## 一，尚德资料：

尚学习1.0模型：

<https://modao.cc/app/9JdhIqqOB9HeuG0NLBeFmnvMd719Cdn#screen=s68737ad3f4153352292400>

<https://modao.cc/app/9JdhIqqOB9HeuG0NLBeFmnvMd719Cdn>

<https://modao.cc/app/Me5M5uk3hb0EjULTZZjN5xwbPWYNRCc>

蓝湖地址：

https://lanhuapp.com/web/#/item/board?pid=4f52e360-48c6-486b-b017-3bd4d832705b

尚学习：后台：admin/admin

/usr/local/redis-3.2.1

科目包—综合包—学习包—5个知识点—两道题（ppt、音频、视频等相关主题资料）

简历：[http://cv.ftqq.com/#](http://cv.ftqq.com/)

查看linux系统版本：cat /etc/redhat-release

**mac使用：**

文稿地址; Users/admin/Documents/mysql\_util/work\_mysql

sublime同时选定多行：Ctrl + Shift + Up

文件已损坏：输入代码：sudo spctl --master-disable 然后在打开

sql文件位置：/Users/admin/Library/Application\ Support/PremiumSoft\ CyberTech/Navicat\ CC/Common/Settings/0/0/MySQL/172.16.164.45/transfer-bank/

打开Library:打开“访达”，shift+command+g，输入“~/Library”

课外扩展：

编程技巧：<https://mp.weixin.qq.com/s/yjbcQY1JjK8U5jAs1YGnYg>

9月19日：

vo->bo->dto->po

<https://blog.csdn.net/qq_41604862/article/details/79675223>

**简单地说Model面向业务，而我们是通过业务来定义Model的。DTO是面向界面UI，是通过UI的需求来定义的。通过DTO实现了表现层与Model之间的解耦，表现层不引用Model，如果开发过程中模型改变了而界面没改变，而我们只需要改Model而不需要去改表现层中的东西。**

## 二，mybatis

Mybatis是ORM（Object Relation Mapping 对象关系映射）框架的一种。

#与$区别：

1）#将传入的数据都当成一个字符串，会对自动传入的数据加一个双引号

­­­ $将传入的数据直接显示生成在sql中

2）#方式能够很大程度防止sql注入。

$方式无法防止Sql注入。

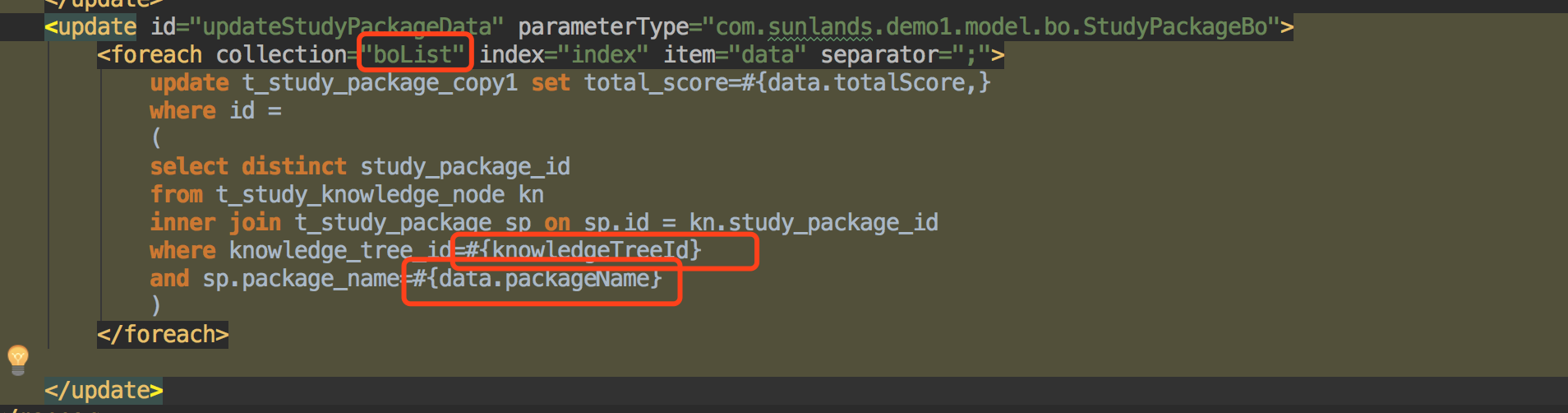
3）$方式一般用于传入数据库对象，例如传入表名.

4）一般能用#的就别用$.

MyBatis排序时使用order by 动态参数时需要注意，用$而不是#

&gt;=  大于等于

&lt;=  小于等于



## 三，mysql

CREATE TEMPORARY TABLE tmp\_table select f.cid client\_user\_id

from (SELECT client\_user\_id FROM t\_follow)

执行顺序：(序号为执行顺序) 聚合函数是在where执行之后having执行之前才会得到执行；参考资料：https://blog.csdn.net/u014044812/article/details/51004754/

(7) SELECT

(8) DISTINCT <select\_list>

(1) FROM <left\_table>

(3) <join\_type> JOIN <right\_table>

(2) ON <join\_condition>

(4) WHERE <where\_condition>

(5) GROUP BY <group\_by\_list> (开始使用select中的别名，之后是avg、sun等聚合函数）

(6) HAVING <having\_condition>

(9) ORDER BY <order\_by\_condition>

(10) LIMIT <limit\_number>

举例：

查询t\_follow表中 client\_user\_id,source\_id重复的数据，只保留一条记录。以下例子为查找出要删除的数据。

select \* from t\_follow

where (client\_user\_id,source\_id) in (select t.client\_user\_id,t.source\_id from t\_follow t group by t.client\_user\_id,t.source\_id having count(\*) > 1)

and id NOT IN (select min(id) from t\_follow t group by t.client\_user\_id,t.source\_id having count(\*) > 1)

随机：

select \*  
 FROM t\_background a  
 where a.type = #{type}  
 and a.delete\_flag = 0  
 order by rand() limit #{count}

在oracle获取记录行号使用rownum获取。

但是在mysql中查询显示行号就没这么方面了

方法如下：

SELECT @rowno:=@rowno+1 as rowno,r.\* from t\_article r,(select @rowno:=0) t

定义一个变量@rowno,让它的初始值为0，每有一行，数值加1。

然后按照你的需求的where条件，看是在当前sql内部或者外部添加@rowno。

优化：

https://mp.weixin.qq.com/s/1XOfPO3KasdJBnDUcla2KA

优化sql和索引‘

用慢查询日志定位执行效率低的 SQL语句

用 explain分析 SQL的执行计划

确定问题，采取相应的优化措施，建立索引啊，等

搭建缓存

读写分离

垂直拆分、水平拆分

**drop、TRUNCATE、DELETE**

DROP用于删除表,数据库等.

TRUNCATE和DELETE都是删除表数据.

但是TRUNCATE相当于初始化,比如如果有自增长ID,用TRUNCATE删除后,自增长Id会从1开始,

TRUNCATE table ent\_pro\_online

 order by convert(name using gbk) asc; 同样，查询的结果也是按照姓氏的升序排序；d多音字会有问题。

9月20日

mysql explain用法和结果的含义

参考资料：<https://www.cnblogs.com/yycc/p/7338894.html>

11月9日

create table en\_exam\_plan\_temp(

id INT NOT NULL AUTO\_INCREMENT,

project2\_name VARCHAR(100) comment '住址',

course\_num INT(3) comment '总科目',

PRIMARY KEY ( id )

)engine=innoDB default charset=utf8;

**mysql数值字符串类型的按照数值进行排序**

对mysql数值字符串类型进行排序，在默认情况下使用order by 字段名称 desc/asc 进行排序的时候，mysql进行的排序规则是按照ASCII码进行排序的，并不会自动的识别出这些数据是数值，可以采用

order by 字段名称+0 desc/asc的形式进行排序(亲测有效)

order by 字段名称\*1 desc/asc的形式进行排序( 没有进行测试 )

**判断是奇数**

case when exam\_session=4 and exam\_session\_num regexp '[13579]$' then 4

**创建临时表**

create temporary table tmp\_table select \* from ent\_project2\_area\_sell

**查看表访问进程：**

show PROCESSLIST

## 四，java

1、Java 实现文件随机读写

RandomAccessFile：<https://www.cnblogs.com/zuochengsi-9/p/6485737.html>

Java高级个人笔记(RandomStringUtils工具类)

//产生5位长度的随机字符串，中文环境下是乱码

RandomStringUtils.random(5);

//使用指定的字符生成5位长度的随机字符串

RandomStringUtils.random(5, new char[]{'a','b','c','d','e','f', '1', '2', '3'});

//生成指定长度的字母和数字的随机组合字符串

RandomStringUtils.randomAlphanumeric(5);

//生成随机数字字符串

RandomStringUtils.randomNumeric(5);

//生成随机[a-z]字符串，包含大小写

RandomStringUtils.randomAlphabetic(5);

//生成从ASCII 32到126组成的随机字符串

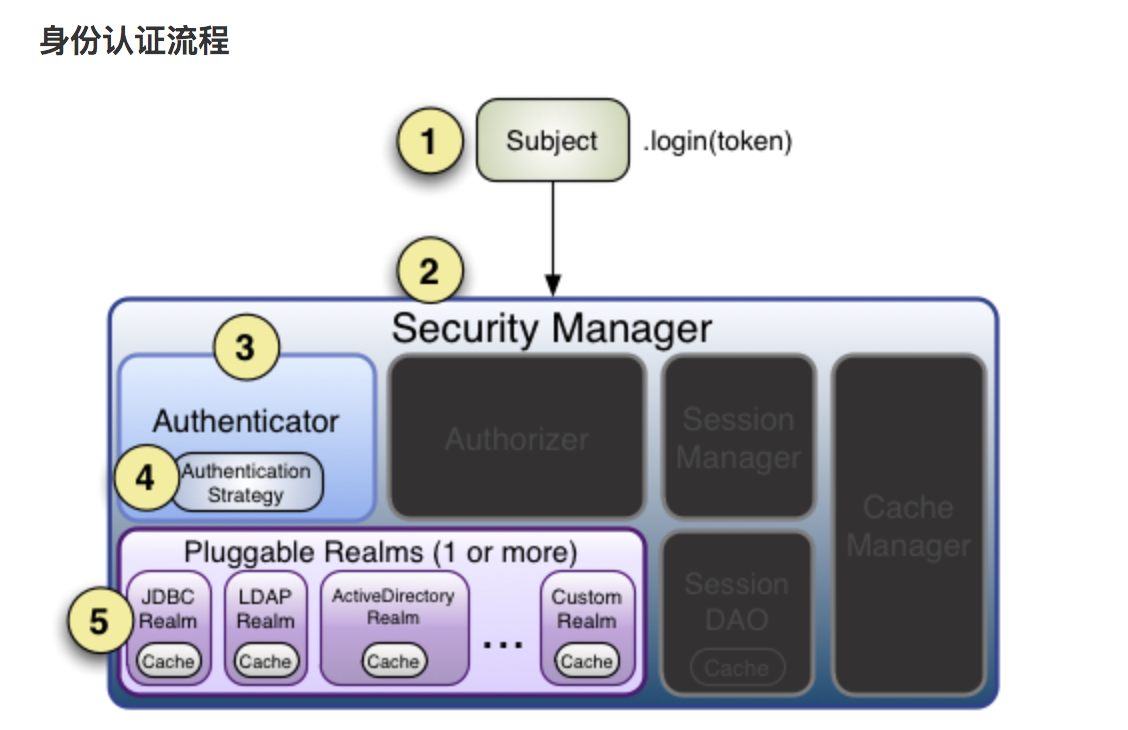
RandomStringUtils.randomAscii(4)

2、extends关键字用于确定泛型的上界。<A extends B>表示类B或者其子类。super关键字用于确定泛型的下界，<A super B>表示类B或者其父类，一直到Object。？则是一个通配符。

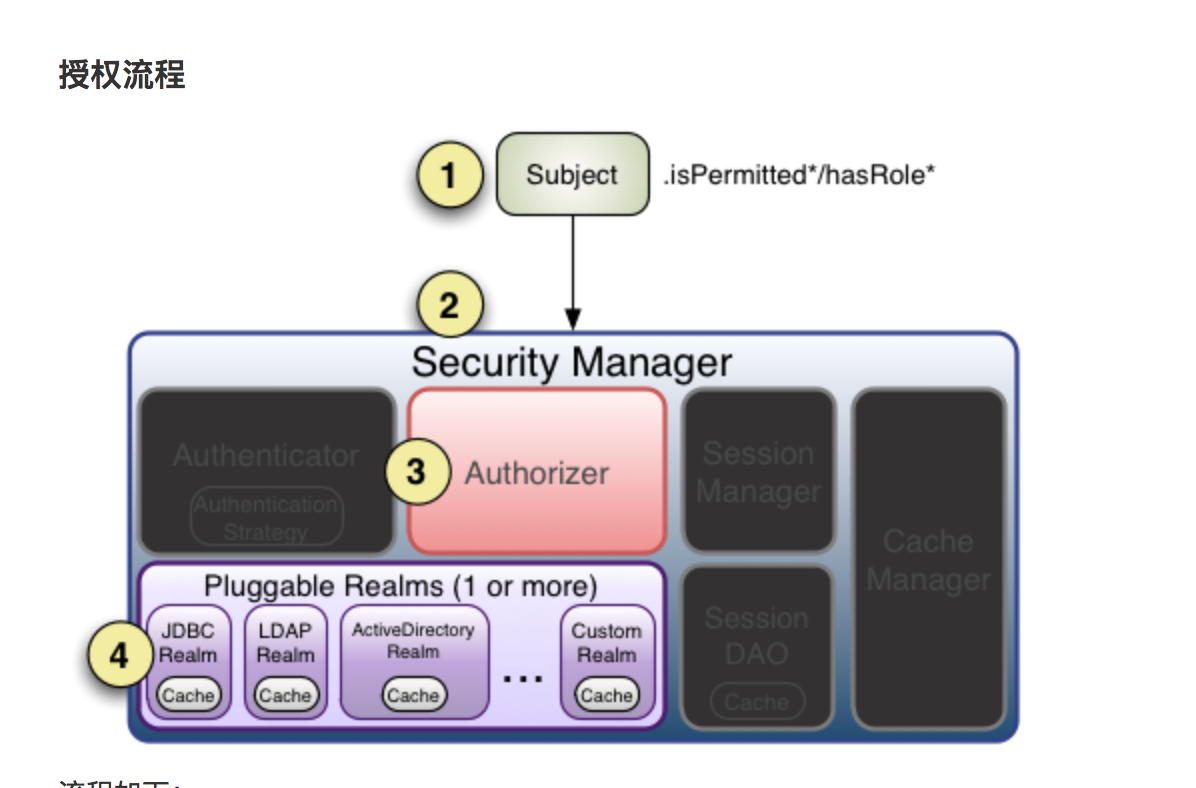
因此，<T extends Compatable<? super T>>表示了上界为实现了Comparable接口的类，<? super T>则表示实现Comparable接口的类的子类也可以，从而确定下界

StopWatch

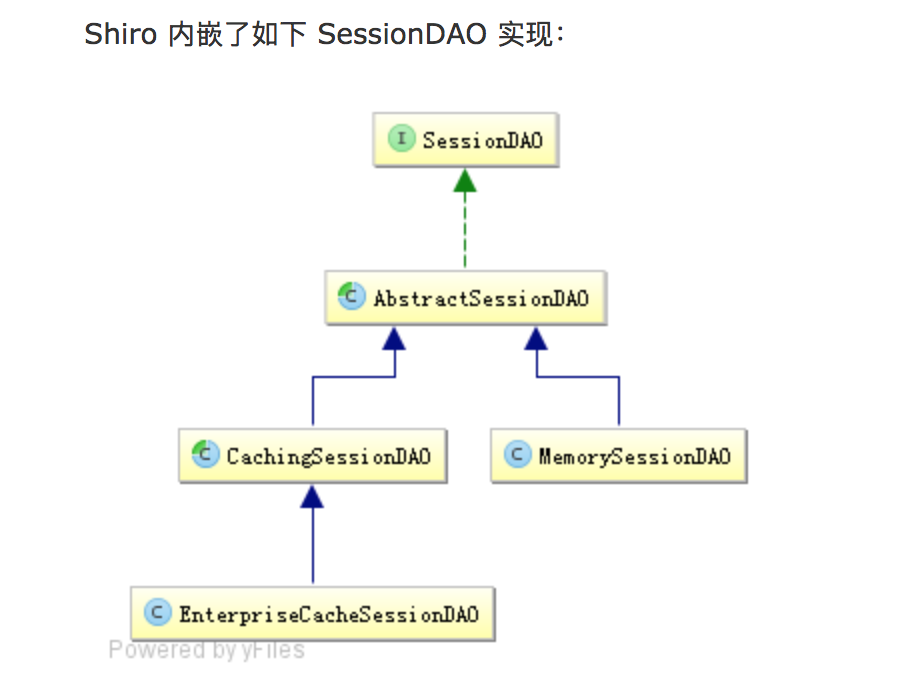
3、shiro

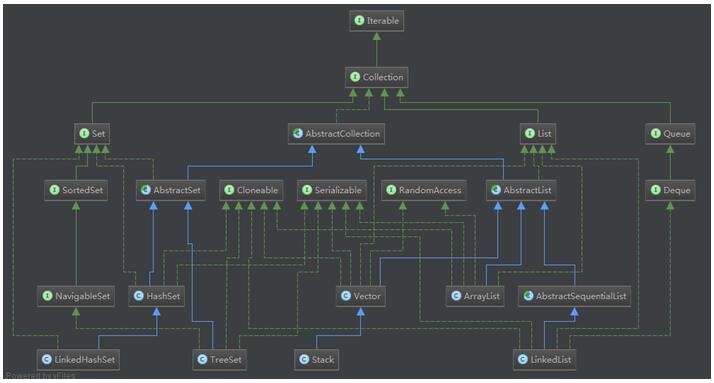


AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken token)：表示获取身份验证信息；AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principals)：表示根据用户身份获取授权信息。这种方式的好处是当只需要身份验证时只需要获取身份验证信息而不需要获取授权信息。对于 AuthenticationInfo 和 AuthorizationInfo 请参考其 Javadoc 获取相关接口信息。



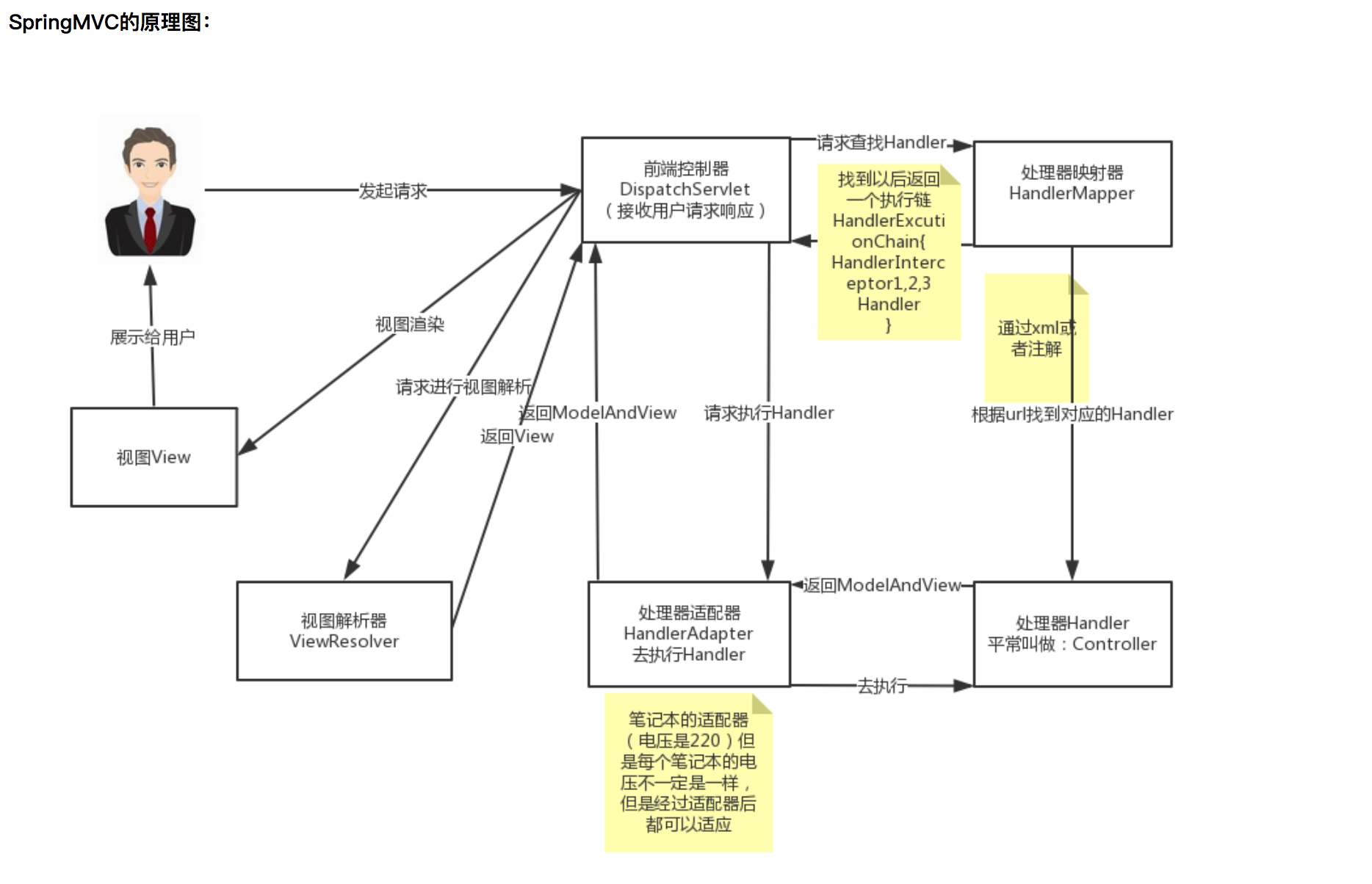






**4、springMVC是什么：**

springMVC是一个MVC的开源框架，springMVC=struts2+spring，springMVC就相当于是Struts2加上sring的整合，但是这里有一个疑惑就是，springMVC和spring是什么样的关系呢？这个在百度百科上有一个很好的解释：意思是说，springMVC是spring的一个后续产品，其实就是spring在原有基础上，又提供了web应用的MVC模块，可以简单的把springMVC理解为是spring的一个模块（类似AOP，IOC这样的模块），网络上经常会说springMVC和spring无缝集成，其实springMVC就是spring的一个子模块，所以根本不需要同spring进行整合。



**看到这个图大家可能会有很多的疑惑，现在我们来看一下这个图的步骤：（可以对比MVC的原理图进行理解）**

第一步:用户发起请求到前端控制器（DispatcherServlet）

第二步：­­前端控制器请求处理器映射器（HandlerMappering）去查找处理器（Handle）：通过xml配置或者注解进行查找

第三步：找到以后处理器映射器（HandlerMappering）像前端控制器返回执行链（HandlerExecutionChain）

第四步：前端控制器（DispatcherServlet）调用处理器适配器（HandlerAdapter）去执行处理器（Handler）

第五步：处理器适配器去执行Handler

第六步：Handler执行完给处理器适配器返回ModelAndView

第七步：处理器适配器向前端控制器返回ModelAndView

第八步：前端控制器请求视图解析器（ViewResolver）去进行视图解析

第九步：视图解析器像前端控制器返回View

第十步：前端控制器对视图进行渲染

第十一步：前端控制器向用户响应结果

5、SpringMVC使用MultipartFile实现文件的上传

6、java.lang.OutOfMemoryError: GC overhead limit exceeded

通常原因都是因为old区占用过多导致频繁Full GC，最终导致GC overhead limit exceed。如果gc log不够可以借助于JProfile等工具查看内存的占用，old区是否有内存泄露。分析内存泄露还有一个方法-XX:+HeapDumpOnOutOfMemoryError，这样OOM时会自动做Heap Dump，可以拿MAT来排查了。

还要留意young区，如果有过多短暂对象分配，可能也会抛这个异常。

<https://www.cnblogs.com/laughingQing/p/5857051.html>

7、stream（）

//排序  
List<ExamInfoNoOneResult> resultClass1 = classLists.stream()  
 .sorted(Comparator.*comparing*(ExamInfoNoOneResult::getYearInt)  
 .thenComparing(ExamInfoNoOneResult::getSort))  
 .collect(Collectors.*toList*());

// 取最大值

Optional<ExamInfoNoOneResult> max = resultExam1.stream()  
 .max(Comparator.*comparing*(ExamInfoNoOneResult::getYearInt)  
 .thenComparing(ExamInfoNoOneResult::getMonthInt));

ExamInfoNoOneResult oneResult = max.get();

// 分组

Map<String, List<ExamInfoNoOneResult>> collectDay = collectMonth.get(monthKey).stream()  
 .collect(Collectors.*groupingBy*(ExamInfoNoOneResult::getStartDate));

**Spring MVC的请求流程：**

第一步：发起请求到前端控制器(DispatcherServlet)

第二步：前端控制器请求HandlerMapping查找Handler可以根据xml配置、注解进行查找

第三步：处理器映射器HandlerMapping向前端控制器返回Handler

第四步：前端控制器调用处理器适配器去执行Handler

第五步：处理器适配器去执行Handler

第六步：Handler执行完成给适配器返回ModelAndView

第七步：处理器适配器向前端控制器返回ModelAndView。ModelAndView是springmvc框架的一个底层对象，包括 Model和view

第八步：前端控制器请求视图解析器去进行视图解析，根据逻辑视图名解析成真正的视图(jsp)

第九步：视图解析器向前端控制器返回View

第十步：前端控制器进行视图渲染。视图渲染将模型数据(在ModelAndView对象中)填充到request域

第十一步：前端控制器向用户响应结果

**spring boot 源码解析2-SpringApplication初始化**

参考文献<https://blog.csdn.net/qq_26000415/article/details/78914752>

**BigDecimal.setScale用法总结**

<https://blog.csdn.net/qq_39101581/article/details/78624617>

**安装tomcat并启动**

[/Users/admin/Library/apache-tomcat-7.0.92/bin](https://blog.csdn.net/qq_35106903/article/details/78860121::::/Users/admin/Library/apache-tomcat-7.0.92/bin)

**springboot打war包部署tomcat**

<https://blog.csdn.net/shu616048151/article/details/81323703>

## 五，linux笔记

|  |  |
| --- | --- |
| Git | 是一种版本控制系统，是一个命令，是一种工具 |
| gitlib | 是用于实现git功能的开发库 |
| github | 是一个基于git实现的在线代码托管仓库，包含一个网站界面，向互联网开放 |
| gitlab | 是一个基于git实现的在线代码仓库托管软件，你可以用gitlab自己搭建一个类似于github一样的系统，一般用于在企业、学校等内部网络搭建git私服 |

10月15日

gitlab安装教程：

<https://blog.csdn.net/qwlovedzm/article/details/80312302>

<https://www.cnblogs.com/TechSnail/p/7745459.html>

gitlab使用说明：<https://blog.csdn.net/Adelly/article/details/79099772>

生成ssh

https://www.jianshu.com/p/5b34b7b34cae

gitlab创建分支

gitlab权限问题：

.配置并启动gitlab-ce

[root@linux-node2 ~]# gitlab-ctl reconfigure

可以使用gitlab-ctl管理gitlab，例如查看gitlab状态：

[root@linux-node2 ~]# gitlab-ctl status

关闭gitlab：[root@linux-node2 ~]# gitlab-ctl stop

启动gitlab：[root@linux-node2 ~]# gitlab-ctl start

重启gitlab：[root@linux-node2 ~]# gitlab-ctl restart

查看linux系统版本：

cat /etc/os-release

cat /proc/version

安装python

https://blog.csdn.net/u012071918/article/details/78817344

**删除现有Python**

[root@test ~]# rpm -qa|grep python|xargs rpm -ev --allmatches --nodeps ##强制删除已安装程序及其关联

[root@test ~]# whereis python |xargs rm -frv ##删除所有残余文件 ##xargs，允许你对输出执行其他某些命令

[root@test ~]# whereis python ##验证删除，返回无结果

**2、删除现有的yum**

[root@test ~]# rpm -qa|grep yum|xargs rpm -ev --allmatches --nodeps

[root@test ~]# whereis yum |xargs rm –frv

redis-cli -h 172.16.164.169 -p 5000 🡪 flushall

mongdb启动：<https://www.cnblogs.com/magicsky0/p/3383812.html>

secureCRT上传文件到服务器：

打开ftp窗口：

打开本地文件：lcd E:\bak\update

lpwd 查看路径 lls 查看文件列表

打开服务器文件：cd /home/mysql

pwd 查看路径 ls 查看文件列表

上传文件：put filename

mysql执行sql文件：

登陆：mysql -h <192.168.0.9> -P 3361 -uroot –p / sunlands@2018

打开数据库：use transfer-bank

执行命令：source /home/mysql/ent\_exam\_video.sql

## 六，快捷键

idea

command+option+B 可以导航到一个抽象方法的实现代码

Command + n 生成set、get方法

shift+cmmand+. ，查看隐藏文件夹

## 七，待查看

Dao层封装数据库逻辑，采用JDBC进行数据库交互

AbstractRoutingDataSource实现类通过AOP切换DataSource

全局异常捕获（参数格式校验）

RestTemplate: <https://www.cnblogs.com/caolei1108/p/6169950.html>

CountDownLatch 和CyclicBarrier的不同之处?

Markdown是一种可以使用普通文本编辑器编写的标记语言

Poi读取大量数据：java.lang.OutOfMemoryError: GC overhead limit exceeded

InputStream和FileInputStream区别

JProfile安装

FastDFS是一个开源的轻量级分布式文件系统

String mon=String.*format*("%tm", nextDate);

## 八，注解

@EnableScheduling：开启对定时任务的支持

@Import：把实例加入Spring容器

@ConfigurationProperties：ConfigurationPropertiesBindingPostProcessior会对给标记的Bean进行属性值配置。

@EnableConfigurationProperties：将@ConfigurationProperties注解的Bean注入Spring的Bean容器.

@ConditionalOnBean：仅仅在当前上下文中存在某个对象时，才会实例化一个Bean）

@ConditionalOnClass：某个class位于类路径上，才会实例化一个Bean）

@ConditionalOnExpression：当表达式为true的时候，才会实例化一个Bean）

@ConditionalOnMissingBean：仅仅在当前上下文中不存在某个对象时，才会实例化一个Bean）

@ConditionalOnMissingClass（某个class类路径上不存在的时候，才会实例化一个Bean）

@ConditionalOnNotWebApplication（不是web应用）