联创镜像交付文档

2022年11月

# 项目架构说明

## 自服务平台

所用开发语言：

JDK 8

技术框架：

SpringMVC+mybatis+redis

## 运营运维管理平台

开发语言：Java + JDK1.8

技术框架：

Spring 3.0.6,

Hibernate 3.6.6

## 接口

开发语言：Java + JDK1.8

技术框架：

Spring 4.2.5,

Mybatis 3.3.1

# 数据库

## 自服务平台

mysql 5.6.24

Redis 3.2.8

## 运营运维管理平台

MySQL 5.7

## 接口

MySQL 5.7

# 前端技术

## 自服务平台

Vue.js v2.2.6

jQuery v3.5.1

Bootstrap v3.3.7

layer-v3.0.1

## 运营运维管理平台

Extjs 4.1.0

## 接口

无

# 接口部署

## 自服务平台

对外开放端口：8082

用于用户登录后台查看绑定、呼叫等记录

## 运营运维管理平台

对外开放端口：8083

用于开户、放号等功能

## 接口

对外开放端口：2081

用于用户绑定、解绑号码

# 部署文件

注：tomcat和JDK解压即可用，JDK解压之后需要配置一下环境变量

## 安装MySQL

注：如果服务器可以上网的话建议使用yum安装，不可以上网才使用以下的安装步骤，yum安装方式见如下网页：<https://www.cnblogs.com/ianduin/p/7679239.html，使用yum安装完成之后可以略过1、2>、3、4，直接执行5、6、7、8

1. 卸载mariadb

rpm -qa | grep postfix

rpm -qa | grep mariadb

rpm -ev postfix-2.10.1-6.el7.x86\_64

rpm -ev mariadb-libs-5.5.52-1.el7.x86\_64

2. 安装mysql

mysql-community-common-5.7.19-1.el7.x86\_64.rpm

mysql-community-libs-5.7.19-1.el7.x86\_64.rpm

mysql-community-client-5.7.19-1.el7.x86\_64.rpm

先安装依赖：

libaio-0.3.109-13.el7.x86\_64.rpm

rpm -ivh perl-5.16.3-291.el7.x86\_64.rpm --nodeps

rpm -ivh perl-libs-5.16.3-291.el7.x86\_64.rpm --nodeps

rpm -ivh perl-Getopt-Long-2.40-2.el7.noarch.rpm --nodeps

再安装：

mysql-community-server-5.7.19-1.el7.x86\_64.rpm

3. 初始化

cd /var/lib

mv mysql mysql\_backup

mysqld --initialize --user=mysql

查看临时密码

grep password /var/log/mysqld.log

# 登录mysql，之后需要输入密码

mysql -p

# 设置新密码

set password = password("123456");

# 退出当前登录

quit;

4. 设置开机自启动

1 [root@localhost ~]# chkconfig mysql on

2 [root@localhost ~]# chkconfig --list | grep mysql

3 mysql 0:off 1:off 2:on 3:on 4:on 5:on 6:off

5、修改mysql的配置文件，修改之后重启mysql

event\_scheduler=ON

max\_connections=2048

innodb\_buffer\_pool\_size=2G

innodb\_flush\_log\_at\_trx\_commit=0

innodb\_log\_buffer\_size=32M

character\_set\_server=utf8

collation\_server='utf8\_general\_ci'

max\_heap\_table\_size=1280M

default-time-zone = '+08:00'

sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES

6、在mysql命令行使用命令create database SAFENUMBER；创建数据库，然后use SAFENUMBER之后使用source导入SAFENUMBER.sql

7、创建供接口、自服务平台、运维运营管理平台使用的账号safenum，密码最好是CvttSafe.95013

8、GRANT SELECT ON mysql.proc TO 'safenum'@'%';

## 安装redis

（1）安装redis：

rpm -ivh jemalloc-3.6.0-1.el7.x86\_64.rpm

rpm -ivh redis-3.2.3-1.el7.x86\_64.rpm

（2）生成配置文件

mkdir /etc/redis

cp /etc/redis.conf /etc/redis/6379.conf

cp /etc/redis.conf /etc/redis/6380.conf

mkdir /var/lib/redis/6379

mkdir /var/lib/redis/6380

(3) 修改配置文件/etc/redis/6379.conf（注意地址）

pidfile /var/run/redis\_6379.pid

logfile /var/log/redis\_6379.log

dir /var/lib/redis/6379

bind 192.168.106.74

port 6379

requirepass Av7W9e06O7wJGmbgDE4D

(4)修改配置文件/etc/redis/6380.conf（注意地址）

pidfile /var/run/redis\_6380.pid

logfile /var/log/redis\_6380.log

dir /var/lib/redis/6380

bind 192.168.106.74

port 6380

requirepass Av7W9e06O7wJGmbgDE4D

（5）将dump.rdb上传至/var/lib/redis/6379/目录下

（6）启动redis：

redis-server /etc/redis/6379.conf &

redis-server /etc/redis/6380.conf &

（7）将redis 6379和6380两个实例加入开机自启动

redis-server /etc/redis/6379.conf &

redis-server /etc/redis/6380.conf &

## 部署接口

1. 将safenumberservicessm.war上传到tomcat的webapps目录下
2. 将conf\_safenumberservice.tar.gz上传到tomcat目录下并解压
3. 修改配置文件jdbc.properties

jdbc.url=jdbc:mysql://ip:port/SAFENUMBER?useUnicode=true&characterEncoding=utf-8&serverTimezone=Asia/Shanghai

jdbc.username=

jdbc.password=

1. 修改配置文件redis.properties（如果安装redis时完全按照本文档操作的，那么此处只需要修改两个IP地址）

redis.master.host=ip

redis.master.port=port

redis.master.password=

redis.lastcaller.host=ip

redis.lastcaller.port=port

redis.lastcaller.password=

1. 启动tomcat

## 自服务平台

* 1. 将safenumberWeb.war上传到tomcat的webapps目录下
  2. 将conf\_safenumberWeb.tar.gz上传到tomcat目录下并解压
  3. 创建两个文件夹

mkdir /home/tcs\_file

mkdir /home/tcs\_file/recharge

* 1. 修改配置文件config.properties

#本服务地址和端口

LOCAL\_BASE\_PATH = [http://x](http://10.10.13.21:8080/)xx

# 接口地址和端口

REQUEST\_BASE\_PATH = [http://x](http://10.10.13.21:8081/)xx

# 接口地址和端口

DOWNLOAD\_BASE\_PATH = [http://x](http://ifs.95013.com:8081/safenumberservicessm/)xx

# 本地服务器公网地址

EXPORT\_BASE\_PATH =

#录音文件

RECORD\_FILE\_PATH =

* 1. 修改配置文件jdbc.properties

jdbc.url=jdbc:mysql://ip:port/SAFENUMBER?allowMultiQueries=true&useUnicode=true&characterEncoding=utf-8

jdbc.username=

jdbc.password=

jdbc2.url=jdbc:mysql://ip:port/SAFENUMBER?useUnicode=true&characterEncoding=utf8&autoReconnect=true&rewriteBatchedStatements=TRUE&useSSL=false

jdbc2.username=

jdbc2.password=

6、修改配置文件redis.properties（如果安装redis时完全按照本文档操作的，那么此处只需要修改IP地址）

redis.master.host=ip

redis.master.port=port

redis.pool.password=

7、启动tomcat

## 运营运维管理平台

* 1. 将safeWeb.war上传到tomcat的webapps目录下
  2. 将conf\_safeWeb.tar.gz上传到tomcat目录下并解压
  3. 修改配置文件app-config.xml，第1、2、4条为接口地址和端口，第3条为自服务平台地址和端口

<param name="safenumUrl" value="http://ip:port/safenumberservicessm/api/manage/dataManage?" />

<param name="safenumReuseUrl" value="http://ip:port/safenumberservicessm/api/manage/dataManageReuse?" />

<param name="safewebUrl" value="http://ip:port/safenumberWeb/sys/user/add\_unit?" />

<param name="recoverUrl" value="http://ip:port/safenumberservicessm/api/manage/recoverRelation?"/>

* 1. 修改配置文件application.properties

jdbc.url=jdbc:mysql://ip:port/SAFENUMBER?useUnicode=true&characterEncoding=utf-8&zeroDateTimeBehavior=convertToNull

jdbc.username=

jdbc.password=

safeWebData.url=jdbc:mysql://ip:port/SAFENUMBER?useUnicode=true&characterEncoding=utf-8&zeroDateTimeBehavior=convertToNull

safeWebData.username=

safeWebData.password=

safeWebStatistical.url=jdbc:mysql://ip:port/SAFENUMBER?useUnicode=true&characterEncoding=utf-8&zeroDateTimeBehavior=convertToNull

safeWebStatistical.username=

safeWebStatistical.password=

shard.jedis.host1=ip

shard.jedis.port1=port

shard.jedis.password1=

启动tomcat

* 1. 启动tomcat
  2. 复制配置文件，进入到conf\_safeWeb目录之后执行：

cp app-config.xml ../webapps/safeWeb/WEB-INF/conf/.

cp application.properties ../webapps/safeWeb/WEB-INF/conf/.

7、重启tomcat

# 配置文档

## 自服务平台

一、jdbc.properties

#主数据库(操作事务)

#数据库驱动

jdbc.driver=

#数据库地址

jdbc.url=

#数据库用户名

jdbc.username=

#数据库密码

jdbc.password=

#定义初始连接数

jdbc.pool.initialSize=10

#定义最大连接数

jdbc.pool.maxActive=100

#定义最大空闲

jdbc.pool.maxIdle=0

#定义最小空闲

jdbc.pool.minIdle=10

#定义最长等待时间

jdbc.pool.maxWait=60000

#连接检测

jdbc.pool.timeBetweenEvictionRunsMillis=60000

jdbc.pool.minEvictableIdleTimeMillis=300000

jdbc.pool.maxPoolPreparedStatementPerConnectionSize=100

#从数据库(不操作事务)

#数据库驱动

jdbc2.driver=

#数据库地址

jdbc2.url=

#数据库用户名

jdbc2.username=

#数据库密码

jdbc2.password=

#从数据库(不操作事务)

#数据库驱动

jdbc3.driver=com.mysql.cj.jdbc.Driver

#数据库地址

jdbc3.url=

#数据库用户名

jdbc3.username=

#数据库密码

jdbc3.password=

#从数据库

#数据库驱动

jdbc4.driver=

#数据库地址

jdbc4.url=

#数据库用户名

jdbc4.username=root

#数据库密码

jdbc4.password=123456

二、redis.properties，redis相关配置

#数据库地址

redis.master.host=

#端口

redis.master.port=

#密码

redis.pool.password=

#最大分配的对象数

redis.pool.maxActive=1024

#最大能够保持idel状态的对象数

redis.pool.maxIdle=200

#当池内没有返回对象时，最大等待时间

redis.pool.maxWait=100000

#当调用borrow Object方法时，是否进行有效性检查

redis.pool.testOnBorrow=true

#当调用return Object方法时，是否进行有效性检查

redis.pool.testOnReturn=true

#超时时间

spring.redis.timeout=

#密码

spring.redis.password=

##哨兵监控主redis节点名称

spring.redis.sentinel.master=

##redis哨兵集群节点

spring.redis.sentinel.nodes1=

spring.redis.sentinel.nodes2=

spring.redis.sentinel.nodes3=

#有两种取值singlePoint标识redis单点，sentinel标识哨兵

redis.redisMode=singlePoint

三、config.properties:一些下载路径、请求路径的配置

# 本地服务器地址

LOCAL\_BASE\_PATH =

# 本地服务器公网地址

EXPORT\_BASE\_PATH =

# 本地文件存储基础路径

LOCAL\_FILE\_PATH =

# 批量任务接口 (更改为批量任务接口地址)

BATCH\_WORK\_PATH =

# 批量结果下载地址 (更改为批量任务接口地址)

DOWNLOAD\_BASE\_PATH =

#录音文件下载地址（没有则保留此字段并置空）

RECORD\_FILE\_PATH =

# 存储加解密开关（0-不进行加解密）

CRYPTIC.SWITCH=0

# 页面显示是否隐藏电话号码(0不隐藏，1隐藏)

TEL\_HIDDEN=1

# 配置一个定时的时间(上午8点到晚上6点每隔30分钟轮巡一次)

QUARTZ.MODEL=

# 存储数据十六进制秘钥

AES.SALT.DB=

# tcs unitid

tcs.unitid=

# tcs充值凭证路径

tcs.recharge.path=

# tcs充值凭证访问路径

tcs.recharge.http=

#SFTP id必须传

SFTP\_UNITID =

# 读取本地目录回传到sftp

SFTP\_RELOADLOCALPATH =

#自动扫描的csv路径 (保证和内网传的地址一样)

POLL\_SAVE\_CSVPATH=

#传入内网的路径

PUSH\_SAVE\_CSVPATH=

#保存备份

PUSH\_SAVE\_CSVPATH\_BACKUP=

#拿取内网日志的路径

GET\_BATCH\_LOG=

#上传企业id

UPLOAD\_CHECK\_UNITID1=

#上传关键字

UPLOAD\_CHECK\_KEYWORDS1=

#上传企业id

UPLOAD\_CHECK\_UNITID2=

#上传关键字

UPLOAD\_CHECK\_KEYWORDS2=

#下载名称

DOWNLOAD\_FILE\_USERNAME=

#下载密码

DOWNLOAD\_FILE\_PASSWORD=

# 录音文件存储基础路径

RECORDING\_FILE\_PATH = files/recordings/

# ip白名单

ALLOW\_IP=0:0:0:0:0:0:0:1

四、log4j.properties:日志相关的配置

#日志记录的优先级

log4j.rootLogger=INFO,stdout,info,warn,error

#控制台日志配置，输出到控制台

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

#控制台日志，把日志输出到系统标准输出设备上

log4j.appender.stdout.Target=System.out

#控制台日志阈值

log4j.appender.stdout.Threshold=INFO

#设置使用日志布局样式，包含日志信息的级别和信息字符串

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

#控制台日志灵活定义输出格式

log4j.appender.stdout.layout.ConversionPattern=[%p][%d{yyyy-MM-dd HH:mm:ss}] %l %m %n

#INFO日志配置的优先级

log4j.logger.info=info

#INFO日志，每天产生一个日志文件

log4j.appender.info=org.apache.log4j.DailyRollingFileAppender

#INFO日志存储路径

log4j.appender.info.File =

#INFO日志日期格式

log4j.appender.info.datePattern=

#INFO日志附加该文件

log4j.appender.info.append=

#INFO日志阈值

log4j.appender.info.Threshold=

#设置使用日志布局样式，包含日志信息的级别和信息字符串

log4j.appender.info.layout=org.apache.log4j.PatternLayout

#INFO日志灵活定义输出格式

log4j.appender.info.layout.ConversionPattern=[%p][%d{yyyy-MM-dd HH:mm:ss}] %l %m %n

#WARN日志配置

log4j.appender.warn=org.apache.log4j.DailyRollingFileAppender

#WARN日志存储路径

log4j.appender.warn.File = ${catalina.base}/logs/safenumberWeb/warn.log

#WARN日志时间格式

log4j.appender.warn.datePattern='.'yyyy-MM-dd

#WARN日志附加该文件

log4j.appender.warn.append=true

#WARN日志阈值

log4j.appender.warn.Threshold=WARN

#WARN日志设置使用日志布局样式，包含日志信息的级别和信息字符串

log4j.appender.warn.layout=org.apache.log4j.PatternLayout

#WARN日志灵活定义输出格式

log4j.appender.warn.layout.ConversionPattern=[%p][%d{yyyy-MM-dd HH:mm:ss}] %l %m %n

#ERROR日志配置

log4j.appender.error=org.apache.log4j.DailyRollingFileAppender

#ERROR日志存储路径

log4j.appender.error.File =

#ERROR日志时间格式

log4j.appender.error.datePattern='.'yyyy-MM-dd

#ERROR日志附加该文件

log4j.appender.error.append=true

#ERROR日志阈值

log4j.appender.error.Threshold=ERROR

#ERROR日志设置使用日志布局样式，包含日志信息的级别和信息字符串

log4j.appender.error.layout=org.apache.log4j.PatternLayout

#ERROR日志灵活定义输出格式

log4j.appender.error.layout.ConversionPattern=[%p][%d{yyyy-MM-dd HH:mm:ss}] %l %m %n

#数据库日志配置的优先级

log4j.logger.com.alibaba.druid.filter.stat.StatFilter=ERROR,slowsql

#数据库日志

log4j.appender.slowsql=org.apache.log4j.DailyRollingFileAppender

#数据库日志存储路径

log4j.appender.slowsql.File =

#数据库日志时间格式

log4j.appender.slowsql.datePattern='.'yyyy-MM-dd

#数据库日志附加该文件

log4j.appender.slowsql.append=true

#数据库日志设置使用日志布局样式，包含日志信息的级别和信息字符串

log4j.appender.slowsql.layout=org.apache.log4j.PatternLayout

#数据库日志灵活定义输出格式

log4j.appender.slowsql.layout.ConversionPattern=[%d{yyyy-MM-dd HH:mm:ss}] %m %n

#Log4j屏蔽指定日志

log4j.logger.org.springframework=OFF

#Log4j屏蔽指定日志

log4j.logger.org.apache.velocity=OFF

## 运营运维管理平台

<?xml version="1.0" encoding="UTF-8"?>

<app-config name="APP" title="安全号运维">

<!--+

+ 日志配置文件更新检测时间，单位：秒。

+ 系统定时检查日志配置文件是否修改，如果发生改变，则自动更新配置。

+-->

<param name="logCheckedTime" value="" />

<!--+

+ 是否是发布版本，经过ANT打包后自动替换为true，开发时不需要修改参数。

+ 在线运行时，如果需要调试可以改为false

+-->

<param name="release" value="@RELEASE@" />

<!--+

+ 接口认证的AppKey

+-->

<param name="appKey" value="" />

<!--+

+ 接口认证的AppSe

+-->

<param name="appSecret" value="" />

<param name="url" value=""/>

<param name="numCount" value="" />

<param name="numCountStr" value="" />

<!-- 安全号解绑配置 -->

<param name="safenumUrl" value="" />

<param name="msgid" value="" />

<param name="msgtype" value="" />

<param name="ver" value="" />

<param name="service" value="" />

<param name="safenumReuseUrl" value="" />

<!-- 号码总量 -->

<param name="uidMax" value="" />

<param name="uidCountStr" value="" />

<param name="safewebUrl" value="" />

<!-- recover\_relation 解冻 安全号 -->

<param name="recover\_relation" value="" />

<param name="recoverUrl" value=""/>

<!--+ 接口认证的AppSe end +-->

<!-- 黄页 渠道号 和SRC -->

<param name="channel" value=""/>

<param name="src" value=""/>

<param name="key" value=""/>

<!-- end -->

<!-- - 外部访问 UseHost 配置true host 才会生效 -->

<param name="host" value=""/>

<param name="UseHost" value=""/>

<param name="appName" value=""/>

<param name="bearUser" value=""></param>

<param name="bearKey" value=""></param>

<!-- redis的两种模式，singlePoint表示单点，sentinel表示哨兵 -->

<param name="redisMode" value="singlePoint"/>

</app-config>

## 接口

#判断是否有效固话的正则表达式

regex.phone\_fixed =

#判断是否有效手机的正则表达式

regex.phone\_mobile =

#判断是否移动手机的正则表达式

regex.mobile\_yd =

#判断是否联通手机的正则表达式

regex.mobile\_lt =

#判断是否电信手机的正则表达式

regex.mobile\_dx =

#判断是否主叫认证外呼的正则表达式

regex.prefix\_authcall =

#判断是否符合95013号码正则表达式

regex.prefix\_95013 =

# 是否进行数据库存储加解密(0-不加密；1-加密)

CRYPTIC.SWITCH =

# 数据库存储加密秘钥

AES.SALT.DB =

# 京东传输加密秘钥

AES.SALT.JD =

# 京东unitid

JD.UNITID =

# 需要设置默认有效期的单位编码

DEFAULT\_EXPIRED\_UNITID =

# 接口请求放的ip信任表

request.ip.whitelist =

# 大网注册基础调用地址

passport.http.url =

# 注销接口基础调用地址

passport.quit.url =

# 主叫认证外呼服务域名

callservice.url =

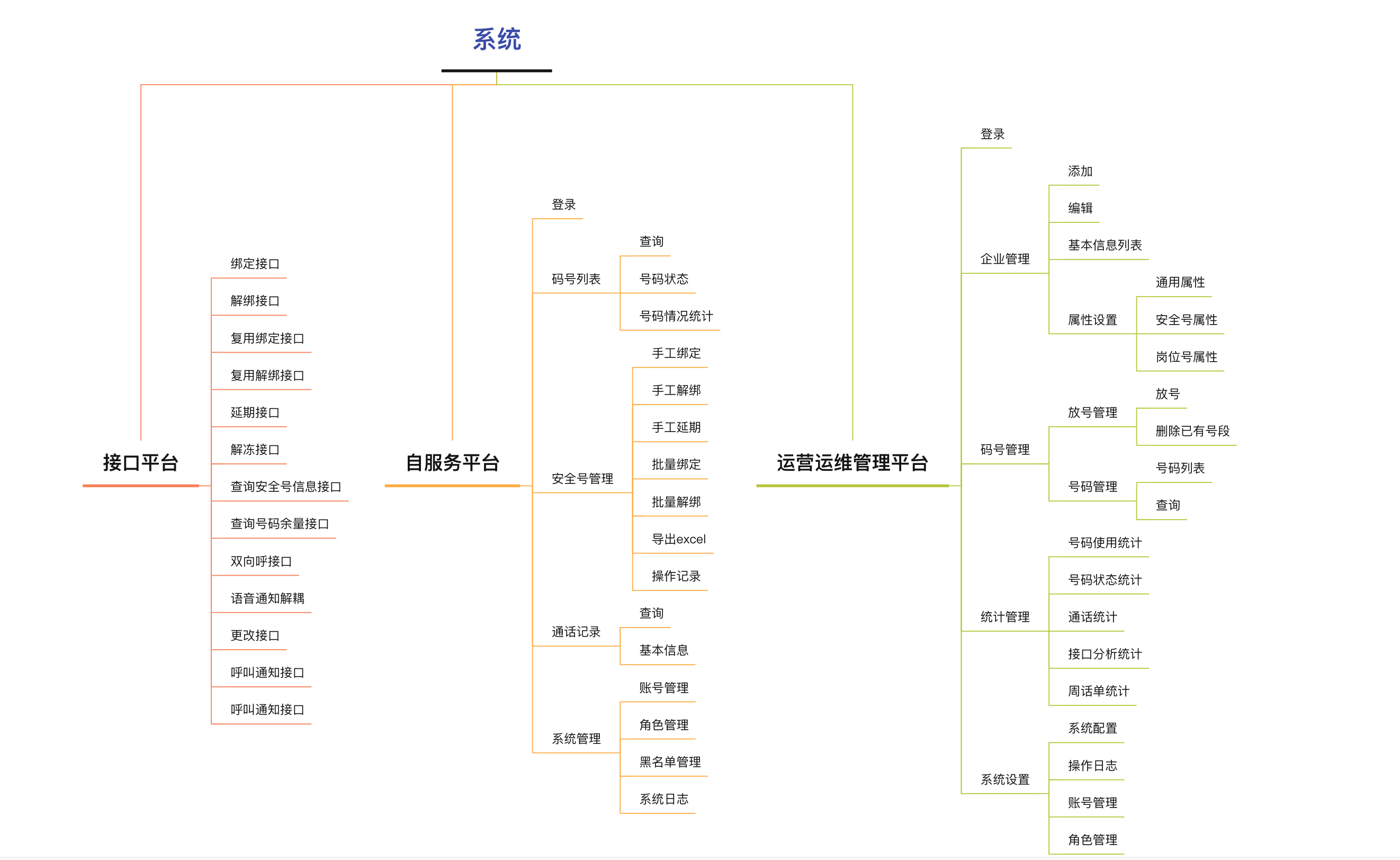
#判断双向呼bleg被叫号码分机号的正则表达式

regex.extnumber =

#企业默认主叫号码

UNIT\_DEFAULT\_CALLER ={}

# 功能结构说明



## 自服务平台

客户平台的WEB服务提供自服务功能，包括：

1. 单个号码绑定/解绑：建立或解除单个95013号码（安全号/岗位号）与对应真实电话的绑定关系；
2. 批量号码绑定/解绑：建立或解除多个95013号码（安全号/岗位号）与对应真实电话的绑定关系；
3. 批量换绑：批量更换客户电话与安全号码的绑定关系；
4. 定期自动换绑：在设定时间，周期性自动批量更换客户电话与安全号码的绑定关系；
5. 运营信息统计报表（本客户）：号码池的使用量、绑定次数统计；系统通话时长，通话次数统计；

## 运营运维管理平台

运维管理工具提供业务运营所需的功能，包括：

1. 开通合作方ID，配置接口所需秘钥；
2. 95013号池的配置（放号）；
3. 系统黑白名单和盲区配置；
4. 配置系统默认参数，包括：号码有效期和冷冻期，号码接听模式，接通时提示音，是否录音，来显控制；
5. 定期自动解绑：在设定时间，周期性自动批量解除客户电话与安全号码的绑定关系；
6. 运营信息统计报表（所有客户）：号码池的使用量、绑定次数统计；系统通话时长，通话次数统计；

## 接口

数据维护接口作为服务器端，供客户自有业务平台调用，实现95013号码（安全号，岗位号）与对应人员号码的绑定，解绑等功能：

1. 绑定：建立95013号码与真实电话号码之间关联关系，并设定是否进行来话控制、是否通话录音，通话来显等参数;
2. 解绑：解除95013号码与真实电话号码之间的关联关系。
3. 更新：变更95013号码与真实电话号码之间关联关系，或变更是否进行来话控制、是否通话录音，通话来显等参数;