,或者是or来连接条件,否则将引擎放弃使用索引而使用全表扫描
g.任何地方都不要使用 select * from T , 用具体的字段来代替(*), 不要返回用不到的字段
h.用not exists代替not in

3.java方面
i.合理利用内存,有的数据要缓存.

SpringMVC的工作原理

1.用户发起请求找到DispatchServlet(控制器)
2.DispatchServlet对URL进行解析,得到请求资源标识符(URL)然后根据URL调用HandlerMapping将请求映射到处理器HandlerExcutionChain
3.DispatchServlet根据获得Handler选择一个具体的HandlerAdapter适配处理器
4.Handler对数据处理完成返回一个ModelAndView（）对象给DisPatchServlet(控制器)
5.Handler返回的ModelAndView（）只是逻辑视图并不是具体的视图,DispatcherSevlet通过ViewResolver视图解析器将逻辑视图转换成真正的视图View
6.DispatcherServle通过model 解析出 ModelAndView()中的参数进行解析最终返回一个具体的view 并返回给客户端



SpringMVC常用注解都有哪些?

@requestMapping : 用于请求URL 路径

@requestBody : 接收http请求的json格式,将json格式转换成java对象

@responseBody : 将controller方法返回的对象转换成json 响应给客户端

springMvc的优化?

1.controller尽量使用单例,可以减少创建对象和回收对象的开销,

2.处理request的方法形参上加@RequestParam注解,可以避免springmvc使用asm框架读取class文件获取方法参数名的过程

谈谈你对Spring的理解

Spring 是一个轻量级开源的JAVAEE 框架，它的核心是: IOC(控制反转)　DI(注入)　AOP(面向切面)

IOC(控制反转): 把创建对象的权利交给Spring , Spring容器使用工厂模式为我们创建了所需要的对象,直接调用就可以了 (线程安全的)

DI(注入) : Spring使用JavaBean对象的Set方法或者构造方法将其属性自动设置所需要的值;

AOP(面向切面) : 将一个对象横向抽成一个切面,对这个切面进行一些如权限控制,事务管理,记录日志等,底层是动态代理;

Spring的启动过程

1.创建一个全局的上下文环境,这个上下文就是ServletContext
2.在web容器启动时,会触发容器初始化事件,Spring 会创建一个上下文,这个上下文被称为根上下文,就是WebApplicationContext
3.监听器初始完毕后,开始初始化web.html配置中的Servlet

Spring中的几种设计模式?

1.单例模式:Spring的配置文件中设置bean默认为单例模式,Spring中有两种方式,若目标对象实现了若干接口,Spring使用JDK的类代理没有实现任何接口,Spring使用cglib 库生成目标类的子类
2.模板方法模式:用来解决代码重复问题.
3.前端控制器模式:Spring提供了前端控制器DispatherServlet来对请求进行分发
4.视图帮助(view) : spring提供了一系列的JSP标签,高效帮助将分散的代码整合在视图中
5.依赖注入:DI贯穿BeanFactory/ApplicationContext接口的核心理念
6.工厂模式: spring中使用beanFactory来创建对象实例.

bean的生命周期

单例对象: 生命周期
1.对象出生: 当应用加载,创建容器时,对象就被创建了
2.对象活着:只要容器在,对象就一直活着
3.对象死亡:当容器销毁时,对象就死亡了

多例对象:生命周期
1.对象出生:当使用对象时,创建新的对象实例(Bean)
2.对象活着:只要对象在使用中就一直活着
3.对象死亡:当对象长时间不使用,Java 的GC 就会自动回收

BeanFactory和FactoryBean的区别?

BeanFactory是IOC最基本的容器,负责生产和管理bean,它为其他具体的IOC容器提供了最基本的规范
FactoryBean是一个接口,在IOC容器中bean实现了FactoryBean之后,通过getBean(String BeanName)获取到Bean对象

spring的事务传播特性?

1.PROPAGATION_REQUIRED : 如果存在一个事务,则支持当前事务,如果没有事务则开启

2.PROPAGATION_SUPPORTS : 如果存在一个事务,则支持当前事务,如果没有事务,则非事务执行

3.PROPAGATION_MANDATORY : 如果存在一个事务,则支持当前事务,如果没有一个活动的事务,则抛出异常

4.PROPAGATION_REQUIRES_NEW : 总是开启一个新事务,如果一个事务已经存在,则将这个存在的事务挂起

5.PROPAGATION_NOT_SUPPORTED : 总是非事务执行,并挂起任何事务

6.PROPAGATION_NEVER : 总是非事务执行,如果存在一个事务则抛出异常

7.PROPAGATION_NESTED : 如果一个活动的事务存在,则运行在一个嵌套的事务中,如果没有活动事务,则按TransactionDefinition.PROPAGATION_REQUIRED执行

HTTP与HTTPS有什么区别?

HTTP : 是一个客户端和服务端请求和答应的标准

HTTPS : 是以安全为目的的HTTP通道,简单讲师HTTP的安全版

区别:

1: HTTPS协议需要到CA申请证书,一般免费的较少,因而要收取一定的费用

2:HTTP是超文本传输协议,信息是明文传输,HTTPS则是具有安全性的SSL加密传输协议

3:HTTP和HTTPS使用的是完全不同的连接方式,用的端口也不一样,HTTP： 80 HTTPS : 443

4.HTTP连接很简单,是无状态的，HTTPS协议是由SSL+HTTP协议构建的可进行加密传输,身份验证的网络协议,比HTTP协议安全

HTTP常见的状态码有哪些

200 OK //客户端请求成功

301 Moved Permanently（永久移除），请求的URL已移走。Response中应该包

含一个Location URL, 说明资源现在所处的位置
302 found 重定向
400 Bad Request //客户端请求语法错误,不能被服务器所理解
401 Unauthorized //请求未经授权,这个状态码必须和WWW-Authenticate报头域一起使用
403 Forbidden //服务器接受到请求,但是拒绝提供服务
404 Not Found //请求资源不存在，输入了错误URL
500 Internal Server Error //服务器内部错误
503 Server Unavailable // 服务器当前不能处理客户端请求,一段时间后可能恢复正常

Mybatis的编程步骤是什么样的?

1:创建SqlSessionFactory
2:通过SqlSessionFactory创建SqlSession
3:通过SqlSession执行数据库操作
4:调用session.commit() 提交事务
5.调用session.close() 关闭会话

Redis 有哪些数据结构

String 字符串
hash 键值对
list 链表
set 集合
zset 有序集合

Redis 持久化?

1. RDB 持久化 (默认) : 可以在指定的时间间隔内生成数据集的时间点快照,适用于进行备份,可以最大化Redis性能
2.AOF 持久化 ：记录服务器执行的所有写操作命令,并在服务器启动时,通过重新执行这些命令来还原数据集,使用AOF持久化会让Redis变的非常耐久,对于相同数量

的数据集而言,AOF文件通常是要大于RDB文件,AOF在运行效率上往往会慢于RDB
两者的区别:
RDB持久化是指在指定的时间间隔之内将内存中的数据集快照写入磁盘,实际操作过程是fork一个子程序,现将数据集写入临时文件.写入成功后,在替换之前的文件,用二进制压缩存储
AOF持久化以日志的形式记录服务器所处理的每一个写/删除操作,查询操作不会记录,以文本形式记录,可以打开文本查看详细的操作记录

StringBoot常用注解

@SpringBootApplication 扫描到Configuration类并把它加入到程序的上下文,启动类
@EnableAutoConfiguration 自动配置
@ComponetScan 组件扫描,可自动发现和装配一些Bean
@Autowierd 自动导入
@PathVariable 获取参数

谈谈你对elasticsearch的理解?

1.是一个基于Lucene的搜索服务器,它提供了一个分布式多功能的全文搜索引擎
2.天然的分布式,无需人工搭集群(solo就需要人工的搭集群,使用Zookeeper作为注册中心)
3.Restful风格,一切API都遵循Rest原则,容易上手
4.近实时搜索,数据更新在elasticsearch中几乎是完全同步的

谈谈你对SpringCloud的理解

SpringCloud是Spring为微服务架构思想做的一个一站式实现,提供了微服务开发所需要的配置管理/服务发现/断路器/只能路由/微代理/控制总线/全局锁/决策竞选/分布式会话和集群状态管理等组件,最重要的是,跟SpringBoot框架一起使用的话,会让你开发微服务架构的云服务非常方便,组件有:
- Eureka： 注册中心

- Zuul： 服务网关
- Ribbon： 负载均衡
- Feign ：服务调用(集成了负载)
- Hystix： 熔断器
- SpringCloudConfig： 它支持配置文件放在配置服务的内存中,也支持放在远程Git仓库里

Eureka的理解

- Eureka：就是服务注册中心(可以是一个集群),对外暴露自己的地址
- 提供者：启动后像Eureka注册自己的信息(地址,提供什么服务)
- 消费者：向Eureka订阅服务,Eureka会将对应服务的所有提供者地址列表发送给消费者,并定期更新
- 心跳(续约)：提供者定期通过HTTP方式向Eureka刷新自己的状态

JWT的理解

JWT,全称是Json Web Token ，是JSON风格轻量级的授权和身份认证规范,可以实现无状态,分布式的WEB应用授权

1.用户登录

2.服务的认证,通过后根据secret生成Token

3.将Token返回给用户

4.用户后续请求登录携带Token

5.服务端解读JWT签名,判断签名有效后,从Payload中获取用户信息

6.请求处理,返回响应结果