JAVA就业课程****

一、面试整体流程

1.1 简单的自我介绍

我是xxxx,工作xxx年.我先后在xxxx公司、yyyy公司工作。先后做个xxxx项目、yyyy项目。

1.2 你简单介绍一下xxxx项目

为了解决xxxx问题，开发了一套xxxx系统，该系统主要有那些部分组成。简单介绍项目的整体架构。参与某个模块的开发。就要求你说一下这个模块的业务及设计。

1.3 会问一下JAVA的专业技能

后面详细讲解

1.4你还有什么需要询问我的吗

公司要做的项目？项目中会使用一下什么技术?

注意：经历了多轮面试后，对于你的自我介绍和项目项目经验面试官就不太关心了。

你说一下你最擅长的什么？你简单说一下?

最终技术面试完成后，都会让你回家等消息，或者等hr来和你谈薪资和福利。

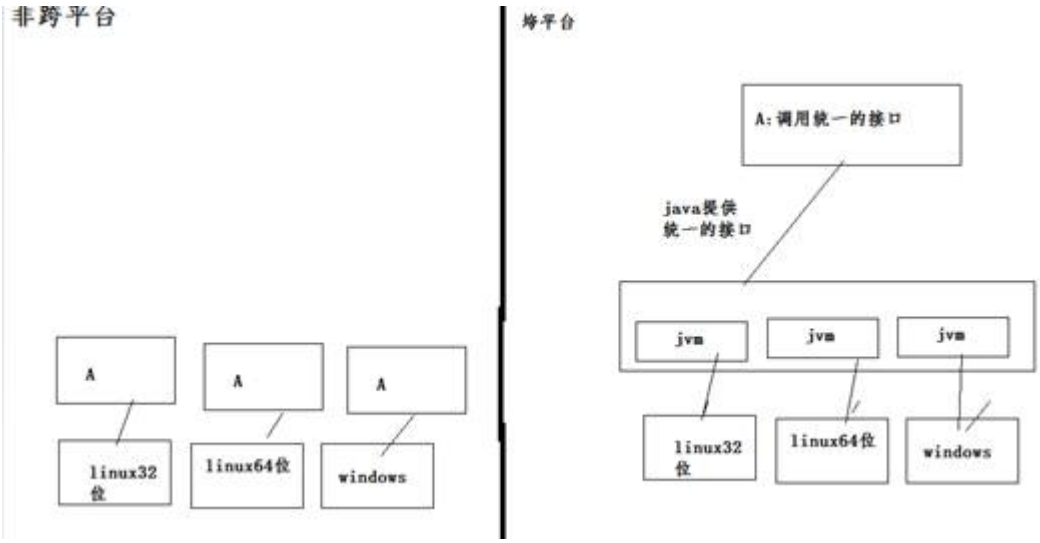
二、java的专业技能

2.1 java的基础部分

2.1.1 简单讲一下java的跨平台原理

由于各操作系统（windows,liunx等）支持的指令集，不是完全一致的。就会让我们的程序在不同的操作系统上要执行不同程序代码。Java开发了适用于不同操作系统及位数的java虚拟机来屏蔽个系统之间的差异，提供统一的接口。对于我们java开发者而言，你只需要在不同的系统上安装对应的不同java虚拟机、这时你的java程序只要遵循java规范，就可以在所有的操作系统上面运行java程序了。

Java通过不同的系统、不同版本、不同位数的java虚拟机(jvm),来屏蔽不同的系统指令集差异而对外体统统一的接口(javaAPI),对于我们普通的java开发者而言，只需要按照接口开发即可。如果我系统需要部署到不同的环境时，只需在系统上面按照对应版本的虚拟机即可。



2.2.2 搭建一个java开发环境的步骤

Java开发环境需要些什么?

1、适用于我们开发环境的jdk

2、对应开发环境eclipse

3、还需要web服务器(tomcat)

一、下载对应组件

二、安装

Jdk,安装正常流程安装即可，配置我们的JAVA_HOME,因为后面的eclispe和tomcat会依赖于这个变量.

Eclispe正常解压就ok,设置workspace的默认编码

Tomcat 正常解压就ok,把tomcat集成到eclispe中，安装插件就OK。

.....

Svn/git

2.1.3讲一下java中int数据占几个字节

Java中有几种基本数据类型？8种

Java 占用字节数

数据类型	大小(二进制位数)	范围	默认值
byte(字节)	8	-128 - 127	0
shot(短整型)	16	-32768 - 32768	0
int(整型)	32	-2147483648-2147483648	0
long(长整型)	64	-9233372036854477808-923337203685447780	0
float(浮点型)	32	-3.40292347E+38-3.40292347E+38	0.0f
double(双精度)	64	-1.79769313486231570E+308-1.79769313486231570E+308	0.0d
char(字符型)	16	' \u0000 - u\ffff '	'\u0000 '
boolean(布尔型)	1	true/false	false

Int占 4个字节，32位

Boolean 1位

2.1.4 面向对象的特征有哪些方面

有四大基本特征:封装、抽象、继承、多态

** **面向对象的封装性，即将对象封装成一个高度自治和相对封闭的个体，对象状态（属性）由这个对象自己的行为（方法）来读取和改变。

张三这个人，他的姓名等属性，要有自己提供的获取或改变的方法来操作。privatename setName getName

抽象就是找出一些事物的相似和共性之处，然后将这些事物归为一个类，这个类只考虑这些事物的相似和共性之处，并且会忽略与当前主题和目标无关的那些方面，将注意力集中在与当前目标有关的方面。 就是把现实生活中的对象，抽象为类。

在定义和实现一个类的时候，可以在一个已经存在的类的基础之上来进行，把这个已经存在的类所定义的内容作为自己的内容，并可以加入若干新的内容，或修改原来的方法使之更适合特殊的需要，这就是继承。遗产继承

多态是指程序中定义的引用变量所指向的具体类型和通过该引用变量发出的方法调用在编程时并不确定，而是在程序运行期间才确定，即一个引用变量倒底会指向哪个类的实例对象，该引用变量发出的方法调用到底是哪个类中实现的方法，必须在由程序运行期间才能决定。

Object obj =new xxx();

UserDaouserDao = new UserDaoJdbcImpl();

UserDaouserDao = new UserDaoHibernateImpl();

靠的是父类或接口定义的引用变量可以指向子类或具体实现类的实例对象，而程序调用的方法在运行期才动态绑定，就是引用变量所指向的具体实例对象的方法，也就是内存里正在运行的那个对象的方法，而不是引用变量的类型中定义的方法。

原则：回答比较抽象问题的时候，要举例说明

2.1.5有了基本数据类型，为什么还需要包装类型?

基本数据类型，java中提供了8中基本的数据类型。boolean int float等

包装类型：每一个基本的数据类型都会一一对应一个包装类型。

boolean ----->Boolean

Int -------->Integer

装箱和拆箱

装箱：把基本的数据类型转换成对应的包装类型.

Integer .valueOf(1)

Integer i = 1;自动装箱，实际上在编译时会调用Integer .valueOf方法来装箱

拆箱：就是把包装类型转换为基本数据类型.基本数据类型 名称 = 对应的包装类型。

Integer i = 1;

int j = i;//自动拆箱//int j = i=intValue();手动拆箱

自动拆箱：实际上会在编译调用intValue

Java是一个面向对象的语言，而基本的数据类型，不具备面向对象的特性。

null Integer--->null int---->0 用Integer和int分别表示Person这个类的ID

Max 最大值

min 最小值

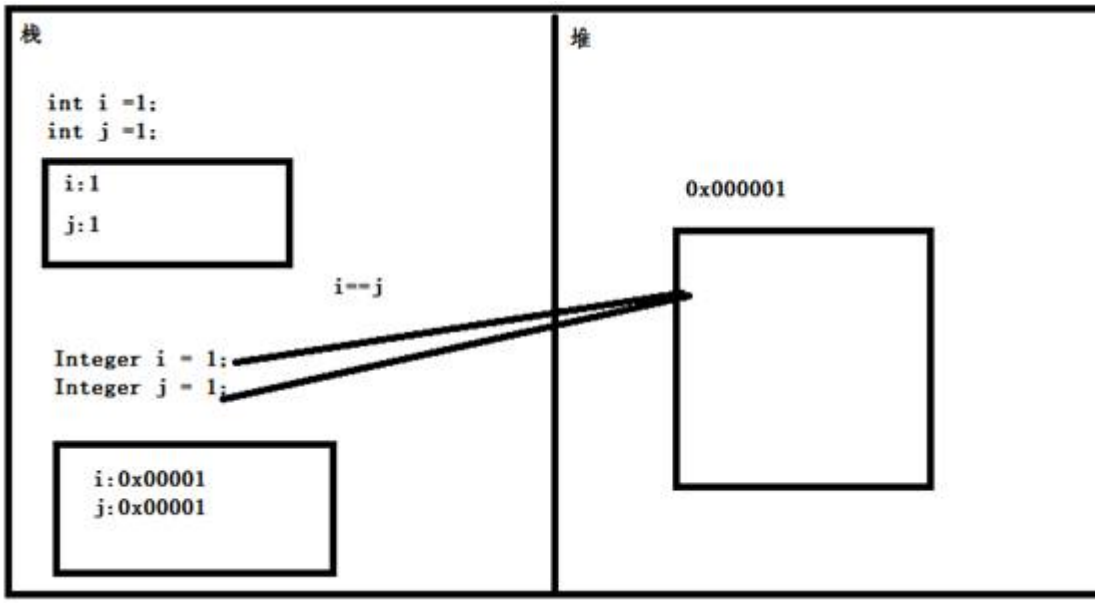
缓存值:对象缓存,Integer i=1; integer j= 1;i ==j

2.1.6、说一下"=="和equals方法究竟有什么区别?

非常经典的一个面试题？先说清楚一个，再来说另一个？

==用来判断两个变量之间的的值是否相等。变量就可以分为基本数据类型变量，引用类型。

如果是基本数据类型的变量直接比较值而引用类型要比较对应的引用的内存的首地址。



equals 用来比较两个对象长得是否一样。判断两个对象的某些特征是否一样。实际上就是调用对象的equals方法进行比较。

2.1.7讲一下String和StringBuilder的区别(final)？StringBuffer和StringBuilder的区别?

1.在java中提供三个类String StringBuillder StringBuffer来表示和操作字符串。字符串就是多个字符的集合。

String是内容不可变的字符串。String底层使用了一个不可变的字符数组(final char[])

```
/** The value is used for character storage. */
private final char value[];
```

Stringstr =new String(“bbbb”);

而StringBuillder StringBuffer,是内容可以改变的字符串。StringBuillder StringBuffer底层使用的可变的字符数组（没有使用final来修饰）

```
/**
 * The value is used for character storage.
 */
char[] value;
```

2.最经典就是拼接字符串。

1、String进行拼接.String c = “a”+”b”;

2、StringBuilder或者StringBuffer

StringBuildersb = new StringBuilder(); sb.apend(“a”).apend(“b”)

拼接字符串不能使用String进行拼接，要使用StringBuilder或者StringBuffer

3.StringBuilder是线程不安全的，效率较高.而StringBuffer是线程安全的，效率较低。

2.1.8、讲一下java中的集合?

Java中的集合分为value，key--vale(Conllection Map)两种。

存储值有分为List 和Set.

List是有序的，可以重复的。

Set是无序的，不可以重复的。根据equals和hashcode判断，也就是如果

一个对象要存储在Set中，必须重写equals和hashCode方法。

存储key-value的为map.

8、ArrayList和LinkedList的区别?

List常用的ArrayList和LinkedList。区别和使用场景?

ArrayList底层使用时数组。LinkedList使用的是链表。

数组查询具有所有查询特定元素比较快。而插入和删除和修改比较慢(数组在内存中是一块连续的内存，如果插入或删除是需要移动内存)。

链表不要求内存是连续的，在当前元素中存放下一个或上一个元素的地址。查询时需要从头部开始，一个一个的找。所以查询效率低。插入时不需要移动内存，只需改变引用指向即可。所以插入或者删除的效率高。

ArrayList使用在查询比较多，但是插入和删除比较少的情况，而LinkedList使用在查询比较少而插入和删除比较多的情况。

2.1.9讲一下HashMap哈HashTable的区别?HashTable和ConcurrentHashMap的区别?

相同点：HashMap和HasheTalbe都可以使用来存储key--value的数据。

区别：

1、HashMap是可以把null作为key或者value的，而HashTable是不可以的。

2、HashMap是线程不安全的，效率较高。而HashTalbe是线程安全的，效率较低。

？我想线程安全但是我又想效率高?

通过把整个Map分为N个Segment（类似HashTable），可以提供相同的线程安全，但是效率提升N倍，默认提升16倍。

2.1.10、实现一个拷贝文件的工具类使用字节流还是字符流?

我们拷贝的文件不确定是只包含字符流，有可以能有字节流(图片、声音、图像等)，为考虑到通用性，要使用字节流。

2.1.11、讲一下线程的几种实现方式?启动方式？区分方式?

①实现方式

1、通过继承Thread类实现一个线程

2、通过实现Runnable接口实现一个线程

继承扩展性不强，java总只支持单继承，如果一个类继承Thread就不能继承其他的类了。

②怎么启动?

Thread thread = new Thread(继承了Thread的对象/实现了Runnable的对象)

thread.setName(“设置一个线程名称”);

thread.start();

启动线程使用start方法，而启动了以后执行的是run方法。

③怎么区分线程？在一个系统中有很多线程，每个线程都会打印日志，我想区分是哪个线程打印的怎么办?

thread.setName(“设置一个线程名称”); 这是一种规范，在创建线程完成后，都需要设置名称。

2.1.12有没有使用过线程并发库?

简单了解过?

JDK5中增加了Doug Lea的并发库，这一引进给Java线程的管理和使用提供了强大的便利性。 java.util.current包中提供了对线程优化、管理的各项操作，使得线程的使用变得的心应手。该包提供了线程的运行，线程池的创建，线程生命周期的控制.

Java通过Executors提供四个静态方法创建四种线程池，分别为:

newCachedThreadPool创建一个可缓存线程池，如果线程池长度超过处理需要，可灵活回收空闲线程，若无可回收，则新建线程。

newFixedThreadPool 创建一个定长线程池，可控制线程最大并发数，超出的线程会在队列中等待。

newScheduledThreadPool 创建一个定长线程池，支持定时及周期性任务执行。

newSingleThreadExecutor 创建一个单线程化的线程池，它只会用唯一的工作线程来执行任务，保证所有任务按照指定顺序(FIFO, LIFO, 优先级)执行

2.1.13线程池的作用?

1、限定线程的个数，不会导致由于线程过多导致系统运行缓慢或崩溃

2、线程池不需要每次都去创建或销毁，节约了资源、

3、线程池不需要每次都去创建，响应时间更快。

连接池也是一样?

2.1.14讲一下什么是设计模式？常用的设计模式有哪些?

设计模式就是经过前人无数次的实践总结出的，设计过程中可以反复使用的、可以解决特定问题的设计方法。

单例(饱汉模式、饥汉模式)

1、构造方法私有化，让出了自己类中能创建外其他地方都不能创建

2、在自己的类中创建一个单实例（饱汉模式是一出来就创建创建单实例，而饥汉模式需要的时候才创建）

3、提供一个方法获取该实例对象(创建时需要进行方法同步)

工厂模式:Spring IOC就是使用了工厂模式.

对象的创建交给一个工厂去创建。

代理模式:Spring AOP就是使用的动态代理。

2.2 java web部分

2.2.1讲一下http get和post请求的区别?

GET和POST请求都是http的请求方式，用户通过不同的http的请求方式完成对资源（url）的不同操作。GET，POST，PUT，DELETE就对应着对这个资源的查，改，增，删 4个操作,具体点来讲GET一般用于获取/查询资源信息，而POST一般用于更新资源信息

1、Get请求提交的数据会在地址栏显示出来，而post请求不会再地址栏显示出来.

GET提交，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，多个参数用&连接；POST提交：把提交的数据放置在是HTTP包的包体中。 因此，GET提交的数据会在地址栏中显示出来，而POST提交，地址栏不会改变

2、传输数据的大小

http Get请求由于浏览器对地址长度的限制而导致传输的数据有限制。而POST请求不会因为地址长度限制而导致传输数据限制。

3、安全性,POST的安全性要比GET的安全性高。由于数据是会在地址中呈现，所以可以通过历史记录找到密码等关键信息。

2.2.2 说一下你对servlet的理解？或者servlet是什么?

Servlet（Server Applet），全称**Java Servlet**， 是用Java编写的服务器端程序。而这些Sevlet都要实现Servlet这个借口。其主要功能在于交互式地浏览和修改数据，生成动态Web内容。Servlet运行于支持Java的应用服务器中。

HttpServlet 重写doGet和doPost方法或者你也可以重写service方法完成对get和post请求的响应

2.2.3简单说一下servlet的生命周期?

servlet有良好的生存期的定义，包括加载和实例化、初始化、处理请求以及服务结束。这个生存期由javax.servlet.Servlet接口的init,service和destroy方法表达。

加载Servlet的class---->实例化Servlet----->调用Servlet的init完成初始化---->响应请求（Servlet的service方法）----->Servlet容器关闭时(Servlet的destory方法)

Servlet启动时，开始加载servlet生命周期开始。Servlet被服务器实例化后，容器运行其init方法，请求到达时运行其service方法，service方法自动派遣运行与请求对应的doXXX方法（doGet，doPost）等，当服务器决定将实例销毁的时候(服务器关闭)调用其destroy方法。

2.2.4 Servlet API中forward() 与redirect()的区别?

1、forward是服务器端的转向而redirect是客户端的跳转。

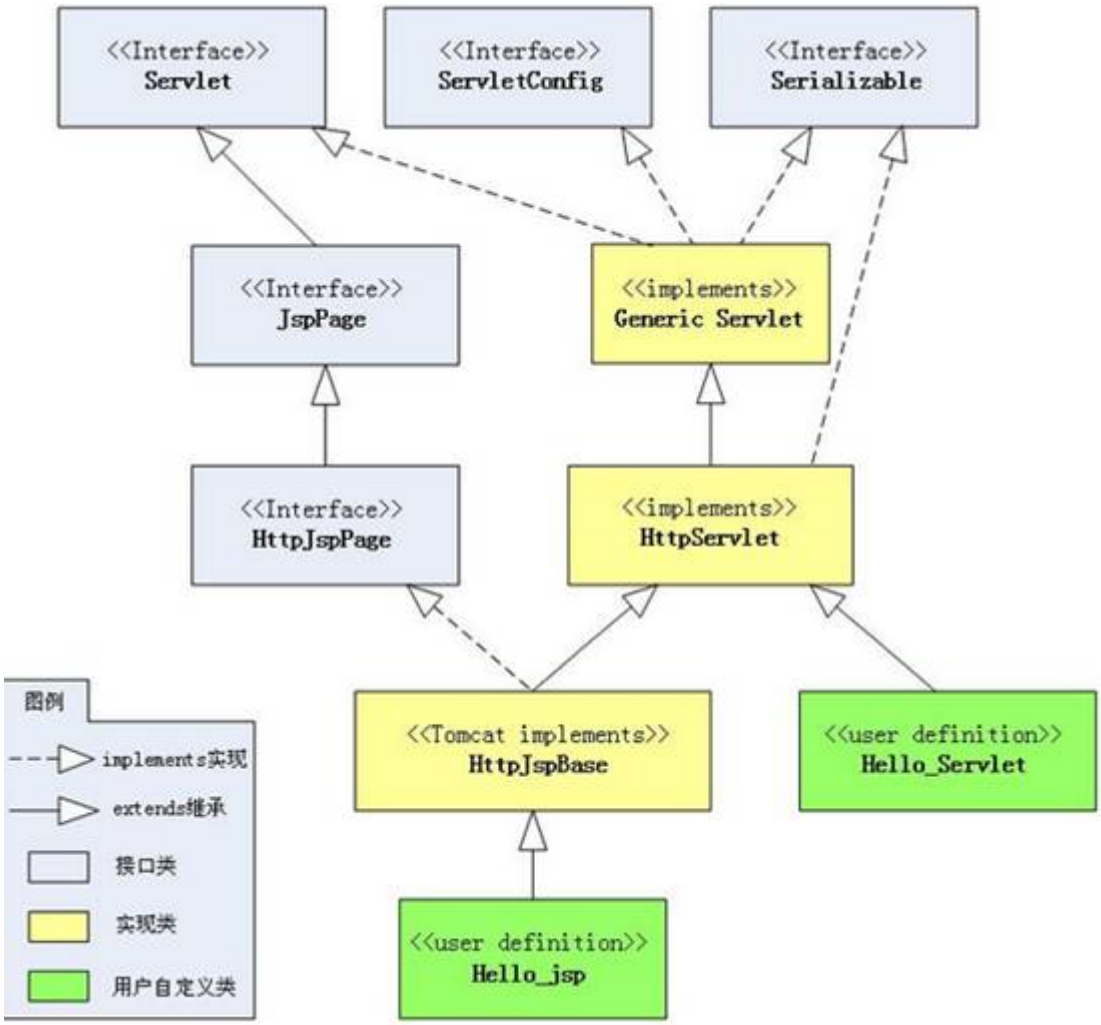
2、使用forward浏览器的地址不会发生改变。而redirect会发生改变。

3、Forward是一次请求中完成。而redirect是重新发起请求。

4、Forward是在服务器端完成，而不用客户端重新发起请求，效率较高。

2.2.5 JSP和Servlet有哪些相同点和不同点?

JSP是Servlet技术的扩展，所有的jsp文件都会被翻译为一个继承HttpServlet的类。也就是jsp最终也是一个Servlet.这个Servlet对外提供服务。



Servlet和JSP最主要的不同点在于JSP侧重于视图，Servlet主要用于控制逻辑。

Servlet如果要实现html的功能，必须使用Writer输出对应的html,比较麻烦。而JSP的情况是Java和HTML可以组合成一个扩展名为.jsp的文件,做界面展示比较方便而嵌入逻辑比较复杂。

2.2.6 jsp有哪些内置对象?作用分别是什么?

9个内置的对象：

request 用户端请求，此请求会包含来自GET/POST请求的参数

response 网页传回用户端的回应

pageContext 网页的属性是在这里管理

session 与请求有关的会话期

application servlet正在执行的内容

out 用来传送回应的输出

config servlet的构架部件

page JSP网页本身

exception 针对错误网页，未捕捉的例外

四大作用域：pageContext request session application 可以通过jstl从四大作用域中取值.

Jsp传递值request session application cookie也能传值

2.2.7说一下session和cookie的区别？你在项目中都有哪些地方使用了?

Session和cookie都是会话(Seesion)跟踪技术。Cookie通过在客户端记录信息确定用户身份，Session通过在服务器端记录信息确定用户身份。但是Session的实现依赖于Cookie,sessionId(session的唯一标识需要存放在客户端).

cookie 和session 的区别：

1、cookie数据存放在客户的浏览器上，session数据放在服务器上。

2、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗 考虑到安全应当使用session。

3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能,考虑到减轻服务器性能方面，应当使用COOKIE。

4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

5、所以个人建议： 将登陆信息等重要信息存放为SESSION 其他信息如果需要保留，可以放在COOKIE中，比如购物车

购物车最好使用cookie，但是cookie是可以在客户端禁用的，这时候我们要使用cookie+数据库的方式实现，当从cookie中不能取出数据时，就从数据库获取。

2.2.8、 MVC的各个部分都有那些技术来实现?

M(Model) 模型 javabean

V(View) 视图 html jsp volicity freemaker

C(Control) 控制器 Servlet,Action

Jsp+Servlet+javabean 最经典mvc模式,实际上就是model2的实现方式，就是把视图和逻辑隔离开来

Model1的方式 jsp+service+dao

MOdel2的方式 jsp+servlet+service+dao

使用struts2和springmvc这样的mvc框架后，jsp+核心控制器+action+javabean

2.3数据库部分

2.3.1数据库的分类及常用的数据库

数据库分为：关系型数据库和非关系型数据库

关系型：mysql oracle sqlserver等

非关系型：redis,memcache,mogodb,hadoop等

2.3.2简单介绍一下关系数据库三范式?

范式就是规范,就是关系型数据库在设计表时，要遵循的三个规范。

要想满足第二范式必须先满足第一范式，要满足第三范式必须先满足第二范式。

第一范式（1NF）是指数据库表的每一列都是不可分割的基本数据项，同一列中不能有多个值，即实体中的某个属性不能有多个值或者不能有重复的属性。列数据的不可分割

二范式（2NF）要求数据库表中的每个行必须可以被唯一地区分。为实现区分通常需要为表加上一个列，以存储各个实例的唯一标识。（主键）

满足第三范式（3NF）必须先满足第二范式（2NF）。简而言之，第三范式（3NF）要求一个数据库表中不包含已在其它表中已包含的非主关键字信息。（外键）

反三范式,有的时候为了效率，可以设置重复或者可以推导出的字段.

订单（总价）和订单项（单价）

2.3.3事务四个基本特征或 ACID 特性。

事务是并发控制的单位，是用户定义的一个操作序列。这些操作要么都做，要么都不做，是一个不可分割的工作单位。

> 一个转账必须 A账号扣钱成功，B账号加钱成功，才算正真的转账成功。

事务必须满足四大特征:原子性,一致性,隔离性持久性/持续性

原子性：表示事务内操作不可分割。要么都成功、要么都是失败.

一致性：要么都成功、要么都是失败.后面的失败了要对前面的操作进行回滚。

隔离性：一个事务开始后，不能后其他事务干扰。

持久性/持续性：表示事务开始了，就不能终止。

2.3.4 mysql数据库的默认的最大连接数?

100

为什么需要最大连接数？特定服务器上面的数据库只能支持一定数目同时连接，这时候我们一般都会设置最大连接数（最多同时服务多少连接）。在数据库安装时都会有一个默认的最大连接数为100

```
# The maximum amount of concurrent sessions the MySQL server will
# allow. One of these connections will be reserved for a user with
# SUPER privileges to allow the administrator to login even if the
# connection limit has been reached.
max_connections=100
```

2.3.5说一下msyql的分页？Oracle的分页?

为什么需要分页？在很多数据是，不可能完全显示数据。进行分段显示.

Mysql是使用关键字limit来进行分页的 limit offset,size 表示从多少索引去多少位.

Oracle的分页，大部分情况下，我们是记不住了。说思路，要使用三层嵌套查询。

Oracle的分页有点儿记不住了，只记得一些大概。是使用了三层嵌套查询。如果在工作中使用了，可以到原来的项目中拷贝或上网查询。

mysql:

Stringsql =

"select* from students order by id limit " + pageSize*(pageNumber-1) +"," + pageSize;

oracle:

Stringsql =

"select * from " +

(select ,*rownum rid from (select * fromstudents order by postime desc) where rid*<=" + *pagesize*pagenumber +") as t" +

"where t>" +pageSize*(pageNumber-1);

2.3.6简单讲一下数据库的触发器的使用场景?

触发器，需要有触发条件，当条件满足以后做什么操作。

触发器用处还是很多的，比如校内网、开心网、Facebook，你发一个日志，自动通知好友，其实就是在增加日志时做一个后触发，再向通知表中写入条目。因为触发器效率高。而UCH没有用触发器，效率和数据处理能力都很低。

每插入一个帖子，都希望将版面表中的最后发帖时间，帖子总数字段进行同步更新，用触发器做效率就很高。

2.3.7 简单讲一下数据库的存储过程的使用场景?

数据库存储过程具有如下优点：

1、存储过程只在创建时进行编译，以后每次执行存储过程都不需再重新编译，而一般 SQL 语句每执行一次就编译一次，因此使用存储过程可以大大提高数据库执行速度。

2、通常，复杂的业务逻辑需要多条SQL 语句。这些语句要分别地从客户机发送到服务器，当客户机和服务器之间的操作很多时，将产生大量的网络传输。如果将这些操作放在一个存储过程中，那么客户机和服务器之间的网络传输就会大大减少，降低了网络负载。

3、存储过程创建一次便可以重复使用，从而可以减少数据库开发人员的工作量。

4、安全性高，存储过程可以屏蔽对底层数据库对象的直接访问，使用EXECUTE 权限调用存储过程，无需拥有访问底层数据库对象的显式权限。

正是由于存储过程的上述优点，目前常用的数据库都支持存储过程，例如 IBM DB2，Microsoft SQL Server，Oracle，Access 等，开源数据库系统 MySQL 也在 5.0 的时候实现了对存储过程的支持。

定义存储过程:

2.3.8 用jdbc怎么调用存储过程?

贾琏欲执事

加载驱动

获取连接

设置参数

执行

释放连接

```
package com.huawei.interview.lym;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Types;

public class JdbcTest {
```

```
/**
* @param args
*/
public static void main(String[]args) {
// TODO Auto-generated method stub
Connectioncn = null;
CallableStatementcstmt = null;
try {
//这里最好不要这么干，因为驱动名写死在程序中了
Class.forName("com.mysql.jdbc.Driver");
//实际项目中，这里应用DataSource数据，如果用框架，
//这个数据源不需要我们编码创建，我们只需Datasource ds = context.lookup()
//cn = ds.getConnection();
cn =DriverManager.getConnection("jdbc:mysql:///test","root","root");
cstmt =cn.prepareCall("{call insert_Student(?,?,?)}");
cstmt.registerOutParameter(3,Types.INTEGER);
cstmt.setString(1,"wangwu");
cstmt.setInt(2,25);
cstmt.execute();
//get第几个，不同的数据库不一样，建议不写
System.out.println(cstmt.getString(3));
} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
finally
{
/*try{cstmt.close();}catch(Exception e){}
try{cn.close();}catch(Exceptione){}*/
try {
if(cstmt != null)
cstmt.close();
```

if(cn != **null**)

cn.close();

} **catch** (SQLException e){

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

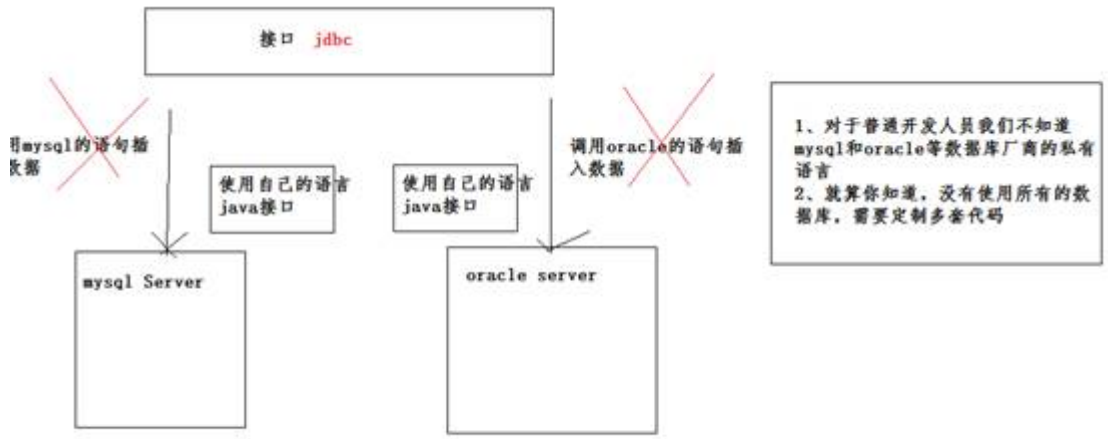
2.3.9常用SQL

略

2.3.10简单说一下你对jdbc的理解?

Java database connection java数据库连接.数据库管理系统(mysql oracle等)是很多，每个数据库管理系统支持的命令是不一样的。

Java只定义接口，让数据库厂商自己实现接口，对于我们者而言。只需要导入对应厂商开发的实现即可。然后以接口方式进行调用.(mysql + mysql驱动（实现）+jdbc)



2.3.11 写一个简单的jdbc的程序。写一个访问oracle数据的jdbc程序?

贾琏欲执事

加载驱动(com.mysql.jdbc.Driver,oracle.jdbc.driver.OracleDriver)

获取连接(DriverManager.getConnection(url,usernam,passord))

设置参数 Statement PreparedStatement

cstmt.setXXX(index, value);

执行 executeQuery executeUpdate

释放连接(是否连接要从小到大，必须放到finnaly)

2.3.12 JDBC中的PreparedStatement相比Statement的好处

大多数我们都使用PreparedStatement代替Statement

1：PreparedStatement是预编译的，比Statement速度快

2：代码的可读性和可维护性

虽然用PreparedStatement来代替Statement会使代码多出几行,但这样的代码无论从可读性还是可维护性上来说.都比直接用Statement的代码高很多档次：

stmt.executeUpdate("insert into tb_name(col1,col2,col2,col4)
values('"+var1+"','"+var2+"',"+var3+",'"+var4+"')");

perstmt = con.prepareStatement("insert intotb_name (col1,col2,col2,col4) values (?,?,?,?)");

perstmt.setString(1,var1);

perstmt.setString(2,var2);

perstmt.setString(3,var3);

perstmt.setString(4,var4);

perstmt.executeUpdate();

不用我多说,对于第一种方法，别说其他人去读你的代码，就是你自己过一段时间再去读,都会觉得伤心。

3：安全性

PreparedStatement可以防止SQL注入攻击，而Statement却不能。比如说：

String sql = "select * from tb_name where name='"+varname+"' and passwd='"+varpasswd+"'";

如果我们把[' or '1' ='1]作为varpasswd传入进来.用户名随意,看看会成为什么?

select * from tb_name = '随意' and passwd = ''or '1' = '1';

因为'1'='1'肯定成立，所以可以任何通过验证，更有甚者：

把[';drop table tb_name;]作为varpasswd传入进来,则：

select * from tb_name = '随意' and passwd ='';drop table tb_name;有些数据库是不会让你成功的，但也有很多数据库就可以使这些语句得到执行。

而如果你使用预编译语句你传入的任何内容就不会和原来的语句发生任何匹配的关系，只要全使用预编译语句你就用不着对传入的数据做任何过虑。而如果使用普通的statement,有可能要对drop等做费尽心机的判断和过虑。

2.3.13 数据库连接池作用

1、限定数据库的个数，不会导致由于数据库连接过多导致系统运行缓慢或崩溃

2、数据库连接不需要每次都去创建或销毁，节约了资源

3、数据库连接不需要每次都去创建，响应时间更快。

2.4 前端部分

2.4.1简单说一下html,css,javascript在网页开发中的定位?

HTML 超文本标记语言定义网页的结构

CSS 层叠样式表，用来美化页面

JavaScript主要用来验证表单，做动态交互(其中ajax)

2.4.2简单介绍一下Ajax?

什么是Ajax? 异步的javascript和xml

作用是什么？通过AJAX与服务器进行数据交换，AJAX可以使网页实现布局更新。

这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

怎么来实现Ajax XmlHttpRequest对象，使用这个对象可以异步向服务器发送请求，获取获取响应，完成局部更新。Open send responseText/responseXML 局部响应.

使用场景登陆失败时不跳转页面，注册时提示用户名是否存在,二级联动等等使用场景

2.4.3 js和jQuery的关系?

jQuery是一个js框架，封装了js的属性和方法。让用户使用起来更加便利。

,并且增强了js的功能.

使用原生js是要处理很多兼容性的问题(注册事件等)，由jQuery封装了底层，就不用处理兼容性问题。

原生的js的dom和事件绑定和Ajax等操作非常麻烦，jQuery封装以后操作非常方便。

2.4.4 jQuery的常用选择器?

ID选择器 通过ID获取一个元素

Class选择器通过类(css)获取元素

标签选择器 通过标签获取元素

通用选择器(*) 获取所以的元素

div.myCls 获取有myCls这个类的div

层次选择器

儿子选择器 > 获取下面的子元素

后代选择器空格 获取下面后代，包括儿子、孙子等后代

属性选择器

Tag[attrName='test'] 获取有属性名为xxxx并且属性的值为test的所有xxx标签

<input type="checkbox" />吃饭

<input type="checkbox" />睡觉

Input[name='hobby'],表示获取属性名为name并且name属性值为hobby的的所有input标签元素

2.4.5 jQuery的页面加载完毕事件?

很多时候我们需要获取元素，但是必须等到该元素被加载完成后才能获取。我们可以把js代码放到该元素的后面，但是这样就会造成js在我们的body中存在不好管理。所有页面加载完毕后所有的元素当然已经加载完毕。一般获取元素做操作都要在页面加载完毕后操作。

第一种：$(document).ready(function());(document)把原生的document这个dom对象转换为jQuery对象，转换完成后才能调用ready方法　ready(fn),表示的是页面结构被加载完毕后执行传入函数fn	**第二种：$(function(){});　当页面加载完毕后执行里面的函数,这一种相对简单，用得最多。**
window.onload的区别 1、jQuery中的页面加载完毕事件，表示的是页面结构被加载完毕。 2、window.onload 表示的是页面被加载完毕。　onload必须等等页面中的图片、声音、图像等远程资源被加载完毕后才调用而jQuery中只需要页面结构被加载完毕。	

2.4.6 Jquery的Ajax和原生Js实现Ajax有什么关系?

jQuery中的Ajax也是通过原生的js封装的。封装完成后让我们使用起来更加便利，不用考虑底层实现或兼容性等处理。

如果采用原生js实现Ajax是非常麻烦的，并且每次都是一样的。如果我们不使用jQuery我们也要封装Ajax对象的方法和属性。有像jQuery这些已经封装完成，并经过很多企业实际的框架，比较可靠并且开源。我们就不需要封装，直接使用成熟的框架(jQuery)即可.

2.4.7简单说一下html5?你对现在的那些新技术有了解?

Html5是最新版本的html,是在原来html4的基础上增强了一些标签。

Html增加一些像画板、声音、视频、web存储等高级功能。但是html5有一个不好的地方，那就是html5太强调语义了，导致开发中都不知道要选择那个标签。

在做页面布局是，无论头部、主题、导航等模块都使用div来表示，但是html5的规范，需要使用不同的标签来表示。(header footer等)

你对现在的那些新技术有了解

Html5css3等

2.4.8 简单说一下css3?

Css3是最新版本的css,是对原理css2的功能增强。

Css3中提供一些原来css2中实现起来比较困难或者不能实现的功能。

1、盒子圆角边框

2、盒子和文字的阴影

3、渐变

4、转换移动、缩放、旋转等

5、过渡、动画都可以使用动画。

6、可以使用媒体查询实现响应式网站。

Css3最大缺点就是要根据不同的浏览器处理兼容性。对应有一些处理兼容性的工具。不用担心.

2.4.9 bootstrap是什么?

BootStrap是一个移动设备优先的UI框架。我们可以不用谢任何css,js代码就能实现比较漂亮的有交互性的页面。我们程序员对页面的编写是有硬伤的，所有要自己写页面的话就要使用类似于bootstrap这样的UI框架。

平时用得很多的:

1、模态框

2、表单，表单项

3、布局

4、删格系统

2.5 框架部分

2.5.1什么是框架?

框架（Framework）是一个框子——指其约束性，也是一个架子——指其支撑性。

IT语境中的框架，特指为解决一个开放性问题而设计的具有一定约束性的支撑结构。在此结构上可以根据具体问题扩展、安插更多的组成部分，从而更迅速和方便地构建完整的解决问题的方案。

1）框架本身一般不完整到可以解决特定问题,但是可以帮助您快速解决特定问题;

没有框架所有的工作都从零开始做,有了框架,为我们提供了一定的功能,我们就可以在框 架的基础上开发,极大的解放了生产力。

不同的框架，是为了解决不同领域的问题。一定要为了解决问题才去学习框架。

2）框架天生就是为扩展而设计的;

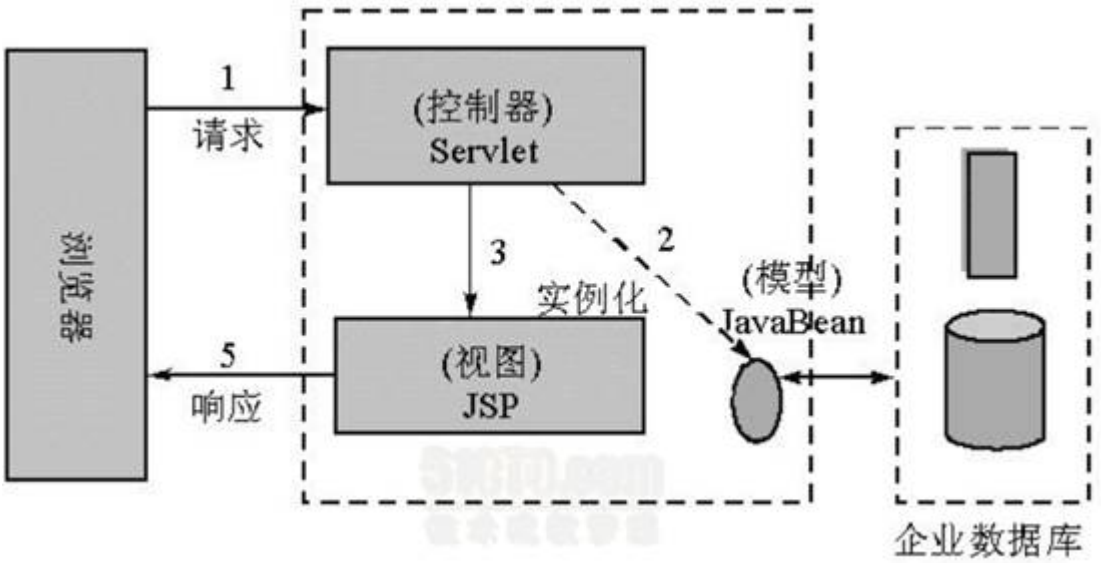
3）框架里面可以为后续扩展的组件提供很多辅助性、支撑性的方便易用的实用工具（utilities），也就是说框架时常配套了一些帮助解决某类问题的库（libraries）或工具（tools）。

java中就是一系列的jar包，其本质就是对jdk功能的扩展.

2.5.2 MVC模式

MVC全名是ModelView Controller，是模型(model) - 视图(view) - 控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。

最简单的、最经典就是Jsp(view) +Servlet(controller) + JavaBean(model)



1、当控制器收到来自用户的请求

2、控制器调用JavaBean完成业务

3、完成业务后通过控制器跳转JSP页面的方式给用户反馈信息

4、Jsp个 用户做出响应。

控制器都是核心

2.5.3 MVC框架

什么是MVC框架?

是为了解决传统MVC模式(Jsp + Servlet + JavaBean)的一些问题而出现的框架。

传统MVC模式问题

1、所有的Servlet和Servlet映射都要配置在web.xml中，如果项目太大，web.xml就太庞大，并且不能实现模块化管理。

2、Servlet的主要功能就是接受参数、调用逻辑、跳转页面，比如像其他字符编码、文件上传等功能也要写在Servlet中，不能让Servlet主要功能而需要做处理一下特例。

3、接受参数比较麻烦(String name = request.getParameter(“name”),User user=new Useruser.setName(name))，不能通过model接收，只能单个接收，接收完成后转换封装model.

4、跳转页面方式比较单一(forword,redirect),并且当我的页面名称发生改变时需要修改Servlet源代码.

现在比较常用的MVC框架有:

struts

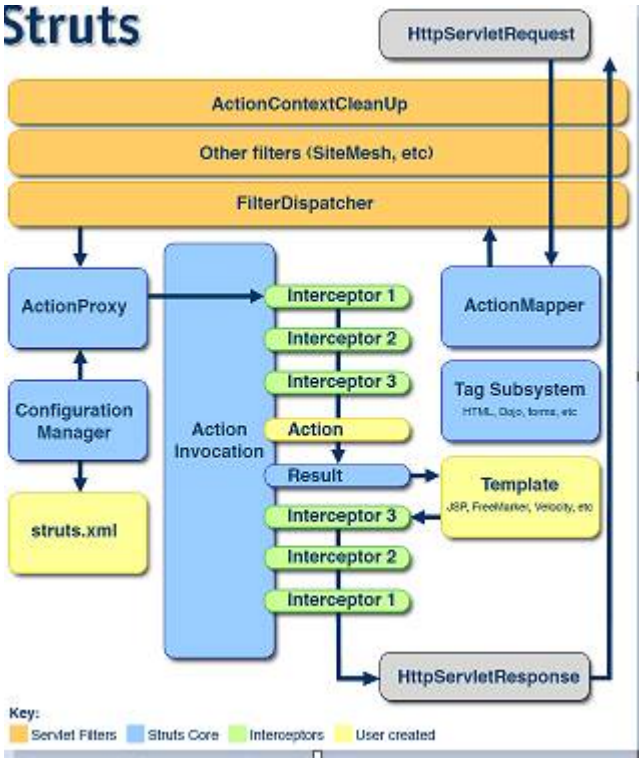
webwork

Struts2

Spring MVC

2.5.4 简单讲一下struts2的执行流程?

Struts2的原理?



一个请求在Struts2框架中的处理大概分为以下几个步骤：

1、客户端浏览器发送请求

2、这个请求经过一系列的过滤器（Filter）（这些过滤器中有一个叫做ActionContextCleanUp的可选过滤器，这个过滤器对于Struts2和其他框架的集成很有帮助，例如：SiteMesh Plugin）；

3、接着FilterDispatcher(StrutsPrepareAndExecuteFilter)被调用，FilterDispatcher

(StrutsPrepareAndExecuteFilter)询问ActionMapper来决定这个请求是否需要调用某个Action；

4、如果ActionMapper决定需要调用某个Action，FilterDispatcher

(StrutsPrepareAndExecuteFilter)把请求的处理交给ActionProxy；

5、ActionProxy通过Configuration Manager询问框架的配置文件，找到需要调用的Action类；

6、ActionProxy创建一个ActionInvocation的实例。

7、ActionInvocation实例使用命名模式来调用，在调用Action的过程前后，涉及到相关拦截器（Intercepter）的调用。

8、一旦Action执行完毕，ActionInvocation负责根据struts.xml中的配置找到对应的返回结果。返回结果通常是（但不总是，也可能是另外的一个Action链）一个需要被表示的JSP或者FreeMarker的模版。在表示的过程中可以使用Struts2框架中继承的标签。在这个过程中需要涉及到ActionMapper。

面试：

1、浏览器发送请求，经过一系列的过滤器后，到达核心过滤器(StrutsPrepareAndExecuteFilter).

2、StrutsPrepareAndExecuteFilter通过ActionMapper判断当前的请求是否需要某个Action处理,如果不需要，则走原来的流程。如果需要则把请求交给ActionProxy来处理

3、ActionProxy通过Configuration Manager询问框架的配置文件(Struts.xml)，找到需要调用的Action类；

4、创建一个ActionInvocation实例，来调用Action的对应方法来获取结果集的name,在调用前后会执行相关拦截器。

5、通过结果集的Name知道对应的结果集来对浏览器进行响应。

拦截、判断、寻找、执行、响应

2.5.5 Struts2中的拦截器，你都用它干什么?

java里的拦截器是动态拦截Action调用的对象。它提供了一种机制可以使开发者可以定义在一个action执行的前后执行的代码，也可以在一个action执行前阻止其执行，同时也提供了一种可以提取action中可重用部分的方式。

在AOP（Aspect-Oriented Programming）中拦截器用于在某个方法或字段被访问之前，进行拦截然后在之前或之后加入某些操作。

面试：

struts2中的的功能（参数处理、文件上传、字符编码等）都是通过系统拦截器实现的。

如果业务需要，当然我们也可以自定义拦截器,进行可插拔配置，在执行Action的方法前后、加入相关逻辑完成业务。

使用场景：

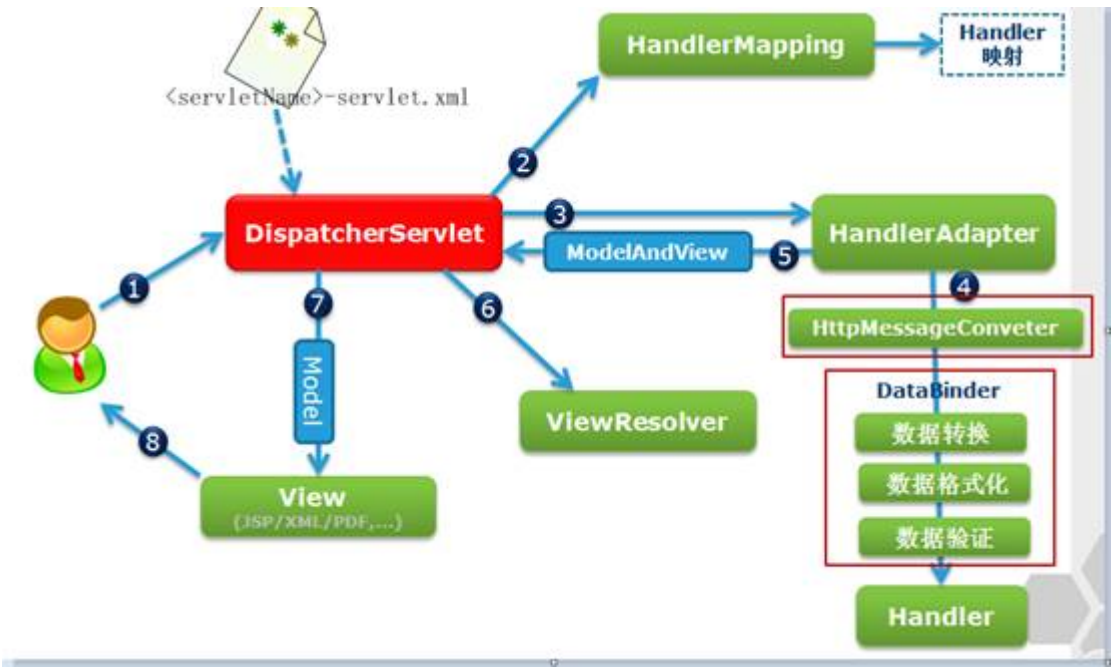
1、用户登录判断，在执行Action的前面判断是否已经登录，如果没有登录的跳转到登录页面。

2、用户权限判断，在执行Action的前面判断是否具有，如果没有权限就给出提示信息。

3、操作日志……

4、……

2.5.6 简单讲一下SpringMVC的执行流程?



1.用户向服务器发送请求，请求被Spring 前端控制Servelt DispatcherServlet捕获；

2. DispatcherServlet对请求URL进行解析，得到请求资源标识符（URI）。然后根据该URI，调用HandlerMapping获得该Handler配置的所有相关的对象（包括Handler对象以及Handler对象对应的拦截器），最后以HandlerExecutionChain对象的形式返回；

3. DispatcherServlet 根据获得的Handler，选择一个合适的HandlerAdapter。（**附注**：如果成功获得HandlerAdapter后，此时将开始执行拦截器的preHandler(...)方法）

4. 提取Request中的模型数据，填充Handler入参，开始执行Handler（Controller)。 在填充Handler的入参过程中，根据你的配置，Spring将帮你做一些额外的工作：

HttpMessageConveter： 将请求消息（如Json、xml等数据）转换成一个对象，将对象转换为指定的响应信息

数据转换：对请求消息进行数据转换。如String转换成Integer、Double等

数据根式化：对请求消息进行数据格式化。如将字符串转换成格式化数字或格式化日期等

数据验证： 验证数据的有效性（长度、格式等），验证结果存储到BindingResult或Error中

5. Handler执行完成后，向DispatcherServlet 返回一个ModelAndView对象；

6. 根据返回的ModelAndView，选择一个适合的ViewResolver（必须是已经注册到Spring容器中的ViewResolver)返回给DispatcherServlet ；

7. ViewResolver 结合Model和View，来渲染视图

8. 将渲染结果返回给客户端。

面试：

1、 用户向服务器发送请求，请求被Spring 前端控制ServeltDispatcherServlet捕获(捕获)

2、 DispatcherServlet对请求URL进行解析，得到请求资源标识符（URI）。然后根据该URI，调用HandlerMapping获得该Handler配置的所有相关的对象（包括Handler对象以及Handler对象对应的拦截器），最后以HandlerExecutionChain对象的形式返回；(查找handler)

3、 DispatcherServlet 根据获得的Handler，选择一个合适的HandlerAdapter。 提取Request中的模型数据，填充Handler入参，开始执行Handler（Controller), Handler执行完成后，向DispatcherServlet 返回一个ModelAndView对象(执行handler)

4、DispatcherServlet 根据返回的ModelAndView，选择一个适合的ViewResolver（必须是已经注册到Spring容器中的ViewResolver) (选择ViewResolver)

5、通过ViewResolver 结合Model和View，来渲染视图,DispatcherServlet 将渲染结果返回给客户端。（渲染返回）

快速记忆技巧：

核心控制器捕获请求、查找Handler、执行Handler、选择ViewResolver,通过ViewResolver渲染视图并返回

2.5.7 说一下struts2和springMVC有什么不同?

目前企业中使用SpringMvc的比例已经远远超过Struts2,那么两者到底有什么区别，是很多初学者比较关注的问题，下面我们就来对SpringMvc和Struts2进行各方面的比较:

\1. 核 心控制器（前端控制器、预处理控制器）：对于使用过mvc框架的人来说这个词应该不会陌生，核心控制器的主要用途是处理所有的请求，然后对那些特殊的请求 （控制器）统一的进行处理(字符编码、文件上传、参数接受、异常处理等等),spring mvc核心控制器是Servlet，而Struts2是Filter。

2.控制器实例：Spring Mvc会比Struts快一些（理论上）。Spring Mvc是基于方法设计，而Sturts是基于对象，每次发一次请求都会实例一个action，每个action都会被注入 属性，而Spring更像Servlet一样，只有一个实例，每次请求执行对应的方法即可(注意：由于是单例实例，所以应当避免全局变量的修改，这样会产生线程安全问题)。

\3. 管理方式：大部分的公司的核心架构中，就会使用到spring,而spring mvc又是spring中的一个模块，所以spring对于spring mvc的控制器管理更加简单方便，而且提供了全注解方式进行管理，各种功能的注解都比较全面，使用简单，而struts2需要采用XML很多的配置参数来管理（虽然也可以采用注解，但是几乎没有公司那样使用）。

4.参数传递：Struts2中自身提供多种参数接受，其实都是通过（ValueStack）进行传递和赋值，而SpringMvc是通过方法的参数进行接收。

5.学习难度：Struts更加很多新的技术点，比如拦截器、值栈及OGNL表达式，学习成本较高，springmvc 比较简单，很较少的时间都能上手。

6.intercepter 的实现机制：struts有以自己的interceptor机制，spring mvc用的是独立的AOP方式。这样导致struts的配置文件量还是比spring mvc大，虽然struts的配置能继承，所以我觉得论使用上来讲，spring mvc使用更加简洁，开发效率Spring MVC确实比struts2高。spring mvc是方法级别的拦截，一个方法对应一个request上下文，而方法同时又跟一个url对应，所以说从架构本身上spring3 mvc就容易实现restful url。struts2是类级别的拦截，一个类对应一个request上下文；实现restful url要费劲，因为struts2 action的一个方法可以对应一个url；而其类属性却被所有方法共享，这也就无法用注解或其他方式标识其所属方法了。spring3 mvc的方法之间基本上独立的，独享request response数据，请求数据通过参数获取，处理结果通过ModelMap交回给框架方法之间不共享变量，而struts2搞的就比较乱，虽然方法之间也是独立的，但其所有Action变量是共享的，这不会影响程序运行，却给我们编码，读程序时带来麻烦。

7.spring mvc处理ajax请求,直接通过返回数据，方法中使用注解@ResponseBody，spring mvc自动帮我们对象转换为JSON数据。而struts2是通过插件的方式进行处理

在SpringMVC流行起来之前，Struts2在MVC框架中占核心地位，随着SpringMVC的出现，SpringMVC慢慢的取代struts2,但是很多企业都是原来搭建的框架，使用Struts2较多。

2.5.8 说一下Spring中的两大核心?

Spring是什么?

spring是J2EE应用程序框架，是轻量级的IoC和AOP的容器框架(相对于重量级的EJB)，主要是针对javaBean的生命周期进行管理的轻量级容器，可以单独使用，也可以和Struts框架，ibatis框架等组合使用。

1、IOC(Inversionof Control)或DI(Dependency Injection)

IOC控制权反转

原来：我的Service需要调用DAO，Service就需要创建DAO

Spring:Spring发现你Service依赖于dao,就给你注入.

核心原理：就是配置文件+反射(工厂也可以)+容器(map)

2、AOP:面向切面编程

核心原理：使用动态代理的设计模式在执行方法前后或出现异常做加入相关逻辑。

我们主要使用AOP来做：

1、事务处理

2、权限判断

3、日志

4、....

2.5.9 AOP是什么？你都拿它做什么？

3、AOP:面向切面编程

核心原理：使用动态代理的设计模式在执行方法前后或出现异常做加入相关逻辑。

我们主要使用AOP来做：

1、事务处理 执行方法前，开启事务、执行完成后关闭事务、出现异常后回滚事务

2、权限判断 在执行方法前，判断是否具有权限

3、日志 在执行前进行日志处理

4、....

2.5.10讲一下Spring的事务传播特性

多个事务存在是怎么处理的策略

\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，支持当前事务。如果没有事务，则非事务的执行 \3. PROPAGATION_MANDATORY: 如果已经存在一个事务，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: 总是开启一个新的事务。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: 总是非事务地执行，并挂起任何存在的事务。 \6. PROPAGATION_NEVER: 总是非事务地执行，如果存在一个活动事务，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 属性执行

Propagation

Required 需要 如果存在一个事务，则支持当前事务。如果没有事务则开启

Supports 支持 如果存在一个事务，支持当前事务。如果没有事务，则非事务的执行

Mandatory 必要的 如果已经存在一个事务，支持当前事务。如果没有一个活动的事务，则抛出异常。

required_new 总是开启一个新的事务。如果一个事务已经存在，则将这个存在的事务挂起。

Not_support 总是非事务地执行，并挂起任何存在的事务。

Never 绝不 总是非事务地执行，如果存在一个活动事务，则抛出异常

Nested 嵌套的 如果有就嵌套、没有就开启事务

2.5.11 Spring事务的隔离级别1. ISOLATION_DEFAULT： 这是一个PlatfromTransactionManager默认的隔离级别，使用数据库默认的事务隔离级别.

另外四个与JDBC的隔离级别相对应 \2. ISOLATION_READ_UNCOMMITTED： 这是事务最低的隔离级别，它充许令外一个事务可以看到这个事务未提交的数据。 这种隔离级别会产生脏读，不可重复读和幻像读。 \3. ISOLATION_READ_COMMITTED： 保证一个事务修改的数据提交后才能被另外一个事务读取。另外一个事务不能读取该事务未提交的数据 \4. ISOLATION_REPEATABLE_READ： 这种事务隔离级别可以防止脏读，不可重复读。但是可能出现幻像读。 它除了保证一个事务不能读取另一个事务未提交的数据外，还保证了避免下面的情况产生(不可重复读)。 \5. ISOLATION_SERIALIZABLE 这是花费最高代价但是最可靠的事务隔离级别。事务被处理为顺序执行。

除了防止脏读，不可重复读外，还避免了幻像读。

其中的一些概念的说明：

** 脏读:** 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: 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。

** 幻觉读:** 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发现表中还有没有修改的数据行，就好象发生了幻觉一样。

2.5.12 什么是ORM?

对象关系映射（Object RelationalMapping，简称ORM）模式是一种为了解决面向**对象**与**关系**数据库存在的互不匹配的现象的技术。简单的说，ORM是通过使用描述对象和数据库之间映射的元数据，将程序中的对象自动持久化到关系数据库中。那么，到底如何实现持久化呢？一种简单的方案是采用硬编码方式(jdbc操作sql方式)，为每一种可能的数据库访问操作提供单独的方法。

这种方案存在以下不足：

1.持久化层缺乏弹性。一旦出现业务需求的变更，就必须修改持久化层的接口

2.持久化层同时与域模型与关系数据库模型绑定，不管域模型还是关系数据库模型发生变化，都要修改持久化曾的相关程序代码，增加了软件的维护难度。

ORM提供了实现持久化层的另一种模式，它采用映射元数据来描述对象关系的映射，使得ORM中间件能在任何一个应用的业务逻辑层和数据库层之间充当桥梁。Java典型的ORM框架
有:Hibernate,ibatis(mybatis),speedframework。

ORM的方法论基于三个核心原则：

简单：以最基本的形式建模数据。

传达性：数据库结构被任何人都能理解的语言文档化。

精确性：基于数据模型创建正确标准化了的结构。

对象关系映射（Object RelationalMapping，简称ORM）模式是一种为了解决面向**对象**与**关系**数据库存在的互不匹配的现象的技术。可以简单的方案是采用硬编码方式(jdbc操作sql方式)，为每一种可能的数据库访问操作提供单独的方法。这种方法存在很多缺陷，使用

使用ORM框架(为了解决解决面向**对象**与**关系**数据库存在的互不匹配的现象的框架)来解决.

Hibernate,ibatis(mybatis),

2.5.13 iBatis(mybatis)与Hibernate有什么不同?

相同点：

都是java中orm框架、屏蔽jdbc api的底层访问细节，使用我们不用与jdbc api打交道，就可以完成对数据库的持久化操作。jdbc api编程流程固定，还将sql语句与java代码混杂在了一起，经常需要拼凑sql语句，细节很繁琐。

ibatis的好处：屏蔽jdbc api的底层访问细节；将sql语句与java代码进行分离;提供了将结果集自动封装称为实体对象和对象的集合的功能.queryForList返回对象集合，用queryForObject返回单个对象；提供了自动将实体对象的属性传递给sql语句的参数。

Hibername的好处：Hibernate是一个全自动的orm映射工具，它可以自动生成sql语句，并执行并返回java结果。

不同点：

1、hibernate要比ibatis功能强大很多。因为hibernate自动生成sql语句。

2、ibatis需要我们自己在xml配置文件中写sql语句，hibernate我们无法直接控制该语句，我们就无法去写特定的高效率的sql。对于一些不太复杂的sql查询，hibernate可以很好帮我们完成，但是，对于特别复杂的查询，hibernate就很难适应了，这时候用ibatis就是不错的选择，因为ibatis还是由我们自己写sql语句。

ibatis可以出来复杂语句，而hibernate不能。

3、ibatis要比hibernate简单的多。ibatis是面向sql的，不同考虑对象间一些复杂的映射关系。

2.5.14 Hibernate映射对象的状态

临时状态/瞬时状态(transient)：刚刚用new语句创建，没有被持久化

不处于session中(没有使用session的方法去操作临时对象)。该对象成为临时对象

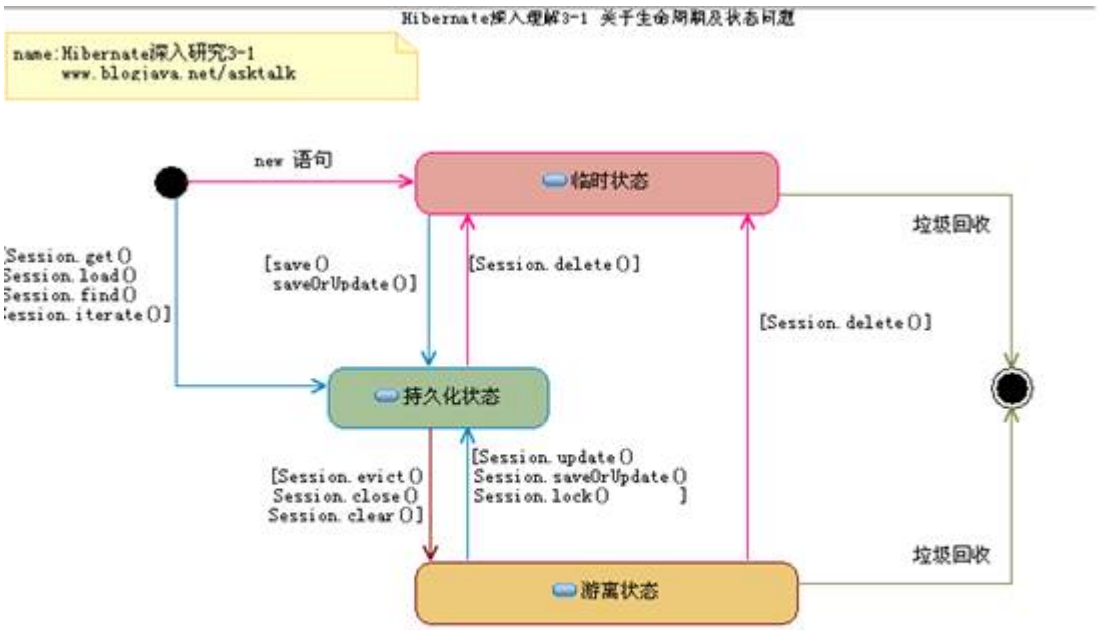
持久化状态/托管状态(persistent)：已经被持久化，加入到session的缓存中。session是没有关闭该状态的对象为持久化对象。

游离状态/脱管状态(detached)：已经被持久化，但不处于session中。

该状态的对象为游离对象。

删除状态(removed)：对象有关联的ID，并且在Session管理下，但是已经被计划(事务提交的时候,commit())删除。如果没有事务就不能删除

相互转换



2.5.15 介绍一下Hibernate的缓存?

一、why（为什么要用Hibernate缓存？）

Hibernate是一个持久层框架，经常访问物理数据库。

为了降低应用程序对物理数据源访问的频次，从而提高应用程序的运行性能。

缓存内的数据是对物理数据源中的数据的复制，应用程序在运行时从缓存读写数据，在特定的时刻或事件会同步缓存和物理数据源的数据。

为了提供访问速度，把磁盘或数据库访问变成内存访问。

二、what（Hibernate缓存原理是怎样的？）Hibernate缓存包括两大类：Hibernate一级缓存和Hibernate二级缓存。

1.Hibernate一级缓存又称为“Session的缓存”。

Session缓存内置不能被卸载，Session的缓存是事务范围的缓存（Session对象的生命周期通常对应一个数据库事务或者一个应用事务）。

一级缓存中，持久化类的每个实例都具有唯一的OID。

2.Hibernate二级缓存又称为“SessionFactory的缓存”。

由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。

第二级缓存是可选的，是一个可配置的插件，默认下SessionFactory不会启用这个插件。

Hibernate提供了org.hibernate.cache.CacheProvider接口,它充当缓存插件与Hibernate之间的适配器。

面试：

Hibernate中的缓存分一级缓存和二级缓存。

一级缓存就是Session级别的缓存，在事务范围内有效是,内置的不能被卸载。二级缓存是SesionFactory级别的缓存，从应用启动到应用结束有效。是可选的，默认没有二级缓存，需要手动开启。

保存数据库后，在内存中保存一份，如果更新了数据库就要同步更新。

什么样的数据适合存放到第二级缓存中?

1) 很少被修改的数据　帖子的最后回复时间

2) 经常被查询的数据 电商的地点

2) 不是很重要的数据，允许出现偶尔并发的数据

3) 不会被并发访问的数据

4) 常量数据

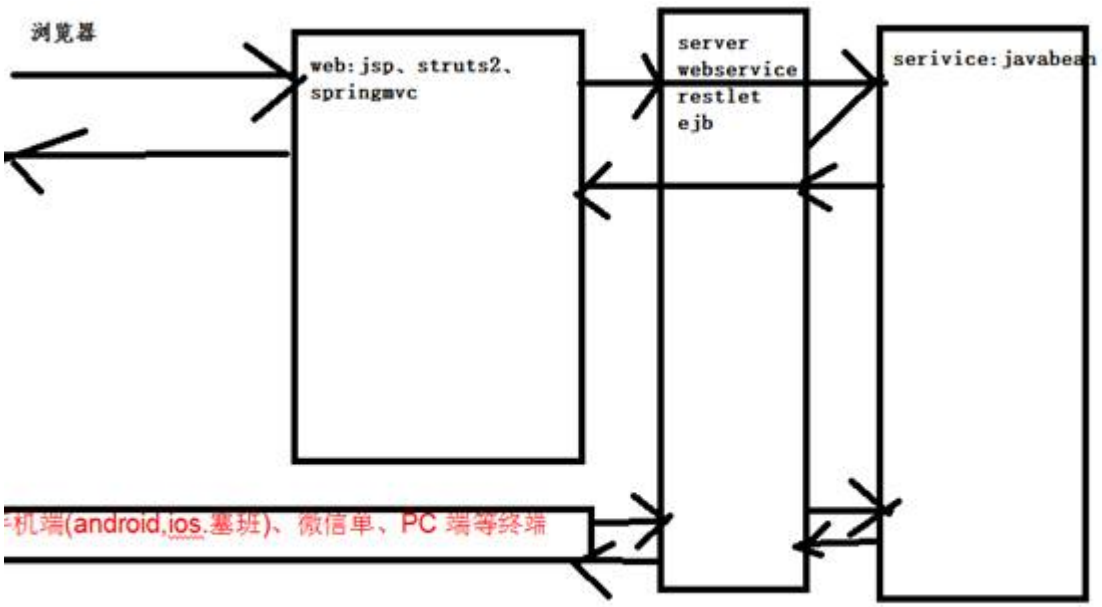
扩展：hibernate的二级缓存默认是不支持分布式缓存的。使用memcahe,redis等中央缓存来代替二级缓存。

2.5.16 简单讲一下webservice使用的场景?

webservice是一个SOA（面向服务的编程）的架构，它是不依赖于语言，不依赖于平台，可以实现不同的语言间的相互调用，通过Internet进行基于Http协议的网络应用间的交互。

1、异构系统(不同语言)的整合

2、不同客户端的整合 浏览器、手机端(android,ios.塞班)、微信单、PC端等终端来访问



3、实实在在的列子:

天气预报：可以通过实现webservice客户端调用远程天气服务实现的。

单点登录：一个服务是所有系统的登录

2.5.17 简单介绍一下activiti?

Activiti是一个业务流程管理(BPM)和工作流系统，适用于开发人员和系统管理员。其核心是超快速，稳定的BPMN2流程引擎。它易于与 Spring集成使用。

主要要在OA中，把线下流程放到线上。把现实生活中一些流程固话定义到系统中，然后通过输入表单数据完成业务。

他可用在OA系统的流程管理中：

请假流程 小于三天，一级主管审批，大于三天二级才能审批。

报销流程 1000 2000 3000>....

如果你想找专门这方面的工作，要下去复习。

2.6 高级部分

2.6.1 有没有用过linux?你都用它来做什么?

Linux是一个长时间运行比较稳定的操作系统，所有我们一般会拿它作为服务器(web,db,app等)。

Linux本身具有C的编译环境、我们的一些软件是没有软件包(redis、nginx等)的，需要在Linux的C编译环境编译得到软件包.

2.6.2 说一下linux下面的一下常用命令?

常用：

Pwd 获取当前路径

Cd 跳转到目录

Su -u 切换到管理员

Ls ls 列举目录

文件操作命令：

文件

tail 查看

rm -rf

vi

文件夹

mkdir

rm -r

2.6.3 你是使用什么来连接远程的Linux服务器的?

需要依赖于Linux服务器安装ssh服务端，一般这个ssh服务的端口22.

需要依赖于Linux服务器安装sftp服务端，一般这个sftp服务的端口25.

使用ssh客户端连接linux服务器，就有点儿像windows下面的远程连接。但是linux通过ssh连接上以后是没有图形界面，全是命令行。

Putty

Xshell

使用sftp客户端来连接sftp服务端，来上传和下载文件.(上传安装包，修改了配置文件上传。)

Winscp

xftp

企业中常用的两种组合:

putty+winscp

Xshell+xftp=xmanager

面试：使用xshell、putty等ssh客户端来连接服务器，使用xftp、winscp等sftp客户端来上传和现在文件。连接和上传、下载必须依赖于服务器的ssh、sftp服务，也就是linux服务器需要启动这两个服务。

2.6.4 有没有使用过云主机?

使用过，在原来的公司，我们没有使用自己的服务器，而是租用阿里的云主机。

没有使用过，但是有所了解。

云主机就是一些云服务运营商(阿里、华为、西部数码、新浪等)，提供的远程的服务器功能，我们开发者或者企业只需按需付费就可以租用对应的服务器。

使用ssh和sftp来进行操作。

2.6.5 有没有做过数据库优化方面的事情?

做过mysql数据库的优化、其他数据库类似

定位：查找、定位慢查询

优化手段：

a) 创建索引:创建合适的索引，我们就可以现在索引中查询，查询到以后直接找对应的记录。

b) 分表 ： 当一张表的数据比较多或者一张表的某些字段的值比较多并且很少使用时，采用水平分表和垂直分表来优化

c) 读写分离：当一台服务器不能满足需求时，采用读写分离的方式进行集群。

d) 缓存:使用redis来进行缓存

e) 一些常用优化技巧

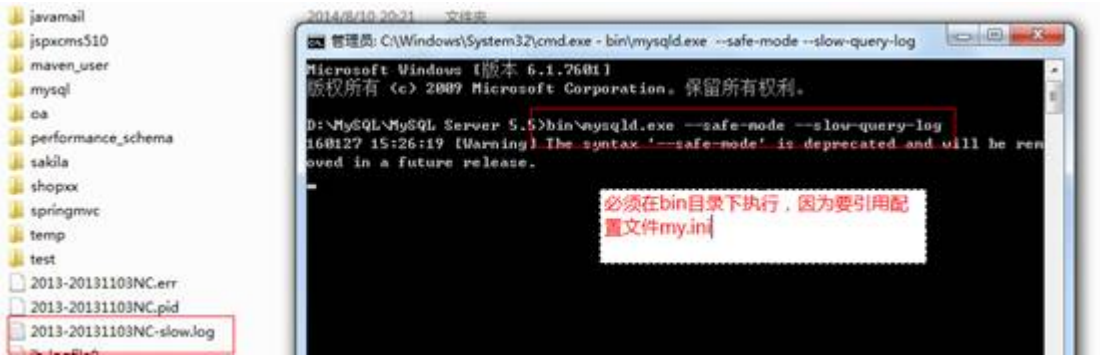
2.6.6 查找慢查询并定位慢查询?

在项目自验项目转测试之前，在启动mysql数据库时开启慢查询，并且把执行慢的语句写到日志中，在运行一定时间后。通过查看日志找到慢查询语句。

要找出项目中的慢Sql时

1、关闭数据库服务器(关闭服务)

2、把慢查询记录到日志中



3、设置慢查询时间

4、找出日志中的慢查询SQL

使用explain 慢查询语句，来详细分析语句的问题.

2.6.6 数据库优化之遵循范式?

数据库表设计时需要遵循方式

表的范式，是首先符合1NF, 才能满足2NF , 进一步满足3NF

1NF: 即表的列的具有原子性,不可再分解，即列的信息，不能分解.只要数据库是关系型数据库(mysql/oracle/db2/sysbase/sql server)，就自动的满足1NF.关系型数据库中是不允许分割列的。

2NF:表中的记录是唯一的.通常我们设计一个主键来实现

3NF:即表中不要有冗余数据, 就是说，表的信息，如果能够被推导出来，就不应该单独的设计一个字段来存放.(外键)

反3NF :没有冗余的数据库未必是最好的数据库，有时为了提高运行效率，就必须降低范式标准，适当保留冗余数据。具体做法是： 在概念数据模型设计时遵守第三范式，降低范式标准的工作放到物理数据模型设计时考虑。降低范式就是增加字段，允许冗余。订单和订单项、相册浏览次数和照片的浏览次数

2.6.7 选择合适的存储引擎

在开发中，我们经常使用的存储引擎 myisam /innodb/ memory

MyISAM存储引擎

如果表对事务要求不高，同时是以查询和添加为主的，我们考虑使用myisam存储引擎. 比如 bbs 中的 发帖表，回复表.

INNODB存储引擎:

对事务要求高，保存的数据都是重要数据，我们建议使用INNODB,比如订单表，账号表.

Memory 存储

我们数据变化频繁，不需要入库，同时又频繁的查询和修改，我们考虑使用memory, 速度极快.

问 MyISAM 和 INNODB的区别(主要)

\1. 事务安全 myisam不支持事务而innodb支持

\2. 查询和添加速度 myisam不用支持事务就不用考虑同步锁，查找和添加和添加的速度快

\3. 支持全文索引 myisam支持innodb不支持

\4. 锁机制 myisam支持表锁而innodb支持行锁(事务)

\5. 外键 MyISAM 不支持外键， INNODB支持外键. (通常不设置外键，通常是在程序中保证数据的一致)

特点	Myisan	InnoDB	BDB	Memory	Archive
批量插入的速度	高	低	高	高	非常高
事务安全		支持	支持		
全文索引	支持				
锁机制	表锁	行锁	页锁	表锁	行锁
存储限制	没有	64TB	没有	有	没有
B树索引	支持	支持	支持	支持	
哈希索引		支持		支持	
集群索引		支持			
数据缓存		支持		支持	
索引缓存	支持	支持		支持	
数据可压缩	支持				支持
空间使用	低	高	低	N/A	非常低
内存使用	低	高	低	中等	低
支持外键		支持			

2.6.8 数据库优化之创建合适的索引?

索引（Index）是帮助DBMS高效获取数据的数据结构。

分类：普通索引/唯一索引/主键索引/全文索引

普通索引:允许重复的值出现

唯一索引:除了不能有重复的记录外，其它和普通索引一样(用户名、用户身份证、email,tel)

主键索引：是随着设定主键而创建的，也就是把某个列设为主键的时候，数据库就会給改列创建索引。这就是主键索引.唯一且没有null值

全文索引:用来对表中的文本域(char，varchar，text)进行索引， 全文索引针对MyIsam

explain select * from articles where match(title,body)against('database');【会使用全文索引】

2.6.9 索引使用小技巧? *

索引弊端

1.占用磁盘空间。

2.对dml(插入、修改、删除)操作有影响，变慢。

使用场景：

a: 肯定在where条件经常使用,如果不做查询就没有意义

b: 该字段的内容不是唯一的几个值(sex)

c: 字段内容不是频繁变化.

具体技巧：

\1. 对于创建的多列索引（复合索引），不是使用的第一部分就不会使用索引。

alter table dept add index my_ind (dname,loc); // dname 左边的列,loc就是右边的列

explain select * from dept where dname='aaa'\G 会使用到索引

explain select * from dept where loc='aaa'\G 就不会使用到索引

\2. 对于使用like的查询，查询如果是'%aaa'不会使用到索引而'aaa%'会使用到索引。

explainselect * from dept where dname like '%aaa'\G不能使用索引

explainselect * from dept where dname like 'aaa%'\G使用索引.

所以在like查询时，'关键字'的最前面不能使用 % 或者 _这样的字符.，如果一定要前面有变化的值，则考虑使用全文索引->sphinx.

\3. 如果条件中有or，有条件没有使用索引,即使其中有条件带索引也不会使用。换言之，就是要求使用的所有字段,都必须单独使用时能使用索引.

\4. 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。否则不使用索引。

expain select * from dept wheredname='111';

expain select * from dept wheredname=111;（数值自动转字符串）

expain select * from dept wheredname=qqq;报错

也就是，如果列是字符串类型，无论是不是字符串数字就一定要用" 把它包括起来.

\5. 如果mysql估计使用全表扫描要比使用索引快，则不使用索引。

表里面只有一条记录

2.6.10 数据库优化之分表?

分表分为水平(按行)分表和垂直(按列)分表

根据经验，Mysql表数据一般达到百万级别，查询效率会很低，容易造成表锁，甚至堆积很多连接，直接挂掉；**水平分表**能够很大程度较少这些压力。

按行数据进行分表。

如果一张表中某个字段值非常多(长文本、二进制等)，而且只有在很少的情况下会查询。这时候就可以把字段多个单独放到一个表，通过外键关联起来。

考试详情，一般我们只关注分数，不关注详情。

水平分表策略：

1.按时间分表

这种分表方式有一定的局限性，当数据有较强的实效性，如微博发送记录、微信消息记录等，这种数据很少有用户会查询几个月前的数据，如就可以按月分表。

2.按区间范围分表

一般在有严格的自增id需求上，如按照user_id水平分表：

table_1 user_id从1~100w

table_2 user_id从101~200w

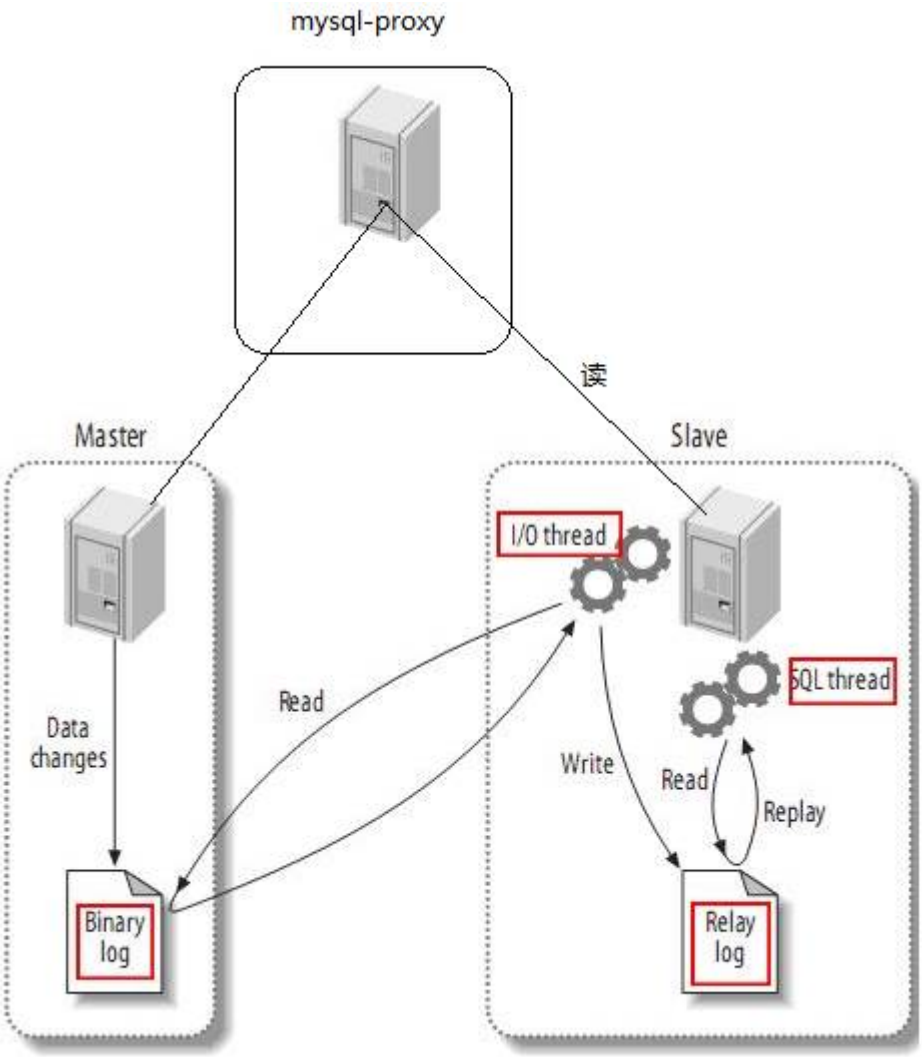
table_3 user_id从201~300w

3.hash分表*

通过一个原始目标的ID或者名称通过一定的hash算法计算出数据存储表的表名，然后访问相应的表。

2.6.11 数据库优化之读写分离

一台数据库支持的最大并发连接数是有限的，如果用户并发访问太多。一台服务器满足不要要求是就可以集群处理。Mysql的集群处理技术最常用的就是读写分离。



主从同步

数据库最终会把数据持久化到磁盘，如果集群必须确保每个数据库服务器的数据是一直的。**能改变数据库数据的操作都往主数据库去写，而其他的数据库从主数据库上同步数据。******

读写分离

使用负载均衡来实现写的操作都往主数据去，而读的操作往从服务器去。

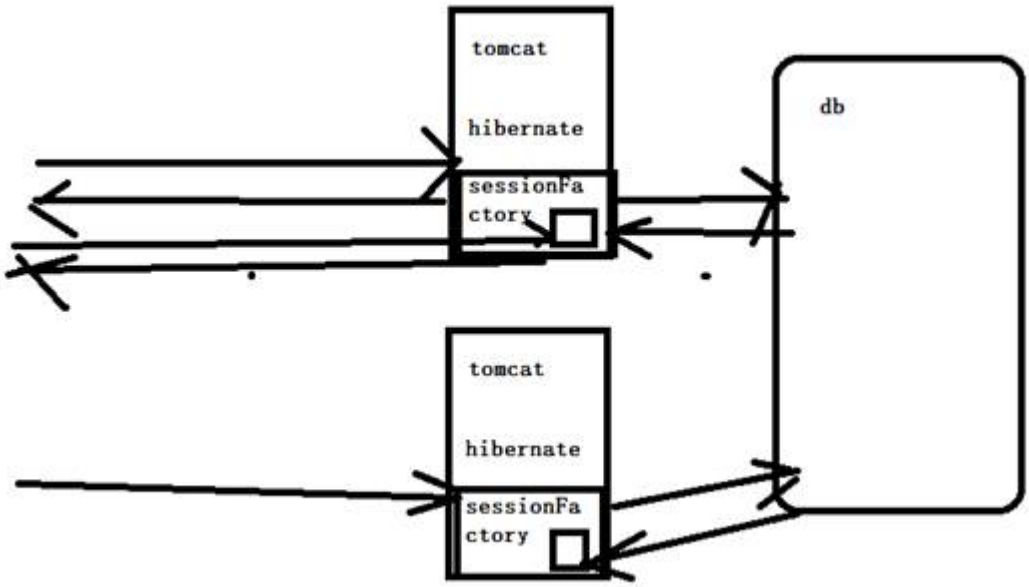
2.6.12 数据库优化之缓存

在持久层(dao)和数据库(db)之间添加一个缓存层，如果用户访问的数据已经缓存起来时，在用户访问时直接从缓存中获取，不用访问数据库。而缓存是在操作内存级，访问速度快。

作用：减少数据库服务器压力，减少访问时间。

Java中常用的缓存有，

1、hibernate的二级缓存。该缓存不能完成分布式缓存。



2、可以使用redis(memcahe等)来作为中央缓存。

对缓存的数据进行集中处理

2.6.13 语句优化小技巧

DDL优化:

1 、通过禁用索引来提供导入数据性能 。 这个操作主要针对有数据库的表，追加数据

//去除键

alter table test3 DISABLE keys;

//批量插入数据

insert into test3 select * from test;

//恢复键

alter table test3 ENABLE keys;

2、 关闭唯一校验

set unique_checks=0 关闭

set unique_checks=1 开启

3、修改事务提交方式(导入)（变多次提交为一次）

set autocommit=0 关闭

//批量插入

set autocommit=1 开启

DML优化（变多次提交为一次）

insert into test values(1,2);

insert into test values(1,3);

insert into test values(1,4);

//合并多条为一条

insert into test values(1,2),(1,3),(1,4)

DQL优化

Order by优化

1、多用索引排序

2、普通结果排序（非索引排序）Filesort

group by优化

是使用order by null,取消默认排序

子查询优化

在客户列表找到不在支付列表的客户

#在客户列表找到不在“支付列表”的客户，查询没买过东西的客户

explain

select * from customer wherecustomer_id not in (select DISTINCT customer_idfrom payment); #子查询 -- 这种是基于func外链

explain

select * from customer c leftjoin payment p on(c.customer_id=p.customer_id)where p.customer_id is null -- 这种是基于“索引”外链

Or优化

在两个独立索引上使用or的性能优于

1、 or两边都是用索引字段做判断，性能好！！

2、 or两边，有一边不用，性能差

3、 如果employee表的name和email这两列是一个复合索引，但是如果是 :name='A' OR email='B' 这种方式，不会用到索引!

limit优化

select film_id,description from film orderby title limit 50,5;

select a.film_id,a.description from film ainner join (select film_id from film order by titlelimit 50,5)b on a.film_id=b.film_id

2.6.14 jdbc批量插入几百万数据怎么实现？*

1、变多次提交为一次

3、使用批量操作

```
conn = DriverManager.getConnection(JDBC_URL
        + "?rewriteBatchedStatements=true", JDBC_USER, JDBC_PASS);
    conn.setAutoCommit(false);
    pstmt = conn
        .prepareStatement("insert into loadtest (id, data) values (?, ?)");
    for (int i = 1; i <= COUNT; i += BATCH_SIZE) {
      pstmt.clearBatch();
      for (int j = 0; j < BATCH_SIZE; j++) {
        pstmt.setInt(1, i + j);
        pstmt.setString(2, DATA);
        pstmt.addBatch();
      }
      pstmt.executeBatch();
      if ((i + BATCH_SIZE - 1) % COMMIT_SIZE == 0) {
        conn.commit();
      }
    }
    conn.commit();
```

省出的时间可观。

像这样的批量插入操作能不使用代码操作就不使用，可以使用存储过程来实现。

2.6.15 有没有使用过redis? Redis是什么

Redis是一个key-value的nosql数据库.先存到内存中，会根据一定的策略持久化到磁盘,即使断电也不会丢失数据。支持的数据类型比较多。

主要用来做缓存数据库的数据和web集群时当做中央缓存存放seesion

2.4.15 Redis和memche的比较?

	mysql	redis	memcache
类型	关系型	非关系型	非关系型
存储位置	磁盘	磁盘和内存	内存
存储过期	不支持	支持	支持
读写性能	低	非常高	非常高

1、 Redis和Memcache都是将数据存放在内存中，都是内存数据库。不过memcache还可用于缓存其他东西，例如图片、视频等等。

2、Redis不仅仅支持简单的k/v类型的数据，同时还提供list，set，hash等数据结构的存储。

3、虚拟内存--Redis当物理内存用完时，可以将一些很久没用到的value 交换到磁盘

2.6.16 简单说一下redis的使用场景?

缓存:

把经常需要查询的、很少修改数据，放到读速度很快的空间(内存)，以便下次访问减少时间。减轻压力，减少访问时间.

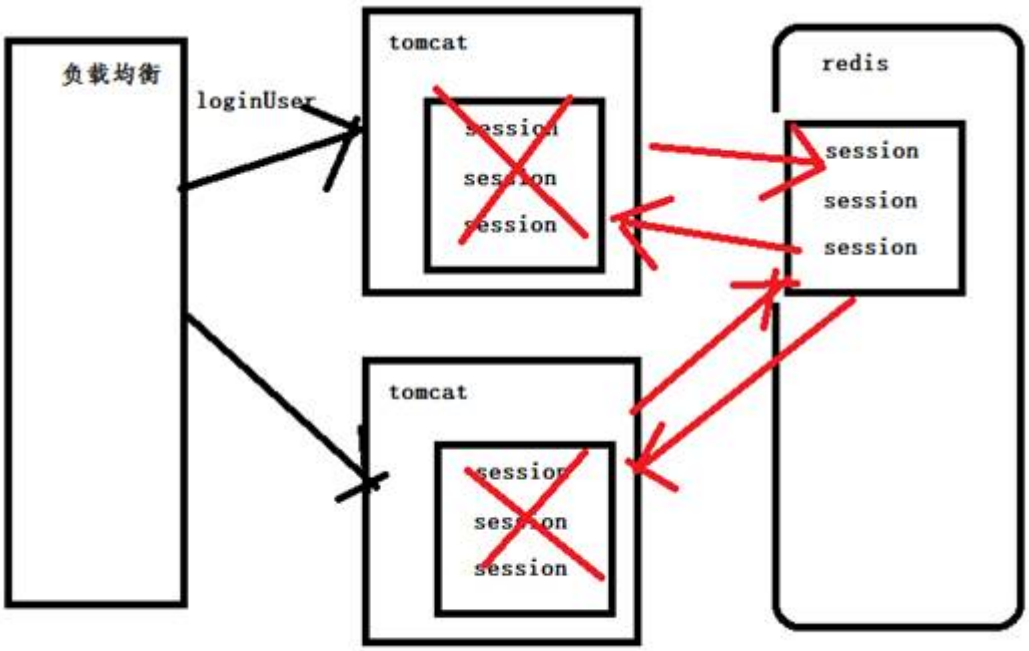
计数器:

redis中的计数器是原子性的内存操作。

可以解决库存溢出问题.进销存系统库存溢出。

session缓存服务器:

web集群时作为session缓存服务器



缓存队列等

2.6.17 redis对象保存方式?

Json字符串:

需要把对象转换为json字符串，当做字符串处理。直接使用set get来设置或者或。

优点：设置和获取比较简单

缺点：没有提供专门的方法，需要把把对象转换为json。(jsonlib)

字节:

需要做序列号，就是把对象序列化为字节保存。

如果是担心JSON转对象会消耗资源的情况，这个问题需要考量几个地方，

第一点：就是使用的JSON转换lib是否就会存在性能问题。

第二点：就是数据的数据量级别，如果是存储百万级的大数据对象，建议采用存储序列化对象方式。如果是少量的数据级对象，或者是数据对象字段不多，还是建议采用JSON转换成String方式。

毕竟redis对存储字符类型这部分优化的非常好。具体采用的方式与方法，还要看你所使用的场景。

2.6.18 Redis数据淘汰机制

在 redis 中，允许用户设置最大使用内存大小 server.maxmemory，在内存限定的情况下是很有用的。譬如，在一台 8G 机子上部署了 4 个 redis 服务点，每一个服务点分配1.5G 的内存大小，减少内存紧张的情况，由此获取更为稳健的服务。

内存大小有限，需要保存有效的数据?

redis 内存数据集大小上升到一定大小的时候，就会施行数据淘汰策略。

redis 提供 6种数据淘汰策略：

volatile-lru：从已设置过期时间的数据集（server.db[i].expires）中挑选最近**最少使用的数据淘汰*****

volatile-ttl：从已设置过期时间的数据集（server.db[i].expires）中挑选将要过期的数据淘汰

volatile-random：从已设置过期时间的数据集（server.db[i].expires）中任意选择数据淘汰

allkeys-lru**：从数据集（server.db[i].dict）中挑选最近最少使用的数据淘汰**

allkeys-random：从数据集（server.db[i].dict）中任意选择数据淘汰

no-enviction（驱逐）：禁止驱逐数据

2.6.19 Java访问Redis

1、使用jedis java客户端来访问redis服务器，有点类似通过jdbc访问mysql一样。

2、当然如果是spring进行集成时，可以使用spring data来访问redis,spring data只是对jedis的二次封装。jdbcTemplatejdbc关系一样

2.6.20 Redis集群

当一台数据无法满足要求，可以使用reids集群来处理，类似于mysql的读写分离。

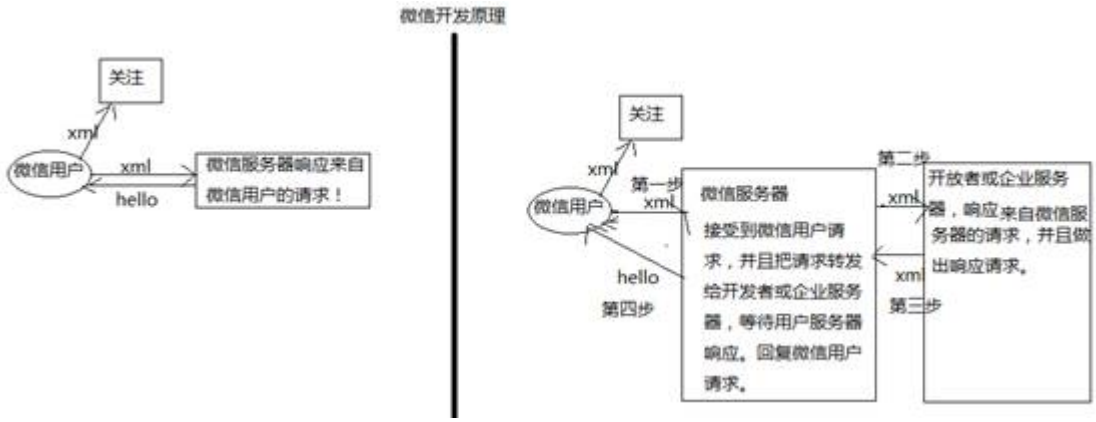
2.6.21简单介绍一下微信公共号的分类?

公众号:个人和企业都能申请

服务号：企业才能申请

企业号:企业才能申请

2.6.22 微信开发原理



微信公众平台开发者，通过接入认证的方式，让我们的服务器能处理来自微信服务器转发的微信用户的请求,处理完成后返回给微信服务器，有微信服务器对用户响应。

2.6.23怎么把微信和业务平台绑定?

微信用户和注册用户绑定?

让微信用户也能完成注册用户的功能。

用户注册实体中包含一个微信号的字段，当我进行绑定时就是修改用户的微信号字段。

当然我们在进行菜单跳转到页面后，我们是无法直接获取微信号的。要通过微信网页授权来获取**微信号。**

第一步：用户同意授权，获取code

https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&scope=SCOPE&state=STATE#wechat_redirect**若提示“该链接无法访问”，请检查参数是否填写错误，是否拥有scope参数对应的授权作用域权限。**

第二步：通过code换取网页授权**openId也就是我们微信号****

获取**code后，请求以下链接获取access_token： https://api.weixin.qq.com/sns/oauth2/access_token?appid=APPID&secret=SECRET&code=CODE&grant_type=authorization_code**

{ "access_token":"ACCESS_TOKEN",

** "expires_in":7200, **

** "refresh_token":"REFRESH_TOKEN", **

** "openid":"OPENID", **

** "scope":"SCOPE" } **

2.7 项目和业务部分

2.7.1项目分类

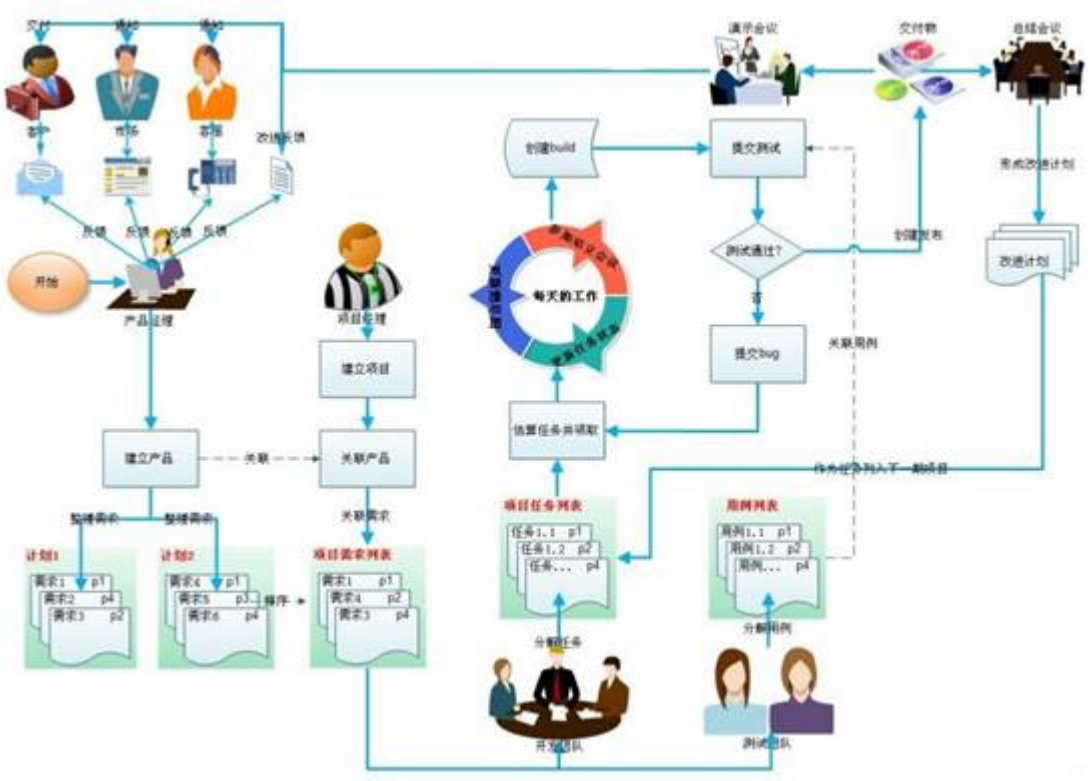
在公司中做的项目可以分为两种产品、项目.

项目：就是给一些公司接的项目，项目开发完成后。就交互，后面这个项目代码就不在维护了

产品：充分考虑扩展性和基本业务，来做一个产品。在这个产品上可以进行定制开发。

2.7.2 项目参与者

产品经理？PM？架构师（SE）？(开发PL? MDE？可能会有多个) (测试PL？ TSE可能会有多个) UI 资料



开发团队:开始代码能完成需求

测试团队：测试功能

UI:负责界面设计、静态代码的编写

资料：负责界面的文字描述。

QA :通过项目质量监控,和PM,SE同级

开发和UI和资料，协同设计和开发，开发完成后转测试(测试策略)交给测试团队进行测试，测试完成后会出一个测试报告。

2.7.3 项目流程

可行性分析和立项和开工会

需求分析

需求设计

项目开发(多个迭代)

迭代开工会

迭代设计

迭代开发

迭代测试

集成测试

迭代发布

迭代总结

....不断迭代

项目验收

项目总结

2.7.4 业务注意事项

1、不要多个项目都说你同一个模块。如果要说，就说后面是进行改进。

在pss,crm,shopping都写权限模块

2、多写点业务