C 接口接出端部署手册

1、中间库创建

增加一个数据库：cinterdb_out（这个是给这个 C 接口专用的中间库），建立 4.0 协议的中间表。

2、C 接口配置

（1）数据库配置

添加 C 接口读取配置

在数据库中执行该语句，其中`dbsvrip`，`dbsvrport`，`dbname`，`dblogname`，`dblogpwd`这几个值要修改为现网的配置数据库对应的 ip，端口，数据库名，数据库账户名，数据库密码

INSERT INTO t_cfg_cserverinfo

(lsc_id, lsc_name, csvrip, csvrport, clogusrname, clogusrpwd, dbtype, dbsvrip, dbsvrport, dbname, dblogname, dblogpwd, access_device_id, protocol_id)

VALUES('8', 'C 接口接出', '10.1.203.119', 8266, 'admin', 'admin@123', 6, '10.1.203.120', 3306, 'cinterdb_out', 'root', 'nZ0qJ8kA1aI9', '00001006000000755298', 400);

（2）配置文件

接出端配置如下所示，app.runMode=server，app.server.clientId 设置为上面数据库配置对应的 lsc_id，app.server.version 设置为 400，通过 app.server.WRITE_TOP_PRECINCT 设置只接入哪个区域下的数据

```
app:
  tempPath: /root
  runMode: server
  server: #接出端配置
    clientId: 8 #上级 SC 的 SCID
    rightLevel : 2  #上级 SC 的权限
    isEnableTerminalId: false #是否启用 terminalID 参数，客户端也要同步开启
    version: 400 #服务端版本号
    WRITE_TOP_PRECINCT: "01-32-09-05-04" #将配置管理该区域下的所有信息写入中间库，没有则是"01"
```

```
10.     syncHisProvince: 01-32 #同步的省份 id
11.     alarmSeqType: cinterface_out_mockserver
12.     mock: true
13.     incrWriteTaskCron: 0 0/1 * * * *
14.     fullWriteTaskCron: 0 0/1 * * * *
15.     syncHisCron: 0 0/1 * * * *
```

3、启动服务

参考其它服务的启动方式，启动 C 接口镜像即可

docker run --name cinterface-service-onelink --net host --log-driver=json-file --log-opt max-size=30m --log-opt max-file=3 --env ENV_APP_NAME=cinterface-service-onelink --env ENV_NACOS=10.1.5.109:8848 --env ENV_TYPE=yunnan --env ENV_NACOS_PASSWORD=r2G%zwoCj#Oz -v /tmp/logs/cinterface:/opt/data/logs/ -d 10.1.6.34:8080/spider/yunnan/cinterface-service:spider1.0.0.0_kernelYunNan_SYT_146

该例子 ENV_APP_NAME 为 cinterface-service-onelink，则对应的配置文件为 ms-cinterface-service-onelink.yml。

4、调试

调用接口全量稽核：

curl --location 'http://localhost:8295/v1/cinterface/checkAndFixWriteCdb?siteId=&checkSignal=true'

SiteId 不为 null 时指定站点稽核同步

调用接口：curl --location 'http://localhost:8295/v1/cinterface/startWriteTask'

开启 C 接口接出定时任务

通过下面的配置可以控制对应的运行频率：

（1）增量稽核定时器

app.server.incrWriteTaskCron

（2）全量稽核定时器

app.server.fullWriteTaskCron

（3）历史数据接出定时器

app.server.syncHisCron