# Git

## 安装git之后,桌面图标出现蓝色问号

解决方法：这是因为我们在桌面创建了版本库了.

这个时候我们在系统中吧隐藏的文件夹显示出来.这个时候会看到桌面上有一个隐藏的git文件夹.把这个文件夹删除掉之后,刷新一下就可以了.

## 检出地址

[git@git.dealmoon.net:api\_java](mailto:git@git.dealmoon.net:api_java)

[git@git.dealmoon.net:libs\_java](mailto:git@git.dealmoon.net:libs_java)

git@git.dealmoon.net:framework

# 帐号密码

Redmine地址：<http://pm.dealmoon.net:8080>

验证：dm secret

账号：wangyunewage123

deploy数据库访问地址：<http://deploy.dealmoon.net/adminer.php>

服务器地址不用写

验证：dmsecret

dealmoon库dm bookface06

dm\_ucenter库 sso bookface06

192.168.1.202 数据库帐号、密码：writeuser 1234

192.168.1.224root：dealmoon Java用户有jdk

后台地址 https://uc.test.dealmoon.net/admin

AdminhxWaDbQt

192.168.1.230

线上数据库服务器地址：198.23.88.245

数据库帐号、密码：dui databaseuserinterface

192.168.1.202数据库：writeuser 1234

新机器数据库

http://106.184.4.162/adminer.php

后台帐号：

认证口令：test: ZqjazndG

后台账号：testadmin: sMgPW5hw

后台评论管理口令：moondeal m00nblackfriday

http://admin.dealmoon.com/a/deals/publish?type=haitao

beta后台

admin.beta.dealmoon.net/a  
 dm  secret  
 xliu  aaaaaa

# 测试

手机上安装app客户端，并设置代理为本机IP，端口号为8888

电脑装上Charles软件

谷歌浏览器安装postman插件

把所有的配置文件的ip改为192.168.1.202

这台虚拟机上装了mysql和redis

Redis是缓存服务

Postman是谷歌浏览器的插件，需要翻墙到工具—>扩展程序中下载，下载client版本，请求方式设置成post，用于模拟页面提交参数，查看应用程返回的数据。

打开SDKserver服务，访问<http://127.0.0.1:8080>，并提交参数

Charles默认监听8888端口，手机app发送请求（手机设置代理http://本机ip，设置端口号为8888），用charles获取请求参数。

打开appServer服务，手机app去访问，可用charles获取请求参数。App的参数用MD5加密。

带main方法的server类开启了线程监听请求，开启该服务需要传递参数。

测试线下接口需要绑定IP和域名

C:\Windows\System32\drivers\etc\hosts

iphone app开发模式

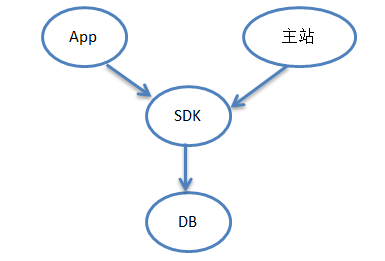
dealmoondebugmode

dealmoon

# 交互过程

旧版本是主站直接访问DB

新版本增加SDK封装，app和主站访问SDK，SDK去调用DB

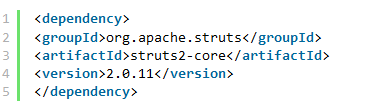


# Maven

Maven是一个项目管理工具，它包含了一个项目对象模型 (Project Object Model)，一组标准集合，一个项目生命周期(Project Lifecycle)，一个依赖管理系统(Dependency Management System)，和用来运行定义在生命周期阶段(phase)中插件(plugin)目标(goal)的。

在pom.xml文件中添加Maven所依赖的Jar的名称，即<dependency></dependency>节点。

例如：添加大家常用的Struts Jar包



<properties>

      <file.encoding>UTF-8</file\_encoding>

      <java.source.version>1.5</java\_source\_version>

      <java.target.version>1.5</java\_target\_version>

</properties>

使用方式如下：${file.encoding}

Repositories：pom里面的仓库与setting.xml里的仓库功能是一样的。主要的区别在于，pom里的仓库是个性化的。比如一家大公司里的setting文件是公用的，所有项目都用一个setting文件，但各个子项目却会引用不同的第三方库，所以就需要在pom里设置自己需要的仓库地址。

## Pom.xml文件

<http://blog.csdn.net/zhuxinhua/article/details/5788546>

<http://www.blogjava.net/jianyue/articles/227932.html>

详细内容参考：

<http://baike.baidu.com/link?url=GI5OjEOmny9H2s-3BjDuk90fiEQqKy9T1-avhNCyT1D2h89AdyJP-Hy9WJdYI68ZtCt94ZXEO6IYExQm1uLEvq>

## GroupID和AtifactID

GroupID是项目组织唯一的标识符，实际对应JAVA的包的结构，是main目录里java的目录结构。

ArtifactID就是项目的唯一的标识符，实际对应项目的名称，就是项目根目录的名称。

一般GroupID就是填com.leafive.test这样子。

## 相关文章

<http://blog.csdn.net/smilevt/article/details/8215558/>

<http://blog.csdn.net/qjyong/article/details/9098213>

<http://huxiaoheihei.iteye.com/blog/1766986>

创建动态的、静态的Maven项目，项目的导入、运行：

<http://blog.csdn.net/jackgaolei/article/details/11332249>

Maven项目的配置文件

<http://blog.csdn.net/defonds/article/details/8004821>

在添加src/main/java，src/test/java ，src/test/resources三个文件夹时，会报错

Cannot nest 'ss/src/main/java' inside 'ss/src/'. To enable the nesting exclude 'main/' from 'ss/src/'；

解决方法：

点击改工程的Properties属性选中Java Build Path项，找到source，发现有两个错误的包，将其选中remove掉；

重新在项目下创建source folder

# Eclipse

## eclipse没有project facets选项

<http://hi.baidu.com/jayson1987/item/6437ef2b2517c142469962d3>

## dynamic web project与web project

dynamic web project是Eclipse的项目，与其对应的有static web project，前者指动态web项目，包含一些动态代码，如java；而static web project则仅包含静态页面文件。

MyEclipse对Eclipse进行了扩展，如：web project可以添加一些开源的框架支持，比如Struts、Hibernate等等。

也就是说，web project是Myeclipse扩展后的项目，而dynamic web project是Eclipse自带的分类，在Myeclipse中，web project具有dynamic web project特性并具有一些方便开发的集成功能。

# Dropwizard

参考文章<http://dreamhead.blogbus.com/tag/dropwizard/>

# MyBatis

参考资料：<http://mybatis.github.io/mybatis-3/zh/index.html>

enctype="multipart/form-data"的意思，是设置表单的MIME编码。默认情况，这个编码格式是application/x-www-form-urlencoded，不能用于文件上传；只有使用了multipart/form-data，才能完整的传递文件数据，进行下面的操作.

JTestCase

## MyBatis/Ibatis中#和$的区别

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。如：order by #user\_id#，如果传入的值是111,那么解析成sql时的值为order by "111", 如果传入的值是id，则解析成的sql为order by "id".

2. $将传入的数据直接显示生成在sql中。如：order by $user\_id$，如果传入的值是111,那么解析成sql时的值为order by user\_id,  如果传入的值是id，则解析成的sql为order by id.

3. #方式能够很大程度防止sql注入。

4.$方式无法防止Sql注入。

5.$方式一般用于传入数据库对象，例如传入表名.

6.一般能用#的就别用$.

ps:在使用mybatis中还遇到<![CDATA[]]>的用法，在该符号内的语句，将不会被当成字符串来处理，而是直接当成sql语句，比如要执行一个存储过程。

String[] ids = {"21","22","23"};

String \_ids = StringUtils.join(ids, ",");

\_ids 是 “21，22，23” 每个元素没有引号，只有最外层一个引号

如果\_ids是由int型组成的字符串且匹配项为int：

in(#{\_ids}) 得到的是 in('21,22,23 ') 可以取出第一条数据

即：select \* from t\_user where id in ('21,22,23') 可以取出id=21 的数据

in(${\_ids}) 得到的是 in(21,22,23) 可以取出三条数据

即：select \* from t\_user where id in (21,22,23)

如果\_ids是由String型组成的字符串且匹配项为String：

in(#{\_ids}) 得到的是 in('aa,bb,cc ') 匹配不到数据

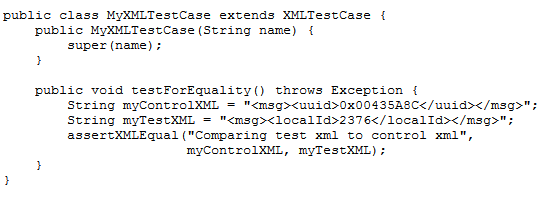
即：select \* from t\_user where userName in ('aa,bb,cc')

in(${\_ids}) 得到的是 in(aa,bb,cc) 取不出数据，报错

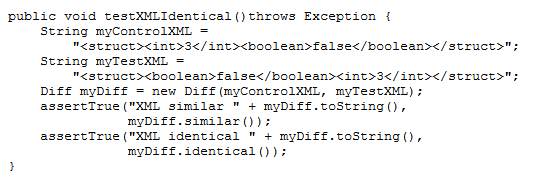
即：select \* from t\_user where userName in (aa,bb,cc) 报错

无论是int还是String组成字符串，用foreach都正确

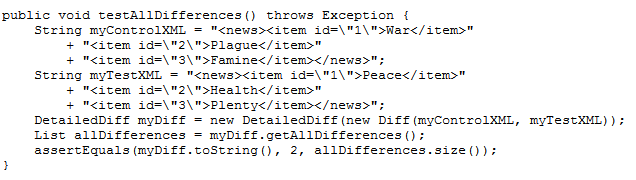
# XMLUnit常用方法



下面的方法只能比较出第一处不同的地方



下面的方法可以比较出所有的不同之处



ElementQualifier

ElementNameQualifier

ElementNameAndTextQualifier

ElementNameAndAttributesQualifier

Transform

# jsonlib

JSONObjectdataJson=new JSONObject("你的Json数据“);  
JSONObject response=dataJson.getJSONObject("response");  
JSONArray data=response.getJSONArray("data");  
JSONObject info=data.getJSONObject(0);  
String province=info.getString("province");  
String city=info.getString("city");  
String district=info.getString("district");  
String address=info.getString("address");  
System.out.println(province+city+district+address);

# jackson基本用法

http://www.blogjava.net/wangxinsh55/archive/2012/09/24/388420.html

# Mysql

## 设置主从数据库

设置mysql远程以root权限访问

grant all privileges on \*.\* to root@'%' identified by '12345678' with grant option;

flush privileges; (运行此句才生效，或者重启MySQL)

GRANT ALL PRIVILEGES ON \*.\* TO ‘root’@'192.168.1.212' IDENTIFIED BY

'bookface06’ WITH GRANT OPTION;

Spring配置读写分离：

<http://blog.csdn.net/huoyunshen88/article/details/36674861>

<http://wenku.baidu.com/link?url=XRL6mBT__9o0ebqZnu6WqCy03DY0yEBhnr62kdUAPMjp0s-tO-ZRJ66Rsclm-uxVagOCE1N1oDRWE_11Qpg7e3FlNpA-u8dzqhJxS6hX7R3>

<http://blog.csdn.net/keda8997110/article/details/16827215>

mysql设置主从数据库：

<http://blog.sina.com.cn/s/blog_8a00577d0101i1up.html>

<http://www.cnblogs.com/luckcs/articles/2543607.html>

Java自定义注解（Annotation）

<http://www.cnblogs.com/peida/archive/2013/04/24/3036689.html>

主数据插入中文，从数据库不更新，同步时报错，从数据库和主数据库编码格式不一致（两处编码），如果之前设置编码格式错误，更改编码格式（注意打勾），把数据库删掉重建、

有dbcp和c3p0常用的数据源

SqlSessionTemplate

LazyConnectionProxy + RoutingDataSource +   Plugin

create table user(id int primary key auto\_increment,userNamevarchar(40),userAgeint,userAddressvarchar(40));

　$ ls -l sobsrc. tgz

chmod ［who］ ［+ | - | =］ ［mode］ 文件名

## InnoDB与Myisam的区别

http://www.cnblogs.com/vicenteforever/articles/1613119.html

optimize table整理表的碎片

## mysql忽略主键冲突，避免重复插入的几种方式

http://zhidao.baidu.com/link?url=OwAdVVrNnphXQT9PSEZm1RiivNPKW-ZroX7DMAc\_m7EZLpGBjCZGhlZj7k7X-QvJauPb4n79m5okCC7vua3WpFHYqa0KYenqdfeU3LiDFnW&qq-pf-to=pcqq.c2c

# Linux配置antjunit

1. 安装JDK

http://johnsz.blog.51cto.com/525379/694052

配置环境变量

<1># vi /etc/profile

<2>在最后加入以下几行：

export JAVA\_HOME=/usr/java/jdk1.6.0\_10

exportCLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

<3># reboot ← 重启机器配置生效

1. 下载junit-4.9.jar

http://blog.csdn.net/blueram/article/details/10357157

http://www.linuxidc.com/Linux/2011-08/41737.htm

把下载的junit-4.9.jar放入Java\_home路径下的lib文件夹里

在CLASSPATH下加入$JAVA\_HOME/lib/junit-4.9.jar:$CLASSPATH

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JAVA\_HOME/lib/junit-4.9.jar:$CLASSPATH

运行单个测试用例用 java xxx

1. 下载Ant apache-ant-1.9.4-bin.tar.gz

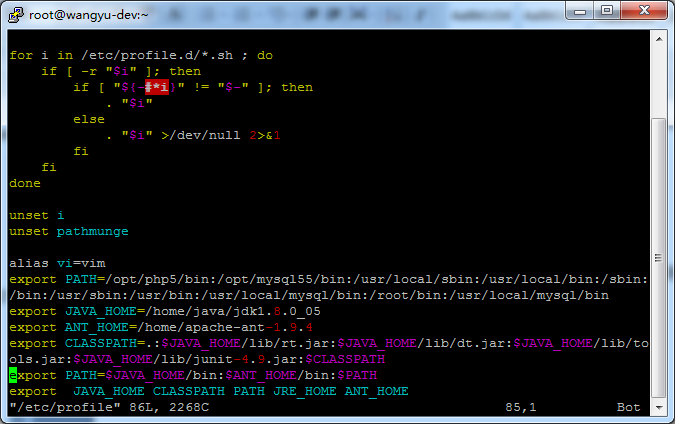
解压tar -zxvf apache-ant-1.9.2-bin.tar.gz

jdk ant 统一都安装 / home/java下

export ANT\_HOME=/usr/ant/apache-ant-1.8.2 同时把刚刚下载的junit-4.9.jar放到里面的lib文件里

PATH下加入$ANT\_HOME/bin:$PATH

最终环境变量为：



查看ant是否安装成功ant -version

拷贝项目到虚拟机，进入工程目录

ant运行build.xmlant -f build.xml

INSERT INTO login\_info (UId, Username, Ip, Code, Login\_time, Count)  
(SELECT id, name,ip,token, NOW(), 1 from user)

java -jar name

创建一个Java类文件的名称TestRunner.java在C：\> JUNIT\_WORKSPACE执行测试案例

可以自定义测试运行器

import org.junit.runner.JUnitCore;

import org.junit.runner.Result;

import org.junit.runner.notification.Failure;

public class TestRunner {

public static void main(String[] args) {

Result result = JUnitCore.runClasses(JunitTestSuite.class);

for (Failure failure : result.getFailures()) {

System.out.println(failure.toString());

}

System.out.println(result.wasSuccessful());

}

}

# Quartz作业调度框架

Map<Integer, String> map = new HashMap<Integer, String>();

先拿所有的键：

Integer[] keys = map.keySet().toArray(new Integer[0]);

然後随机一个键，找出该值：

Random random = new Random();

Integer randomKey = keys[random.nextInt(keys.length)];

String randomValue = map.get(randomKey);

//getRepeatInterval()指：执行任务间的间隔时间,getRepeatCount()表示总共执行多少次,-1表示未知

SHOW CREATE VIEW 视图名

在abator中可生成iBatis的代码。其中，Sql\_map中带有的两个函数是：

updateByPrimaryKeySelective

updateByPrimaryKey

前者只是更新新的model中不为空的字段。

后者则会将为空的字段在数据库中置为NULL。

# 代码结构

dmsdk controllers --> model -->dao和mapper

dao和mapper地位相当，dao没有用mybatis，访问的dealmoon数据库，mapper主要是ugc的，用的mybatis，访问ugc和ucenter两个库。

controllers是控制层，model是业务逻辑层，dao和mapper是数据层

entity是实体类

cache是缓存

commandinfo定义的请求指令

data定义的请求和返回指令

dbutil定义的数据库操作

# crontab

192.168.1.230 root dealmoon crontab -l crontab -e

查看crontab日志 tail /var/log/cron\*

用以下的方法启动、关闭这个cron服务：

/sbin/service crond start //启动服务

/sbin/service crond stop //关闭服务

/sbin/service crond restart //重启服务

/sbin/service crond reload //重新载入配置

查看crontab服务状态：service crond status

手动启动crontab服务：service crond start

\*/1 \* \* \* \* /home/java/jdk1.8.0\_05/bin/java -jar

/home/java/statistics.jar

\*/1 \* \* \* \* cd /home/java/autolike && java -jar autolike.jar

\*/5 \* \* \* \* cd /root/RedisLive/src/; python /./redis-monitor.py --duration 5

# redis

/opt/redis/bin/redis-cli 进入redis  
keys xxxx\*  
get xxx\*  
set xxxx  8  
incr(key) value值自增

ps -ef | grep redis

/opt/redis/bin/redis-server /opt/redis/conf/redis.conf

maven jar包

<dependency>

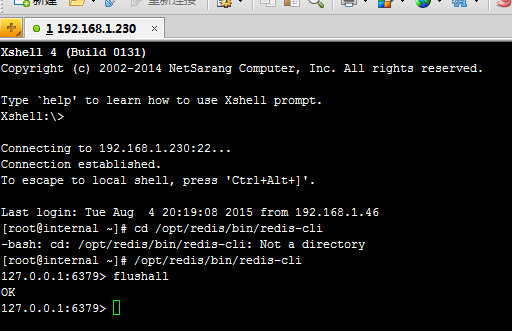
<groupId>redis.clients</groupId>

<artifactId>jedis</artifactId>

<version>2.6.0</version>

</dependency>

清缓存



1、redis-benchmark   
redis基准信息，redis服务器性能检测   
**redis-benchmark -h localhost -p 6379 -c 100 -n 100000**   
100个并发连接，100000个请求，检测host为localhost 端口为6379的redis服务器性能

2、redis-cli   
**redis-cli -h localhost -p 6380 monitor**   
Dump all the received requests in real time; 监控host为localhost，端口为6380，redis的连接及读写操作

**redis-cli -h localhost -p 6380 info**   
Provide information and statistics about the server ; 提供host为localhost，端口为6380，redis服务的统计信息

3、redis-stat   
**redis-stat host localhost port 6380 overview**   
Print general information about a Redis instance; 实时打印出host为localhost，端口为6380，redis实例的总体信息

**redis-stat host localhost port 6380 overview**   
Measure Redis server latency; 输出host为localhost，端口为6380，redis服务中每个请求的响应时长

redis监控工具：

Redis-rdb-tools

http://www.ttlsa.com/python/redis-rdb-tools-analysis-of-reids-dump-file-and-memory-usage/

angel

http://www.oschina.net/p/angel

redis-live

http://www.nkrode.com/article/real-time-dashboard-for-redis

# linux指令

在Linux（unix）中，以波浪线“~”开始的文件名有特殊含义。单独使用它或者其后跟一个斜线（~/），代表了当前用户的宿主目录（在shell下可以通过命令“echo ~(~\)”来查看，在slackware上要用 "echo ~"来查看）。例如“~/bin”代表“/home/username/bin/”（当前用户宿主目录下的bin目录）。波浪线之后跟一个 单词（~word），代表由这个“word”所指定的用户的宿主目录。例如“~john/bin”就是代表用户john的宿主目录下的bin目录。

screen -S [name]

screen -ls

screen -r [name/pid]

exit 终止

ctrl + a d

ss -tnl

ss -tnlp

ps aux |grep 2363

lsof -i:8533

kill pid

curl -d '{"command":"Search/indexDeal","commandInfo":{"id":467736},"lang":"en","udid":"","type":"email","ip":"192.168.1.96","ua":"Mozilla/5.0","editorId":1}' 127.0.0.1:8080

curl -H "Content-Type:application/json" -X POST --data '{"mobileInfo":{"device":"iPhone","height":"2208","udid":"25D36CAD-2DC6-44BF-BE3D-1A742771A0C8","model":"iPhone7,1","width":"1242","systemVersion":"9.1","accessToken":"a9d6c7842e9bc57f37","scale":"3"},"protocol":"1.0.0","command":"search\/search","protocolType":"request","token":"125105|a66ab3b585f1fb5a6384fb6ee99b12e4","clientInfo":{"platformId":"10006","softLanguage":"31","productId":"1000","editionId":"5.5.0 Build 1"},"commandInfo":{"pageNum":"1","pageSize":"20","expired":false,"type":"deal","keyword":" 海淘攻略"},"serviceInfo":{}}' 127.0.0.1:8081

curl -H "Content-Type:application/json" -X POST --data '{"command": "Deal/search","lang": "en","udid": "","type": "email","ip": "127.0.0.1","ua": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.85 Safari/537.36","editorId": "","commandInfo":{"keyword": "海淘攻略","category": "All","expired": false,"order": {"published\_date": "desc"},"pageNum": 1,"pageSize": 80,"isHaitao": false,"language": ["cn\_body"],"index": "deal\_cn","showByLocation": ["all","cn"]}}' 127.0.0.1:8080

# 部署

su java

cd~

./deploy\_java/deploy deploy test\_ugc\_438

切换到Java用户,去宿主目录（cd ~）执行下面指令

scd

tail -f rundir/dmsdk1/logs/rest.log

查看sql语句输出：tail -f /opt/mysql-5.5.35/log/general\_query.log

后台地址：

http://www12.test.dealmoon.net/v2/admin/index.php?r=Post/index

帐号：xliu

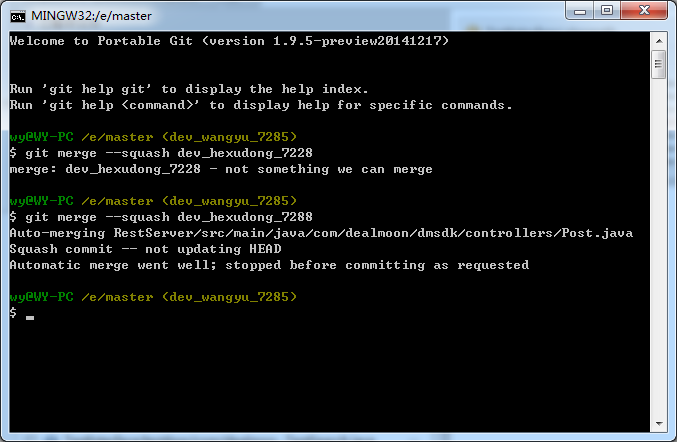
密码：aaaaaa

合并分支到自己的分支

root@50.116.3.240  deploy1

wangyu@50.116.3.240

173.255.246.189 java用户



1. java 运行目录结构

/home/java/rundir/

|-- jar

| |-- appapi-0.0.1-SNAPSHOT.jar --> appapi-0.0.1-${tag}.jar

|-- app

| |-- app1-${tag}

| |-- app2-${tag}

| |-- appN-${tag} ...

|-- dmsdk

| |-- dmsdk1-${tag}

| |-- dmsdk2-${tag}

| |-- dmsdkN-${tag} ...

|-- version

|-- oldversion

|-- appN ... --> ${rundir}/app/appN-${tag} #N为实例数

|-- dmsdkN... --> ${rundir}/dmsdk/dmsdkN-${tag}

NOTE: 其中 appapi-0.0.1-SNAPSHOT.jar、appN...、dmsdkN... 是软连接，连接到当前运行API的实际目录上，不用人手动干预，部署程序自动生成和调整

2. 部署

部署脚本: deploy

启动脚本：startapi

配置文件: main.conf

这三个文件最好在一个目录下

3. 配置文件

instance=2 #指定当前要部署和运行的实例数，部署前必须先给此实例准备好配置文件，放在${rundir}/(app|dmsdk)/目录下，命名规则: app${instance}-${tag}，第一次可命名为app${instance}-old

autorestart=yes/no #部署完后，是否要自动重启所有实例

rhosts="IP1 IP2 IP3" #指定需要远程部署的机器IP，如果不指定，则不进行远程部署,如指定多个注意在引号内用空格隔开

4. 使用方法

deploy 使用方法：

Usage: ./deploy deploy tag #部署

./deploy makelink tag #修改软连接，可快速回滚到任意版本

./deploy rollback #快速回滚到上次部署

startapi 使用方法:

Usage:

./startapi [start|stop|restart|status] app #启动和关闭所有app实例

./startapi [start|stop|restart|status] dmsdk #启动和关闭所有dmsdk实例

./startapi [start|stop|restart|status] {app1|app2|...} #启动和关闭某一个app实例

./startapi [start|stop|restart|status] {dmsdk1|dmsdk2|...} #启动和关闭某一个dmsdk实例

5. 注意：

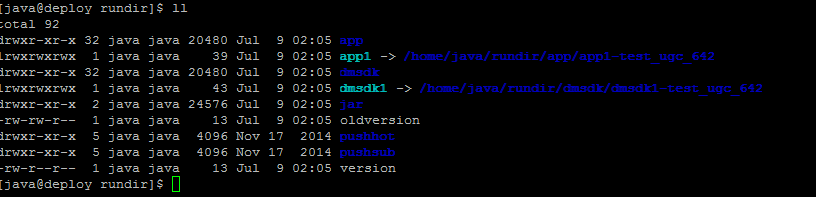
5.1 若要启动新实例，必须先在 ${rundir}/(app|dmsdk)/ 新建此实例的配置文件。

如增加 app3实例，则先新建 ${rundir}/app/app3-old 或${rundir}/app/app3-${previousTag} 目录，将修改过的配置文件copy到此目录下，在编辑 main.conf 配置文件改"instance=3" 这样实例便能正常被部署和启动，否则会因找不到配置文件，而忽略此实例的部署和启动。

5.2 远程部署需要两台机器提前配置好基于java用户的SSH互信(秘钥认证).

5.3 远程部署必须要将这一套脚本也放在远程的机器上，放在 /home/java/deploy\_java/ 目录下.

查看部署的是哪个版本sdk



# 开源类库整理

<http://blog.jobbole.com/> （job伯）

<http://runjs.cn/code>（在线js编辑）

<http://www.searchtb.com/?spm=0.0.0.0.fCArxj>（淘宝搜索博客）

<http://stblog.baidu-tech.com/>（百度搜索博客）

<http://blog.fnil.net/>（庄周梦蝶博客）

<http://www.alidata.org/p_d/tech>（阿里数据平台团度）

<http://www.chinaunix.net/>（china unix）

<http://tech.163.com/school/>（网易学院）

<http://rdc.taobao.com/team/jm/> 淘宝中间件

<http://www.mhtml5.com/> html5

<http://nec.netease.com/>（页面css框架）

<http://jsbin.com/> 正则表达式

[www.oschina.net](http://www.oschina.net/)

[http://sourceforge.net](http://sourceforge.net/)

[http://www.codeplex.com](http://www.codeplex.com/)

<https://github.com/>

<https://code.google.com/intl/zh-CN/>

<http://apache.org/>

作业相关

<http://www.oschina.net/p/alibaba-zeus> 淘宝作业平台 （<https://github.com/alibaba/zeus/wiki> 官网）

<http://www.oschina.net/p/quartz> 定时执行框架 （<http://www.cnblogs.com/phinecos/archive/2008/09/03/1282747.html> 快速入门）

<http://www.sauronsoftware.it/projects/cron4j/> 类似quartz（<http://www.sauronsoftware.it/projects/cron4j/manual.php> 文档）

web开发框架

<http://www.oschina.net/p/jfinal>

<http://www.oschina.net/p/nutz>

<http://www.oschina.net/p/argo>（58同城 <https://github.com/58code/Argo> 官网）

<http://www.fireflysource.com/>

<http://www.playframework.com/> playframework

脚本语言

<http://www.oschina.net/p/clojure> java方言（<http://clojure.org/>  官网，<http://huangz.iteye.com/blog/1325228> 快速入门）

<http://www.oschina.net/p/ruby> ruby（<http://www.ruby-lang.org/en/documentation/> 官网，<http://www.ostools.net/apidocs/apidoc?api=ruby-library> 文档）

<http://www.lua.org/> lua（<http://www.ostools.net/apidocs/apidoc?api=lua> 文档，<https://github.com/nrk/redis-lua> redis的lua客户端）

<http://www.jython.org/> python的java实现（<http://www.jython.org/Project/userguide.html> 文档）

<http://groovy.codehaus.org/Chinese+Home> java方言

<http://www.erlang.org/> erlang语言

web服务器

<http://www.oschina.net/p/tomcat> tomcat

<http://www.oschina.net/p/lambdaprobe> tomcat manager的增强版

<http://www.oschina.net/p/jetty>

<http://www.oschina.net/p/resin>

<http://www.oschina.net/p/apache+http+server>

<http://www.oschina.net/p/nginx> http服务器

<http://tengine.taobao.org/> 淘宝web服务器，类nginx

<http://www.lighttpd.net/> lightpd

基础类库

<http://jodd.org/doc/index.html> java常用操作类

<https://code.google.com/p/guava-libraries/> ，<https://code.google.com/p/google-collections/> google的java常用操作类

<https://github.com/facebook/jcommon> facebook的java常用操作类

<http://www.oschina.net/p/commons-daemon> java转后台服务

<http://asm.ow2.org/doc/developer-guide.html#examples> java原生码编辑

<http://www.oschina.net/p/httpclient> java的http协议实现

日志分析

<http://flume.apache.org/> apache flume（start文档<https://cwiki.apache.org/confluence/display/FLUME/Getting+Started>）

性能检测

<https://github.com/alibaba/TProfiler> 淘宝性能分析（<https://github.com/alibaba/TProfiler/wiki> 文档）

<http://jmeter.apache.org/index.html> java性能检测

消息队列

<http://www.oschina.net/p/activemq> （<http://www.cnblogs.com/xwdreamer/archive/2012/02/21/2360818.html> 入门）

<http://www.oschina.net/p/kafka>（<http://www.oschina.net/translate/kafka-design> 架构）

<https://github.com/alibaba/metaq> metaq（<https://github.com/alibaba/metaq/wiki> 文档）

模板引擎

<http://www.oschina.net/p/velocity>  oschina采用的模板（<http://velocity.apache.org/engine/releases/velocity-1.6.1/user-guide.html> 官网）

<http://www.oschina.net/p/beetl> （<http://ibeetl.com:8080/beetlonline/> 官网）

<http://www.oschina.net/p/freemarker>（<http://freemarker.org/docs/index.html>）

JSON

<http://www.oschina.net/p/fastjson>**java的json解析**（<http://www.cnblogs.com/zhenmingliu/archive/2011/12/29/2305775.html> 用法，<http://www.iteye.com/topic/1113183>，技术内幕）

<http://www.oschina.net/p/google-json> google的json解析

XML

<http://www.oschina.net/p/dom4j> dom4j

<http://www.oschina.net/p/xerces> 有c++版本

分布式

<http://www.oschina.net/p/zookeeper> [分布式系统协调](http://www.oschina.net/p/zookeeper) （<http://stblog.baidu-tech.com/?p=1164> 原理，<http://www.ibm.com/developerworks/cn/opensource/os-cn-zookeeper/> 权威，<http://zookeeper.apache.org/doc/r3.3.2/zookeeperOver.html?spm=0.0.0.0.fCArxj> 概述）

<http://www.oschina.net/p/hadoop> hadoop（<http://hadoop.apache.org/> 官网，<http://subject.csdn.net/hadoop/> 博客，<http://www.cnblogs.com/skyme/archive/2011/10/26/2223984.html> 入门）

nosql数据库

<http://www.oschina.net/p/couchdb>

<http://pig.apache.org/> sql生成map-reduce

<http://www.oschina.net/p/cassandra>

<http://www.oschina.net/p/mongodb>

<http://www.oschina.net/p/lealone> 淘宝分布式sql引擎（<https://github.com/codefollower/Lealone> 文档）

<http://code.taobao.org/p/OceanBase/wiki/index/> 淘宝oceanbase

<http://www.oschina.net/p/sqlite> 嵌入式小型数据库

缓存

<http://www.oschina.net/p/memcached>

<http://www.oschina.net/p/redis>

<http://www.oschina.net/p/tair> 淘宝缓存

<http://www.oschina.net/p/ehcache> java缓存框架（<http://www.ehcache.org/documentation/user-guide/getting-started> 在线文档）

ORM

<http://www.oschina.net/p/mybatis>（<http://www.ibm.com/developerworks/cn/java/j-lo-ibatis-principle/> 深入分析）

<http://www.oschina.net/p/dbutils>（简单封装）

<http://www.oschina.net/p/sql2o>（简单封装，<http://www.sql2o.org/learn> 官网）

数据库连接池

<http://www.oschina.net/p/c3p0>

<http://www.oschina.net/p/druid> 监控

<http://www.oschina.net/p/dbcp>

<http://www.oschina.net/p/bonecp> 最快？

RPC（<http://blog.163.com/xiechunhao@126/blog/static/110411383201073102017368/>）

<http://www.oschina.net/p/dubbo>（淘宝）

<https://github.com/58code/Gaea>（58同城）

<http://avro.apache.org/>  数据序列化

<http://bgcc.baidu.com/tech.html> 百度多语言，跨平台rpc框架

实时计算

<http://www.oschina.net/p/twitter-storm>

设计相关

<http://www.oschina.net/p/guice> 轻量ioc（<http://www.blogjava.net/xylz/archive/2009/12/22/306955.html> 入门，<https://code.google.com/p/google-guice/downloads/list> pdf文档）

<http://www.oschina.net/p/aspectj> 面向切面 （<http://blog.csdn.net/starchu1981/article/details/15557> 分析，<http://www.eclipse.org/aspectj/doc/released/progguide/starting.html> 官方文档）

java nio

<http://www.oschina.net/p/grizzly>（<http://www.2cto.com/kf/201108/101973.html>）

<http://mina.apache.org/>（<http://mina.apache.org/mina-project/quick-start-guide.html> 快速入门）

js

<http://www.oschina.net/p/seajs> 模块加载（<http://witcheryne.iteye.com/blog/1166331>）

<http://www.oschina.net/p/modulejs> 模块化

<http://www.oschina.net/p/bootstrap> 前端ui（<http://www.ostools.net/apidocs/apidoc?api=bootstrap>）

<http://www.oschina.net/p/lesscss> css预处理编译器（<http://www.ostools.net/less>，在线编译）

<http://www.oschina.net/p/nodejs> node.js

<http://www.oschina.net/p/coffeescript> js上层语言（<http://coffeescript.org/> 官网）

<http://www.oschina.net/p/scala-js> js上层scala语言

<http://www.oschina.net/p/xregexp> js正则表达式库 （<http://www.oschina.net/p/xregexp> 官网）

<https://github.com/reyesr/fullproof> js搜索引擎

<http://www.oschina.net/p/lunrjs> 搜索引擎

<http://documentcloud.github.io/backbone/> mvc框架

<http://www.taffydb.com/> js小型数据库

js前端框架

<http://www.oschina.net/p/qatrix>（小型jquery <http://qatrix.com/> 官网）

<http://www.oschina.net/p/tangram>（百度，类似jquery，<http://tangram.baidu.com/> 官网）

<http://www.oschina.net/p/yui>（yui <http://yuilibrary.com/yui/quick-start/> 官网）

<http://www.oschina.net/p/nej> （网易前端 <http://nej.netease.com/document/> 官网）

<http://www.oschina.net/p/jx> 腾讯前端（<http://www.alloyteam.com/2012/06/jx-framework/> 官网）

<http://docs.kissyui.com/> 淘宝前端

<http://aralejs.org/> 支付宝前端

<http://prototypejs.org/> prototype

js前端工具

<http://ueditor.baidu.com/website/index.html> 百度编辑框

<http://www.oschina.net/p/tinymce> 小型编辑框

<http://www.oschina.net/p/datav-js> 数据展示（<https://github.com/TBEDP/datavjs>官网）

<http://wangxiao.github.io/BaiduTemplate/>  百度模板引擎

<https://github.com/janl/mustache.js/> 模板引擎

<https://github.com/yinhang/Elapse> 国产模板引擎

<http://www.oschina.net/p/codemirror> 语法高亮（<http://codemirror.net/> 官网）

代理服务器

<http://www.squid-cache.org/> squid （中文简介）

文件系统

<https://code.google.com/p/fastdfs/>  分布式文件系统（<http://www.oschina.net/p/fastdfs> 简单介绍）

<http://datafarm.apgrid.org/> 分布式文件系统（<http://datafarm.apgrid.org/document/> 简单介绍）

<https://github.com/handsomestone/FUFS> 新浪微盘（学习用）

linux工具

<http://www.oschina.net/p/tsar> 淘宝系统信息采集（<http://code.taobao.org/p/tsar/wiki/guide/> 官网）

<http://code.google.com/p/inotify-sync/> 文件更改监控

c++工具类

<https://github.com/facebook/folly> facebook工具库

<http://www.oschina.net/p/dsal> 数据结构算法库

<https://code.google.com/p/rlib/> 轻量级工具库

# ElasticSearch

ElasticSearch可以理解成一个数据库，需要往里面扔东西，然后搜索

查看运行状态指令：ss -tnl

开启服务指令：

su java

/etc/init.d/elasticsearch start|stop|restart

建索引把数据库里东西放到elasticsearch中

su java

cd ~

php index.php 建索引

php index.php rebuild 重建索引

php index.php &后台运行

fg 打开后台运行的程序

ctrl+z bg

nohup

curl 'http://127.0.0.1:9200/dealmoon/deal\_en/\_search?q=Nike&pretty=true'

curl http://127.0.0.1:9200/dealmoon/deal\_en/ 469245?pretty=true

curl http://127.0.0.1:9200/dealmoon/deal\_cn/\_mapping?pretty=true

deal用elasticsearch搜索的时候筛选了state = published 内部类 Filter

php index\_deal.php rebuild

# SSDB

数据缓存服务

phpssdbadmin 绑host

http://phpssdbadmin/login

test 12345678

# SourceTree

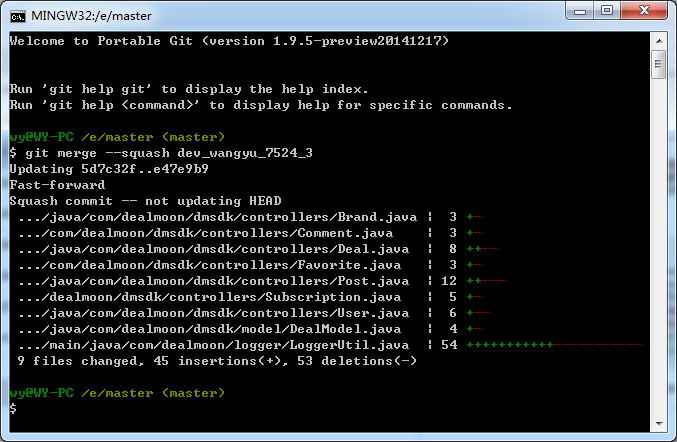
获取fetch 相当于是从远程获取最新版本到本地，不会自动merge

拉取pull 相当于是从远程获取最新版本并merge到本地

合并分支：

在dev\_ugc分支上在分支中选择dev\_wangyu\_7285\_2 遴选解决冲突可以用编译器打开编辑冲突的位置

若分支上的改变没有提交，选择要恢复的版本，重建当前分支到此次提交



# geoip2

下载数据库文件GeoLite2-Country.mmdb

maven jar包

<dependency>

<groupId>com.maxmind.geoip2</groupId>

<artifactId>geoip2</artifactId>

<version>2.1.0</version>

</dependency>

public static Country getCountryInfoByIp(String ip){

File database = new File("db/GeoLite2-Country.mmdb");

DatabaseReader reader = null;

try {

reader = new DatabaseReader.Builder(database).build();

} catch (IOException e2) {

e2.printStackTrace();

}

InetAddress ipAddress = null;

try {

ipAddress = InetAddress.getByName(ip);

} catch (UnknownHostException e) {

e.printStackTrace();

}

CountryResponse response = null;

try {

response = reader.country(ipAddress);

} catch (Exception e) {

return new Country();

}

if(response != null){

return response.getCountry();

}else{

return new Country();

}

}

# ZHConverter

简繁体中文互换的Java开源类库。

# Integer和String

Integer.valueof(String s)和Integer.parseInt(String s)的区别

Integer.valueof(String s)是将一个包装类是将一个实际值为数字的变量先转成string型再将它转成Integer型的包装类对象(相当于转成了int的对象)这样转完的对象就具有方法和属性了。

而Integer.parseInt(String s)只是将是数字的字符串转成数字，注意他返回的是int型变量不具备方法和属性。

String.valueOf() 方法的使用

由 基本数据型态转换成 String

StringUtils用法

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

<version>3.1</version>

</dependency>

int类型范围

1437463095

Integer time = 1437463095;

System.out.println((long)time\*1000);

System.out.println((long)(time\*1000));

# orika对象间Mapping

# Python

python strip()函数

声明：s为字符串，rm为要删除的字符序列

s.strip(rm) 删除s字符串中开头、结尾处，位于 rm删除序列的字符

s.lstrip(rm) 删除s字符串中开头处，位于 rm删除序列的字符

s.rstrip(rm) 删除s字符串中结尾处，位于 rm删除序列的字符

split是分割字符串

join是拼接字符串

## re模块是正则表达式是相关的

注意：正则表达式前面加r

re.match 尝试从字符串的开始匹配一个模式

re.match的函数原型为：re.match(pattern, string, flags)

第一个参数是正则表达式，这里为"(\w+)\s"，如果匹配成功，则返回一个Match，否则返回一个None；

第二个参数表示要匹配的字符串；

第三个参数是标致位，用于控制正则表达式的匹配方式，如：是否区分大小写，多行匹配等等。

re.search函数会在字符串内查找模式匹配,只到找到第一个匹配然后返回，如果字符串没有匹配，则返回None。

re.match与re.search的区别：re.match只匹配字符串的开始，如果字符串开始不符合正则表达式，则匹配失败，函数返回None；而re.search匹配整个字符串，直到找到一个匹配。

re.sub用于替换字符串中的匹配项

re.sub的函数原型为：re.sub(pattern, repl, string, count)

re.split可以使用re.split来分割字符串，如：re.split(r'\s+', text)；将字符串按空格分割成一个单词列表。

re.findall可以获取字符串中所有匹配的字符串。如：re.findall(r'\w\*oo\w\*', text)；获取字符串中，包含'oo'的所有单词。

re.compile可以把正则表达式编译成一个正则表达式对象。可以把那些经常使用的正则表达式编译成正则表达式对象，这样可以提高一定的效率。下面是一个正则表达式对象的一个例子：

text = "JGood is a handsome boy, he is cool, clever, and so on..."

regex = re.compile(r'\w\*oo\w\*')

print regex.findall(text) #查找所有包含'oo'的单词

re.compile(pattern[, flags])

作用：把正则表达式语法转化成正则表达式对象

flags定义包括：

re.I：忽略大小写

re.L：表示特殊字符集 \w, \W, \b, \B, \s, \S 依赖于当前环境

re.M：多行模式

re.S：' . '并且包括换行符在内的任意字符（注意：' . '不包括换行符）

re.U：表示特殊字符集 \w, \W, \b, \B, \d, \D, \s, \S 依赖于 Unicode 字符属性数据库

## os模块

os模块包含普遍的操作系统功能，与具体的平台无关。

os.name()——判断现在正在实用的平台，Windows 返回‘nt'; Linux 返回’posix'。

os.getcwd()——得到当前工作的目录。

os.listdir()——指定所有目录下所有的文件和目录名。

os.remove(filename)——删除指定文件

os.rmdir()——删除指定目录

os.mkdir()——创建目录

os.path.isfile()——判断指定对象是否为文件。是返回True,否则False

os.path.isdir()——判断指定对象是否为目录。是True,否则False

os.path.exists()——检验指定的对象是否存在。是True,否则False

os.path.split()——返回路径的目录和文件名

os.system(shell命令)——执行shell命令。

os.chdir()——改变目录到指定目录

os.path.getsize()——获得文件的大小，如果为目录，返回0

os.path.abspath()——获得绝对路径

os.path.join(path, name)——连接目录和文件名

os.path.basename(path)——返回文件名

os.path.dirname(path)——返回文件路径

os.path.realpath(sys.argv[0])——获取程序所的实际目录

os.path.realpath(sys.argv[0])

os.path.split(os.path.realpath(sys.argv[0]))

将一个路径名分解为目录名和文件名两部分

fpath , fname = os.path.split( "要分解的路径")

例如：

a, b = os.path.split( "c:\\123\\456\\test.txt" )

print a

print b

os.path.split(os.path.realpath(sys.argv[0]))[0]

## configParser 模块

用于操作配置文件

注意：文件路径引号前加u

1、config=ConfigParser.ConfigParser()

创建ConfigParser实例

2、config.sections()

返回配置文件中节序列

3、config.options(section)

返回某个项目中的所有键的序列

4、config.get(section,option)

返回section节中，option的键值

5、config.add\_section(str)

添加一个配置文件节点(str)

6、config.set(section,option,val)

设置section节点中，键名为option的值(val)

7、config.read(filename)

读取配置文件

8、config.write(obj\_file)

写入配置文件

import configparser

import sys

config = configparser.ConfigParser()

'''

config.read('E:\workspace\HelloPython\my.ini')

print (config.get("book", "title"))

print (config.sections())

#遍历配置文件的项

for section in config.sections():

print (section)

for option in config.options(section):

print (" ",option,"=",config.get(section,option))

'''

config.add\_section("book")

config.set("book","title","这是标题")

config.set("book","author","大头爸爸")

config.add\_section("size")

config.set("size","size","1024")

config.write(sys.stdout)

config.write(open('E:\workspace\HelloPython\my.ini','r+'))

## logging模块

import logging

'''

logging.basicConfig(level=logging.DEBUG,

format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',

datefmt='%a, %d %b %Y %H:%M:%S',

filename='../test.log',

filemode='w')

'''

logging.debug('debug message')

logging.info('info message')

logging.warning('warning message')

logging.error('error message')

logging.critical('critical message')

# SpringBoot

resttemplate

http://start.spring.io/

http://blog.csdn.net/wwwihpccn/article/details/30496089

如果controller与main方法不在一个包下，则需配置

@Configuration

@ComponentScan("com.dealmoon.rss.controllers")

@EnableAutoConfiguration

public class Application {

public static void main(String[] args) {

SpringApplication.run(Application.class);

}

}

# MongoDB

show dbs 列出所有的数据库

use databasesName //use test

db 显示当前的数据库

show collections 列出数据库下的所有集合

db.wy.find() 查询wy集合里所有的文档

## 插入文档：

MongoDB 使用 insert() 或 save() 方法向集合中插入文档，语法如下：

db.COLLECTION\_NAME.insert(document)

实例

以下文档可以存储在 MongoDB 的 runoob 数据库 的 col集合中：

>db.col.insert({title: 'MongoDB 教程',

description: 'MongoDB 是一个 Nosql 数据库',

by: '菜鸟教程',

url: 'http://www.runoob.com',

tags: ['mongodb', 'database', 'NoSQL'],

likes: 100

})

## 更新文档

update() 方法用于更新已存在的文档。语法格式如下：

db.collection.update(

<query>,

<update>,

{

upsert: <boolean>,

multi: <boolean>,

writeConcern: <document>

}

)

**参数说明：**

* **query**: update的查询条件，类似sql update查询内where后面的。
* **update**: update的对象和一些更新的操作符（如$,$inc...）等，也可以理解为sql update查询内set后面的
* **upsert**: 可选，这个参数的意思是，如果不存在update的记录，是否插入objNew,true为插入，默认是false，不插入。
* **multi**: 可选，mongodb 默认是false,只更新找到的第一条记录，如果这个参数为true,就把按条件查出来多条记录全部更新。
* **writeConcern**:可选，抛出异常的级别。

# 需求

## 后台数据统计

http://pm.dealmoon.net:8080/projects/ugc\_1/wiki/Ugc%E5%90%8E%E5%8F%B0%E7%BB%9F%E8%AE%A1%E6%96%87%E6%A1%A3

## 评论的几种状态

unpublished\_to\_review 被拦截

published\_to\_review 待审

published 审核通过

interception/delete 被删除

## 晒单的几种状态

recommend\_to\_review

block

published

reported

delete

unpublished\_to\_review未发待审（前台看不到）

published\_to\_review已发待审（前台能看到，二级以上用户）

## 自动点赞需求整理

在最新晒单的列表中，取前20条；

发5个以内晒单的视为新用户；

需要请求用户信息接口和点赞接口；

判断最新列表中用户是新人，发帖时间大于10秒的，未被点过赞的；

已发待审的post；

从点赞账号列表中，随机安排1-3个用户来为符合要求的晒图点赞；

linux下 host文件/etc/hosts

## api 1.0重写

数据库表

dm\_subscriber\_client

dm\_user\_like\_deals

deal/topics/getbanners 仅在 iPad 客户端使用, 是获取 iPad 横屏时右侧 banner 的接口. 无特殊参数.

deal/client/active 接口 iPhone 和 iPad 客户端均未使用.仅android在用

## 折扣分地区显示需求整理

根据访问者所在区域显示符合要求的deal

function getParamsByLocation($location = 'all'){

$params = array('all', 'no\_us', 'no\_cn');

switch ($location) {

case 'CN':

$params = array('all', 'cn', 'no\_us');

break;

case 'US':

$params = array('all', 'us', 'no\_cn');

break;

}

return $params;

}

涉及的mapper:

DmHotDealRankEntityMapper

HotdealsEntityMapper

CnDealsEntityMapper

DealCategoryEntityMapper

涉及的缓存处理：

App端：AppCacheManager

DMSDK：DealModelCache

非中国和美国地区的 array('all');

中国地区 array('all', 'cn');

美国地区 array('all', 'us');

加拿大array('all', 'ca');

条件之间是or 的关系

1.改造list接口，点击排行、deal\_list

2.CommandInfo中不需要加字段，req中有ip

3.Dao层继续用sql2o

4.APP两个cache需要处理，AppClickRankCache、AppDealCache，DMSDK的cache也要处理，增加一个地区标识字段

5.涉及三个表，hotdeals、cn\_deals、deal\_category

6.APP层就做ip处理，然后把地区标识传到DMSDK

7.在location2分支做，hotdeals已经做完

8.APP的两个cache在AppApiResource中做的拦截处理

9.DMSDK的cache处理只加DealIdList

10.deal/list deal/click有相同的排行榜逻辑，都要处理

Logback和SFL4J

Grant all privileges on \*.\* to 'root'@'%' identified by 'bookface06' with grant option;

mysqladmin -uroot -p1234 flush-hosts;

## 日志处理需求整理

问题：分享暂时无法统计

每条记录都添加下面数据：

clientType 设备类型：iphone, ipad, android

udid

ip

userId

addTime 添加时间

location 地区(国家)

typeadd\_deal\_comment|add\_post\_comment

deal评论添加日志： add\_deal\_comment

评论id, deal\_id, 内容，评论状态，

晒图评论添加日志：add\_post\_comment

评论id, post\_id, 内容，评论状态，添加积分，添加金钱

deal点赞部分添加日志：add\_deal\_like

deal\_id

?deal评论点赞部分添加日志：

评论id， deal\_id

deal收藏日志add\_deal\_fav

deal\_id

晒图收藏日志 add\_post\_fav

post\_id

deal点击deal\_click detail

deal\_id，来源（普通，订阅）

晒图点击post\_click info 没有来源字段

post\_id

订阅规则subscription

分类id（多个用，分割），店铺id，关键字

关注品牌 add\_brand\_like

品牌id

关注人 add\_user\_follow

关注人id

？晒货分享（访问情况）

post\_id，作者， tags，图片（多张）

发表晒图post\_create

post\_id，添加积分，添加金币，tags name，图片（多张）url，状态

用户注册 user\_register

用户id，用户email, 注册类型（email（正常注册），微博，微信，人人，facebook）

推送

设备udid，推送类型，匹配规则，推送内容， deal\_id, 数量

## Task #7632: deal/subscription/unread

接口增加对设备ip变动的修改

7632是mybatis版本

7632\_1是sql2o版本

## Task #7376: deal/deal/unLikeDeals

7376\_1是deal点赞，取消赞

## Bug #7685: “最新”页面拉到最底端后无法加载更多（有图）

7685\_2

## Bug #7620: 客户端搜索简体字，繁体字内容也应该出现（见图）

7620 是mapper版本的 7620\_1是sql2o版的

## Task #7390: deal/subscription/unread 接口改造

## Task #7384: deal/client/active 设备激活

7390和7384都在7390里边

## BBS需求

http://www.moonbbs.com/forum.php?api=dm\_list\_forum\_refresh  
取数据：http://www.moonbbs.com/forum.php?api=dm\_list\_forum  
如果location=2 那么请求http://www.moonbbs.com/forum.php?mod=forumdisplay&fid=69&filter=sortid&sortid=5&searchoption[33][value]=2&searchoption[33][type]=select

http://apim.test.dealmoon.net/

http://apim.test.dealmoon.net/forum.php?mod=forumdisplay&fid=69&filter=sortid&sortid=5&searchoption[33][value]=2&searchoption[33][type]=select

http://apim.test.dealmoon.net/forum.php?api=dm\_list\_forum

http://apim.test.dealmoon.net/forum.php?mod=forumdisplay&api=yes&searchoption[33][value]=3&searchoption[33][type]=select&filter=sortid&sortid=5&forumdefstyle=yes&submit\_f=1&page=1&tpp=10

## 海淘需求

http://pm.dealmoon.net:8080/projects/dealmoon-related/wiki/2015-08-31

搜索调整（中文加入海淘信息搜索关键词）

小贴士加入搜索（仅中国地区搜索小贴士） 中国站。建索引建进去，搜索的时候只有中国搜索。

商家管理 支付方式 运输方式

小贴士和商家挂钩

支付方式存名称。

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

接口：

小贴士 增，删，列表（新增）

搜索调整（中文搜索，索引的时候）

deal添加，修改，读详情（前台，后台）。

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

新增prepublish

编辑不处理状态

状态要统一（英文，中文，海淘）

单品分组的id

图片有相对对地址

分类 关系表。hot\_pick,hot\_buy,不发布，不插入category\_deal

中英文。

renew不需要处理

log不需要管

分类逗号点起来。

weibo只入库

exclusioons appley

seanson如果选中，需要特殊处理

main\_url 替换 exec?d=:dealId

store\_ name store\_id

publishNote关系表，

//deal\_note

Admin/deal/create

hotdeals h cn\_deals c seasonal\_deal s

commandInfo {

price: "价格", h price

listPrice : "原价", h 原价 - 价格 = price\_dropped

storeName : "店铺名字", h store\_name

storeId : "店铺id", h store\_id

isHotPick : "true", h is\_hotpick

isExclusive : "false", h is\_exclusive

isHotbuy : "true", h is\_hotbuy

tipWeibo : "true", h tip\_weibo

isDead : "true", h flags = dead

noExpire : "true", h flags = no\_expire优先级高

author : "创建者", create时，author=created create时，author= modified

category: "1231232,123231231", h category

showByLocation : "all", h show\_by\_location

startTime : "2015-09-09 12:34:21", h start\_time

expirationTime : "2015-09-09 12:34:21", h expiration\_time

publishedDate : "2015-09-09 12:34:21", h published\_date

url : "购买链接", h url

spGroupId : "123", h sp\_group\_id

recipient : "Women,Men", h recipient

language : "cn",

seasonTag: "aa", 对应s表，有就更新，没有再插入

spDefaultView : "less", h sp\_default\_view

deal\_en : {

title : "英文标题", h title

titleEx : "标题前缀", h title\_ex

imageUrl : "图片1"， h image\_url

imageUrl1 : "图片2", h image\_url1

bannerUrl : "链接地址", h en\_banner\_url

body : "简介" h body

},

deal\_cn : {

title : "中文标题", c title\_cn

titleEx : "标题前缀", c title\_ex\_cn

imageUrl : "图片1"， c image\_url\_cn

bannerUrl : "链接地址", c cn\_banner\_url

displayOrder : "排序", c display\_order

hideImage : "true", c hide\_image

cnMurlDst : "esingle", c cn\_murl\_dst

body : "简介" c body\_cn

},

deal\_haitao : {

countryCode : "us", 不用考虑，拿来直接用

price : "12.12",

chinaPrice : “22.22”,

directDelivery : "true", 是否支持直邮

payment : "银联,支付宝,zidingyi",

alipay\_shipping\_desc : "",

shipping : "direct\_delivery,transport" //direct\_delivery|transport|us\_address

}

}

deal create:

1. en cn season\_deal状态都是prepublished
2. hotdeals cn\_deal haitao\_deals 表都会有数据，如果缺少某个结构，则存空值和默认值
3. 插入seasonTag，状态保持一致
4. modified = Createed = author

deal edit:

1. 如果deal state=published 插到deal\_category表，添加的时候不用考虑，如果不是published 对应删掉
2. 如果create没传seasonTag，edit时传了，则插入记录，如果编辑的时候没传，则对应删掉
3. 只要en cn 中有published状态就插入deal\_caregory，如果非published则删掉

公用:

1. language字段对应en cn haitao的状态，对应h表state cn\_state haitao\_state
2. 不传的字段相应的deal设置成hidden，传的要判断编辑之前的en cn非hidden的状态
3. cn\_deals表需要把简体字段转成繁体，存到\_bg字段
4. show\_by\_location ， publish\_date三个表一致
5. 先查出之前的en cn deal 的状态，如果en为hidden状态，则用cn的状态，以这个状态设置编辑后的en cn haitao（language传的字段） seasontag 的状态
6. language中，不传的对应的deal设置成hidden，传的要判断编辑之前的状态
7. hotdeals 的flags字段要分开处理，没传的对应去掉，编辑的时候去掉need\_mod，插入是去重
8. 返回结构的state 给en deal状态
9. 如果haitao\_deal中shipping中，存在alipay检查hotdeal里面是否有storeid，如果有，要affiliate\_store给出这个store结构中的 alipay\_shipping\_desc
10. 中文deal create和update时，body.replace("http://main\_url/", "/exec/j/?d="+dealId);

## App haitao需求

[http://pm.dealmoon.net:8080/projects/dealmoon-related/wiki/DEAL](http://pm.dealmoon.net:8080/projects/dealmoon-related/wiki/DEAL" \t "_blank)

433758

## RSS需求

对于dealmoon入驻Flipboard的事情，我们进行了讨论，初步决定将北美省钱快报旗下三个热门栏目内容作为三本杂志同步到Flipboard中，具体如下：  
杂志1：热门折扣C:\Users\wy\AppData\Local\Temp\%W@GJ$ACOF(TYDYECOKVDYB.pnghttp://cn.dealmoon.com/Hot-Picks 及旗下三个微信账号内容  
杂志2：晒单与测评 C:\Users\wy\AppData\Local\Temp\%W@GJ$ACOF(TYDYECOKVDYB.pnghttp://cn.dealmoon.com/Buying-Guide  
杂志3：美国畅销榜 C:\Users\wy\AppData\Local\Temp\%W@GJ$ACOF(TYDYECOKVDYB.pnghttp://cn.dealmoon.com/Best-Seller  
   
微信账号信息  
北美省钱快报 ID：dealmoon     粉丝数：30万+     更新频率：每日更新，每次7-8条内容  
北美时尚   ID：fashionmoon    粉丝数：12万+      更新频率：一周5次推送，每次4条内容  
美妆败家控 ID：dealmoonbeauty 粉丝数：15万+   更新频率：一周5次推送，每次8条内容  
关于RSS技术文档，已经发给我们的产品技术同学，具体上线需要准备的事宜请随时与我联系。

RFC822 时间格式

SimpleDateFormat fm=new SimpleDateFormat("EEE,d MMM yyyy hh:mm:ss Z", Locale.ENGLISH);

System.out.println( fm.format(new Date())

{

"command": "Deal/getDealIds",

"lang": "en",

"udid": "",

"type": "email",

"ip": "221.122.18.11",

"ua": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.0",

"editorId": "",

"commandInfo": {

"categoryid": "1175715850",

"pageLimit": 80,

"pageNum": 1,

"timerange": null,

"endtime": null,

"notInID": null,

"lang": "cn",

"showByLocation": [

"all",

"cn"

]

}

}

{

"result": {

"code": 0

},

"responseData": [

457189,

455946,

457132,

458120,

443367,

443230,

442923

]

}

img.dealmoon.com/images/c/15/04/09/5526fee051f26.jpg 原图地址

http://198.23.88.246:8010/dealmoon/category/Hottest

http://198.23.88.246:8010/dealmoon/category/Buying-Guide

http://198.23.88.246:8010/dealmoon/category/Best-Seller

<http://flipboard.dmadm.com/dealmoon/category/Hottest>  
<http://flipboard.dmadm.com/dealmoon/category/Buying-Guide>  
<http://flipboard.dmadm.com/dealmoon/category/Best-Seller>

http://rss.dealmoon.com/cn/

## 搜索需求

前台:  
英文页面:  
英文: 只搜英文, 结果显示在英文页面 \*  hotdeals     en\_body , cn\_body   
中文: 搜英文和中文, 结果显示在中文页面  
中文页面:  
英文: 搜英文和中文, 结果显示在中文页面  
中文: 只搜中文, 结果显示在中文页面  
  
后台:  
全部:  
英文: 搜英文和中文, 结果显示在英文页面//  
中文: 搜英文和中文, 结果显示在中文页面  
若折扣隐藏了英文,就用中文的  
  
仅中文:  
英文: 搜英文和中文, 结果显示在中文页面//   从英文索引里搜  
中文: 搜英文和中文, 结果显示在中文页面//  
若折扣隐藏了中文,就用英文的  
  
  
  
$commandInfo = array(  
    'keyword' => $keyword,  
    'category' => $category,  
         language => [’en\_body’,’cn\_body’]，  
         index ＝> ‘deal\_en’  全部：deal\_cn   中文：deal\_en :英文页面:deal\_en中文页面:deal\_cn  
    'expired' => $expired,  
    'order' => $order,  
    'pageNum' => $pageNum,  
    'pageSize' => $pageSize,  
    'isHaitao' => $isHaitao,  
showByLocation =>[‘all’,’cn’]

## 接口优化

超过1s的访问次数

'deal/subscription/storeList' => 1,

'deal/deal/unLikeDeals' => 1,

'hashTag/recommends' => 1,

'hashtag/recommends' => 1,

'post/getlikelist' => 1,

'user/follow' => 1,

'search/keyword' => 2,

'comment/like' => 2,

'deal/subscription/delAlert' => 2,

'deal/push/hotRegister' => 2,

'post/getfavoritelist' => 2,

'user/forgetPassword' => 2,

'hashtag/search' => 2,

'brand/list' => 2,

'post/getcomment' => 3,

'deal/comment/gethot' => 3,

'deal/deal/likeDeals' => 3,

'post/addcomment' => 4,

'deal/subscription/mine' => 4,

'deal/deal/delFavDeal' => 7,

'user/bind' => 7,

'post/addlike' => 7,

'user/login' => 8,

'message/getlist' => 11,

'deal/click/getCategroyClickRankList' => 18,

'category/list' => 20,

'deal/click/getClickRankIndex' => 21,

'user/sendActiveMail' => 22,

'deal/topics/getBanners' => 22,

'deal/deal/addFavDeal' => 22,

'deal/store/getStoreList' => 26,

'deal/deal/getFavDealList' => 33,

'search/index' => 40,

'deal/bbs/getbbshot' => 44,

'user/info' => 50,

'post/explore' => 79,

'user/profile' => 89,

'setting/system/getSystemConfig' => 94,

'deal/deal/search' => 126,

'user/register' => 166,

'post/info' => 196,

'deal/comment/add' => 202,

'tag/info' => 206,

'post/recommend' => 267,

'post/getlist' => 373,

'deal/subscription/myAlerts' => 376,

'deal/topics/getSplashScreen' => 404,

'search/search' => 525,

'deal/push/register' => 542,

'deal/comment/get' => 979,

'deal/bbs/getbbsList' => 1151,

'deal/deal/list' => 3223,

'deal/deal/detail' => 3473,

'deal/subscription/getUnRead' => 4042,

一天各接口访问次数

array (  
  'deal/subscription/delAllAlert' => 2,  
  'post/delfavorite' => 3,  
  'post/delcomment' => 4,  
  'hashTag/recommends' => 4,  
  'hashTag/search' => 5,  
  'user/forgetPassword' => 5,  
  'subject/detail' => 6,  
  'subject/list' => 6,  
  'user/unfollow' => 6,  
  'message/clear' => 8,  
  'user/unbind' => 8,  
  'post/delete' => 13,  
  'user/logout' => 15,  
  'outlet/outlet/getOutletInfo' => 15,  
  'user/bind' => 15,  
  'deal/push/hotUnRegister' => 18,  
  'post/share' => 18,  
  'brand/addlike' => 20,  
  'post/addfavorite' => 21,  
  'user/sendActiveMail' => 22,  
  'Vsdeal/list' => 26,  
  'outlet/outlet/getOutletList' => 27,  
  'user/fanlist' => 28,  
  'deal/subscription/update' => 31,  
  'user/follow' => 37,  
  'deal/subscription/categoryList' => 38,  
  'fashionmoon/fashion/getGroup' => 41,  
  'user/update' => 43,  
  'post/dellike' => 46,  
  'comment/dellike' => 48,  
  'deal/subscription/delete' => 49,  
  'user/followlist' => 63,  
  'hashtag/recommends' => 86,  
  'fashionmoon/fashion/getArticleList' => 88,  
  'post/getlikelist' => 88,  
  'brand/likelist' => 88,  
  'post/getfavoritelist' => 141,  
  'deal/subscription/submit' => 152,  
  'deal/deal/unLikeDeals' => 165,  
  'deal/client/active' => 206,  
  'post/addcomment' => 295,  
  'deal/comment/gethot' => 318,  
  'product/search' => 361,  
  'deal/subscription/storeList' => 409,  
  'comment/like' => 422,  
  'deal/topics/gettopics' => 423,  
  'user/register' => 456,  
  'post/addlike' => 469,  
  'user/login' => 497,  
  'deal/deal/likeDeals' => 548,  
  'brand/list' => 647,  
  'deal/comment/add' => 737,  
  'message/getlist' => 847,  
  'deal/deal/delFavDeal' => 882,  
  'deal/subscription/mine' => 936,  
  'hashtag/search' => 939,  
  'post/getcomment' => 990,  
  'deal/push/hotRegister' => 1095,  
  'deal/subscription/delAlert' => 1524,  
  'search/keyword' => 2177,  
  'deal/deal/getFavDealList' => 3702,  
  'deal/click/getCategroyClickRankList' => 4615,  
  'deal/bbs/getbbshot' => 5147,  
  'deal/deal/addFavDeal' => 5710,  
  'deal/click/getClickRankIndex' => 6181,  
  'post/explore' => 6973,  
  'tag/info' => 7522,  
  'user/profile' => 7527,  
  'user/info' => 7590,  
  'deal/version/update' => 8577,  
  'category/list' => 10907,  
  'deal/store/getStoreList' => 11103,  
  'deal/deal/search' => 12679,  
  'deal/topics/getBanners' => 14648,  
  'search/index' => 16886,  
  'deal/bbs/getbbsList' => 18889,  
  'post/getlist' => 18979,  
  'deal/subscription/myAlerts' => 22476,  
  'post/info' => 24078,  
  'post/recommend' => 32317,  
  'search/search' => 52457,  
  'setting/system/getSystemConfig' => 71429,  
  'deal/topics/getSplashScreen' => 202180,  
  'deal/comment/get' => 230142,  
  'deal/push/register' => 253612,  
  'deal/deal/detail' => 417267,  
  'deal/deal/list' => 665404,  
  'deal/subscription/getUnRead' => 725136,  
)

## MailService

<http://pm.dealmoon.net:8080/projects/moonbbs/wiki/%E9%82%AE%E4%BB%B6%E5%8F%91%E9%80%81%E6%9C%8D%E5%8A%A1MailService>

<http://pm.dealmoon.net:8080/projects/sysadmin/wiki/20150415-5816_%E5%BE%AE%E8%AE%BA%E5%9D%9B%E9%82%AE%E4%BB%B6%E5%8F%91%E9%80%81%E6%9C%8D%E5%8A%A1%E4%B8%8A%E7%BA%BF>



database:

sourceUrl: jdbc:mysql://localhost:3306/dealmoon?useUnicode=true&characterEncoding=utf8

user: dm

pass: bookface06

sendcloud pm@mail.dealmoon.com   dealmoon2010   
mailgun david.wang@dealmoon.com bookface123

http://sendcloud.sohu.com/  
http://www.mailgun.com/

发送post请求格式

curl -d 'api\_user=\*\*\*&api\_key=\*\*\*&from=test@test.com&fromname=来自测试发送&subject=测试&html=这是一封测试邮件&to=ben@ifaxin.com;joe@ifaxin.com&cc=bida@ifaxin.com&bcc=lianzimi@ifaxin.com&replyto=reply@test.com&label=16800&resp\_email\_id=true&x\_smtpapi={"to":["mary@ifaxin.com", "karl@ifaxin.com"]}' http://sendcloud.sohu.com/webapi/mail.send.json

send\_cloud 付费 国内邮箱

1.需要添加apiUser吗?有几个可以共用吗? postmaster@dealmoon.sendcloud.org

2.apiKey可以共用吗? 共用

3.url 可以共用吗? 共用

mail\_gun 免费 10000封以内 国外邮箱

1.apiKey可以共用吗? 共用

2.url 可以共用吗? https://api.mailgun.net/v3/sso.dealmoon.net

{

"source":"dealmoon",

"from":"712642832@qq.com",

"to":"1106066794@qq.com",

"subject":"test",

"message":"dfddfdfd"

}

http://54.215.9.31:8533/mail/send

url2: ALTER  TABLE  hotdeals  ADD  url2 text NOT NULL;

## RSS重写需求

问题：

1、上面那个数据接口非常不稳定，慢，而且总是断，断了之后数据不完整。通过海外的代理可以正常访问。

2、rss中缺失图片

我们的解决办法：最好是北美那边提供一个稳定可靠且有图片的rss，否则的话，我们需要通过 proxy＋抓取的方式来提取，很麻烦

## 后台重构提供接口

管理员登录，验证 半天

获取所有管理员。 1小时

获取season列表 1小时

prepublish deal列表， 根据分类，状态，时间，操作人等信息进行筛选，支持分页。 2

publish

myworkspace

submitteddeal

bookmark

详情信息获取

deal日志

网站基本信息接口

小贴士接口

publish note接口

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

roundup 几个deal合集

roundup image 主图合集

prepublish it 取消发布

bookmark 状态

clone 克隆到prepublished myworkspace

renew 更新发布时间为当前时间

new 创建新的

pick it 检出到workspace

trash 改状态 放弃编辑

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

prepublish my workspace publish(英文) publish(中文) publish(海淘) publish(晒单) publish(畅销榜) publish(攻略) 中文置顶CN page SubmittedDeal fetchdeal bookmark

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

功能点：

1、category 获取分类（一二级分类）

2、store 店铺列表（有自动补全）

3、editor 编辑人列表

4、season season列表

5、根据category、store、types、editor、status、date、url convert、season、ip、keyword筛选deal

6、根据date、clicks、views、bookmarked排序（正序或倒序）

7、符合条件的deal总记录数

8、deal 列表分页

9、筛选all或haitao

10、筛选all、shared、picked

11、单个deal操作日志，按编辑人、时间（区间）、时区筛选

12、修改deal信息（修改状态、字段、分类、显示区域）

13、new deal数、today deal数（异步请求，定时）

14、管理员登录验证、权限验证

15、new/edit deal

store 列表

store 最近发布的10个deal

deal tip 增、删、查

16、单品管理

17、deal 添加评论

18、焦点、add to wap、add to top、add to email、push message、single email

19、编辑store信息

19、submittedDeal

search

deal 信息

edit deal、skip deal

copy to prepublish

20、fetch deal

copy to prepublish

copy to myworkspace

21、publish note

-------------------------------------------------------------------------------------------------------------------------------v接口列表：

1、category 获取分类（一二级分类） （0.25天）

2、store 店铺列表，根据店铺名模糊查找 （0.25天）

3、editor 编辑人列表 （0.25天）

4、season 列表 （0.25天）

5、根据category、store、types、editor、status、date、url convert、season、ip、keyword筛选deal，根据date、clicks、views、bookmarked排序（正序或倒序），符合条件的deal总记录数，支持分页，返回dea信息 筛选all或haitao，筛选all、shared、picked （2天）

6、单个deal操作日志，按编辑人、时间（区间）、时区筛选 （0.5天）

7、修改deal信息（修改状态、字段、分类、显示区域） （0.5天）

8、new deal数、today deal数（异步请求，定时） （0.5天）

9、管理员登录验证、权限验证 （1天）

10、new/edit deal （1天）

store 列表

store 最近发布的10个deal

deal tip 增、删、查

11、单品管理 （2天）

12、deal 添加评论 （1天）

13、焦点、add to wap、add to top、add to email、push message、single email （2天）

14、编辑store信息 （0.5天）

15、submittedDeal （1天）

search deal 信息

edit deal、skip deal

16、fetch deal （1天）

17、copy to prepublish ，copy to myworkspace （0.5天）

18、publish note （0.5天）

## 微博需求

weibo相关的代码（抓取weibo，评论 ）的代码在 代码库 git.dealmoon.net:weibo 中，

由于历史的原因， 目前的代码库里有两个分支在使用，

一个是 master， 主要用来更新北美省钱快报的 weibo相关内容，

另一个是 multi\_channel， 主要用来抓取子频道的weibo相关内容，

区别是 multi\_channel 支持抓取多个微博帐号的内容。