Gihupt账号密码:

账号:a18574392661ssss

密码:tangzhengwu123S

Git：tzw [18574392664@163.com](mailto:18574392664@163.com)

三种方式创建本地仓库

Git init

Hear

小乌龟右键创建

提交

ssh keygen -t rsn

Sql语句练习:https://blog.csdn.net/fashion2014/article/details/78826299

Java项目 eclise

Class文件在bulid下面

Pack环境不显示 nati..显示

Tomcat 开的过多

解决方案https://www.cnblogs.com/guoguochong/p/10641615.html

Js方法传字符串

$(".pagination").append("<li ><a href='#' onclick='queryUserPageList("+(--pageno)+","+pagesize+",\""+$("#loginacctHidden").val()+"\")'>上一页</a></li>");

视图解析器可以至多个 优先级在前面

分布式:

就是把一个项目拆分的多个模块部署在不同的服务器上面 哪个有压力加哪个服务器 解耦合

前台模块:实名认证 后台审核通过 发布项目 查看项目 投资项目

?useUnicode=true&characterEncoding=UTF-8&serverTimezone=GMT%2B8

Springboot项目可以设置多个端口号 哪个启动设置哪个

redis启动命令

<https://www.cnblogs.com/gzdlh/p/8148644.html>

出现:Could not connect to Redis at 127.0.0.1:6379: Connection refused

解决方案:./redis-server redis.conf redis-cli

Redis配置文件redis.conf有两个是因为一个备份覆盖默认

命令:

select 下标 指定数据库 默认0

redis为什么这么快?

1,基于内存的数据库

2,单线程

底层是队列 每次获取一个创建一个单线程 不影响

BIO 一次连接一个线程

NIO 一次请求一个线程

AIO

Session共享(心得)

就是多台服务器通过session存数据保存到redis 通过session来共享

Redis整合sprigboot简单的crud

<https://blog.csdn.net/u012050299/article/details/88311784>

<https://blog.csdn.net/qq_22211217/article/details/80463053>

Exists key 查看键盘存不存在

ttl key 查看过期时间 -1永久 -2已经过期了

Expire key 键的时间

dbsize key 查看键的数量

Flushdb 清空当前库

Flushall 清空所有库

append key value 追加到键后面原来的

strlen 键的长度

setnx 键的长度不存在的时候在设置键的值

Inct key 数量+1

Incr key 数量-1

Inct/incr key 步长(数量)相当于循环到加多少次

Getrange key 开始位置 截取位数

123456 0 4

结果12345

Setrange key 哪个诶位置 值

123456 3 789

123789

Set集合 不能存重复

Hash集合 key field value (适合存储对象)

Redis的原子性

就是所谓操作不会被打断

Redis事务是一个单独隔离操作 事务中的所有命令都会序列化

组成队列

Multi,exec,discard

业务场景:

比如说多个人有一个账号 同时买东西 价格不同

查看文档:

<https://www.runoob.com/redis/redis-lists.html>

随机数不用random(随机种子不是随机) 用math

锁:

悲观锁:性能低 只能自己用 别人用要等待

乐观锁:哪个先执行 哪个升级版本 老版本过时 （乐观锁会失败）

Watch打断事务

### Redis的事务三大特性:

三特性:单独的隔离操作

事务中的所有命令都会序列化,按顺序执行,事务在执行的过程中不会被其他客户端发送的命令打断

没有隔离级别的概念:

队列的命令没有提交之前都不会被执行,因为事务提交任何指令都不会被实际执行,也就不存在,事务内的查询要看到事务的更新,在事务外查询不能看到 这个让人很头疼的问题

不保证原子性:

Redis一个失败 其他的会执行 不会回滚

三个过程:开始事务 命令入队 执行事务

Redis:持久化

RDB

在指定的间隔时间内将内存中数据集写入磁盘,快照直接到内存

配置文件:dump.conf （redis.conf里面配置 76byte就代表没数据）

问题:如果还没到时间段 没达到要求就数据丢失 可以用save手动快照 (生产环境不允许用)

优点:备份块,节省空间

缺点:有可能丢失数据因为是间隔

AOF:已日志的形式记录每个写的操作,

开启Aof redis.conf的 apendonly改为yes

appendfilename aof文件名字

aof跟RDB都开启 优先读取Aof

如果遇到Aof文件损坏,可以通过

redis-check-aof --fix appendonly.aof

1 每秒保存

2 每次添加都保存 性能低

优点:备份更稳健,丢失数据的几率低

缺点:比RDB速度要慢 占用更多的空间

不建议单独用 有可能全部不能恢复

每次读写同步都有一点压力

# 解决性能问题(每次存)

重写机制命令:

**主从复制:**

1:读写分离,容错恢复快

主机写 从机读

可以多级节点 有点分担压力

查看状态: info replication

Slaveof <ip> <port>

哨兵机制：

主机挂了从机变主机

1 优先级 2 数据大小 3 id(集群)

**集群开启:cluster-enabled**

好处:分担压力 实现扩容 高可用

# Activemq：异步,解耦合

生产者 →队列中间件 消费者

1 点对点

2 发布订阅 (先启动消费者订阅才行)

消息类型:TextMessage、MapMessage、BytesMessage、StreamMessage和ObjectMessage。

# <https://www.cnblogs.com/hxun/p/11245987.html>

Linux永久开放端口号:

https://blog.csdn.net/zhe\_csdn/article/details/95789554

# 微服务:

把一个完整的应用垂直拆分成一个个小的服务 独立 独立扩张 远程调用

#粒谷商城项目

电商特点:技术新,技术多,业务广,用户量多,高可用

Springboot+dubbo+mysql+mybatis+redis+git+es+activemq +fts（文件上传分部署）

前后端分离:前台后台分开 单独独立的 互不影响

Npm run dev运行前端项目

(自己linux)启动service fdfs\_trackerd start

service fdfs\_storaged start

# <id> <group\_name> <ip\_or\_hostname>

# 100001 group1 192.168.0.196

# 100002 group1 192.168.0.116

Linux下面带Mod都是测试文件

巨坑:

Eclipse

IDEA

#第六天:小难点

查询sku信息 多张图片

根据sku拿到spuid 查询spu的属性 属性值 显示

每个sku对应自己的属性值

去sku存在表里面查询它包含的属性跟属性值

Left join 包含 选中

处理高并发:

缓存 nginx负载均很 限制流量

# 业务重点:

1. redis的击穿 高并发情况下某个key失效或者过期 给DB造成压力

解决办法:使用redis自带的数据库锁 nx

或者使用Redission的lock();

1. 穿透 通过不存在的key一直访问 造成DB压力 不存在设置””值 (增删改会清空)
2. 雪崩 高并发情况下多个个key失效或者过期 给DB造成压力 解决办法给key设置时间

Es:put delte post get

PUT /movie\_index/movie/2

{

"id":2,

"name":"xuexiaoxue张三nimabi",

"acList":[

{"id":1,"name":"xiaoming","age":15},

{"id":2,"name":"xzing","age":12}

]

}

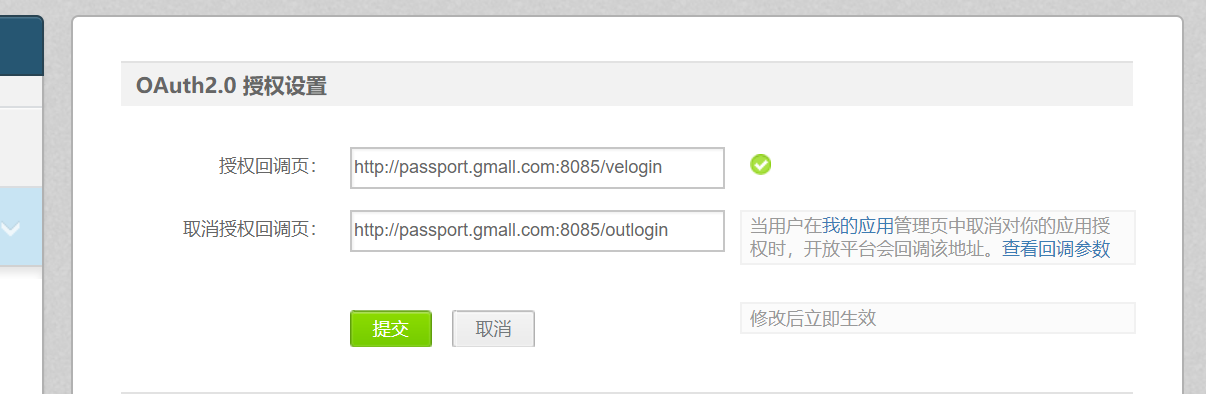
Es6中一个索引index只能建一个type(表)

Es的数据类型 long boolearn doble text(可拆分) keywork(不可拆分)

"analyzer": "ik\_max\_word",

我的新浪微博账号:18574392661 pwd:tangzhengwu123S



授权的路径(开发者告诉客户)

2.00AA1KEIujyQCC30493065bag9TKfD

防止订单重复提交；

每次去订单页面 都保存一个商品码 hidden 提交比较 存在提交去支付页面 不存在重复了

Lua脚本解决并发问题

订单编号川过去 总价格

返回支付表单

添加支付记录

支付成功以后 修改叮当 减少库存 修改用户支付的信息 redis订单改

分布式事物:

在分布式的情况下,保持数据的一致性.

#总体练习 登录 jwt第三方登录 redis 击穿穿透雪崩 mq事务 队列解耦

Activemq 发布了服务 订阅他的(监听)会同时只需 要么全部成功 回滚

二. 幂等性检查

调用相同的服务,返回的结果始终一样,叫做幂等性检查.

解决:如果是对DB操作 那可以拿到id作为中间 添加会冲突

2 可以使用id为key存到redis 有的话return 没有添加

如果防止消失丢失

1 使用事务

2 持久化(三种)

3 手动确认消息

什么是死信队列:重新发消息失败超过指定的次数 (默认6次)

Activemq.xml里面配置

ActiveMQ的无法数据类

TextMessage, map,object,bytes,stearm

Mq数据消息持久化

基于内存(重启消失) 日志文件 DB(消费完成就删除数据库)

确认消息 手动 自动 延时

持久消息才会进入了死队列(发送失败)

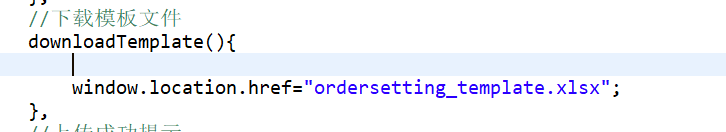
Springboot的手动确定消息是4

Git

拉取 提交 创建分支

合并 解决冲突

下载文件第一种



第二种 生成模板已IO流写出

页面静态化(理解了完全)

适用于不经常改变的数据的小网站

增删改的时候查询了数据通过th模板生成到静态页面到指定的路径下面

每次访问通过nginx直接放回静态页面(静态页面生成到nginx目录下面)

没有就返向代理到服务器访问数据库