

2018java基础面试题（我自己用的，面试好多家公司基本会问到的问题和技术） - CSDN博客

原

2018java基础面试题（我自己用的，面试好多家公司基本会问到的问题和技术）

2018年03月01日 09:51:35 一念永恒 阅读数：12726 更多

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/u013305864/article/details/79383225>

1，webservice是什么？

webservice是一种跨编程语言和跨操作系统的远程调用技术,遵循SOPA/WSDL规范。

2，springCloud是什么？

springcloud是一个微服务框架，并提供全套分布式系统解决方案。支持配置管理，熔断机制，leader选举，服务治理，分布式session，微代理，控制总线，智能路由，一次性token。

Java中堆和栈有什么不同？

每个线程都有自己的栈内存，用于存储本地变量，方法参数和栈调用，一个线程中存储的变量对其它线程是不可见的。而堆是所有线程共享的一片公用内存区域。对象都在堆里创建，为了提升效率线程会从堆中弄一个缓存到自己的栈，如果多个线程使用该变量就可能引发问题，这时volatile 变量就可以发挥作用了，它要求线程从主存中读取变量的值。

堆：（对象）

引用类型的变量，其内存分配在堆上或者常量池（字符串常量、基本数据类型常量），需要通过new等方式来创建。

堆内存主要作用是存放运行时创建(new)的对象。

（主要用于存放对象，存取速度慢，可以运行时动态分配内存，生存期不需要提前确定）

栈：（基本数据类型变量、对象的引用变量）

基本数据类型的变量（int、short、long、byte、float、double、boolean、char等）以及对象的引用变量，其内存分配在栈上，变量出了作用域就会自动释放。

Spring的Scope有以下几种，通过@Scope注解来实现：

- (1) Singleton：一个Spring容器中只有一个Bean的实例，此为Spring的默认配置，全容器共享一个实例。
- (2) Prototype：每次调用新建一个Bean实例。
- (3) Request：Web项目中，给每一个 http request 新建一个Bean实例。
- (4) Session：Web项目中，给每一个 http session 新建一个Bean实例。

(5) GlobalSession：这个只在portal应用中有用，给每一个 global http session 新建一个Bean实例。

Spring事务传播行为

所谓事务的传播行为是指，如果在开始当前事务之前，一个事务上下文已经存在，此时有若干选项可以指定一个事务性方法的执行行为。在TransactionDefinition定义中包括了如下几个表示传播行为的常量：

- TransactionDefinition.PROPAGATION_REQUIRED：如果当前存在事务，则加入该事务；如果当前没有事务，则创建一个新的事务。这是默认值。
- TransactionDefinition.PROPAGATION_REQUIRES_NEW：创建一个新的事务，如果当前存在事务，则把当前事务挂起。
- TransactionDefinition.PROPAGATION_SUPPORTS：如果当前存在事务，则加入该事务；如果当前没有事务，则以非事务的方式继续运行。
- TransactionDefinition.PROPAGATION_NOT_SUPPORTED：以非事务方式运行，如果当前存在事务，则把当前事务挂起。
- TransactionDefinition.PROPAGATION_NEVER：以非事务方式运行，如果当前存在事务，则抛出异常。
- TransactionDefinition.PROPAGATION_MANDATORY：如果当前存在事务，则加入该事务；如果当前没有事务，则抛出异常。
- TransactionDefinition.PROPAGATION_NESTED：如果当前存在事务，则创建一个事务作为当前事务的嵌套事务来运行；如果当前没有事务，则该取值等价于TransactionDefinition.PROPAGATION_REQUIRED。

Spring的声明式事务管理力度是什么级别？

Struts2是类级别的，Spring是方法级别的

spring事务可以分为编程式事务和声明式事务

spring MVC与struts2的区别:

参考: <http://blog.csdn.net/chenleixing/article/details/44570681>

- 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
- 2、SpringMVC的方法之间基本上独立的，独享request response数据
- 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interceptor机制，SpringMVC用的是独立的AOP方式
- 5、SpringMVC的入口是servlet，而Struts2是filter
- 6、SpringMVC集成了Ajax
- 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
- 9、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

Spring框架中的核心思想包括什么？

主要思想是IOC控制反转，DI依赖注入，AOP面向切面

ArrayList和LinkedList的大致区别如下：

- 1.ArrayList是实现了基于动态数组的数据结构，LinkedList基于链表的数据结构。
- 2.对于随机访问get和set，ArrayList觉得优于LinkedList，因为LinkedList要移动指针。
- 3.对于新增和删除操作add和remove，LinedList比较占优势，因为ArrayList要移动数据。

ArrayList，Vector主要区别为以下几点：

- （1）：Vector是线程安全的，源码中有很多的synchronized可以看出，而ArrayList不是。导致Vector效率无法和ArrayList相比；
- （2）：ArrayList和Vector都采用线性连续存储空间，当存储空间不足的时候，ArrayList默认增加为原来的50%，Vector默认增加为原来的一倍；

HashSet与HashMap的区别：

HashMap	HashSet
实现了Map接口	实现Set接口

存储键值对	仅存储对象
调用put（）向map中添加元素	调用add（）方法向Set中添加元素
HashMap使用键（Key）计算Hashcode	HashSet使用成员对象来计算hashcode值， 对于两个对象来说hashcode可能相同， 所以equals()方法用来判断对象的相等性， 如果两个对象不同的话，那么返回false
HashMap相对于HashSet较快，因为它是使用唯一的键获取对象	HashSet较HashMap来说比较慢

HashMap和Hashtable的区别:

HashMap和Hashtable都实现了Map接口，但决定用哪一个之前先要弄清楚它们之间的分别。主要的区别有：线程安全性，同步(synchronization)，以及速度。

1. HashMap几乎可以等价于Hashtable，除了HashMap是非synchronized的，并可以接受null(HashMap可以接受为null的键值(key)和值(value)，而Hashtable则不行)。
2. HashMap是非synchronized，而Hashtable是synchronized，这意味着Hashtable是线程安全的，多个线程可以共享一个Hashtable；而如果没有正确的同步的话，多个线程是不能共享HashMap的。Java 5提供了ConcurrentHashMap，它是HashTable的替代，比HashTable的扩展性更好。
3. 另一个区别是HashMap的迭代器(Iterator)是fail-fast迭代器，而Hashtable的enumerator迭代器不是fail-fast的。所以当有其它线程改变了HashMap的结构（增加或者移除元素），将会抛出ConcurrentModificationException，但迭代器本身的remove()方法移除元素则不会抛出ConcurrentModificationException异常。但这并不是一个一定发生的行为，要看JVM。这条同样也是Enumeration和Iterator的区别。
4. 由于Hashtable是线程安全的也是synchronized，所以在单线程环境下它比HashMap要慢。如果你不需要同步，只需要单一线程，那么使用HashMap性能要好过Hashtable。
5. HashMap不能保证随着时间的推移Map中的元素次序是不变的。

线程安全是什么？线程不安全是什么？

线程安全就是多线程访问时，采用了**加锁机制**，当一个线程访问该类的某个数据时，进行保护，其他线程不能进行访问直到该线程读取完，其他线程才可使用。不会出现数据不一致或者数据污染。(Vector,HashTable)

线程不安全就是**不提供数据访问保护**，有可能出现多个线程先后更改数据造成所得到的数据是脏数据。（ArrayList, LinkedList, HashMap等）

线程和进程的区别？

进程和线程都是一个时间段的描述，是CPU工作时间段的描述，不过是颗粒大小不同；

- (1) 进程是资源的分配和调度的一个独立单元，而线程是CPU调度的基本单元
- (2) 同一个进程中可以包括多个线程，并且线程共享整个进程的资源（寄存器、堆栈、上下文），一个进程至少包括一个线程。
- (3) 进程的创建调用fork或者vfork，而线程的创建调用pthread_create，进程结束后它拥有的所有线程都将销毁，而线程的结束不会影响同个进程中的其他线程的结束
- (4) 线程是轻两级的进程，它的创建和销毁所需要的时间比进程小很多，所有操作系统中的执行功能都是创建线程去完成的
- (5) 线程中执行时一般都要进行同步和互斥，因为他们共享同一进程的所有资源
- (6) 线程有自己的私有属性TCB，线程id，寄存器、硬件上下文，而进程也有自己的私有属性进程控制块PCB，这些私有属性是不被共享的，用来标示一个进程或一个线程的标志

黑盒测试、灰盒测试、白盒测试、单元测试有什么区别？

黑盒测试关注程序的功能是否正确，面向实际用户；

白盒测试关注程序源代码的内部逻辑结构是否正确，面向编程人员；

灰盒测试是介于白盒测试与黑盒测试之间的一种测试。

单元测试(Unit Testing)是对软件基本组成单元进行的测试，如函数或是一个类的方法。这里的单元，就是软件设计的最小单位。

怎么对数据库百万级数据进行优化？

使用读写分离技术（

让主数据库（master）处理事务性增、改、删操作（INSERT、UPDATE、DELETE），而从数据库（slave）处理SELECT查询操作

）

Spring Bean的生命周期：

Bean的建立， 由BeanFactory读取Bean定义文件，并生成各个实例

Setter注入 执行Bean的属性依赖注入

BeanNameAware的setBeanName(), 如果实现该接口, 则执行其setBeanName方法

BeanFactoryAware的setBeanFactory(), 如果实现该接口, 则执行其setBeanFactory方法

BeanPostProcessor的processBeforeInitialization(), 如果有关联的processor, 则在Bean初始化之前都会执行这个实例的processBeforeInitialization()方法

BeanPostProcessor的processAfterInitialization(), 如果有关联的processor, 则在Bean初始化之前都会执行这个实例的processAfterInitialization()方法

InitializingBean的afterPropertiesSet(), 如果实现了该接口, 则执行其afterPropertiesSet()方法

Bean定义文件中定义init-method

BeanPostProcessors的processAfterInitialization(), 如果有关联的processor, 则在Bean初始化之前都会执行这个实例的processAfterInitialization()方法

DisposableBean的destroy(), 在容器关闭时, 如果Bean类实现了该接口, 则执行它的destroy()方法

Bean定义文件中定义destroy-method, 在容器关闭时, 可以在Bean定义文件中使用“destroy-method”定义的方法

简单回答springbean生命周期:

(1) 实例化 (必须的) 构造函数构造对象

(2) 装配 (可选的) 为属性赋值

(3) 回调 (可选的) (容器-控制类和组件-回调类)

(4) 初始化(init-method=" ")

(5) 就绪

(6) 销毁 (destroy-method=" ")

springmvc生命周期:

1A) 客户端发出http请求，只要请求形式符合web.xml文件中配置的*.action的话，就由DispatcherServlet来处理。

1B) DispatcherServlet再将http请求委托给映射器的对象来将http请求交给对应的Action来处理

2) 映射器根据客户的http请求，再对比<bean name="/hello.action如果匹配正确，再将http请求交给程序员写的Action

3) 执行Action中的业务方法，最终返回一个名叫ModelAndView的对象，其中封装了向视图发送的数据和视图的逻辑名

4) ModelAndView对象随着响应到DispatcherServlet中了

5) 这时DispatcherServlet收到了ModelAndView对象，它也不知道视图逻辑名是何意，又得委托一个名叫视图解析器的对象去具体解析ModelAndView对象中的内容

6) 将视图解析器解析后的内容，再次交由DispatcherServlet核心控制器，这时核心控制器再将请求转发到具体的

视图页面，取出数据，再显示给用户

servlet生命周期?

- Servlet 通过调用 **init ()** 方法进行初始化。
- Servlet 调用 **service()** 方法来处理客户端的请求。
- Servlet 通过调用 **destroy()** 方法终止（结束）。
- 最后，Servlet 是由 JVM 的垃圾回收器进行垃圾回收的

ajax怎么解决跨域?

参考：<http://blog.csdn.net/u014727260/article/details/72793459>

1, 代理（通过后台操作）

2, JSONP（添加响应头，允许跨域）

```
addHeader('Access-Control-Allow-Origin:'); //允许所有来源访问
```

```
addHeader('Access-Control-Allow-Method:POST,GET'); //允许访问的方式
```

3, 在ajax的dataType方式改为“jsonp”

Mysql数据类型：

- 1.普通索引
- 2.唯一索引
- 3.主键索引
- 4.组合索引
- 5.全文索引

参考：<https://www.cnblogs.com/luyucheng/p/6289714.html>

Eureka和zookeeper的区别？

- 1，做分布式下的服务发现还是使用eureka更好，也就是AP特性的分布式协调工具（zookeeper因为网络故障就无法返回可用的主机）
- 2，zookeeper技术更加成熟，资料更多
- 3，Eureka。是spring cloud之下一个专门负责微服务服务注册和发现的组件，Eureka就是为了服务发现而设计的
- 4，Zookeeper。是用来保证分布式一致性的一个软件。不是为了服务发现注册而设计的，只不过它的特性也可以被二次开发成服务发现注册中心罢了

SpringCloud都有哪些组件？

Spring Cloud为微服务架构开发涉及的配置管理，服务治理，熔断机制，智能路由，微代理，控制总线，一次性token，全局一致性锁，leader选举，分布式session，集群状态管理等操作提供了一种简单的开发方式。

组件列：

- **Spring Cloud Config**: 配置管理工具，支持使用Git存储配置内容，支持应用配置的外部化存储，支持各客户端配置信息刷新、加解密配置内容等。
- **Spring Cloud Bus**: 事件、消息总线，用于在集群（例如，配置变化事件）中传播状态变化，可与Spring Cloud Config联合实现热部署。
- **Spring Cloud Netflix**: 针对多种Netflix组件提供的开发工具包，其中包括Eureka、Hystrix、Zuul、Archaius等。
- **Netflix Eureka**: 一个基于rest服务的服务治理组件，包括服务注册中心、服务注册与服务发现机制的实现，实现了云端负载均衡和中间层服务器的故障转移。
- **Netflix Hystrix**: 容错管理工具，实现断路器模式，通过控制服务的节点,从而对延迟和故障提供更强大的容错能力。
- **Netflix Ribbon**: 客户端负载均衡的服务调用组件。
- **Netflix Feign**: 基于Ribbon和Hystrix的声明式服务调用组件。
- **Netflix Zuul**: 微服务网关，提供动态路由，访问过滤等服务。
- **Netflix Archaius**: 配置管理API，包含一系列配置管理API，提供动态类型化属性、线程安全配置操作、轮询框架、回调机制等功能。
- **Spring Cloud for Cloud Foundry**: 通过OAuth2协议绑定服务到CloudFoundry，CloudFoundry是VMware推出的开源PaaS云平台。
- **Spring Cloud Sleuth**: 日志收集工具包，封装了Dapper,Zipkin和HTrace操作。
- **Spring Cloud Data Flow**: 大数据操作工具，通过命令行方式操作数据流。
- **Spring Cloud Security**: 安全工具包，为你的应用程序添加安全控制，主要是指OAuth2。
- **Spring Cloud Consul**: 封装了Consul操作，consul是一个服务发现与配置工具，与Docker容器可以无缝集成。
- **Spring Cloud Zookeeper**: 操作Zookeeper的工具包，用于使用zookeeper方式的服务注册和发现。
- **Spring Cloud Stream**: 数据流操作开发包，封装了与Redis,Rabbit、Kafka等发送接收消息。
- **Spring Cloud CLI**: 基于 Spring Boot CLI，可以让你以命令行方式快速建立云组件

Hibernate的三种状态是什么？怎么将游离状态转换为持久化状态？

transient(瞬时状态), persistent(持久化状态)以及detached(离线状态)

转换: update() saveOrUpdate() lock()

