1. hibernate和mybatis的区别?

首先、Hibernate和Mybatis虽然都是对JDBC的封装，但两者

的设计理念不太相同，Hibernate侧重于ORM【对象关系映射】，所以

Hibernate将生成SQL语句，执行SQL语句，返回结果的工作全部都做了

而MyBatis的侧重点在于SQL语句的映射上，所以Mybatis【ibatis最初

被称作SQL映射框架】的重点在于我们业务中执行的SQL语句上。

然后、不同的理念导致不同的设计方案，不同的设计方案有不

同的使用场景，我们使用Hibernate主要看中他的"全自动"ORM,可以省

去很多JDBC的细节工作，但是这势必在效率上有些影响【尤其在查询效

率上，因为Hibernate需要将HQL或者QBC翻译成SQL】，而我们选中

MyBATIS主要看中他的SQL自定义、灵活的特点，【因为开发中可能会用

到很多复杂的业务运算SQL】，这样就可以根据自己的需求来设计比较优

化的SQL语句，但是这样也会导致MYBATIS必须在SQL开发工作量和数据

库移植性上做出让步。

2.spring中的事务用过没？

------------------------------------------------------------------------------------------

(1) (那你谈谈什么是事务，你的项目中哪里用到事务了)

------------------------------------------------------------------------------------------

在一次请求过程中，对多次数据库的连接操作进行统一的提交或回滚操作，以此保证数据执行的一致性 。 或者是对多个sql语句进行 统一的提交或回滚操作。

比如涉及到批量添加、删除的操作，或者银行转账这种一次请求涉及多次数据库连接修改的操作。

------------------------------------------------------------------------------------------

(2)SPRING有哪些事务？

------------------------------------------------------------------------------------------

spring提供了编程式事务支持 TransactionTemplate

也提供了声明式事务支持 底层采用AOP实现的事务通知bean

也有注解形式的 @Transactional

\*\*\* 还是问spring内置了哪些事务管理器？？？

Spring的事务管理器有5个，都实现了PlatformTransactionManager接口，如下所示：

DataSourceTransactionManager JDBC事务管理器

HibernateTransactionManager Hibernate事务管理器

JdoTransactionManager JDO事务管理器

JtaTransactionManager JTA事务管理器

PersistenceBrokerTransactionManager Apache的OJB事务管理器

------------------------------------------------------------------------------------------

(3)SPRING中怎么使用事务？

------------------------------------------------------------------------------------------

1)编程式事务 在DAO中通过TransactionTemplate模板类封装事务的管理过程

2)声明式事务：底层采用AOP思想实现，将事务的管理过程（包括事务对象的创建，提交和回滚操作）封装在一个事务通知bean中，然后通过配置AOP切入点的方式来描述需要将事务通知bean所提供的事务管理服务引用给哪些核心对象/方法。

配置方式依赖于AOP的配置

<tx:advice id="" transaction-manager="指向一个事务管理器">

<tx:attributes> 为不同的方式设置不同的事务属性

<tx:method name="表达式匹配方法" 四个属性...rollback-for="监听到什么异常回滚事务"/>

....

</tx:attributes>

</tx:advice>

<aop:config>

<aop:pointcut id="" expression="表达式指向需要引用事务的方法列表"/>

<aop:advisor advice-ref="" pointcut-ref=""/>

</aop:config>

声明式事务的特性 ： 不需要在核心代码中编写事务的管理过程，以此将核心代码与事务管理服务动态分离。

声明式事务回滚的方式 ： 运行时异常

------------------------------------------------------------------------------------------

3.使用数据库连接池的优点？

数据库连接池可以重复使用，避免了频繁创建

数据库连接池在初始化时就已经在连接池中创建连接用以备用，因此提高了响应

4.说出数据连接池的工作机制是什么？

当服务器启动时，初始化一些连接放在连接池中，等待客户端调用，

如果客户端请求连接，则首先判断连接池中是否有可用连接，

如果有则返回，并将该连接标记为使用中的状态，如果没有，

则判断当前连接数是否超过了最大可用连接数，如果当前连接数超过了

最大可用连接，则抛出无可用连接的异常，如果没有超过，

则创建新得连接并返回，当使用完连接之后，再将连接放回到连接池中，

从而实现连接的复用。

1. jquery的方法

each()：是jQuery中专用于迭代(遍历)数组的方法，参数为一个处理函数，this表示当前需要迭代的js对象

append()：追加到父元素之后

prepend()：追加到父元素之前

after()：追加到兄弟元素之后

before()：追加到兄弟元素之前

attr(name)：获取属性值

attr(name,value)：给符合条件的标签添加key-value属性对, 设置属性值

$("<div id='xxID'>HTML代码</div>")：创建元素，属性，文本

remove()：删除自已及其后代节点

val()：获取value属性的值

val("")：设置value属性值为""空串，相当于清空

text()：获取HTML或XML标签之间的值

text("")：设置HTML或XML标签之间的值为""空串

clone()：只复制样式，不复制行为

clone(true)：既复制样式，又复制行为

replaceWith()：替代原来的节点

removeAttr()：删除已存在的属性

addClass()：增加已存在的样式

removeClass()：删除已存在的样式

hasClass()：判断标签是否有指定的样式，true表示有样式，false表示无样式

toggleClass()：如果标签有样式就删除，否则增加样式

offset()：获取对象的left和top坐标

offset({top:100,left:200})：将对象直接定位到指定的left和top坐标

width()：获取对象的宽

width(300)：设置对象的宽

height()：获取对象的高

height(500)：设置对象的高

children()：只查询子节点，不含后代节点

next()：下一个兄弟节点

prev()：上一个兄弟节点

siblings()：上下所有兄弟节点

动画API:

show()：显示对象

hide()：隐藏对象

fadeIn()：淡入显示对象

fadeOut()：淡出隐藏对象

slideUp()：向上滑动

slideDown()：向下滑动

slideToggle()：上下切换滑动，速度快点

6.用过的前端技术easyui里面都有什么布局，有哪些组件？

布局:面板,选项卡,分类,布局

组件:菜单,按钮, 表单,验证框,下拉列表框,日期输入框,窗口,数据表格,树形表格,树

1. servlet的九个内置对象

JSP九种内置对象：request, response, session, application, page, pageContext, out, exception, config

(1)request对象

客户端的请求信息被封装在request对象中，通过它才能了解到客户的需求，然后做出响应。它是HttpServletRequest类的实例

(2)response对象

response对象包含了响应客户请求的有关信息，但在JSP中很少直接用到它。它是HttpServletResponse类的实例。

(3)session对象

session对象指的是客户端与服务器的一次会话，从客户连到服务器的一个WebApplication开始，直到客户端与服务器断开连接为止。它是HttpSession类的实例.

(4)out对象

out对象是JspWriter类的实例,是向客户端输出内容常用的对象

(5)page对象

page对象就是指向当前JSP页面本身，有点象类中的this指针，它是java.lang.Object类的实例

(6)application对象

application对象实现了用户间数据的共享，可存放全局变量。它开始于服务器的启动，直到服务器的关闭，在此期间，此对象将一直存在；这样在用户 的前后连接或不同用户之间的连接中，可以对此对象的同一属性进行操作；在任何地方对此对象属性的操作，都将影响到其他用户对此的访问。服务器的启动和关闭 决定了application对象的生命。它是ServletContext类的实例。

(7)exception对象

exception对象是一个异常对象，当一个页面在运行过程中发生了异常，就产生这个对象。如果一个JSP页面要应用此对象，就必须把isErrorPage设为true，否则无法编译。他实际上是java.lang.Throwable的对象

(8)pageContext对象

pageContext对象提供了对JSP页面内所有的对象及名字空间的访问，也就是说他可以访问到本页所在的SESSION，也可以取本页面所在的application的某一属性值，他相当于页面中所有功能的集大成者，它的本 类名也叫pageContext。

(9)config对象

config对象是在一个Servlet初始化时，JSP引擎向它传递信息用的，此信息包括Servlet初始化时所要用到的参数（通过属性名和属性值构成）以及服务器的有关信息（通过传递一个ServletContext对象）

8.Hibernate的缓存有哪些？

Hibernate缓存是为了提高查询效率[通过减少与数据库的交互频次]设计的

分为Session维护的一级缓存和SessionFactory维护的二级缓存[默认没用]

如果使用需要打开:Hibernate并没有实现商用级的二级缓存{HashMap}

Hibernate打包了很多开源的缓存实现,如:EhCache\OSCache

如果使用二级缓存; hibernate.cfg.xml

1.声明启用二级缓存

2.声明用哪个[缓存的供应商]

3.添加jar

4.添加缓存配置文件

1. 怎样提高hibernate的效率。

使用单例模式,用过的session要及时关闭

10.怎样创建索引?

create index 索引名 on 表名(列名,列名..);

1. 简述IOC和AOP

spring的两大核心是什么 / 简述spring的IOC和AOP

spring框架提供的两大核心是 : IOC和AOP

ioc和aop是spring框架提供的用于实现对象管理的两种技术。

IOC:

全称 : inversion of control?

直译为控制反转，其核心思想为依赖注入

ioc是一种用于描述对象创建 及 对象依赖形成的一种技术。

首先，ioc底层采用工厂模式实现，所以其首要目的-也就是工厂的目的-创建对象；但是ioc创建对象的方式和普通工厂不同，普通工厂是在工厂类中new对象；而ioc则是将需要其创建的对象配置在ioc容器中，使用工厂类解析ioc容器，然后以反射的方式创建对象。ioc的直译'控制反转'正是描述了ioc环境下对象创建的特点：将对象的创建权利从内部代码转移到外部容器，这个转移的过程即为反转。

其次，ioc的核心思想则是他的被动注入。我们使用ioc的方式创建对象的目的就是为了采用其提供的被动注入思想，以此形成对象之间的依赖关系。ioc为了实现被动注入提供了setter、构造器和方法注入。

AOP:

全称 : aspect oriented programming 面向切面编程思想

底层采用代理模式实现的一种技术

首先，AOP是OOP的一种延续思想，用于解决OOP开发过程中对象封装、继承、多态所造成的对象之间高度耦合性而提出的一种新的编程思想。

AOP通过分离组成软件的核心业务逻辑和系统级服务进行内聚性开发。其关注点在于系统级服务部分。

AOP的实现策略是将系统级服务封装成为一种AOP组件，采用XML配置及引用的方式将这个AOP组件所提供的系统级服务添加给需要该服务的核心业务逻辑。从而实现核心逻辑与通用服务之间的解耦。

事务的特性

事务四大特性(简称ACID)

原子性(Atomicity)：事务中的全部操作在数据库中是不可分割的，要么全部完成，要 么均不执行。

一致性(Consistency)： 事务必须是使数据库从一个一致性状态变到另一个一致性状态。 一致性与原子性是密切相关的。

隔离性(Isolation)：事务的执行不受其他事务的干扰。

持久性(Durability)：能够将提交的数据持久化到数据库中。

1. 对spring的理解

其实就是spring的优势 ： 轻量级的IOC和AOP容器框架

轻量级主要针对重量级的EJB而言，

它的非侵入性和对容器的低依赖性

(1)提供了IOC依赖注入管理开发中的业务对象，管理对象的生命周期,降低组件之间的耦合性，让我们更专注于应用逻辑

(2)提供了对AOP思想的支持，通过将通用服务从业务逻辑中分离出来，实现高内聚开发。应用对象只需要关注业务逻辑，不需要再负 责其他系统问题。

(3)对主流的框架提供了很好的集成支持，如Hibernate,Struts2,JPA等

(4)Spring的高度可开放性，并不强制依赖于Spring，开发者可以自由选择Spring部分或全部（分层理念）

14.Springmvc和Struts有什么区别？

首先spring-mvc和struts1.x/2.x都是以servlet为模型衍生出的技术

a 核心控制器的不同

对于所有的基于mvc模式的框架来说，核心控制器的主要用途在于拦截某一特定的请求，然后对这些请求进行统一的处理

spring-mvc的核心控制器是servlet(DispatcherServlet);struts2的核心控制器是filter (StrutsPrepareAndExecuteFilter)

b 请求处理器的不同

struts2是基于对象设计，每次发送的请求都会为其分配一个action实例，每个action都会被注入属性(表单项)

spring-mvc是基于方法设计，只提供一个实例，每次请求执行实例中的不同方法。

c 对annotation的支持

spring-mvc完全实现了零配置，也就是全注解方式管理；

struts2则需要采用xml配置来进行管理(虽然也提供了零配置概念，但很少有人会用到)

d 表单项封装

struts2提供了两种封装方式(属性驱动and模型驱动),但都是通过action的属性进行封装。

spring-mvc则是通过方法参数进行封装(当然也会用到第三方javaBean或者直接使用@RequestParam注解)

e 拦截器支持

struts2底层采用拦截器设计实现，几乎所有的服务都被封装在Interceptor中；虽然这种可插拔式的设计在使用上比较方便， 但xml中的配置量过于庞大(这还是在interceptor之间可以继承的前提下)。

spring-mvc的拦截器则是采用独立的aop支持，并且他的拦截器是方法级别的拦截，一个方法对应一个request上下文。

f ajax支持

struts2为了实现对ajax支持内置了json插件，通过返回结果或者拦截器实现对json数据的封装

spring-mvc则提供了@ResponseBody注解来实现对ajax支持

g servlet-api支持

struts2是类级别的拦截，一个类对应一个request上下文；虽然方法之间也是相互独立的，但其所有的变量都是共享的，虽然不 影响程序运行，但给我们的编码造成困扰。

spring-mvc属于方法级别的拦截，一个方法对应一个request上下文；因此spring-mvc的方法之间基本上是相互独立的，独享 request和response数据，方法之间不共享变量。

h 最主要的....

spring-mvc和spring之间是无缝的(spring-mvc最初是作为spring的一个层提出)。在基于spring管理的项目上来说，管理性 和安全性要比Struts2高

[总结 : 两个框架的设计目的都是为了处理HTTP请求,成为优秀的MVC框架；但设计思想是截然不同的...随便拿出一点就是他们之间 的区别...]

15.spring和mvc理解、

spring是一种开源的相对于EJB而言轻量级的ioc和aop容器框架

ioc控制反转/依赖注入,在spring中实现对象的创建和依赖的被动注入

aop面向切面编程思想，将业务逻辑和服务代码动态分离，提高代码的重用性

spring的核心在于bean工厂，用于构建我们想要的对象；在此基础上，通过引入aop支持，提供一些声明式事务、安全上的服务。

spring中也提供了对常用框架或技术的整合管理，包括dao层管理jdbc，orm层管理hibernate和mybatis，web层管理struts，mvc层提供了对表现层的另外一种支持

spring的mvc是相对于struts等表现层框架，由spring提供的一个用于处理http请求的mvc框架；提供了一个请求入口DispatcherServlet 以及两种请求解析器HandlerMapping和MultipartResolver 请求处理器的Controller 视图解析器的ViewResolver 和 视图对象 View

16.Jquery选择器、

基本的id class 元素选择器

层级选择器 > ~ +

基本选择器 odd even eq lt gt...

内容选择器 contains empty has parent..

属性选择器 attr

表单选择器 input text password submit ...

17.数据库索引原理

索引类似于目录，但它绝非目录，目录只是排序不能提高查询效率，

索引能够提高查询效率，索引查询数据类似于平衡树的形式：

首先从数据内容中间查找数据大体位置，如果比当前中间数据小则往前取四分之一位置，

否则往后取四分之一位置数据再次判断,然后以此类推..这就是一种二分法的形式也叫平衡树。

18.sql优化

(1) 可以使用索引

(2) SQL语句如果是多表关联可以将关联条件写在左侧,过滤条件尽量写在右侧

(3) 避免在子查询语句中使用in some any all等关键字

可以使用多表关联替代

(4) 避免使用like '%%' 写法,这种写法或造成全盘扫描

like 之后是 '字符%'

如果必须是 like '%字符%' 可以使用instr函数替代

where ename like '%E%';

where instr(ename,'E')>0;

(5) 复杂的业务功能可以使用存储过程(如果对存储过程熟悉的话回答)

19.list和map都是继承collection吗? list 和 set 的区别

首先List和Set是Collection接口的子接口，属于单值类型，List集合要求元素有序，不唯一，而Set集合要求元素无序,唯一 Map是键值对集合的父接口

20.如何实现多线程,实现线程的方式

1.extends Thread @Override public void run(){...}

2.implements Runnable @Override public void run(){...}

3.implements Callable<X> @Override public X call(){...}

\*:弥补了原本run()返回类型定义为void

方法签名后有throws Exception

1. servlet的生命周期

Servlet是一个单实例多线程的Java程序

构造方法（一次）===》init()方法（一次）===》service()(一个请求一次)

===》doGet()或者doPost()(多次)===》destory()(一次)

22.Stringbuilder和Stringbuffer和String的区别？

String不会操作原本的字符串，而是将符合要求的新字符串返回回来，因为String没预留缓存空间，而Stringbuilder 和Stringbuffer直接在原本的字符串进行操作，因为他们 预留了缓存空间，默认为16块

23.Struts流程

Struts2处理请求的流程:容器启动时，首先启动Struts2的核心过滤器(StrutsPrepareAndExecuteFilter)，来加载配置文件等待客户端请求，如果客户端发出请求，Struts2首先根据请求的名字来到struts.xml中寻找对应action【根据name属性来查找】，如果找到对应的，则创建class属性指定的类，并且根据method属性值调用对应的方法【如果没有method属性值，则调用execute方法】，然后再根据方法返回的字符串找到元素的子元素，根据其中的name 和type属性跳转到其他的action或者页面。

Action:不是单例

24.Oracle和MySQL的区别\*\*\*\*\*\*

> Oracle使用rownum来实现分页，MySQL使用limit实现分页

> Oracle没有主键自增特性,可以使用调用序列的方式来实现主键自增；

mySQL可以在主键列之后添加auto\_increment设置自增特性

> Oracle有表空间的概念，MySQL没有表空间

> Oracle的分组函数中select之后不能直接写非分组的其他列；

mySQL如果没有聚组函数则可以写其他列

> Oracle的表存放在用户（其实是schema对象）所属的表空间的数据文件中，

mySQL的表存放在某个database中（mysql一个root用户下可以有很多database）

> Oracle中的一些函数和MySQL也不一样

> Oracle中的字符类型可以使用varchar2 MySQL不能使用该类型

1. ArrayList 和 LinkedList的区别。

ArrayList底层是数组实现 优势在于查找遍历 随机访问!

LinkedList底层是双向循环链表实现 优势在于添加和删除

劣势在于查找遍历 最大的劣势在随机访问 (get())

\*:使用LinkedList务必回避它的get(int)

1. 说说left join 、right join 和inner join 的区别。

left join 左外连接

\* from 原表 join 新表 on 关联关系

原表和新表能够关联的数据+无法通过关联关系关联到原表的数据

right join 右外连接

原表和新表能够关联的数据+无法通过关联关系得到的新表的数据

inner join 内连接

内连接只查询到两张表有关联关系的数据

1. hashMap 和 hashtable 的区别

同步特性不同

首先HashMap同一时间允许多个线程同时进行访问，效率相对较高，但是可能 出现并发错误

Hashtable底层大量的使用了synchronized修饰方法，同一时间只允许一个线程 对其操作，效率相对较低，但是不会出现并发 错误

\*：从jdk5.0开始集合的工具类Collections当中出现了一个静态方法 synchronizedMap方法可以将一个不安全的map变成线 程安全的

\*：在高并发的场景下推荐使用java.util.concurrent.ConcurrentHashMap 有更高 的性能~

对于null的处理不同

HashMap无论是主键还是值都能存放null，

但是由于主键要求唯一，所以主键只能存放一个null，但是值能存放多个空

Hashtable无论是主键还是值都不能添加null，会触发空指针异常

底层实现的区别

底层有点不同 HashMap分组组数可以指定，默认分为16个小组，但是最终的 分组组数一定是2的n次方数，

在计算散列的时候，底层使用&(分组组数-1) [按位与运算]

Hashtable分组组数可以随意指定，默认分11组，可以随意指定分组组数

出现的版本不同

HashMap jdk1.2 Hashtable jdk1.0

1. 在js中如何创建一个数组对象？

两种方法:

1. var array = [1,2,3] //直接定义并初始化
2. var array = new Array(1,2,3) //调用构造函数创建数组并初始化
3. 数据库中in和exists的区别 哪个效率大？

exists效率高：exist只要存在就能被查到,不需要像in一样逐一比较子查询结果。

31.spring-mvc处理请求的流程 :

客户端提交请求，被web.xml中注册的请求接收器DispatcherServlet接收到 ； DispatcherServlet将拦截到的请求转发给ioc容器中注册的请求解析器(spring-mvc提供了两种请求解析器：字符请求解析器HandlerMapping 和 字节请求解析器 MultipartResolver 根据请求类型选择不同的解析器进行解析)；请求解析完毕之后，在ioc容器中查找一个请求处理器（使用@Controller注解标注的类）解析请求；在请求解析器中根据@RequestMapping(path)查找能够处理该请求的方法；请求处理完毕之后，（在ioc容器中查找视图解析器 根据方法的返回值[String or ModelAndView]渲染视图）

32.从两亿数据中查出两条一模一样的数据（要快，sql执行时间15秒最多）?

对数据进行分组，具有相同数据的数据会分为一组然后进行拿取

1. 定时任务分几种

a (oracle)触发器

b jdk - Timer计时器

任务对象 需要继承TimerTask 线程对象 run(){}

Timer触发任务对象

schedule(TimerTask , long , long)

何时准备调度的过程 ? tomcat启动时

ServletContextListener

\* 周期不固定

c quartz

任务对象 普通javaBean

a 配置任务对象 <bean>

b 描述任务细节 MethodInvokingJobDetailFactoryBean

setTargetObject() setTargetMethod

c 触发器

为任务设置触发的时间/周期

CronTriggerFactoryBean

setJobDetail(JobDetail jd)

setCronExpression(String cronExpression)

d 配置调度类

SchedulerFactoryBean

setTriggers(Trigger[])

1. 什么是缓存框架or缓存技术? 你接触过哪些缓存技术? 在什么地方用?怎么用?怎么在缓存中存储自定义对象?需要注意什么?

缓存技术:就是将程序或系统经常调用的对象存在内存中,以便其使用时可以快速调用,不 必再去创建新的重复的实例

接触过的缓存技术:

1. 浏览器缓存 cookie
2. hibernate 一级/二级缓存
3. Redis

高性能的key-value型数据库,在内存中存储一个Map对象

在该对象中以key-value形式存储数据库中的数据

java提供jedis开发包连接redis服务器

spring提供spring-data组件[spring-data-redis]

1) 开启redis服务器 redis-serve.exe redis.conf

2) 提供一个连接池文件 .properties

3) 配置连接池 JedisPoolConfig

4) 配置连接工厂 JedisConnectionFactory

setPort() setPoolConfig() setHostName()

5) 配置模板类 RedisTemplate

setConnectionFactory()

jt.execute(new RedisCallback(){

doInRedis(RedisConnection con){

con.set(key , value)

con.get(key)

}

})

opsForList() ListOperations

opsForSet() SetOperations

opsForHash() HashOperations.put(key , map-key , map-value)

get(key , map-key) -> map-value

35.http和web service请求的区别？

(1)HTTP是个网络协议 超文本传输协议

webservice - 底层采用SOAP[简单的对象访问协议]

(2) http为应用程序之间

Web service为多个服务器之间,多个工程之间的数据交互

SOAP是一种基于HTTP POST的协议

(3)HTTP的效率更高

SOAP基于XML的数据传输方式

1. get和post的区别?

method:表单提交的方式

get：速度快安全性较低，不支持中文

通过浏览器url进行值得传递，用户可以从

浏览器地址栏直接查看值传递的情况

base1.html?key1=value1&key2=value2&key3=value3\*\*\*

最多传递256个字符

post：速度慢安全性高，通过消息体传递值

无法查看，没有大小限制支持中文

37.spring如何整合hibernate

ORM提供的模板模式的实现类 : HibernateTemplate 封装传统hibernate链接数据库操作数据的流程

通过在DAO中对模板类的使用，实现对传统开发流程的代替。

要在DAO中使用模板类对象，需要先获取一个模板类实例，而获取实例的方式是采用了IOC的被动注入

因此需要在ioc容器中配置一个HibernateTemplate-Bean对象

而HibernateTemplate的使用需要一个SessionFactory实例的支持。

因此还需要在IOC容器中配置一个SessionFactory实例 - 配置方式：LocalSessionFactoryBean

通过其提供的三个setter方法 ： setDataSource setMappingResouces setHibernateProperties 实现SessionFactory的配置过程！

1. maven的使用

Maven是一个采用纯Java编写的开源项目管理工具, Maven采用了一种被称之为Project Object Model (POM)概念来管理项目，所有的项目配置信息都被定义在一个叫做POM.xml的文件中, 通过该文件Maven可以管理项目的整个生命周期，包括清除、编译，测试，报告、打包、部署等等.

Maven能够做什么:Jar的声明式依赖性管理 , 项目自动构建

1. 你用过线程里的什么锁?用线程怎么解决高并发?(如何实现线程安全)

①使用synchronized(锁标记/临界资源){需要连续执行的操作;}

②Lock lock = new java.util.concurrent.locks.Reentrantlock();

通过调用lock()和unlock()方法对临界资源进行操作

40.session和cookie的区别？

两种会话跟踪机制

Cookie和HttpSession

1:Cookie由服务器创建，是浏览器提供的一种会话

跟踪机制，所有浏览器都支持cookie会话跟踪机制

cookie存在构造方法，直接可以在服务器端new出来

传递一个键值对，数据类型必须都为String，默认不

支持中文，一般我们将可以分辨权限的数据放置在

cookie中用来对用户的权限等进行解析，当服务器

返回响应时，cookie一并返回客户端，并最终保存在

用户本地，我们从浏览器的设置中可以找到本地的

cookie实体文件

2:HttpSession由服务器创建，是web容器提供的一种

安全性很高的会话跟踪机制，通过request.getSession()

创建新的HttpSession(中文称之为会话)，支持中文

通过setAttribute(String,Object)封装数据，通过

getAttribute(String)来拿取数据。HttpSession在创建后

开始进行倒计时，默认最大不活动周期为1800s（可以更改）

，在这期间如果用户再次提交请求，则重新倒计时。

HttpSession是一段时间概念，我们无法从本地找到一个具体的

HttpSession实体

41.接口和抽象类的区别

(1)接口当中定义的方法 默认添加三个修饰符public static final

接口当中定义的变量默认就是常量

抽象类当中定义的变量是普通的属性 每个这个类型的对象都有一份

(2)接口当中定义的方法 自动添加两个修饰符 public abstract

抽象类当中定义的方法可以是非抽象方法 也可以是抽象方法

\*:从JDK8.0开始 接口当中的方法可以有具体的方法实现

a> static修饰的静态方法

b> default修饰的默认方法

42.Spring和Spring-MVC的区别

①spring-mvc框架是作为spring框架的一个层提出的，mvc的运行和使用方式依 赖spring的ioc容器。

③在web(web.xml)环境下对两个框架的注册方式不同

Spring需要ContextLoaderListener,

springmvc需要DispatcherServlet

④在web工程中既需要spring框架提供的管理对象特性也 需要spring-mvc框架 提供的处理请求功能

⑤两个ioc容器加载的时机不同

Spring在tomcat启动时加载,tomcat启动完毕后,ioc容器中所有 <bean>全部被实例化

Springmvc是在客户端首次发送请求时加载

43.oracle中用过哪些与时间有关的函数？怎么查询每个月第一天的数据？

add\_months(d1,i2) 在d1时间上增减 i2 个月份

months\_between(d1,d2) 计算两个日期之间的月份

last\_day(d1) 得到给定日期所在月份的最后一天是几号

next\_day(d1,'周期') 得到给定日期d1之后指定的周期是哪天 周期可以是:星期日 星期一 星期二.... 星期六

查询给定日期所在月份的第一天,可以使用trunc(d1,'mm') 得到,

trunc能够精确日期,第二个参数'mm'表示精确到当前的月份,天数默认为1 也就是该 月第一天

44.触发器 存储过程

存储过程 procedure：在服务器端，能够被多个程序调用的一段SQL语句块。

简单来说就是事先用数据库语言写好的一段功能，能够像函数一样被程序调用，

但是存储过程本身没有返回值，是通过返回参数的形式返回数据。

触发器 trigger：一种特殊的存储过程，不需要调用也没有参数，在满足条件时触发：

触发条件：在对某个表做insert 、 update 或delete操作之前或之后（取决于如何 实现）自动执行

45.视图

视图 View ： 将编译后的SQL语句保存在数据库中，这就是视图。视图可以将查询语句保存起来以供下次执行而无需编译，还能屏蔽某些敏感字段保护隐私。

46.如何创建序列 如何指定步长(每次增长大小)?

创建序列:create sequence 序列名;

minvalue 最小值

maxvalue 最大值

increment by 每次增长多大;

47.悲观锁与乐观锁?

悲观锁: 在操作表时认为另外一个对表的操作会修改数据

乐观锁: 在操作表时认为别的操作不会对表有修改数据的操作

48.一个table如何得到第二行第三列的值?

在JDBC中,我们可以通过ResultSet对象得到:

ResultSet rs对象中的方法absolute(2)能够定位到第二条数据,

rs.getObject(3)可以得到第2条第3列的值.

49.spring切面可以应用5种类型的通知,简单介绍?

Before 在核心方法执行之前执行

After 在核心方法执行之后执行

AfterReturning 会在核心方法执行完毕并且获得返回值之后执行

afterThrowing 会在核心方法出现异常时执行

Throws 会环绕核心方法执行,也就是before和after的组合

50.spring的7种事务传播行为?

请写出Spring的七种事务传播行为

mandatory

支持当前事务，如果当前没有事务抛出异常

（我要找个有钱的男票，屌丝滚粗）

never

不支持当前事务，如果当前有事务抛出异常

（我不喜欢有钱的[太花心了?）

supports

支持当前事务，如果当前没有则以非事务环境运行

（有钱过有钱的日子，没钱过没钱的日子）

not\_support

不支持当前事务，总是以非事务环境运行

（不喜欢有钱的，就想过没钱的日子）

required

支持当前事务，如果没有则创建一个新的（常用&&默认）

(喜欢有钱的，没钱我可以自己挣钱O(∩\_∩)O）

requires\_new

总是创建一个新事务，如果当前有事务则挂起当前事务（不是吊死他，是等新 事物运行完了再运行原实物)

(不喜欢花你的钱，等花完我的再用你的）

nested

如果当前有事务则把新事物作为原事务的内嵌事务进行执行；否则以required 方式运行

（我破产了你可以罩着我，你破产了我也玩完了...）

51.使用spring框架的好处是什么?

Spring是轻量级的ioc和aop框架,它的轻量级主要是相对于重量级的EJB而言的.

Spring具有非入侵性和对服务器相对低的依赖性

1. 提供了ioc容器实现对象的创建和依赖注入,降低组件之间的耦合性
2. 提供了Aop思想,将核心业务逻辑与非核心业务逻辑分离,实现了将非核心业务添加给需要的核心业务逻辑,从而让程序员更好的关注核心业务逻辑
3. 实现了对主流框架的集成:hibernate,mybatis,struts
4. 开发者根据分层理念,可以任意选择spring的某个部分使用

52.简单介绍一下你对springmvc的理解

(1)Springmvc是spring提供的用来处理http请求的技术

Springmvc的运行和使用依赖于spring的ioc容器

(2)Springmvc定义了一套用于处理请求的<bean>对象:

DispatcherServlet请求接收器

HandlerMapping(字符)和MultipartResolver(字节) 请求解析器

请求处理器@Controller

视图解析器ViewResolver和视图对象view

拦截器HandlerInterceptor

1. Springmvc处理请求的流程:

客户端的请求首先被注册在web容器中的DispatcherServlet统一拦截，并转发给ioc容器中的请求解析器（这里会根据请求的类型[字符or字节]自动选择匹配的解析器），接下来在ioc容器环境下，会根据解析之后的字符串[/前缀]查找用于处理请求的处理器Controller，通过@RequestMapping(path)在Controller中查找与之映射的处理方法。请求处理完毕之后，会根据处理请求方法的返回值[String or ModelAndView]以ViewResolver为基础返回一个View视图对象。

53.使用springmvc是不是每次请求都创建一个controller对象

由于springmvc的请求处理是依赖于ioc容器的,而用于处理请求的@Controller也属于ioc容器的<bean>,而ioc容器中的bean对象默认是单例的singleton,所以在默认情况下,springmvc只维护一个controller来处理多个客户端提交的请求

也可以将其设置为非单例状态，以保证每次请求都可以创建一个新的controller对象进行处理。只需要将controller对象的scope属性设置为prototype。或者使用@Scope注解

54.spring如何整合struts?

使用spring的ioc容器管理struts中用于处理请求的action,将action配置成ioc容器中的<bean>

1. 客户端提交的请求被struts拦截,通过struts.xml的<action>来配置其映射关系,name属性指向请求名,class属性指向ioc容器中某个<bean>的id值,表示使用该bean指向的处理器处理当前请求.
2. 将action配置在ioc容器之后,该action默认会在tomcat启动时初始化,并且初始化的状态为：单例（只维护一个Action实例处理多用户提交的请求）。区别于单独使用struts时客户端每次发送请求创建一个新的action处理。

55.为什么说spring是轻量级的?

Spring的提出是为了解决传统的javaEE开发的复杂性而提出的,其中最重要的一点是:使用普通的javabean代替传统的javaEE开发中的EJB技术

Spring的轻量级主要体现在相比于重量级的EJB而言的

1. 对服务器没有任何依赖性,spring作为javaEE框架,既可以在web环境下使用,也可以脱离web环境单独使用
2. 对代码的低侵入性:spring管理对象采用普通的javaBean,不需要让其管理的对象实现springApi对象或接口

56.aop原理

底层采用代理模式实现,(jdk动态代理和cglib代理的组合)

将非核心的封装在aop组件中,采用xml配置的方式将aop组件提供的服务引用给需要的核心业务对象或方法

典型的aop实现技术:interceptor

57.事务中用到的声明标签,注解有哪些

@Transactional(

Isolation,

propagation,

readOnly,

timeout,

rollback-for

)

58.spring mvc 中的c在项目中是怎么实现的？

c - Controller 控制器

分为三部分 :

请求接收器 : DispatcherServlet 注册在web容器中,负责请求的拦截

请求解析器 : HandlerMapping 或 MultipartResolver 注册在IOC容器中，负责请求的 解析工作

请求处理器 : 实现Controller接口 或 使用@Controller标注 ，负责请求的处理

59.spring如何整合mybatis?

使用spring的IOC容器管理mybatis的接口实例

(1)SqlSessionFactory - bean 接口

SqlSessionFactoryBean

数据源 setDataSource(DataSource ds)

映射文件 setMapperLocations(String[] mappers) , 为该对象注入数据源和映射文件

(2)工厂bean

MapperFactoryBean mybatis-spring-1.1.0.jar

通过该工厂获取哪个接口的实例bean

setMapperInterface(Class cls)

如何根据该接口获取对应的实例

通过SqlSessionFactory解析接口对应的映射文件获取

setSqlSessionFactory(SqlSessionFactory ssf)

60.hql和sql最根本的区别是什么？

HQL:面向对象

SQL:面向关系数据库

61.mybatis - 怎么用动态标签 做三个判断，如果三个判断都符合则不返回值！同时不报错！

<if test="classEntity != null and classEntity.classId !=null and classEntity.classId !=' ' ">

62.Mybatis中的里面的id对应的是什么

id对应的是要执行的SQL的标示，当我们在DAO中想调用

SQL语句时，就是通过id来寻找对应的SQL语句

63.hibernate的映射文件有哪些

Hibernate配置文件一般包括两种，一种是属性文件另一种是映射文件，属性文件一般包含连接数据库的信息，方言信息以及其他可选信息，映射文件中包括类与表的映射，类中属性与表中字段的映射,如果有多个实体，则可能包含一些实体关联映射。

64.问bootstrap，用的最多的是什么？

平常使用最多的就是删格系统、流式布局、表格、

表单以及一些组件比如按钮、分页、导航条、列表组、轮播等。

注意:最好包含 组件 布局 表格 珊格。

65.bootstrap中的核心组件都有哪些？

表格、表单、珊格系统、流式布局、按钮、导航、分页、列表等

66.bootstrap,easyui？

首先、Bootstrap跨平台更准确的描述应该是跨不同的设备。

BootStrap的最大特点就是响应式设计。所谓响应式设计：就是让网站

针对不同的浏览器和设备响应不同的显示效果的策略。

然后、Bootstrap之所以能实现在不同设备之间显示不同的效果，最

大的特点在于应用了CSS3的Media Query(媒体查询)技术，即能够针对

不同的媒体和设备显示不同的效果

再次、Bootstap和EasyUI侧重点不同，Bootstrap侧重前端CSS效果，即页面如何展示，并且提供了很多组件供我们使用，而EASYUI则为我们提供了一中后台管理软件开发中模板，并且也提供了很多DataGrid之类的组件。

67.Struts2是单线程还是多线程?Spring是单线程还是多线程?

Struts2的Action需要传递一些客户端参数，所以每次都会请求都会创建一个Action 对象，每个对象中存放本次请求的参数。所以从Action角度看 Struts2是单线程 的。

Spring的bean可以通过 prototype控制 bean是单例的还是多例的。

68.HttpSession和cookie是两种会话跟踪机制，

表面上两者没有任何联系，HttpSession的应用场合

一般较高，两者没有任何关系吗?答案是否定的，

HttpSession并不能独立从一个页面传递到另外一个

页面，一般情况下它是通过浏览器提供的cookie会话

跟踪机制进行跳转的，

Cookie cookie = new Cookie("jsessionid",\*\*\*)

key为jsesssionid

value为session的id数

当开启的session会话跟踪机制后，cookie会自动

帮助session进行传递，我们不需要手动进行设置

这也是为什么在禁用cookie后，HttpSession会话跟踪

机制也会失效

69.禁用Cookie就无法维持Session会话跟踪机制了吗?

重写url的方式可以解决

String path = "/ServletDay4\_cookie\_session/servlet/TestSession2";

//重写url

String finalPath = response.encodeURL(path);

response.sendRedirect(finalPath);

答案:通过response.encodeURL(路径)

将sessionid封装进提交的url路径

则sessionid可以传递到新的页面，从而web容器

根据id提供的会话，这个会话就是在跳转前创建好的

（因为id一致），从而继续维持会话跟踪机制

70.servlet API中forward()与redirect()的区别？

forward方式：request.getRequestDispatcher("/somePage.jsp").forward(request,response);

redirect方式：response.sendRedirect("/somePage.jsp");

请求转发(forward)

由浏览器发出，经程序转发，浏览器并不知道

跳转到哪里，所以浏览器地址栏依然是跳转之前的路径

跳转之前是do\*\*\*跳转之后依然是do\*\*\*

因为是同一次请求，所以request范围有效

重定向(redirect)

程序跳转之后返回响应给浏览器，浏览器发出最后一次请求

因为是浏览器发出所以浏览器地址栏就是最终跳转的路径

跳转之前是do\*\*\*(do post或者doget)跳转之后肯定是doGet,因为浏览器默认使用get 发送请求

因为不再是同一次请求，所以request范围无效

71.JavaScript的数据类型都有什么？

String null number undefined boolean

Array Object

72.已知ID的input输入框，希望获得这个输入框的输入值，怎么做？（不使用第三方框架）

document.getElementById(ID).value

73.jquery怎么全选一个多选框

$(":checkbox").attr("checked",true);

74.jquery对象怎么转化为js对象，js对象怎么转为jquery对象

jquery->js

$().get(0)

$()[0]

js->jquery

$(js对象)

75.平时上传用的什么？前端的上传框架用过什么？

SmartUpload

ajaxFileUpload

uploadify

76.页面如何实现延迟加载

1：延迟加载js代码

<script type="text/javascript">

function demo(){}

$(function(){

setTimeout('demo()',1000); //延迟1s

})

</script>

2：将js代码放置在body底部最后加载

77.jquery中$.ajax中参数

$.ajax({

//要提交到目的地

url:"servlet/Page",

//1.2新特性，设置为false则不缓存此页面

cache:false,

//默认true，为异步请求，更改为false则变为同步请求，浏览器被锁住

async:true,

//异步请求的类型

type:"post",

//要传递的值 ,常使用serialize()方法序列化表单

data:"key="+value+"&key2="+value2,

//返回的值的类型支持 json html text script 和 xml

dataType:"json",

success:function(data){

//服务器无误是回调函数

}，

error:function(){

//服务器出现异常时回调函数

}

);

78.Jquery遍历方法

$.each(循环体,function(index,element){

})

index:表示索引值

element:表示每一个被遍历的元素

79.mvc的controller怎么处理分页的(\*\*\*\*\*)

接受前台传递过来的当前页默认值以及每页记录数，调用

数据拿取总记录数，通过计算取出总页数，根据分页查询公式

拿取分页查询的集合，将总记录数 总页数 集合封装进json中

发送给页面，页面获取后拿取数据

80.当一个dom节点被点击的时候，我们希望能够执行一个函数，应该怎么做？

$(dom).click(function(){

})

81.Js选择器由哪些？

getElementById()

根据id拿取一个元素

getElementsByName()

根据name属性拿取多个元素

getElementsByTagName()

根据标签名拿取多个元素

82.ajax同步请求和异步请求如何实现？有什么区别？

(1)原生ajax

在open(url,data,async,username,password)

异步：async 填写 true

同步：async 填写 false

(2)jquery的ajax

$.ajax({

同步（默认）

async:true

异步

async:false

})

异步请求：用户发出请求之后不需要等待响应的返回

同步请求：用户发出请求之后浏览器锁死，必须等待

响应的返回

83.document load 和 document ready的区别?

load 当整个页面全部加载完毕之后触发

ready 当整个页面的结构全部加载完毕之后触发

84.jQuery的选择器、怎么用

$("标签名")

直接使用标签名作为选择元素的依据

$(".class名")

直接使用.class属性名作为选择元素的依据

$("#id名")

直接使用#id属性名作为选择元素的依据

$("标签#id名")

$("标签.class名")

必须同时满足标签带有id或者相应class

的元素

$("选择器,选择器,选择器")

只要符合其中任意一个选择器就可以成功选取

$("选择器 选择器 选择器")

根据左祖先右后代的原则精确选取子元素

$("\*")

选择全部元素

$("选择器 > 选择器")

仅仅选取字辈元素

$("选择器 + 选择器")

选取的条件

1：从上向下选取

2：必须紧邻

3：必须互为兄弟

$("选择器 ~ 选择器")

向下选取所有的兄弟元素，不需要紧邻

$("选择器:first")

拿取指定元素序列中第一个

$("选择器:last")

拿取指定元素中最后一个

注意以上两个选择器选择都是

全文中的第一个和最后一个，

如果要选择每个ul中的第一个或者最后一个

请使用

:first-child

:last-child

$("选择器:eq(index)")

根据索引值拿取元素

$("选择器:gt(index)")

拿取索引值大于index的指定元素

&gt;

$("选择器:lt(index)")

&lt;

拿取索引值小于index的指定元素

$("选择器:contains('文本内容')")

拿取含有特定内容的元素

$("选择器:has(选择器)")

拿取含有特定内容得元素

$("选择器:hidden")

拿取被隐藏的特定元素

注意如果条件符合也包含

<input type="hidden" />

.text()拿取已经获得元素中的文本

$("选择器:visible")

拿取所有可见元素

$("选择器[属性名='属性值']")

拿取属性名为属性值的特定元素

$("选择器[属性名!='属性值']")

拿取属性名不为属性值的特定元素

html() 效果类似于 javascript中 innerHTML

$("选择器[属性名\*='部分属性值']")

拿取属性名部分匹配的特定元素

$("选择器[属性名^='属性的开头']")

拿取以属性名为\*\*\*开头的特定元素

$("选择器[属性名$='属性的结尾']")

拿取以属性名为\*\*\*结尾的特定元素

$(选择器:even)

拿取索引值为偶数和0的特定元素

$(选择器:odd)

索取索引值为奇数的特定元素

$(":header")

拿取h1~h6的标题

$(":input")

拿取所有的input表单元素

注意包含<textarea> <button> <select>

$(":text")

拿取单行文本输入框

$(":password")

拿取单行文本密码框

$(":radio")

拿取单选框

$(":checkbox")

拿取多选框

$(":image")

拿取input中type="image"的元素

注意此选择器不能拿取<img>标签

$(":submit")

拿取所有的提交按钮

val()拿取所获得元素的value属性

$(":reset")

拿取所有重置按钮

$(":button")

拿取button按钮

此选择器能够选择：

<input type="button" />和<button>

$(":checked")

拿取已经选中的元素

$(":selected")

下拉框被选中的元素

\*\*\*\*\*还没有打印的内容\*\*\*\*\*

85.xml文档定义有几种形式？他们之间有何本质区别？解析XML文档有哪几种方式？

(1)XML文档定义:dtd和schema

(2)区别:schema要比DTD更加的细腻规范，比如Shame会定义整个文档的层次接口， DTD没有层次结构，Schema有数据类型，DTD没有数据类型

(3)解析XML:JDK中提供的有SAX和DOM

第三方:DOM4J /JDOM/XStream等

86.sturts和json分别是哪家的？

Struts：是ASF(Apache Software Foundation)的即apache的。

Json：JSON是javascript的一个子集。而解析json一般使用net.sf.json包下的东东， 当然在Spring中内置了json解析工具jackson工具包里。