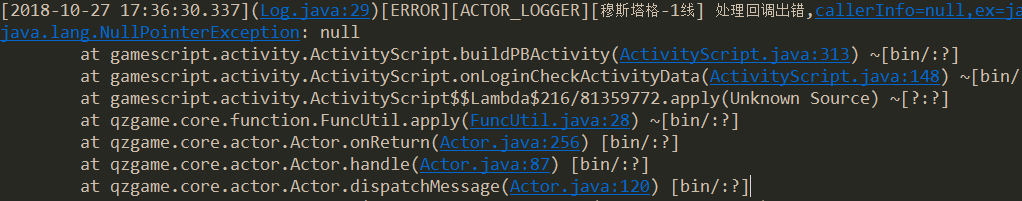
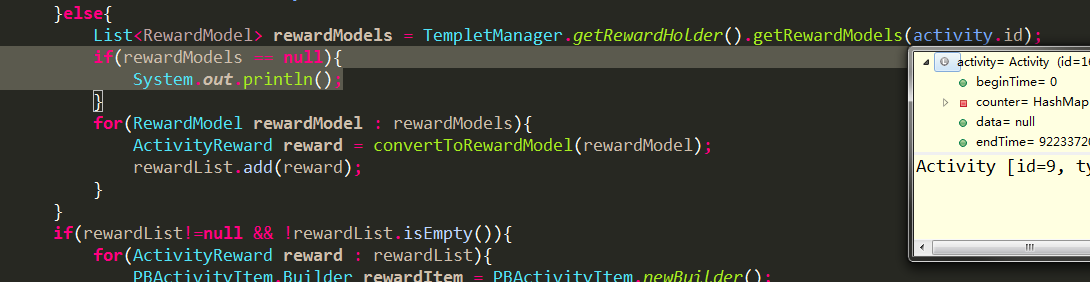
# 2018-10-27

一个报空错误





# 2018-10-28

比拼活动协议设计

客户单：

1.对应 活动id.

服务端：

1.活动是否结束

2.活动基本奖励是否领取

3.活动排名奖励是否领取

4.活动的排行榜

5.我的排名

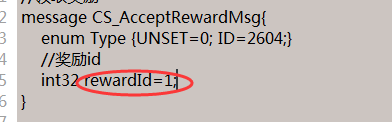
6.活动信息

# 2018-10-29

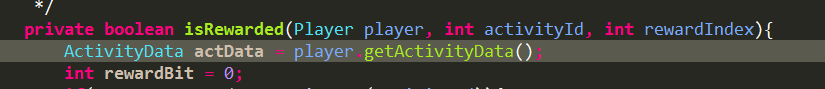
## 在线奖励流程

判断是否领过奖励

此时客户端发送的协议：

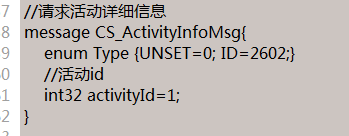


服务端验证奖励领取状态：

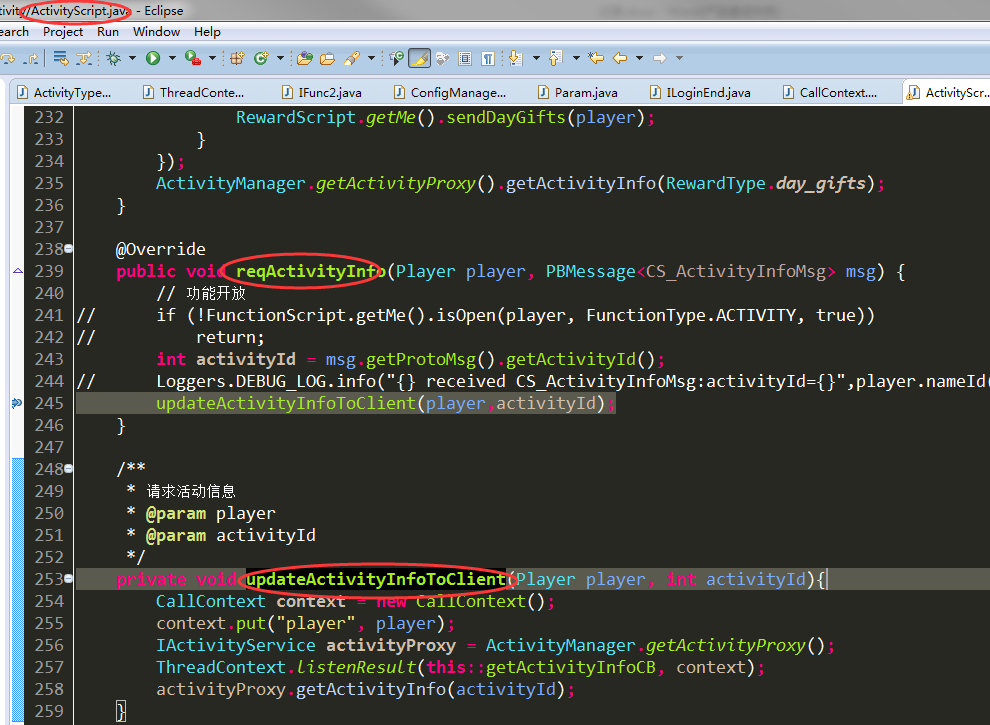


点击在线奖励按钮时，客户端会主动请求该活动的信息：



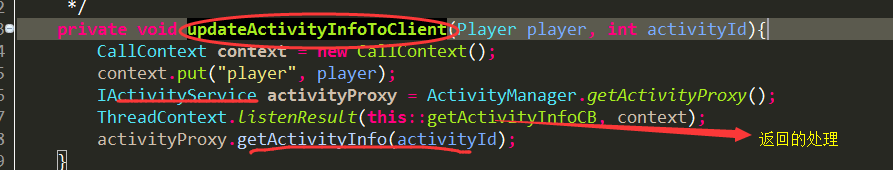


服务端会查询该活动信息，并返回给客户端：



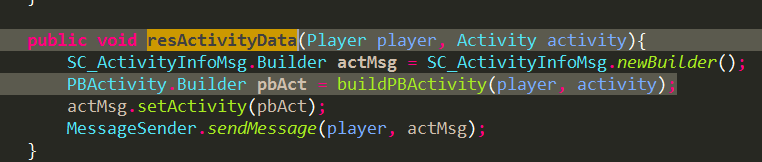
服务器获取活动信息的流程：

从 服务器内存的activityservice 获取该活动的信息activity对象

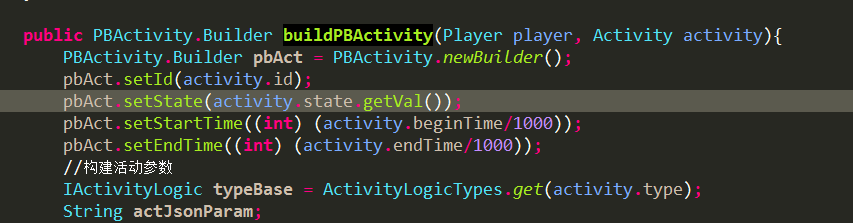


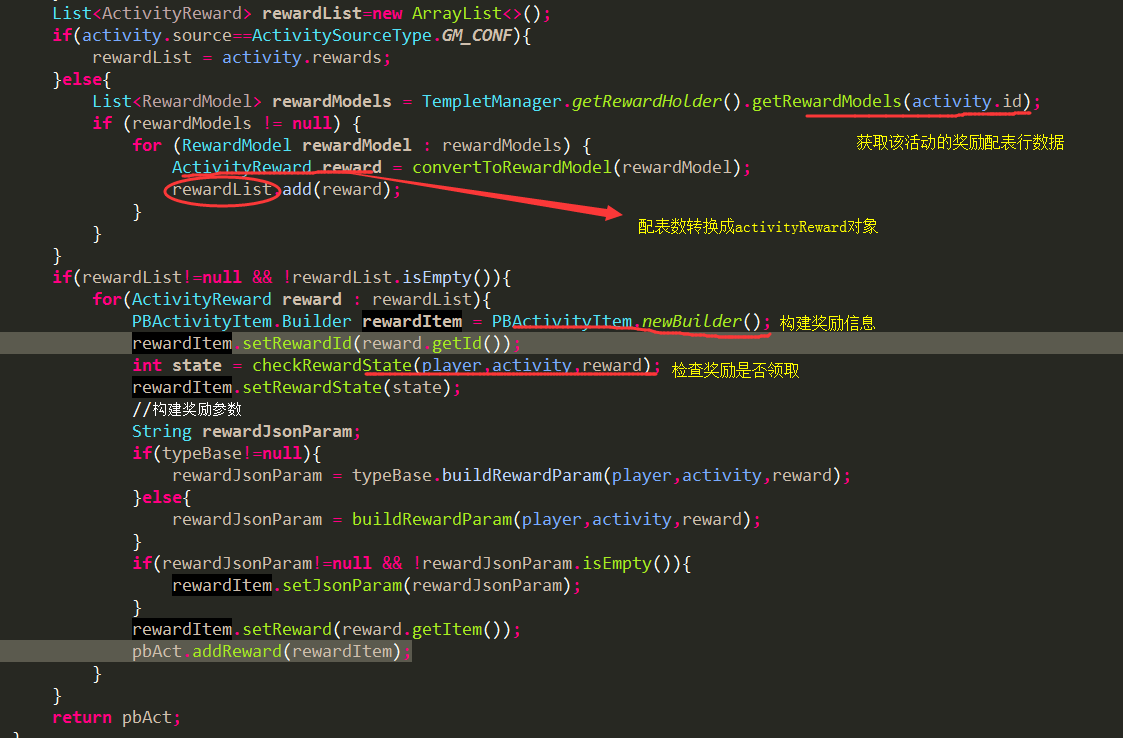


最终返回给客户端：

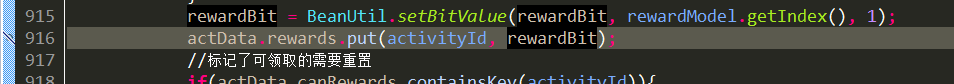


构建返回的activity协议：





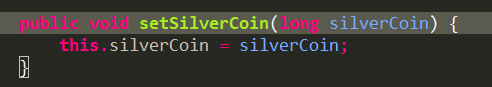
在线奖励领取状态，是记录在数据库何处？

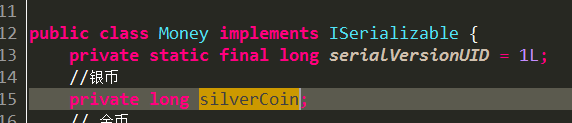


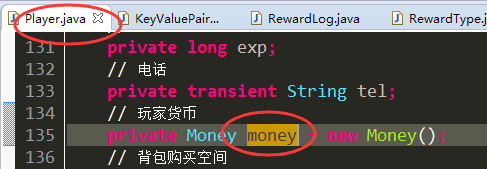
上面的代码是如何确定数据放入数据库的？

货币发生改变：









此处可以改变数据库的数据

**由上面可以看出，数据发生改变时，首先改变 player对象 中 的money对象**

**So….player对象的数据，如何持久化到数据库？**

**Player 对象是何时生成的，当角色登陆，当创角的时候**

## 用位来保存数值的领取状态：

原始值0，求1到5领取后的值的改变：

1<<1 = 10

0|10 = 10 = 2

1<<2 = 100

10|100 =110 = 6

1<<3 = 1000

110|1000 = 1110 = 14

1<<4 = 10000

1110|10000 = 11110 = 30

1<<5 = 100000

11110|100000 =111110 = 62

原始值0，5领取后的值的改变：

1<<5 = 100000

0|100000 = 100000

1<<2 = 100

100000|100 = 100100

判断是否领取计算：

原始值 右移 POS位

已知原始值111110=62

判断位置1是否领取：

62>>>1 = 111110>>> 1 = 11111

res = 11111|1 = 11111|00001 = 1 = true

已知原始值111110=62

判断位置2是否领取：

62>>>1 = 111110>>> 2 = 1111

res = 1111|1 = 1111|0001 = 1 = true

**用一个long数据类记录各个活动的登陆情况。（用位登记）:**

说明：数值<<移动的位数

初始值 0，设置 2

public static void setBool(Player player, 2, boolean flag) {

long bools = player.getBools(); 0

long bitAss = 1<<pbv.ordinal(); 1<<2 = 100

if (flag) { true

bools |= bitAss; 0 |= 100 =100

} else if ((bools&bitAss)>0){

bools ^= bitAss;

}

player.setBools(bools); bools = 100 =4

}

public static boolean getBool(Player player, PlayerBoolVar 2) {

return (player.getBools()&(1<<pbv.ordinal()))!=0;

}

100 & 1<<2 = 1 说明设置过了

初始值 100 ，设置 3

setBool(3)

100

1<<3 = 1000

100|1000 = 1100

初始值 1100

查看各个为的记录情况：

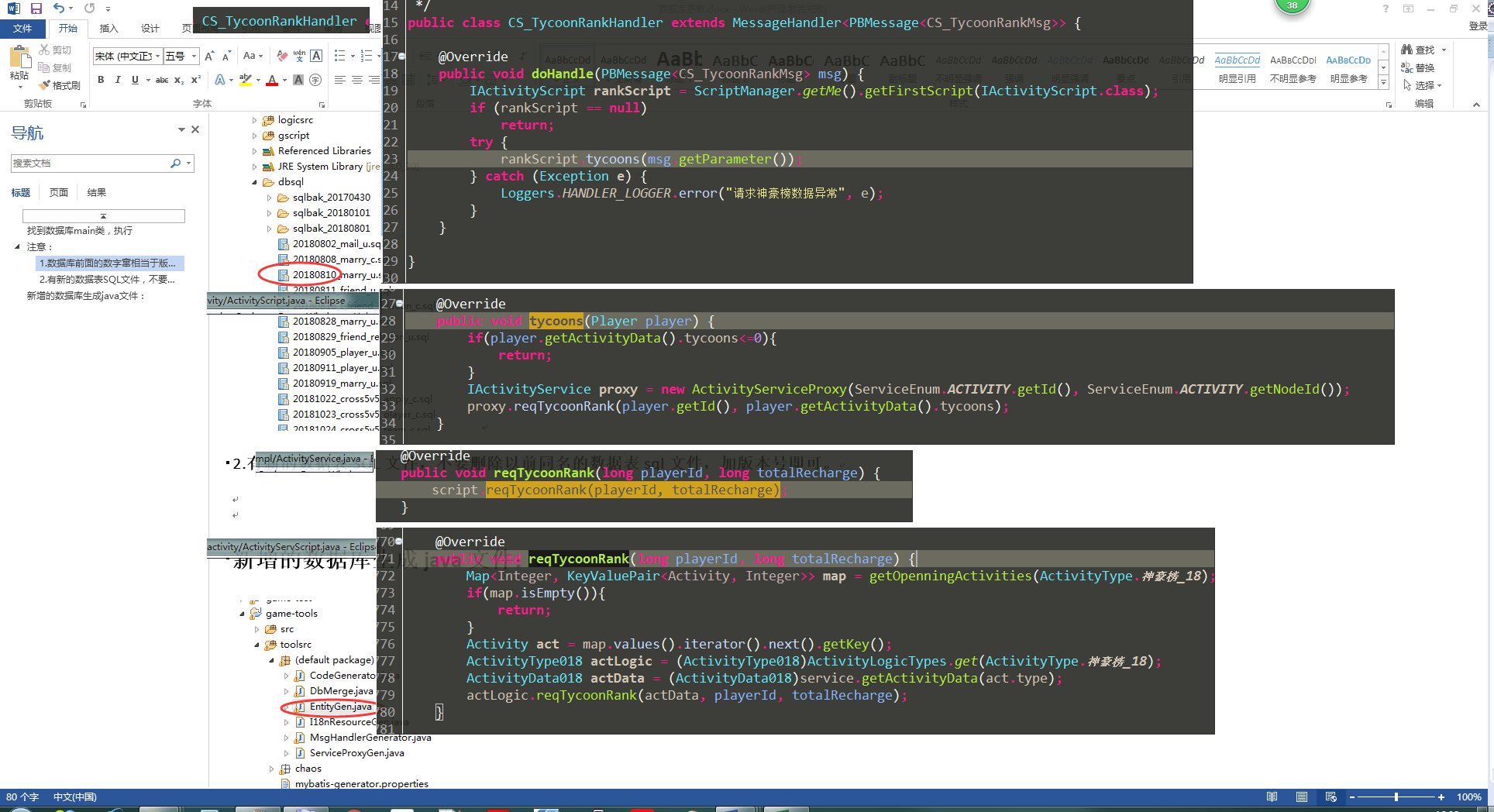
1100 & 1<<2 0100 100

1100 & 1<<3 1000 1000

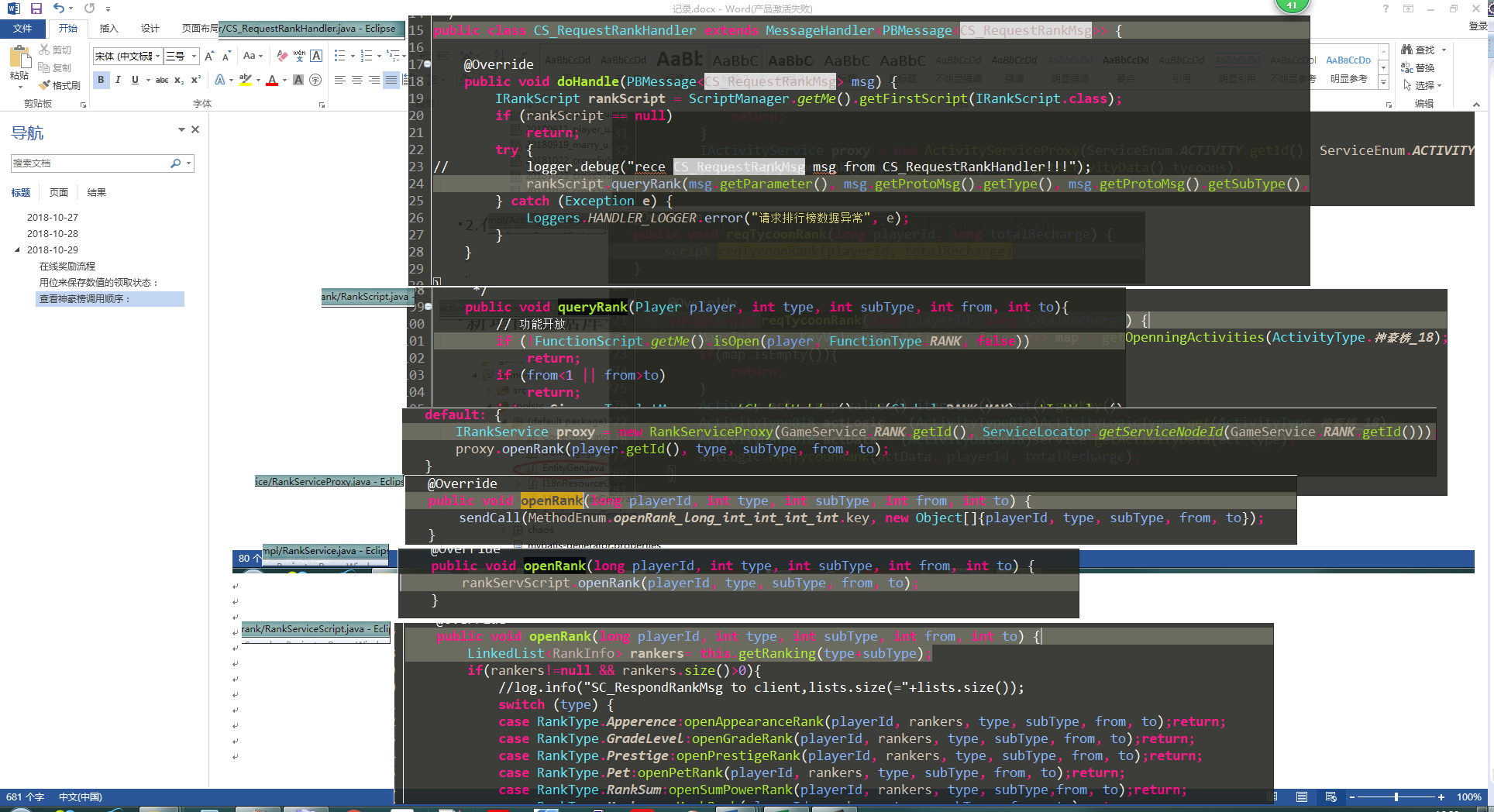
1100 & 1<<0 1 0

1100 & 1<<1 10 0

## 查看神豪榜调用顺序：

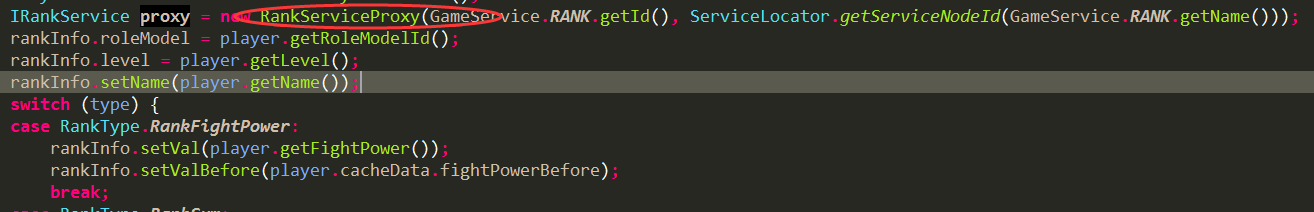


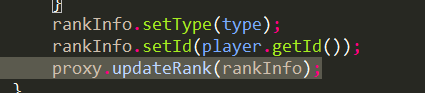
## 获取战力排行榜的调用顺序

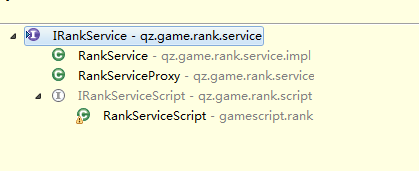


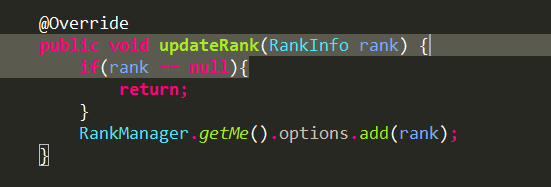
## 战力排行榜之更新：



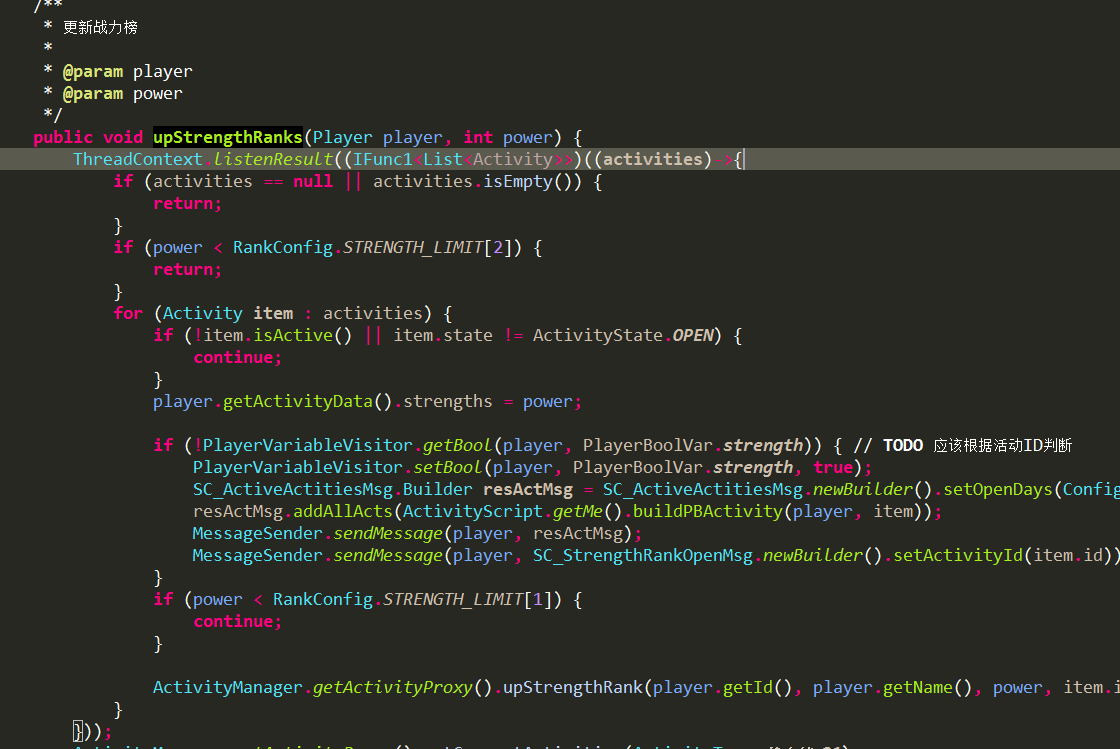








## 未知问题：listenResult是监听哪里的



List<RankInfo> rankers size 10

求 1-5的数据

for (ListIterator<RankInfo> iter=rankers.listIterator(from-1); iter.hasNext(); from++) {

if ((!flag&&from>to) || from>Const.RANK\_SIZE)

break;

RankInfo info = iter.next();

AppearanceRanker ranker = null;

if (flag && info.getId()==playerId) {

ranker = toAppearanceRanker(info, from);

msg.setSelfRank(TopRanker.newBuilder().setAppearanceRanker(ranker));

}

if (from <= to) {

if (ranker == null) {

ranker = toAppearanceRanker(info, from);

}

msg.addTopRankers(TopRanker.newBuilder().setAppearanceRanker(ranker));

}

}

# 2018-10-31

活动结束后一直报错。活动结束的方法没有写完整？

# 2018-11-1

MyTime [timeFormatStr=6 1-8 \*, list=[], index=-1, beginTime=1520611201000, endTime=1521302400000, state=UN\_START]

beginTime=1520611201000 = 2018-03-10 00:00:01

endTime=1521302400000 = 2018-03-18 00:00:00

MyTime [timeFormatStr=1 2018-10-31 00:00:00,2018-11-07 23:59:59, list=[TimeInfo [startTime=1541001600000, endTime=1541088000000]], index=0, beginTime=1540915200000, endTime=1541606399000, state=ACTIVE]

\* 打开某一个具体的比拼活动时，实时加载排行榜

\* 比拼活动加载的时候，必须判断活动是否已经结束

\* 已过期，数据从数据库里获取。

\* 未结束，数据从实时榜单获取

\* 活动结束后，数据保存在数据库。并且发放奖励给用户

# 2018-11-2

再测一遍：

未过期时：服务器开启，角色登陆，角色点击活动

过期时：服务器开启，角色登陆，角色点击活动

如何保存自己的排名和战力？不需要保留！

## 太乙bug：

重生：

种类别的猎魂没有返还

消耗的经验材料没有返还

# 2018-11-3

**自定义活动排行榜**

当活动不存在的时候，排行榜不开启

Map<Integer, KeyValuePair<Activity, Integer>> **map** = ActivityServScript.*getMe*().getOpenningActivities(ActivityType.***比拼\_战力32***);

**if**(map.isEmpty()){

**return**;

插入方法不执行

活动开启的时候，清空以前的排行榜

当活动结束后，保存排行榜信息

Eclipse报错：

#

# A fatal error has been detected by the Java Runtime Environment:

#

# EXCEPTION\_ACCESS\_VIOLATION (0xc0000005) at pc=0x00000000771da365, pid=71436, tid=62232

#

# JRE version: Java(TM) SE Runtime Environment (8.0\_20-b26) (build 1.8.0\_20-b26)

# Java VM: Java HotSpot(TM) 64-Bit Server VM (25.20-b23 mixed mode windows-amd64 compressed oops)

# Problematic frame:

# C [ntdll.dll+0x2a365]

#

# Failed to write core dump. Minidumps are not enabled by default on client versions of Windows

#

# An error report file with more information is saved as:

# D:\yingling\trunk\server\game-server\hs\_err\_pid71436.log

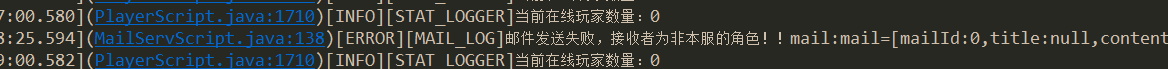
#

# If you would like to submit a bug report, please visit:

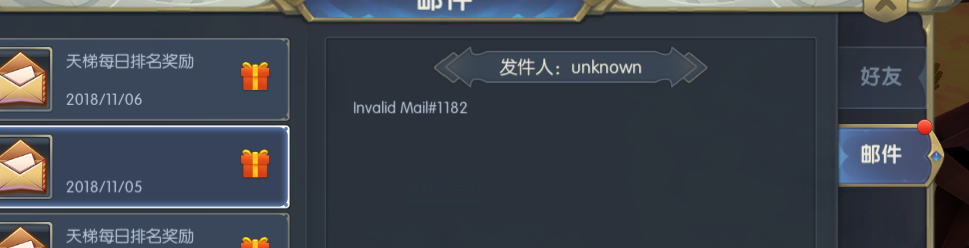
# http://bugreport.sun.com/bugreport/crash.jsp

#

# 2018-11-5







**int** **fightPower** = PlayerAttributeScript.*getMe*().calEquipAttributeFightPower(itemModel, equip);

Equip[] **equips** = player.getEquips();

## 问题：是如何保存数据的？

1.player上的参数有设置，立刻关掉服务器。Player上数据会丢失

2.状态设置器的记录也是如此

3.宝石的穿戴也是如此

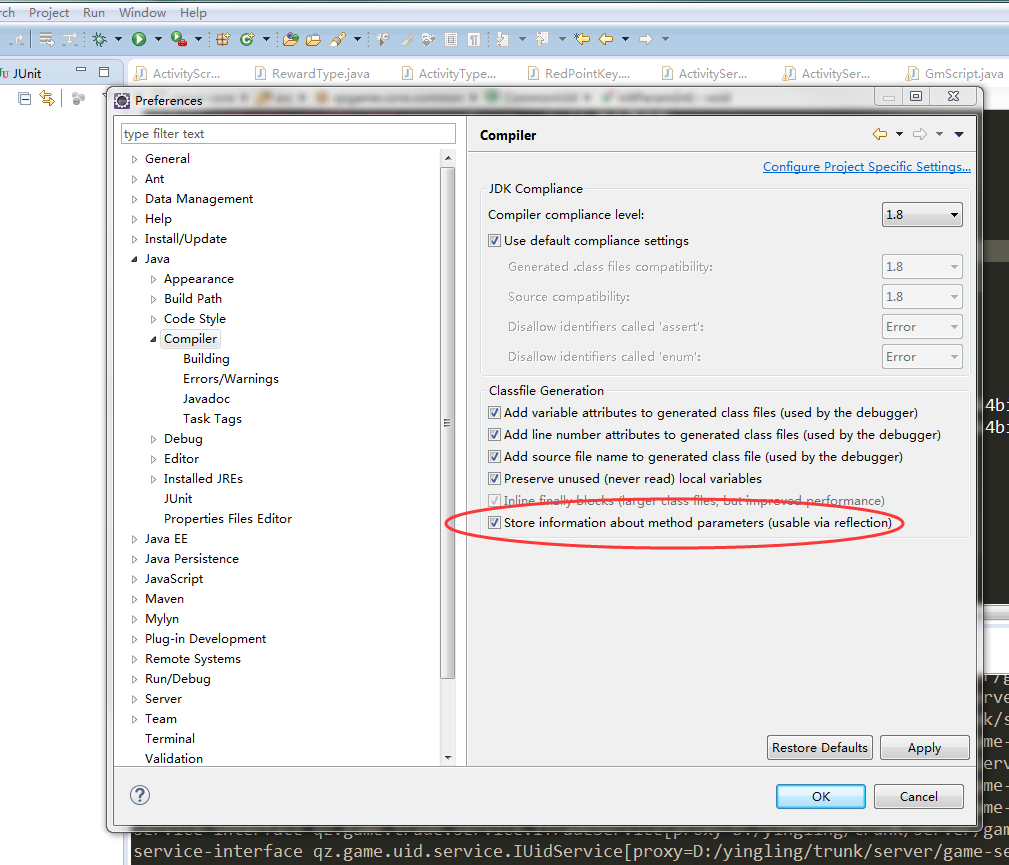
提问：是如何保存数据的？没有做到实时保存数据

# 2018-11-7

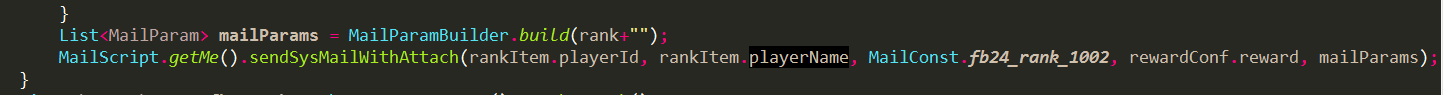
## 小红点

RedPointScript.getMe().addRedPoint(player, RedPointKey.ActivityRedPointKey(activity.type));

## 设置java的编译参数



## MailScript.getMe().sendSysMailWithAttach(b.getPlayerId(), null, mailId, b.getItemId() + "\_" + b.getItemNum(), null);



# 2018-11-7

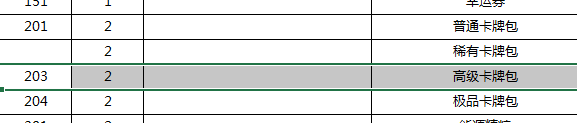
## 一些gm命令：

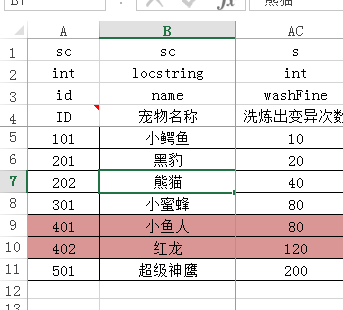
关机：$stop 30

添加卡牌包：$item 201 10

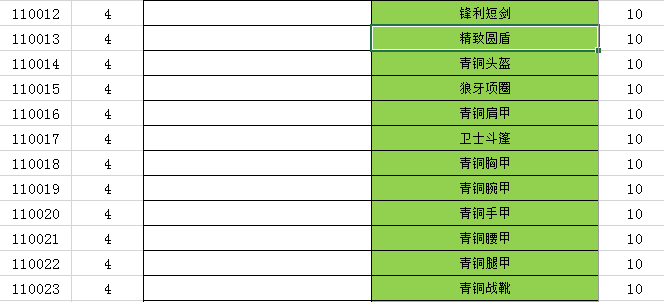
开启所有功能： $function

添加战宠：$pet add 101 战宠表





添加装备：$item 110012 1



添加水晶：

$item 10108 1



# 2018-11-8

## 皇家骑士战斗观察：

坐骑的升星，属性将会加到主角身上

所有坐骑的属性都会加到主角身上！





登陆流程

玩家移动

玩家战斗

地图怪物生成

装备

# 2018-11-12

15级：

activity {

reward {

rewardId: 52011

jsonParam: "{}"

reward: "1\_550000;5153\_25;5103\_25;790\_2;3122\_1"

require: "15"

rewardType: 143

}

jsonParam: "{\"myRank\":0,\"myValue\":15}"

startTime: 1541433600

activityType: 33

}

16级：

reward {

rewardId: 52011

**rewardState: 1**

jsonParam: "{}"

reward: "1\_550000;5153\_25;5103\_25;790\_2;3122\_1"

require: "15"

rewardType: 143

}

# 2018-11-13

## 比拼平台配置奖励设置

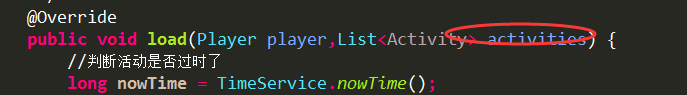
## Gm命令的研究

$item 11 3

## Ant build的研究

## 比拼活动的判空

实时





**if**(activities!=**null**){

**if**(rankers!=**null**){

自定义排行榜：

## 比拼日期bug

活动开启时，时间没有下发



## 邮件bug：





比拼活动的数据：

卡牌（图鉴）

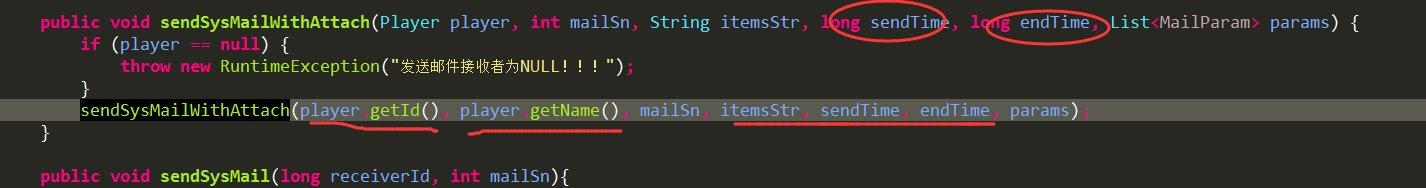
60005 [{"val":8279,"name":"耐性的路罗斯","rank":1,"id":1098001085},{"val":3845,"rank":2,"id":109800110},{"val":3845,"rank":3,"id":109800809}]

竞技场

60007 [{"val":44774,"name":"天梯守护者","rank":1,"id":-101},{"val":42384,"name":"天梯守护者","rank":2,"id":-201},{"val":42384,"name":"天梯守护者","rank":3,"id":-401},{"val":30938,"name":"天梯守护者","rank":4,"id":-402},{"val":30938,"name":"天梯守护者","rank":5,"id":-202},{"val":28662,"name":"天梯守护者","rank":6,"id":-102},{"val":28662,"name":"天梯守护者","rank":7,"id":-103},{"val":26456,"name":"天梯守护者","rank":8,"id":-203},{"val":260339,"name":"高尚的背叛者","rank":9,"id":10980077},{"val":26456,"name":"天梯守护者","rank":10,"id":-403}]

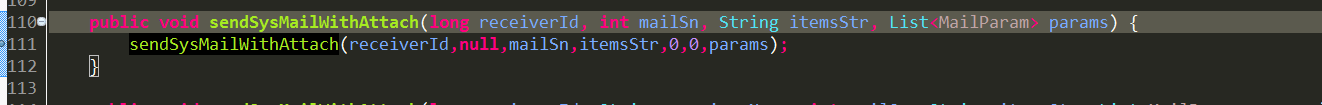
天梯的邮件发送方式：

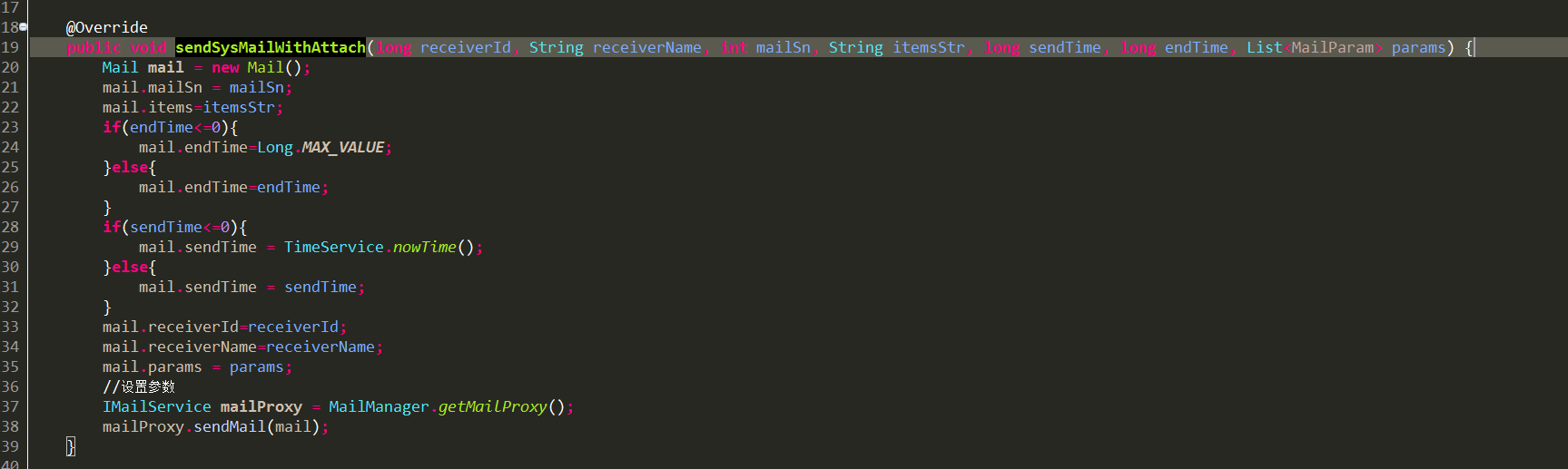




比拼的邮件发送方式：







## 电脑C盘爆满清理：

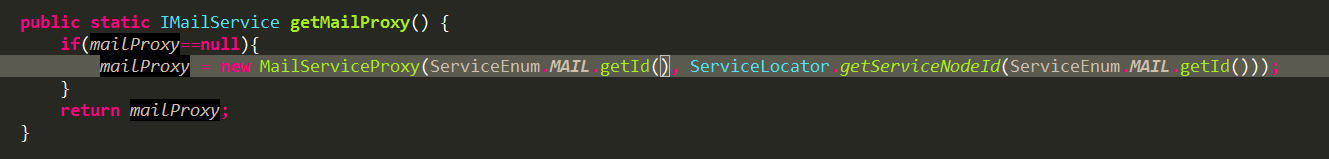
360系统瘦身：



# 2018-11-14

## 太乙的活动更新

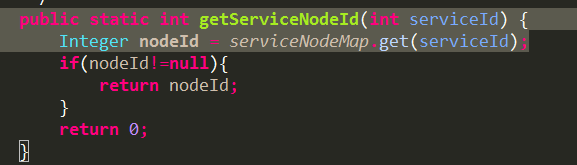
## 邮件服务跟踪：



 = 20；

ServiceNodeMap的内容：

{1=20101, 2=20101, 3=20101, 4=20101, 5=20101, 6=20101, 7=20101, 8=20101, 9=20101, 10=20101, 11=20101, 12=20101, 13=20101, 14=20101, 15=20101, 16=20101, 17=20101, 18=20101, 19=20101, 20=20101, 21=20101, 23=20101, 24=20101, 25=20101, 26=20101, 27=20101, 28=20101, 29=20101, 30=20101, 32=20101, 33=20101, 34=982001, 35=982001, 36=20101, 39=982001, 42=982001, 43=20101}

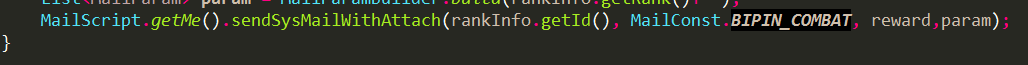




## 排行榜改成自己的排行榜

战力榜，等级榜，卡牌帮

邮件修改：



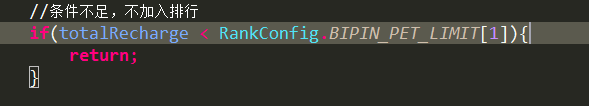
类型修改：



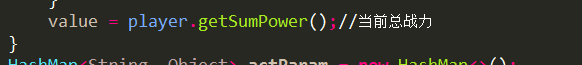
数据修改：



条件修改：



当前值修改：



## 取消修改，还原。







# 2018-11-15

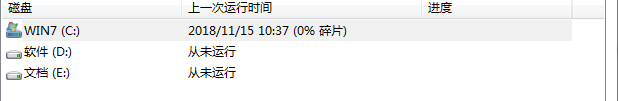
## C盘整理

碎片整理









## Player.loadNum断线重连时，要设置为0

## 比拼活动达标设置断点bug：

RedPointScript.*getMe*().addRedPoint(player, RedPointKey.*ActivityRedPointKey*(activityModel.getType()));

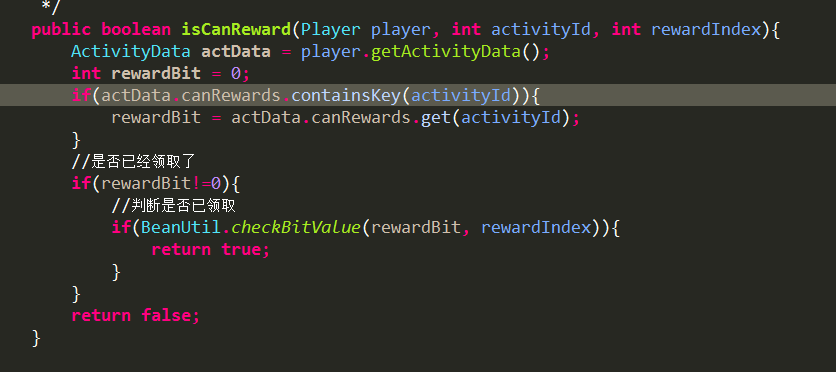
判断是否领取过，判断是否可以发小红点

## 战力达标领取奖励的判断和设置状态

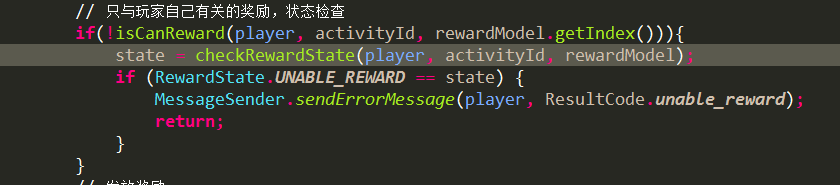
判断能否领奖：

此时activityId = 60001

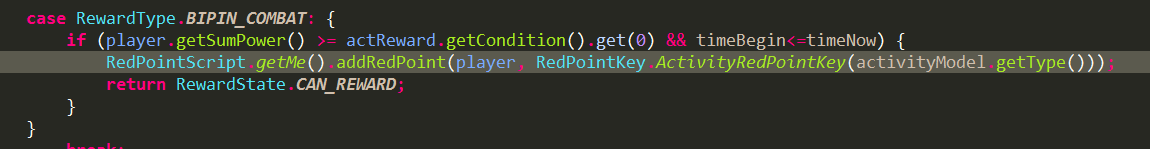
canRewards没有包含60001



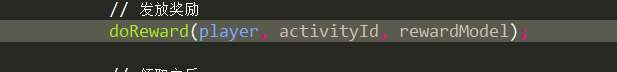
检查状态：



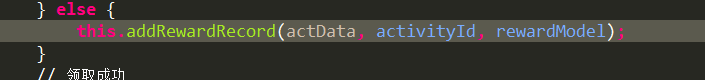
判断是否达标：



发放奖励：



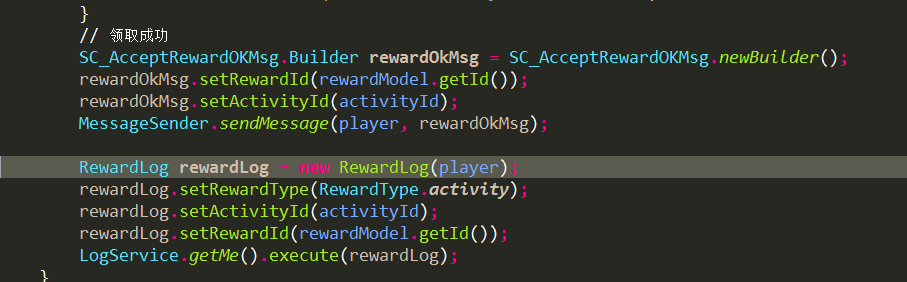




设置状态领取：



返回消息，并记录日志



在插入排行的地方，都要判断比拼活动的相关内容的达标奖励

判断是否领取过奖励：



未领取，则发送小红点

///加指定的小红点

RedPointScript.getMe().sendRedPoint(player, RedPointKey.ActivityRedPointKey(32));

插入排行和检测小红点

IActivityLogic actLogic = ActivityLogicTypes.get(ActivityType.比拼\_装备37);

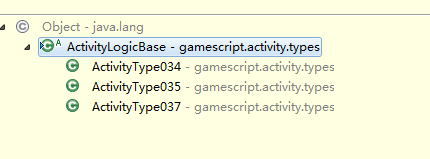
if(actLogic != null){

ActivityType037 type = (ActivityType037)actLogic;

type.upActivityRank(player.getId(), equipPower,player);

}

已经拥有的小红点判断的比拼活动



32,33,36,38

比拼活动的奖励，放到Activity对象内部

奖励可能来自后台配置和配表读取。。。。

## 活动对应的所有奖励：

List<ActivityReward> rewardList=new ArrayList<>();

List<RewardModel> rewardModels = TempletManager.getRewardHolder().getRewardModels(activity.id);

if (rewardModels != null) {

for (RewardModel rewardModel : rewardModels) {

ActivityReward reward = convertToRewardModel(rewardModel);

rewardList.add(reward);

}

}

# 2018-11-15

## 活动奖励与后台配置兼容！

邮件下发的奖励兼容

达标奖励兼容

# 2018-11-19

## 股票买卖：

买入：

特变：7.05 200 1410

特变：6.45 1100 7095

许继：9.35 100 935

兴业：4.74 100 474

总买入：9914

卖出：

特变：7.19 1300 9347

许继：9.41 100 941

兴业：5.61 100 561

总卖出：10849

总收益：935

转到银行：65.82

2018/11/22 股价下跌

特变：6.96 1300 9347 9048 -299

许继：8.97 100 941 897 -44

兴业：5.35 100 561 535 -26

## 小7新渠道SDK接入：

1.SDK登陆服务器

2.SDK购买商品

小7 域名：https://api.x7sy.com



服务端必须获得小7的appkey。（是公匙吗？）

服务端如何接受客户端的tokenkey？

需要游戏方在开放平台中给游戏添加白名单？

客户端给服务端tokenkey

【游戏服务器】将tokenkey和tokenkey与appkey（小7给游戏方提供的参数）处理后的sign信息请求【小7服务器】得到当前登录游戏的用户信息并且在【游戏服务器】中保存

如果不成功，服务端要返回给客户端消息

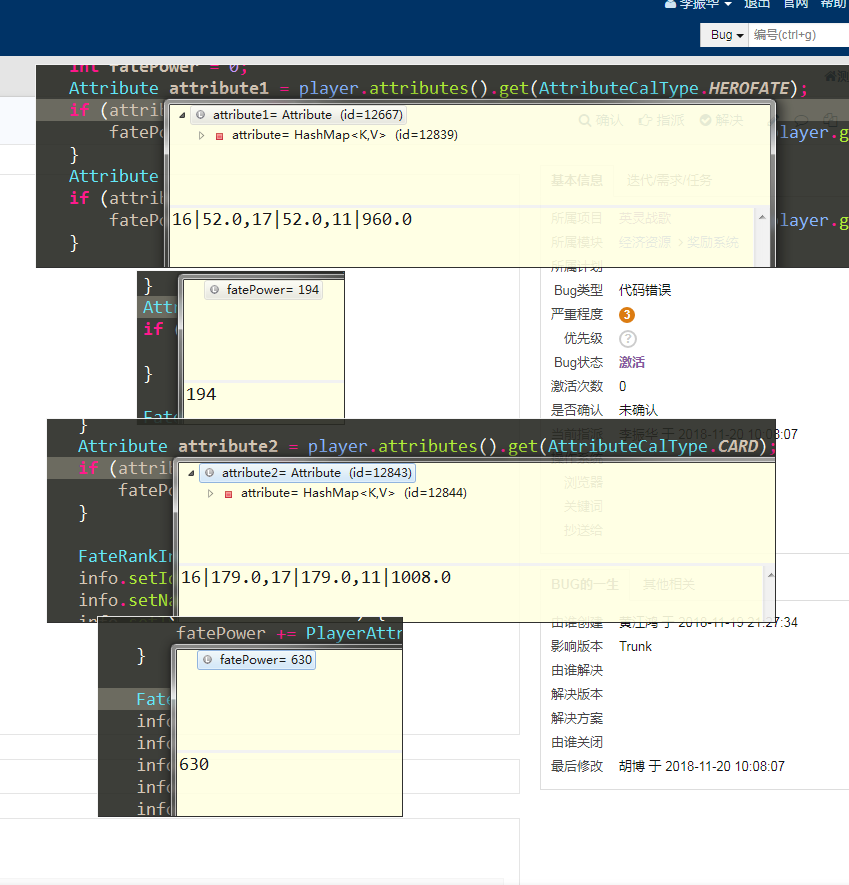
服务端把验证的消息发送到小七服务器验证：<https://api.x7sy.com/user/check_logion>

<https://api.x7sy.com/user/check_logion>/tokenkey=de28191a4c808d36787d97f1a3825be1&sign=e7f1d806155b64483611042a23167448

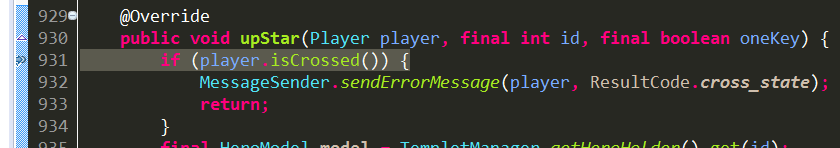
卡牌战力排行榜：

已激活卡组：





激活小恐龙



# 2018-11-21

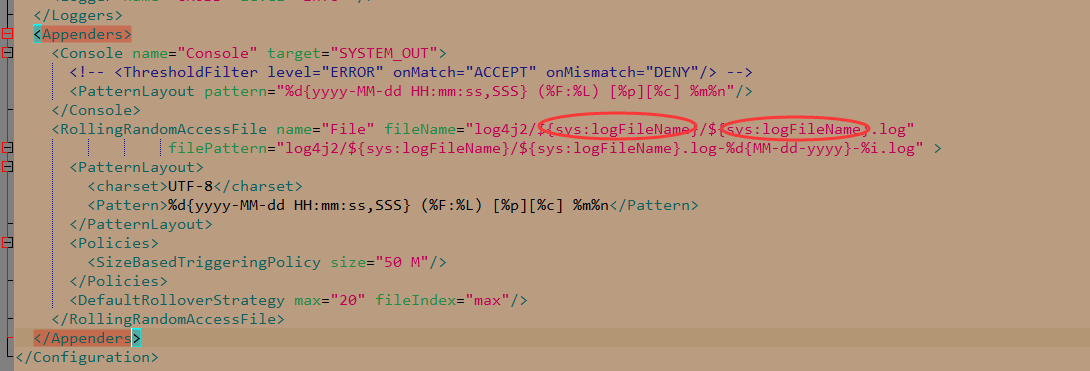
## 平台服的日志报错

单独执行main文件时，平台的日志报错

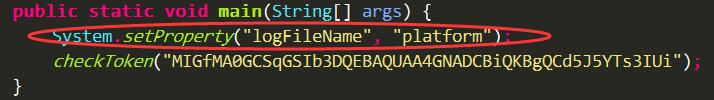
虽然在配置中有平台的配置，但当启动太乙服时，打印的日志是在world0里面



原因：



在log4j的配置文档中，有sys:logFileName参数，这个参数没有设置就会报错



在main开始时设置好久可以了

## 正常登陆流程，与SDK的关系

情况1:

未在游戏后端注册生成账号：

CSLogin

version: "2.1.0"

loginType: PC

account: "11111"

serverId: 30103

SCLoginResult

resultCode: 0

vipFlag: false

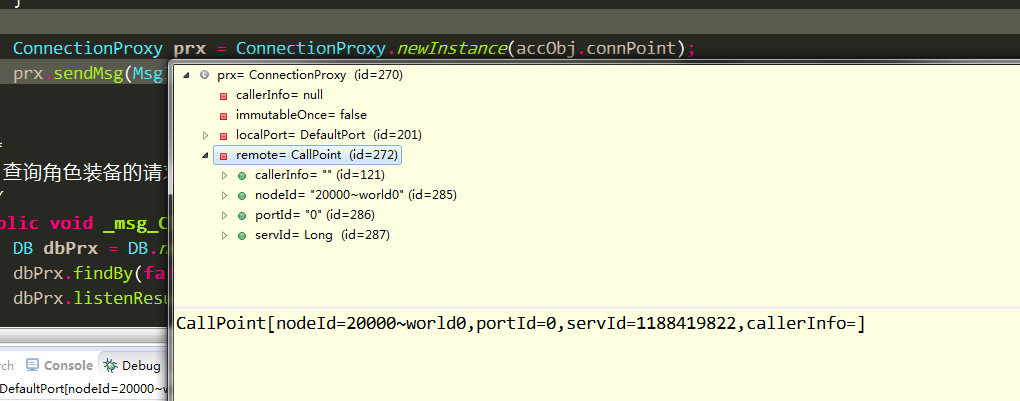
CSQueryCharacters

message CSQueryCharacters {

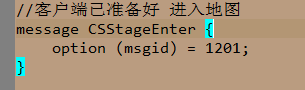
option (msgid) = 1003;

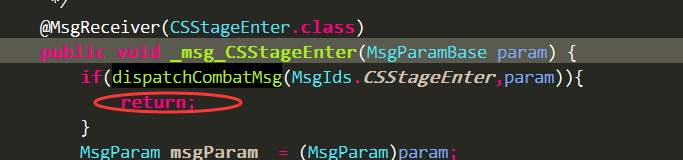
}

SCQueryCharactersResult



CSStageEnter





客户端进入创角界面：



{"errorno":-1,"errormsg":"tokenkey或者sign信息不完整！请检查tokenkey或者sign参数是否有传递，或者检查参数名称是否正确！"}

## 太乙支付流程：

## \*\*\*\*太乙活动表格更新

Http命令过来是怎么触发service的方法的

# 2018-11-22

## 用户的下单记录：

2018-11-22 05:07:11,906 (Log.java:142) [INFO][GAME] ===VipStoreManager.java:527 \_msg\_CSPayCheckCode 返回充值校验码:human=HumanObject[id=300400000211000012,name=皇甫光熙],id=300400000266294368,sn=6

2018-11-22 05:07:30,056 (HttpIpFilter.java:27) [INFO][PLATFORM] 请求者的Ip=120.92.216.197

2018-11-22 05:07:30,056 (HttpServerHandler.java:80) [INFO][PLATFORM] [-201783232]收到HTTP请求:{}

2018-11-22 05:07:30,057 (HttpServerHandler.java:125) [INFO][PLATFORM] POST请求：uri=/payNotify, param={accountId=603204173, amount=9800, quantity=1, productId=6, rate=0.0, orderId=201811220507124f46pw, roleId=300400000211000012, amountReal=9800, sandbox=false, currency=rmb, userId=300400000211000012, paymentType=PPTV渠道}

2018-11-22 05:07:30,070 (PayManager.java:58) [INFO][PLATFORM] onPay

2018-11-22 05:07:30,094 (PFService.java:44) [INFO][PLATFORM] payNotice {accountId=603204173, amount=9800, quantity=1, productId=6, orderId=201811220507124f46pw, rate=0.0, roleId=300400000211000012, amountReal=9800, sandbox=false, currency=rmb, userId=300400000211000012, paymentType=PPTV渠道}

2018-11-22 05:07:30,094 (Log.java:142) [INFO][GAME] ===PF\_PAY\_Manager.java:75 onPay 充值开始jo={accountId=603204173, amount=9800, quantity=1, productId=6, orderId=201811220507124f46pw, rate=0.0, roleId=300400000211000012, amountReal=9800, sandbox=false, currency=rmb, userId=300400000211000012, paymentType=PPTV渠道}

2018-11-22 05:07:30,175 (Log.java:142) [INFO][GAME] ===PF\_PAY\_Manager.java:163 \_result\_onPay1 充值中

2018-11-22 05:07:30,175 (Log.java:142) [INFO][GAME] ===PF\_PAY\_Manager.java:181 \_result\_onPay1 充值中，查询玩家jo={accountId=603204173, amount=9800, quantity=1, productId=6, orderId=201811220507124f46pw, rate=0.0, roleId=300400000211000012, amountReal=9800, sandbox=false, currency=rmb, userId=300400000211000012, paymentType=PPTV渠道}

2018-11-22 05:07:30,195 (Log.java:142) [INFO][GAME] ===PF\_PAY\_Manager.java:283 \_result\_onPay2 充值中

2018-11-22 05:07:30,195 (Log.java:142) [INFO][GAME] ===PF\_PAY\_Manager.java:300 \_result\_onPay2 充值中，玩家在线，正在充值，jo={accountId=603204173, amount=9800, quantity=1, productId=6, orderId=201811220507124f46pw, rate=0.0, roleId=300400000211000012, amountReal=9800, sandbox=false, currency=rmb, userId=300400000211000012, paymentType=PPTV渠道}

2018-11-22 05:07:30,215 (Log.java:142) [INFO][GAME] ===PF\_PAY\_Manager.java:346 \_result\_onPay3 充值结束

2018-11-22 05:07:30,215 (Log.java:142) [INFO][GAME] ===PF\_PAY\_Manager.java:348 \_result\_onPay3 充值结束，充值成功jo={accountId=603204173, amount=9800, quantity=1, productId=6, orderId=201811220507124f46pw, rate=0.0, roleId=300400000211000012, amountReal=9800, sandbox=false, currency=rmb, userId=300400000211000012, paymentType=PPTV渠道}

2018-11-22 05:07:36,921 (StageGlobalService.java:98) [ERROR][STAGE\_COMMON] pulseStageLog--------------------------------

2018-11-22 05:07:36,921 (StageGlobalService.java:107) [ERROR][STAGE\_COMMON] pulseStageLog countAllHuman 0--------------------------------

2018-11-22 05:07:36,921 (StageGlobalService.java:108) [ERROR][STAGE\_COMMON] pulseStageLog countAllStage 2--------------------------------

2018-11-22 05:07:36,928 (StatisticsMSG.java:219) [INFO][CORE\_STATIS]

2018-11-22 05:07:36,929 (StatisticsOB.java:135) [INFO][CORE\_STATIS]

2018-11-22 05:07:36,929 (StatisticsRPC.java:133) [INFO][CORE\_STATIS]

2018-11-22 05:07:36,984 (HumanGlobalService.java:138) [INFO][GAME] ========= 每分钟打印下在线人数：3 =========

2018-11-22 05:07:50,278 (DataScanProcess.java:128) [INFO][GAME] game-data.jar开始一次扫描key=11390361542802976000,jarValue=11390361542802976000

POST请求：uri=/payNotify,

param={

accountId=603204173,

amount=9800,

quantity=1,

productId=6,

rate=0.0,

orderId=201811220507124f46pw,

roleId=300400000211000012,

amountReal=9800,

sandbox=false,

currency=rmb,

userId=300400000211000012,

paymentType=PPTV渠道

}

## 太乙平台所有地址：

120.92.216.59 商务一服, 

120.92.136.109 

120.92.138.131 

118.89.173.60 

120.92.155.124 

111.231.70.40  

120.92.217.199  

120.92.217.41  

123.206.175.102  

120.92.216.197 

ty10001.qzgame.cc 



## 太乙在线用户和服务器CPU和内存的关系

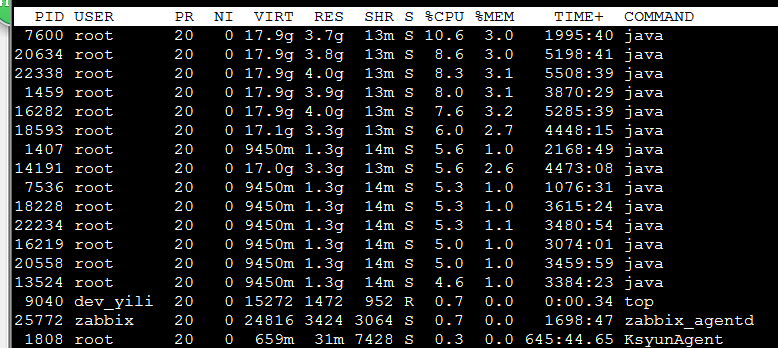
一个游戏服同时在线人数是多少？

一个云服务器上有多个的游戏服，能够配置多少个游戏服？

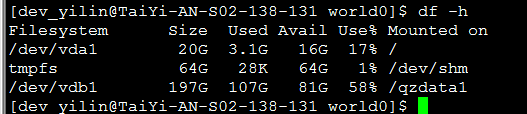
游戏服和数据服是分别放在不同 云服务器上？

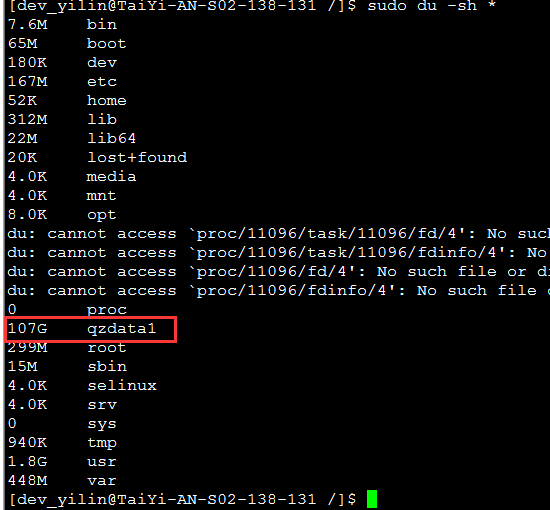
Top 命令：



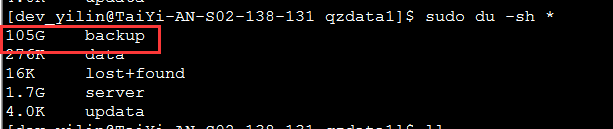
Df –h 命令：



查看各个文件的大小，可以看到qzdata1所占空间最大



进入qzdata1查看，可见backup所占空间最大



Backup里面主要放着的是数据库的备份

## CPU核数

CPU个数=CPU颗数\*CPU核数。

grep -c 'processor' /proc/cpuinfo

16

## 内存：

#cat /proc/meminfo | grep MemTotal

MemTotal: 132157984kB //内存126.035675千兆字节(gb)

## 硬盘大小：

[dev\_yilin@TaiYi-AN-S02-138-131 /]$ sudo fdisk -l | grep Disk

Disk /dev/vda: 21.5 GB, 21474836480 bytes

Disk identifier: 0x000d98f0

Disk /dev/vdb: 214.7 GB, 214748364800 bytes

Disk identifier: 0xd5e01732

## 查看内存？？

8\*16384 MB = 128千兆字节(gb)

[dev\_yilin@TaiYi-AN-S02-138-131 /]$ sudo dmidecode |grep -A16 "Memory Device$"

Memory Device

Array Handle: 0x1000

Error Information Handle: 0x0000

Total Width: 64 bits

Data Width: 64 bits

Size: 16384 MB

Form Factor: DIMM

Set: None

Locator: DIMM 0

Bank Locator: Not Specified

Type: RAM

Type Detail: None

Handle 0x1101, DMI type 17, 21 bytes

Memory Device

Array Handle: 0x1000

Error Information Handle: 0x0000

Total Width: 64 bits

Data Width: 64 bits

Size: 16384 MB

Form Factor: DIMM

Set: None

Locator: DIMM 1

Bank Locator: Not Specified

Type: RAM

Type Detail: None

Handle 0x1102, DMI type 17, 21 bytes

Memory Device

Array Handle: 0x1000

Error Information Handle: 0x0000

Total Width: 64 bits

Data Width: 64 bits

Size: 16384 MB

Form Factor: DIMM

Set: None

Locator: DIMM 2

Bank Locator: Not Specified

Type: RAM

Type Detail: None

Handle 0x1103, DMI type 17, 21 bytes

Memory Device

Array Handle: 0x1000

Error Information Handle: 0x0000

Total Width: 64 bits

Data Width: 64 bits

Size: 16384 MB

Form Factor: DIMM

Set: None

Locator: DIMM 3

Bank Locator: Not Specified

Type: RAM

Type Detail: None

Handle 0x1104, DMI type 17, 21 bytes

Memory Device

Array Handle: 0x1000

Error Information Handle: 0x0000

Total Width: 64 bits

Data Width: 64 bits

Size: 16384 MB

Form Factor: DIMM

Set: None

Locator: DIMM 4

Bank Locator: Not Specified

Type: RAM

Type Detail: None

Handle 0x1105, DMI type 17, 21 bytes

Memory Device

Array Handle: 0x1000

Error Information Handle: 0x0000

Total Width: 64 bits

Data Width: 64 bits

Size: 16384 MB

Form Factor: DIMM

Set: None

Locator: DIMM 5

Bank Locator: Not Specified

Type: RAM

Type Detail: None

Handle 0x1106, DMI type 17, 21 bytes

Memory Device

Array Handle: 0x1000

Error Information Handle: 0x0000

Total Width: 64 bits

Data Width: 64 bits

Size: 16384 MB

Form Factor: DIMM

Set: None

Locator: DIMM 6

Bank Locator: Not Specified

Type: RAM

Type Detail: None

Handle 0x1107, DMI type 17, 21 bytes

Memory Device

Array Handle: 0x1000

Error Information Handle: 0x0000

Total Width: 64 bits

Data Width: 64 bits

Size: 16384 MB

Form Factor: DIMM

Set: None

Locator: DIMM 7

Bank Locator: Not Specified

Type: RAM

Type Detail: None

Handle 0x1300, DMI type 19, 15 bytes

Memory Array Mapped Address

Starting Address: 0x00000000000

Ending Address: 0x000DFFFFFFF

## 太乙游戏在线的一些分析：

========================

## == 游戏运行统计 - 消息协议 ==

========================

### 收到总消息数量2180501次, 高占比列表:

次数 总大小 均大小 数量% 消息ID

323132 3786KB 12B 14.82% 5604

156303 1221KB 8B 7.17% 1201

142122 1387KB 10B 6.52% 2101

82204 802KB 10B 3.77% 5391

76059 965KB 13B 3.49% 10208

75290 588KB 8B 3.45% 10212

75054 2418KB 33B 3.44% 1203

59039 774KB 13B 2.71% 2209

57653 563KB 10B 2.64% 10210

54059 527KB 10B 2.48% 10216

49197 672KB 14B 2.26% 10207

48233 376KB 8B 2.21% 2202

47353 524KB 11B 2.17% 2201

46273 361KB 8B 2.12% 2203

44237 345KB 8B 2.03% 10213

40821 491KB 12B 1.87% 3201

39222 547KB 14B 1.80% 2103

33870 330KB 10B 1.55% 2105

32416 321KB 10B 1.49% 3203

27659 1133KB 41B 1.27% 1502

### 收到总消息大小29MB, 高占比列表:

次数 总大小 均大小 大小% 消息ID

323132 3786KB 12B 12.59% 5604

75054 2418KB 33B 8.04% 1203

142122 1387KB 10B 4.61% 2101

156303 1221KB 8B 4.06% 1201

27659 1133KB 41B 3.77% 1502

76059 965KB 13B 3.21% 10208

7057 884KB 128B 2.94% 111

82204 802KB 10B 2.67% 5391

59039 774KB 13B 2.57% 2209

49197 672KB 14B 2.24% 10207

21130 632KB 30B 2.10% 1912

25888 610KB 24B 2.03% 1606

75290 588KB 8B 1.96% 10212

57653 563KB 10B 1.87% 10210

39222 547KB 14B 1.82% 2103

54059 527KB 10B 1.75% 10216

47353 524KB 11B 1.74% 2201

9396 498KB 54B 1.66% 1610

40821 491KB 12B 1.63% 3201

1669 416KB 255B 1.38% 10511

### 收到消息均大小16B, 高占比列表:

次数 总大小 均大小 最大 最小 消息ID

1669 416KB 255B 1231B 42B 10511

1263 95KB 77B 484B 22B 1617

7057 884KB 128B 284B 50B 111

3010 42KB 14B 248B 12B 4802

192 7KB 37B 136B 13B 1506

27659 1133KB 41B 119B 15B 1502

439 22KB 51B 106B 24B 10405

14 688B 49B 104B 8B 10408

9481 320KB 34B 88B 28B 1626

3741 218KB 59B 78B 30B 1905

538 26KB 50B 66B 37B 121

9396 498KB 54B 63B 35B 1610

4971 204KB 42B 60B 20B 3601

2765 113KB 41B 60B 20B 3603

4724 194KB 42B 60B 20B 3605

2602 107KB 42B 60B 20B 3607

9505 349KB 37B 58B 37B 1608

2221 125KB 58B 58B 58B 1625

21130 632KB 30B 49B 27B 1912

2 76B 38B 47B 29B 5310

### 发送总消息数量11412182次, 高占比列表:

次数 总大小 均大小 数量% 消息ID

1702227 99MB 61B 14.92% 1112

965423 65MB 71B 8.46% 1601

857824 38MB 47B 7.52% 10205

720887 210MB 306B 6.32% 1205

720815 169MB 247B 6.32% 1216

605465 11MB 20B 5.31% 1217

587570 623MB 1112B 5.15% 1902

505423 13MB 29B 4.43% 1119

495187 7MB 16B 4.34% 2001

320364 6MB 20B 2.81% 5605

274604 49MB 188B 2.41% 1504

242043 1997MB 8KB 2.12% 4803

201099 7MB 37B 1.76% 2011

190884 51MB 282B 1.67% 2102

186625 2183KB 11B 1.64% 1114

186235 2206KB 12B 1.63% 10222

173510 1355KB 8B 1.52% 1000

155393 126MB 852B 1.36% 1202

148993 5020KB 34B 1.31% 1204

128058 1834KB 14B 1.12% 10204

### 发送总消息大小3970MB, 高占比列表:

次数 总大小 均大小 大小% 消息ID

242043 1997MB 8KB 50.32% 4803

587570 623MB 1112B 15.70% 1902

720887 210MB 306B 5.30% 1205

720815 169MB 247B 4.28% 1216

155393 126MB 852B 3.18% 1202

23348 108MB 4890B 2.74% 4805

1702227 99MB 61B 2.52% 1112

965423 65MB 71B 1.65% 1601

190884 51MB 282B 1.30% 2102

274604 49MB 188B 1.24% 1504

6788 40MB 6KB 1.03% 1692

75462 39MB 555B 1.01% 10201

15445 39MB 2706B 1.00% 4302

857824 38MB 47B 0.98% 10205

40577 28MB 746B 0.73% 1691

6906 23MB 3625B 0.60% 2205

6129 23MB 4034B 0.59% 5354

11971 22MB 2008B 0.58% 5304

39585 19MB 526B 0.50% 3402

505423 13MB 29B 0.35% 1119

### 发送消息均大小32B, 高占比列表:

次数 总大小 均大小 最大 最小 消息ID

242043 1997MB 8KB 50KB 149B 4803

6788 40MB 6KB 36KB 8B 1692

587570 623MB 1112B 21KB 150B 1902

6906 23MB 3625B 11KB 20B 2205

23348 108MB 4890B 11KB 46B 4805

39585 19MB 526B 6KB 8B 3402

155393 126MB 852B 6KB 8B 1202

274604 49MB 188B 5KB 8B 1504

6129 23MB 4034B 4774B 8B 5354

19708 10MB 543B 4439B 14B 5704

59 168KB 2919B 3363B 77B 5915

216 613KB 2907B 3327B 515B 3507

24 60KB 2573B 3129B 315B 3510

305 674KB 2264B 3086B 67B 3509

2547 6MB 2662B 2985B 60B 3505

1200 3130KB 2671B 2984B 189B 3504

231 515KB 2283B 2874B 63B 3508

15445 39MB 2706B 2859B 1923B 4302

12269 5MB 512B 2849B 8B 3102

11971 22MB 2008B 2693B 152B 5304

2018-11-22 21:50:20,382 (StatisticsOB.java:135) [INFO][CORE\_STATIS]

========================

## == 游戏运行统计 - OB类事件 ==

========================

### Event总耗时1003s, 高占比列表:

耗时 抛次数 接次数 均耗时 耗时% 最大耗时 关键字

291s 142716 142716 2.03ms 29.08% 1804ms 4130

211s 264945 264945 0.79ms 21.10% 1804ms 4146

75s 2081065 2081065 0.04ms 7.57% 249ms 12326

59s 455891 455891 0.12ms 5.92% 4151ms 24578

33s 96528 96528 0.35ms 3.37% 143ms 12329

26s 186685 186685 0.14ms 2.61% 10ms 12325

19s 41208 41208 0.48ms 1.96% 28ms 28678

18s 6796 6796 2.65ms 1.80% 51ms 4128

18s 6755 6755 2.65ms 1.81% 263ms 12292

17s 66608 66608 0.25ms 1.72% 413ms 24593

16s 89145 89145 0.18ms 1.60% 261ms 28674

14s 9576 9576 1.51ms 1.46% 272ms 12321

13s 6774 6774 1.98ms 1.34% 26ms 12294

13s 33954 33954 0.39ms 1.32% 12ms 1638416

12s 88315 88315 0.14ms 1.26% 256ms 4144

10s 33870 33870 0.30ms 1.01% 13ms 36867

9s 39207 39207 0.24ms 0.97% 261ms 36866

9s 4863 4863 1.87ms 0.91% 27ms 16388

9s 33648 33648 0.27ms 0.91% 20ms 1638420

8s 20994 20994 0.39ms 0.81% 21ms 16385

### Msg总耗时2048s, 高占比列表:

耗时 抛次数 均耗时 耗时% 最大耗时 关键字

250s 323052 0.74ms 12.21% 10s game.msg.MsgFengmo$CSFight

173s 155393 1.12ms 8.48% 265ms game.msg.MsgStage$CSStageEnter

168s 46272 3.63ms 8.22% 279ms game.msg.MsgInstance$CSInstanceEnd

106s 10177 10ms 5.22% 1092ms game.msg.MsgCompete$CSCompeteOpen

105s 59038 1.77ms 5.17% 1469ms game.msg.MsgInstance$CSInstanceAuto

98s 25888 3.79ms 4.80% 421ms game.msg.MsgItem$CSEquipIntensify

78s 44236 1.77ms 3.84% 261ms game.msg.MsgTurnbasedFight$CSTurnbasedQuickFight

65s 6295 10ms 3.22% 83ms game.msg.MsgCompete$CSCompeteFight

59s 14039 4.22ms 2.90% 321ms game.msg.MsgHoroscope$CSHoroSelect

57s 39222 1.47ms 2.82% 269ms game.msg.MsgShop$CSShopBuy

51s 142071 0.36ms 2.54% 142ms game.msg.MsgShop$CSShopOpen

41s 21130 1.94ms 2.01% 168ms game.msg.MsgGeneral$CSGeneralAddLevel

38s 33870 1.13ms 1.87% 30ms game.msg.MsgShop$CSShopFlush

35s 47100 0.75ms 1.74% 22ms game.msg.MsgInstance$CSInstanceEnter

30s 9396 3.17ms 1.46% 200ms game.msg.MsgItem$CSAccessoryIntensify

29s 4864 6.07ms 1.45% 84ms game.msg.MsgGeneral$CSGeneralReplace

25s 9481 2.70ms 1.25% 53ms game.msg.MsgItem$CSOnekeyItemEquipReplace

24s 27658 0.88ms 1.20% 64ms game.msg.MsgInform$CSInformChat

22s 32355 0.71ms 1.12% 14ms game.msg.MsgGuide$CSChangeGuidePlotStatus

22s 75047 0.30ms 1.09% 18ms game.msg.MsgStage$CSStageSwitch

2018-11-22 21:50:20,384 (StatisticsRPC.java:133) [INFO][CORE\_STATIS]

========================

## == 游戏运行统计 - RPC调用 ==

========================

### 主动调用总耗时7397s, 高占比列表:

总耗时 次数 均耗时 耗时% 最大耗时 关键字

2939s 302479300 0.01ms 39.74% 1268ms class game.worldsrv.human.HumanGlobalService$55

2167s 2156497 1.00ms 29.31% 10s class game.worldsrv.character.HumanObjectService$31

1070s 928004 1.14ms 14.47% 11s class game.worldsrv.activity.ActivityService$3

596s 11232434 0.05ms 8.07% 10s class core.connsrv.Connection$5

113s 2245396 0.05ms 1.53% 22ms class game.worldsrv.character.HumanObjectService$7

50s 213932 0.23ms 0.68% 1340ms class game.worldsrv.rank.RankGlobalService$1

38s 760 49ms 0.52% 496ms class game.platform.gift.GiftService$2

21s 15449 1.40ms 0.29% 25ms class game.worldsrv.compete.CompeteService$9

19s 46871 0.41ms 0.26% 23ms class game.worldsrv.stage.StageService$4

18s 148994 0.12ms 0.25% 252ms class game.worldsrv.stage.StageObjectService$4

18s 11987 1.45ms 0.25% 1216ms class game.worldsrv.guild.GuildService$21

16s 34418 0.47ms 0.22% 26ms class game.worldsrv.compete.CompeteService$13

15s 148994 0.10ms 0.21% 20ms class game.worldsrv.character.HumanObjectService$27

15s 53848 0.28ms 0.21% 276ms class game.worldsrv.guild.GuildService$68

14s 198408 0.07ms 0.20% 9.14ms class game.worldsrv.compete.CompeteService$27

14s 23590 0.60ms 0.19% 52ms class game.seam.account.AccountService$4

13s 46871 0.28ms 0.18% 11ms class game.worldsrv.stage.StageGlobalService$2

12s 81156 0.15ms 0.17% 15ms class game.worldsrv.guild.GuildService$12

12s 28293 0.43ms 0.16% 11ms class game.worldsrv.human.HumanGlobalService$39

12s 73947 0.16ms 0.17% 283ms class game.worldsrv.stage.StageGlobalService$13

### 请求回调总耗时2284s, 高占比列表:

总耗时 次数 均耗时 耗时% 最大耗时 关键字

1090s 148994 7.24ms 47.74% 12s class game.worldsrv.stage.StageGlobalService$$Lambda$710/763976113

435s 928004 0.46ms 19.08% 12s class game.worldsrv.activity.ActivityManager$$Lambda$650/705778656

116s 88315 1.29ms 5.08% 1804ms class game.worldsrv.general.GeneralManager$$Lambda$744/725541375

70s 88315 0.79ms 3.08% 329ms class game.worldsrv.item.ItemBodyManager$$Lambda$746/1905755641

62s 2103 29ms 2.75% 330ms class game.worldsrv.item.ItemBagManager$$Lambda$961/728298422

55s 9182 6.03ms 2.43% 127ms class game.worldsrv.invade.InvadeManager$$Lambda$983/1648506702

37s 23348 1.61ms 1.65% 72ms class game.worldsrv.activity.ActivityManager$$Lambda$726/2044175413

27s 88315 0.31ms 1.22% 338ms class game.worldsrv.general.GeneralManager$$Lambda$745/1346107846

27s 2304 11ms 1.22% 248ms class game.worldsrv.guild.GuildManager$$Lambda$912/72722170

23s 6179 3.64ms 1.04% 1176ms class game.worldsrv.compete.CompeteManager$$Lambda$774/1577054852

20s 2245306 0.01ms 0.91% 238ms class core.connsrv.Connection$$Lambda$703/529264422

20s 6880 2.87ms 0.88% 278ms class game.worldsrv.activity.ActivityManager$$Lambda$658/193428116

19s 148994 0.13ms 0.87% 250ms class game.worldsrv.stage.StageGlobalService$$Lambda$711/377243086

18s 1658 11ms 0.81% 28ms class game.worldsrv.mine.MineManager$$Lambda$813/771280473

18s 6796 2.71ms 0.81% 52ms class game.worldsrv.human.HumanManager$$Lambda$628/81168588

14s 5990 2.32ms 0.62% 262ms class game.worldsrv.general.GeneralManager$$Lambda$742/974353363

11s 6178 1.25ms 0.52% 4152ms class game.worldsrv.compete.CompeteManager$$Lambda$776/520026074

11s 2446 4.68ms 0.51% 172ms class game.worldsrv.instTrial.InstTrialManager$$Lambda$931/561180277

8s 4252 2.09ms 0.39% 29ms class game.worldsrv.invade.InvadeManager$$Lambda$985/1854544515

8s 81156 0.11ms 0.38% 19ms class game.worldsrv.guild.GuildManager$$Lambda$701/1343509272

# 2018-11-23

# 太乙游戏服在云服务器上的配置

游戏服，数据库