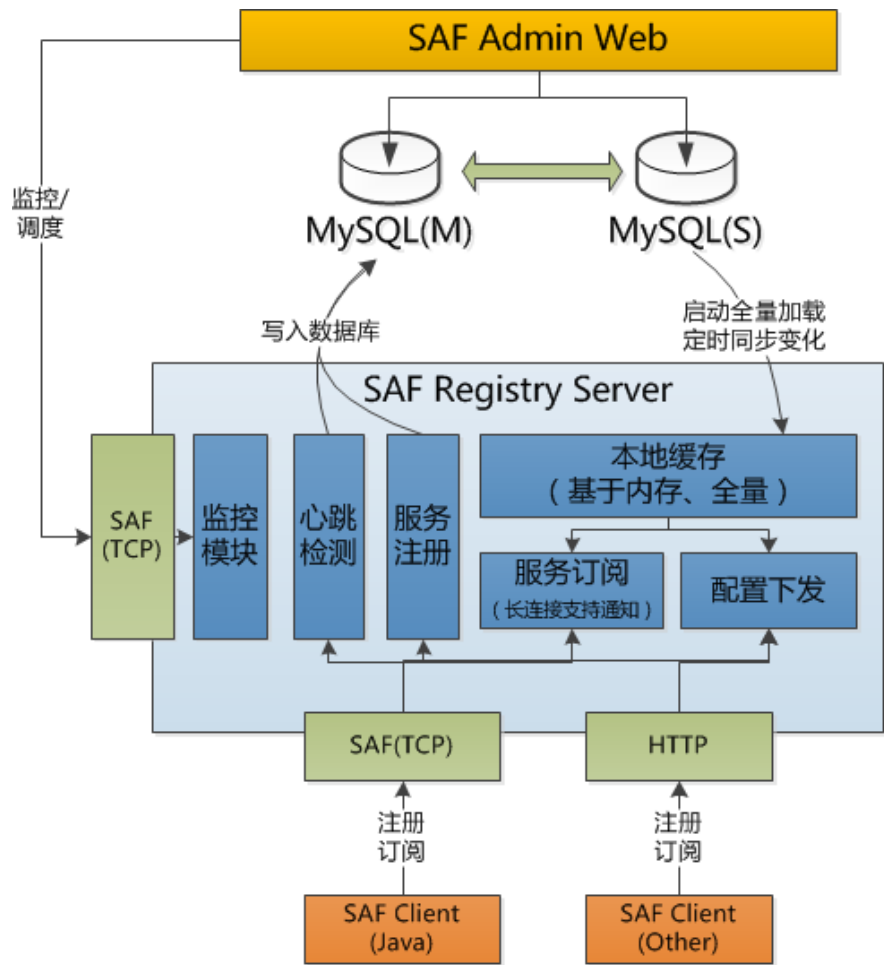


# SAF2.1注册中心设计

被章耿添加，被鲍宁天最后更新于五月 05, 2014

## 1.总体架构图



## 2.接口设计

注册中心对外暴露2个服务，

1. 针对客户端的注册订阅服务
2. 针对管理端的监控调度服务

### 注册订阅服务

主要是针对客户端调用，

提供服务的注册，服务的订阅（长连接支持事件通知），心跳发送，saf配置下发等服务。

将发布两种协议的服务：

1. saf协议，基于tcp的长连接，主要针对java客户端支持服务端发现服务列表变化后的主动通知（callback）。
2. http协议，基于http的短连接，主要针对saf未实现的语言的客户端。服务订阅只能客户端定时去取，无法做到服务端主动通知。

接口参考：

## com.jd.saf.registry.service.RegistryService

```

1  public interface RegistryService {
2      /**
3       * 注册数据，比如：提供者地址，消费者地址，路由规则，覆盖规则，等数据。
4       *
5       * 注册需处理契约：<br>
6       * 1. 当URL设置了check=false时，注册失败后不报错，在后台定时重试，否则抛出异常。<br>
7       * 2. 当URL设置了dynamic=false参数，则需持久存储，否则，当注册者出现断电等情况异常退出时，需自动删除。<br>
8       * 3. 当URL设置了category=routers时，表示分类存储，缺省类别为providers，可按分类部分通知数据。<br>
9       * 4. 当注册中心重启，网络抖动，不能丢失数据，包括断线自动删除数据。<br>
10      * 5. 允许URI相同但参数不同的URL并存，不能覆盖。<br>
11      *
12      * @param safUrl
13      *      注册信息，不允许为空，如：dubbo://10.20.153.10/com.alibaba.foo.BarService?
14      *      version=1.0.0&application=kylin
15      */
16      public void register(SafUrl safUrl);
17      /**
18      * 取消注册。
19      *
20      * 取消注册需处理契约：<br>
21      * 1. 如果是dynamic=false的持久存储数据，找不到注册数据，则抛IllegalStateException，否则忽略。<br>
22      * 2. 按全URL匹配取消注册。<br>
23      *
24      * @param safUrl
25      *      注册信息，不允许为空，如：dubbo://10.20.153.10/com.alibaba.foo.BarService?
26      *      version=1.0.0&application=kylin
27      */
28      public void unregister(SafUrl safUrl);
29      /**
30      * 订阅符合条件的已注册数据，当有注册数据变更时自动推送。
31      *
32      * 订阅需处理契约：<br>
33      * 1. 当URL设置了check=false时，订阅失败后不报错，在后台定时重试。<br>
34      * 2. 当URL设置了category=routers，只通知指定分类的数据，多个分类用逗号分隔，并允许星号通配，表示订阅所有
35      分类数据。<br>
36      * 3. 允许以interface,group,version,classifiser作为条件查询，如：
37      interface=com.alibaba.foo.BarService&version=1.0.0<br>
38      * 4. 并且查询条件允许星号通配，订阅所有接口的所有分组的所有版本，或：
39      interface=*&group=*&version=*&classifiser=*<br>
40      * 5. 当注册中心重启，网络抖动，需自动恢复订阅请求。<br>
41      * 6. 允许URI相同但参数不同的URL并存，不能覆盖。<br>
42      * 7. 必须阻塞订阅过程，等第一次通知完后再返回。<br>
43      *
44      * @param safUrl 订阅条件，不允许为空，如：consumer://10.20.153.10/com.alibaba.foo.BarService?
45      version=1.0.0&application=kylin
46      * @param listener 变更事件监听器，不允许为空
47      */
48      public void subscribe(SafUrl safUrl, CallbackListener listener);
49      /**
50      * 取消订阅。
51      *
52      * 取消订阅需处理契约：<br>
53      * 1. 如果没有订阅，直接忽略。<br>
54      * 2. 按全URL匹配取消订阅。<br>
55      *
56      * @param safUrl 订阅条件，不允许为空，如：consumer://10.20.153.10/com.alibaba.foo.BarService?
57      version=1.0.0&application=kylin
58      * @param listener 变更事件监听器，不允许为空
59      */
60      public void unsubscribe(SafUrl safUrl, CallbackListener listener);
61      /**
62      * 查询符合条件的已注册数据，与订阅的推模式相对应，这里为拉模式，只返回一次结果。
63      *
64      * @param safUrl 查询条件，不允许为空，如：consumer://10.20.153.10/com.alibaba.foo.BarService?
65      version=1.0.0&application=kylin
66      * @return 已注册信息列表，可能为空，含义同{@link
67      com.jd.service.registry.NotifyListener#notify(List<URL>)}的参数。
68      */
69      public List<SafUrl> lookup(SafUrl safUrl);
70      //public HeartbeatResult doHeartbeat(Heartbeat heartbeat);
71      /**
72      * 配置下发，在saf运行时，可以改变一些配置
73      * @param safUrls 查询的信息
74      * @return 带更新的配置信息
75      */

```

```
        public List<SafUrl> getConfig(List<SafUrl> safUrls);
    }
}
```

## 监控调度服务

主要针对管理端，  
提供注册中心服务状态的监控，注册中心的调度，配置中心一些参数的下发（也可以注册中心从db定时加载）等其它功能。  
由于管理端为Java的，所以只要发布saf协议接口。  
接口参考：

```
public interface ControlService {
    /**
     * 查询状态，包括连接数，请求数，数据文件大小等信息
     *
     * @param safUrl
     */
    public StatusInfo stat(SafUrl safUrl);
    /**
     * 暂停<br>
     * 已有请求不中断，不再接受新的请求，新请求返回暂停服务标识
     *
     * @param safUrl
     * @return
     */
    public StatusInfo pause(SafUrl safUrl);
    /**
     * 下线<br>
     * 已有请求中断，客户端将重连，不再接受新的请求，新请求返回下线服务标识
     *
     * @param safUrl
     * @return
     */
    public StatusInfo offline(SafUrl safUrl);
    /**
     * 配置下发，在saf注册中心运行时，可以改变一些配置
     *
     * @param safUrls
     *          查询的信息
     * @return 带更新的配置信息
     */
    public List<SafUrl> pushConfig(List<SafUrl> safUrls);
}
```

## 3.注册中心模块拆解

- 数据库结构大致如下：
- saf\_interface 接口表
  - saf\_interface\_group 接口分组（服务别名）表，一行数据表示唯一的一组服务
  - saf\_interface\_node 接口节点表，区分服务端和客户端
  - saf\_serviceinfo 接口表
  - saf\_server\_alias 接口分组表，表示用户
  - saf\_server 服务端节点表
  - saf\_client 客户端节点表

saf\_routerule 路由规则表

saf\_ipwb ip黑白名单表

## 服务注册

---

Provider客户端调用此方法完成provider节点注册：

1. 写入saf\_server
2. 更新saf\_server\_alias，表示有数据变更，等待同步
3. 是否按照实例的方式进行注册？？？

Consumer客户端调用此方法完成consumer节点注册：

1. 写入saf\_client

## 服务订阅

---

Consumer客户端调用此方法完成服务节点订阅：

1. 返回内存中对应的provider列表（路由后的provider列表。并检查当前订阅者是否在黑白名单中）。如何做到增量订阅？？？
2. 保留订阅到服务别名的listener（等变化的时候通知客户端）

## 配置下发

---

简单的配置查询接口。

提供定时检查配置，此配置为一些全局的运行时配置，例如心跳间隔，监控数据发送间隔等数据。

可以和心跳包结合返回。

## 心跳检测

---

客户端调用此方法发送心跳信息。

如果是长连接，可以只发少量数据。

如果是短连接，则需要带上客户端标识（ip、端口、pid等）。

步骤如下：

1. 得到客户端标识
2. 更新服务节点的最后心跳时间
3. 同时有一个worker 检查最后心跳时间，如果长时间未心跳，则置为死亡状态，发报警。

## 本地缓存

---

注册中心实例内存中缓存全部的服务列表和配置数据。

1. 在系统启动时候一次性加载全部数据到内存中
2. 数据更新采取一个定时服务每隔指定时间去数据库查询最新数据，同步最新数据。
3. 如果有变化，则需要触发回调机制。

## 监控模块

---

注册中心的统计功能，包括连接数，心跳数，注册数等。

同时有一些开关，包括状态开发，是否提供服务等。

# 4.注册中心数据库设计

---

## registry.sql-注册中心表结构

```

CREATE TABLE saf_serviceinfo (
  serviceinfo_id int(10) NOT NULL AUTO_INCREMENT,
  interface_id varchar(200) NOT NULL UNIQUE,
  application varchar(64) NOT NULL,
  is_important tinyint(1) NOT NULL,
  owner_group varchar(32) NOT NULL,
  data_version varchar(32) NOT NULL,
  subscribe_sttype tinyint(2) DEFAULT 0 NOT NULL comment '0-订阅online和offline状态, 1-只能订阅online',
  serviceinfo_desc varchar(512),
  create_date datetime NOT NULL,
  PRIMARY KEY (serviceinfo_id)) comment='接口表' CHARACTER SET UTF8 engine innodb;

CREATE TABLE saf_serviceinfo_depend (
  id int(10) NOT NULL AUTO_INCREMENT,
  serviceinfo_id int(10) NOT NULL,
  serviceinfo_childid int(10) NOT NULL,
  PRIMARY KEY (id)) comment='接口关联表' CHARACTER SET UTF8 engine innodb;

CREATE TABLE saf_group (
  group_id int(10) NOT NULL AUTO_INCREMENT,
  group_name varchar(64) NOT NULL UNIQUE comment '该字段做唯一值约束, 并且字段的数据需要做大小写区分。该字段的
  BINARY需要手动添加, 不能通过VP工具生成sql',
  group_desc varchar(512),
  creator_id int(10) NOT NULL,
  create_date datetime NOT NULL,
  update_date datetime NOT NULL,
  PRIMARY KEY (group_id)) comment='group表' CHARACTER SET UTF8 engine innodb;

CREATE TABLE saf_server_group (
  id int(10) NOT NULL AUTO_INCREMENT,
  server_id int(10) NOT NULL,
  serviceinfo_id int(10) NOT NULL,
  group_id int(10) NOT NULL,
  src_type tinyint(2) NOT NULL comment '添加类型: 1-proxy, 2-manual',
  PRIMARY KEY (id)) comment='server和group关联表' CHARACTER SET UTF8 engine innodb;

CREATE TABLE saf_server (
  server_id int(10) NOT NULL AUTO_INCREMENT,
  server_uniquekey varchar(128) NOT NULL comment 'ip_port_group_version_protocol_serviceInfoId , 将这6个字段作
  为provider的唯一判断和索引, 在此不做唯一约束, 通过程序进行判断',
  server_group varchar(64) NOT NULL comment '默认group',
  server_version varchar(16) NOT NULL comment 'server的版本号',
  server_ip varchar(32) NOT NULL comment 'server的IP地址',
  server_port int(6) NOT NULL comment 'server的端口号',
  server_pid int(8) comment 'server的PID',
  server_token varchar(32),
  saf_version varchar(16) NOT NULL,
  server_status tinyint(2) NOT NULL,
  server_room tinyint(2) NOT NULL,
  server_timeout int(8),
  server_subnet tinyint(2),
  server_weight int(6),
  server_monitor tinyint(2),
  server_apppath varchar(128) NOT NULL,
  protocol varchar(16),
  protocol_context varchar(32),
  serviceinfo_id int(10) NOT NULL,
  is_random tinyint(1) NOT NULL,
  src_type tinyint(2) NOT NULL comment 'data type: 1-proxy, 2-manual, 3-zookeeper 用于数据同步',
  attr_url varchar(255) comment '保存safurl中的attr参数, 过滤掉一些在表中已经存在的参数',
  safurl_desc varchar(1023),
  create_date datetime NOT NULL,
  update_date datetime NOT NULL,
  PRIMARY KEY (server_id)) comment='服务提供者表' CHARACTER SET UTF8 engine innodb;

CREATE TABLE saf_client (
  client_id int(10) NOT NULL AUTO_INCREMENT,
  client_uniquekey varchar(128) NOT NULL comment 'ip_pid_group_version_serviceInfoId , 将这5个字段作为consumer
  的唯一判断和索引。为了降低唯一约束对在此不做唯一约束, 通过程序进行唯一判断',
  client_ip varchar(32) NOT NULL,
  client_apppath varchar(128),
  client_pid int(8),
  client_group varchar(64) NOT NULL,
  client_token varchar(32),

```

```
client_version    varchar(16) NOT NULL,
saf_version       varchar(16) NOT NULL,
client_status     tinyint(2) NOT NULL,
serviceinfo_id    int(10) NOT NULL,
group_id          int(10) NOT NULL,
src_type          tinyint(2) NOT NULL comment '数据来源: 1-proxy, 2-manual, 3-zookeeper',
safurl_desc       varchar(1023),
start_time        varchar(31) NOT NULL comment '节点启动时间',
create_date       datetime NOT NULL,
update_date       datetime NOT NULL,
PRIMARY KEY (client_id)) comment='服务调用者表' CHARACTER SET UTF8 engine innodb;
```

```
CREATE TABLE saf_routerule (
routerule_id      int(10) NOT NULL AUTO_INCREMENT,
interface_id      varchar(200) NOT NULL,
source_ip         varchar(32),
source_version    varchar(32),
source_group      varchar(32),
source_room       varchar(32),
source_subnet     varchar(32),
dest_ip           varchar(32),
dest_port         varchar(8),
dest_version      varchar(16),
dest_group        varchar(32),
dest_room         varchar(32),
dest_subnet       varchar(32),
expression        varchar(128),
routerule_type    tinyint(2) NOT NULL,
is_valid          tinyint(1) NOT NULL,
PRIMARY KEY (routerule_id)) comment='路由表' CHARACTER SET UTF8 engine innodb;
```

```
CREATE TABLE saf_room (
id                int(10) NOT NULL AUTO_INCREMENT,
computer_room     tinyint(2) NOT NULL,
ip_section        varchar(32) NOT NULL,
ip_regular        varchar(255),
PRIMARY KEY (id)) comment='机房表' CHARACTER SET UTF8 engine innodb;
```

```
CREATE INDEX query_index_server_uniquekey
ON saf_server (server_uniquekey);
CREATE INDEX query_index_client_uniquekey
ON saf_client (client_uniquekey);
```

## admin.sql 管理端表结构

```

CREATE TABLE saf_upload_jar_info (
  id int(10) NOT NULL AUTO_INCREMENT comment '主键',
  erp_id varchar(63) NOT NULL,
  upload_user_name varchar(63) NOT NULL comment '上传用户名, 冗余字段',
  create_date datetime NOT NULL,
  update_date datetime NOT NULL,
  jar_version_name varchar(63) NOT NULL comment '上传jar包名称',
  save_path varchar(127) NOT NULL comment '保存路径',
  md_five_value varchar(127) NOT NULL comment '文件上传时md5值',
  PRIMARY KEY (id)) comment='上传jar包信息表' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_serviceinfo_jar (
  id int(10) NOT NULL AUTO_INCREMENT comment '主键',
  ip_regular varchar(63) NOT NULL comment 'ip表达式',
  create_date datetime NOT NULL,
  update_date datetime NOT NULL,
  jar_version_name varchar(255) NOT NULL comment 'jar包名称',
  app_path varchar(255) NOT NULL comment '实例路径, 对应一个jvm',
  PRIMARY KEY (id)) comment='下载jar规则表' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_monitor_property (
  id int(10) NOT NULL AUTO_INCREMENT comment '自增id',
  interface_id varchar(200) NOT NULL comment '接口名称',
  monitor_type varchar(20) NOT NULL comment '监控类型',
  monitor_value varchar(511) NOT NULL comment '值',
  PRIMARY KEY (id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_monitor_proxy (
  id int(10) NOT NULL AUTO_INCREMENT,
  proxy_ip varchar(32) NOT NULL,
  port int(6) NOT NULL,
  monitor_type int(10) NOT NULL comment '1-每分钟注册数, 2-每分钟心跳数, 3-当前连接数',
  monitor_time datetime NOT NULL,
  m_value int(10) NOT NULL,
  create_date datetime NOT NULL,
  PRIMARY KEY (id)) comment='监控proxy实例' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_service_tracelog (
  id int(10) NOT NULL AUTO_INCREMENT,
  interfaceID varchar(200) comment '接口名字',
  group_name varchar(64) comment '组',
  version varchar(16) comment '版本',
  ip_addr varchar(32) comment 'ip地址',
  port int(6) comment '端口',
  protocol varchar(16) comment '协议',
  pid int(6) comment '进程id',
  saf_version varchar(16),
  saf_action tinyint(1) comment 'saf上下线 0下线 1上线',
  isprovider tinyint(1) comment '是否服务端 1服务端0客户端',
  url varchar(1024),
  create_time datetime comment '创建时间',
  update_time datetime comment '更新时间',
  params varchar(120) comment '拓展字段, 用于功能拓展使用, 保存格式为key:value;',
  msg_src varchar(20) comment '日志来源, 1 状态扫描, 2 服务注册',
  PRIMARY KEY (id)) comment='接口上下线记录表' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_trace_tps (
  id int(10) NOT NULL AUTO_INCREMENT comment 'id',
  interface_id varchar(200) NOT NULL comment '接口名',
  method varchar(80) NOT NULL comment '方法名',
  avg_tps double NOT NULL comment '平均tps',
  total_tps double NOT NULL comment '总tps',
  create_time timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP NOT NULL comment '发送时间',
  PRIMARY KEY (id)) CHARACTER SET UTF8;
CREATE TABLE saf_trace_perform (
  id int(10) NOT NULL AUTO_INCREMENT comment '自增id',
  host varchar(80) NOT NULL comment 'ip',
  port int(6) NOT NULL comment '端口',
  interface_id varchar(200) NOT NULL comment '接口名',
  method varchar(80) NOT NULL comment '方法名',
  elapsed int(5) NOT NULL comment '耗时',
  send_time timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP NOT NULL comment '发送时间',
  PRIMARY KEY (id)) comment='单个接口trace性能' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_monitor_error (
  id int(11) NOT NULL AUTO_INCREMENT,
  interface_id varchar(200) NOT NULL comment '接口名',
  method varchar(100) NOT NULL comment '方法名',
  `group` varchar(60) comment '组名',
  version varchar(60) comment '版本',
  error_name varchar(100) comment '错误类型',

```



```

times          int(10) comment '异常次数',
provider       varchar(100) NOT NULL comment '提供者',
is_provider    tinyint(1) NOT NULL comment '是否提供者',
port          int(5) comment '端口',
error_message  varchar(5000) comment '异常信息',
error_stack    varchar(10000) comment '异常堆栈',
create_time    timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP NOT NULL comment '发送时间',
consumer      varchar(100) NOT NULL comment '调用者',
protocol       varchar(40) comment '协议',
PRIMARY KEY (id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_statistics (
  id          int(10) NOT NULL AUTO_INCREMENT,
  weekend      tinyint(2) NOT NULL comment '第几周',
  selection   tinyint(2) NOT NULL comment '统计标准 实例: ip',
  sum        int(4) NOT NULL comment '数量',
  create_date timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP NOT NULL,
  PRIMARY KEY (id)) CHARACTER SET UTF8;
CREATE TABLE saf_invocation_monitor (
  id          int(10) NOT NULL AUTO_INCREMENT comment '自增id',
  client_ip   varchar(40),
  host        varchar(40),
  `group`     varchar(60),
  interface_id varchar(200),
  method      varchar(80),
  version     varchar(40),
  record_time datetime comment '发送时间',
  protocol    varchar(20),
  application varchar(80),
  port        int(6) comment '服务端口',
  success_num int(10) comment '调用成功数量',
  fail_num    int(10) comment '调用失败数量',
  concurrent  int(11) comment '当前并发数',
  elapsed     int(11) comment '调用服务平均耗时',
  max_elapsed int(11) comment '最长耗时',
  record_by   tinyint(1) comment '记录着 0 provider 1 consumer',
  save_time   datetime,
  PRIMARY KEY (id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_dept (
  id          int(6) NOT NULL AUTO_INCREMENT comment '主键',
  ref_id      int(6) NOT NULL comment '所属一级部门',
  dept_name   varchar(200) comment '部门名称',
  remark      varchar(400) comment '备注',
  create_time datetime comment '创建时间',
  update_time datetime comment '更改时间',
  PRIMARY KEY (id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_user_action (
  id          int(10) NOT NULL AUTO_INCREMENT,
  user_name   varchar(32) NOT NULL comment '用户名',
  user_erp_id varchar(32) NOT NULL comment 'erp帐户',
  action_type varchar(32) NOT NULL comment '操作类型',
  create_time datetime NOT NULL,
  detail      varchar(5000),
  PRIMARY KEY (id)) comment='用户操作日志' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_alarmrule_user (
  id          int(10) NOT NULL AUTO_INCREMENT comment '记录ID',
  rule_id     int(11) comment '所触发的规则(外键)',
  user_id     int(11) comment '规关联用户(外键)',
  PRIMARY KEY (id)) comment='报警规则用户关联表' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_alarmrule (
  id          int(10) NOT NULL AUTO_INCREMENT comment '规则ID',
  name        varchar(128) NOT NULL comment '规则名称',
  object_type varchar(32) comment '对象类型(service-p-count,service-c-count,service-up-down)',
  object_id   varchar(200) comment '对象ID',
  threshold   varchar(256) comment '阈值',
  durable_time int(11) comment '最长允许连续异常时间',
  alarm_min_interval int(11) comment '警报最短间隔时间,用于防止频繁发警报',
  create_date datetime comment '规则创建时间',
  created_user varchar(32) comment '规则创建人',
  modified    datetime comment '规则修改时间',
  modified_user varchar(30) comment '规则修改人',
  status      tinyint(3) comment '状态(1:启用;2:停用;3:删除)',
  supplement  varchar(1024) comment '规则补充字段, 用于拓展(key:value形似存储)',
  ip_list     varchar(1024) comment '关联ip,多个逗号分隔',
  common      tinyint(1) comment '标记是否是共有规则',
  PRIMARY KEY (id)) comment='报警规则表' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_alarmevent (
  id          int(10) NOT NULL AUTO_INCREMENT comment '报警记录ID',
  object_type varchar(32) NOT NULL comment '对象类型(service-p-count,service-c-count,service-up-down)',

```

```

object_id    varchar(200) NOT NULL comment '对象ID',
rule_id      int(11) NOT NULL comment '所触发的规则(外键)',
detail       varchar(2048) NOT NULL comment '警告内容',
status       tinyint(3) NOT NULL comment '状态',
created      datetime NOT NULL comment '创建时间',
sent_time    datetime comment '发送时间',
try_times    tinyint(10) NOT NULL comment '尝试发送报警次数',
task_group   int(11) comment '任务组号',
PRIMARY KEY (id)) comment='报警任务,status为0没有处理, 1处理但不报警, 2处理且报警' CHARACTER SET UTF8 engine
innodb;
CREATE TABLE saf_suggestion (
  id          int(10) NOT NULL AUTO_INCREMENT comment '主键ID',
  title       varchar(128) comment '建议名称',
  context     varchar(2048) comment '内容',
  create_time datetime comment '创建时间',
  create_user_id int(11) comment '创建人',
  update_time datetime comment '修改时间',
  update_user_id int(11) comment '修改人',
  parent_id   int(11) comment '父id',
  PRIMARY KEY (id)) comment='建议问题表' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_param (
  id          int(10) NOT NULL AUTO_INCREMENT,
  param_key   varchar(32) NOT NULL,
  param_name  varchar(32) NOT NULL,
  param_value varchar(128),
  param_type  tinyint(2) NOT NULL,
  note        varchar(512),
  PRIMARY KEY (id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_notice (
  id          int(10) NOT NULL AUTO_INCREMENT comment '自增id',
  content     text NOT NULL comment '公告内容',
  title       varchar(128) NOT NULL comment '标题',
  is_top      tinyint(1) comment '是否置顶',
  default_show tinyint(1) comment '是否默认显示',
  create_date datetime NOT NULL comment '创建时间',
  update_date datetime NOT NULL comment '更新时间',
  PRIMARY KEY (id)) comment='公告通知表' CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_user_resource (
  user_id int(10) NOT NULL,
  resource varchar(64) NOT NULL,
  PRIMARY KEY (user_id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_user_ip (
  id          int(10) NOT NULL AUTO_INCREMENT,
  user_id int(10) NOT NULL,
  ip          varchar(32) NOT NULL,
  PRIMARY KEY (id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_provider_log (
  id          int(10) NOT NULL AUTO_INCREMENT comment '主键ID',
  provider    varchar(512) NOT NULL comment '操作日志:变化的节点',
  log_type    varchar(16) NOT NULL comment '日志类型:zk_add;zk_del;redisAdd;redisDel;db_add;db_del;',
  creator     varchar(32) NOT NULL comment '日志的来源: 例如redisSynZkWorker',
  create_date datetime NOT NULL comment '创建时间',
  log_desc    varchar(256) comment '备用字段',
  PRIMARY KEY (id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_user (
  user_id          int(10) NOT NULL AUTO_INCREMENT,
  user_erpid       varchar(32) NOT NULL UNIQUE,
  user_name        varchar(64) NOT NULL,
  user_department  varchar(256) NOT NULL,
  user_email       varchar(32) NOT NULL,
  user_cellphone   varchar(16) NOT NULL,
  PRIMARY KEY (user_id)) CHARACTER SET UTF8 engine innodb;
CREATE TABLE saf_serviceinfo_user (
  id          int(10) NOT NULL AUTO_INCREMENT,
  serviceinfo_id int(10) NOT NULL,
  user_id     int(10) NOT NULL,
  user_type   tinyint(1) NOT NULL comment '1- watch user; 2- owner user',
  PRIMARY KEY (id)) CHARACTER SET UTF8 engine innodb;
CREATE INDEX query_index
  ON saf_invocation_monitor (interface_id, host, client_ip, record_time);
CREATE INDEX query_index_group
  ON saf_invocation_monitor (`group`);
CREATE INDEX query_index_version
  ON saf_invocation_monitor (version);
CREATE INDEX query_index_method
  ON saf_invocation_monitor (method);
CREATE INDEX query_index_interface
  ON saf_invocation_monitor (interface_id);

```

无