API 使用说明：

Note：

目前请勿将程序应用sun主机环境, 总部主机的sun主机程序编译存在问题!可以用于 linux aix hp

测试报文汇总:

1:调用ESB的客户信息校验接口:

1.1 webservice服务

http://172.21.1.95:51900/esbWS/services/sCustInfoValid

1.1.1测试报文:

<ROOT><PHONE\_NO type="string">15799881753</PHONE\_NO><TEST type="string">#####中文测试，校验是否有乱码出现，ESB虽然透传，但是改变了编码，需要在发送数据的时候将字符串转换成UTF-8编码#####</TEST></ROOT>

1.2 rest 服务

http://192.168.0.11:51900/esbWS/rest/sCustInfoValid

1.2.1 测试报文

{"ROOT":{"HEADER":{"POOL\_ID":"","DB\_ID":"","ENV\_ID":"","ROUTING":{"ROUTE\_KEY":"","ROUTE\_VALUE":""}},"BODY":{"PHONE\_NO":"15799881753","TEST":"ESB在rest服务的时候,也没有进行透传,输入报文需要是UTF-8的格式,返回的报文不需要进行UTF-8转换,需要和WEB服务进行核对中文测试"}}}

2:直接调用zhaojie提供的wsdl:

http://172.21.0.122:21106/services/WsGetSaleAction?wsdl

对应服务中的url:

http://172.21.0.122:21106/services/WsGetSaleAction.WsGetSaleActionHttpEndpoint/

2.1 关于如何获取URL地址的方法

1):浏览器中输入:

http://172.21.0.122:21106/services/WsGetSaleAction?wsdl

2):找到这一项:

<wsdl:port name="WsGetSaleActionHttpSoap11Endpoint" binding="ns:WsGetSaleActionSoap11Binding"><soap:address location="http://172.21.0.122:21106/services/WsGetSaleAction.WsGetSaleActionHttpSoap11Endpoint/"/></wsdl:port>

location的值就是url的地址.

2.1.2 测试报文:

<?xml version="1.0" encoding="GBK" standalone="no" ?>

<ROOT>

<COMMON\_INFO>

<PROVINCE\_GROUP type="string">10011</PROVINCE\_GROUP>

<SESSION\_ID type="string">23CGRBDMttwgBpsh6mSyTJqtggVc2pYLmGSVCLXF8ypG8L59jdGQ!-1625009959!1363248044628</SESSION\_ID>

<SER\_NAME type="string"></SER\_NAME >

<ROUND\_AUDIT type="string">Y</ROUND\_AUDIT >

<DEST\_IP type="string">172.21.0.123</DEST\_IP>

<DEST\_PORT type="string">11001</DEST\_PORT>

<SRC\_PORT type="string">4567</SRC\_PORT>

<APP\_NAME type="string">NGBOSS</APP\_NAME>

<APP\_ID type="string">2000</APP\_ID>

<FROM\_SYS type="string">BOSS</FROM\_SYS>

</COMMON\_INFO>

<REQUEST\_INFO>

<PHONE\_NO type="string">13546799494</PHONE\_NO>

<ID\_NO type="string">16310001001540</ID\_NO>

<BRAND\_ID type="string">ys</BRAND\_ID>

<LOGIN\_NO type="string">faaa48</LOGIN\_NO>

<REGION\_CODE type="string">16</REGION\_CODE>

<CHANNEL\_TYPE type="string">0</CHANNEL\_TYPE>

<GROUP\_ID type="string">13765</GROUP\_ID>

<OP\_CODE type="string">1147</OP\_CODE>

<BUSI\_INFO type="string"></BUSI\_INFO>

<DISTRICT\_CODE type="string">10145</DISTRICT\_CODE>

<FLAG type="string">4</FLAG>

<CUST\_FLAG type="string">0</CUST\_FLAG>

<RED\_FLAG type="string">N</RED\_FLAG>

<BLACK\_FLAG type="string">N</BLACK\_FLAG>

<ACT\_CLASS type="string"></ACT\_CLASS>

<MARKOBJ\_TYPE type="string">00</MARKOBJ\_TYPE>

<TMP\_TEST type="string">#######中文测试#######</TMP\_TEST>

</REQUEST\_INFO>

</ROOT>

其中对api的说明:

@in: url webservice服务的url地址,如何获取详见上文;

@in: conn\_timeout 连接超时时间;

@in: recv\_timeout 接受超时时间;

@in: inmsg 调用服务的入参报文

@in: need2utf 入参报文是否需要转UTF-8编码

@in: outmsg 指针的地址,用于保存服务返回的信息,错误信息也保存在内;

@in: need2gbk 出惨是否转为 GBK编码;

@out: 0表示成功, 成功报文在 \*outmsg 中保存,使用完毕请free调,否则内存泄漏;

非0表示失败, 失败报文在 \*outmsg 中保存,使用完毕请free调,否则内存泄漏;

int call\_esb\_ws(char \*url, int conn\_timeout, int recv\_timeout,

char \*inmsg, int need2utf,char \*\*outmsg, int need2gbk);

int call\_esb\_rest(char \*url,int conn\_timeout, int recv\_timeout,

char \*inmsg, int need2utf, char \*\*outmsg, int need2gbk);

详细的例子请看:

demo.cpp

demo.sh

#调用服务集成平台发布的 webservice 服务 客户信息校验服务;

./demo 0 "http://172.21.1.95:51900/esbWS/services/sCustInfoValid" esba.txt

#调用服务集成平台发布的 rest 服务 客户信息校验服务;

./demo 1 "http://172.21.1.95:51900/esbWS/rest/sCustInfoValid" esbb.txt

#调用 zhaojie 提供的 webservice 服务 ;

./demo 0 "http://172.21.0.122:21106/services/WsGetSaleAction.WsGetSaleActionHttpEndpoint/" esbc.txt

usage: demo 0|1 url msg\_filename

>>>>>: demo 0|1 其中: 0为ws服务; 1为rest服务

>>>>>: demo url 其中: url为ws服务或者rest服务的地址

>>>>>: demo url 其中: msg\_filename为一个文件名称,存放服务入参的报文

程序的编译和连接:

目前程序已经在 Linux, AIX , HP 主机上测试通过,

因为总部 Sun 主机的问题, 程序只能编译为32位程序,目前无法测试通过, SUN主机慎用,带测试通过后

可以应用于SUN主机.

demo的实例编译为:

mk.AIX mk.CYGWIN\_NT-6.1 mk.HP-UX mk.Linux

运行demo:

1): 复制 esb\_cli 目录到 主机上;

2): 执行 sh b.sh

3): sh demo.sh

注意服务入参和出参中文是否是乱码的问题.

1:根据 httpposttest.h 文件生成服务端的相关代码;soapcpp2 -S -c httpposttest.h2:将gsoap提供的 httppost.c stdsoap2.c 和 httppost.h stdsoap2.h复制到当前目录中;3:执行 b.sh (build.sh)编译程序;4:mm.sh检测内存是否有泄漏;如何根据wsdl生成客户端代码：

1：

esb\_cli 目录程序相关:/mnt/hgfs/lpstu/open/gsoap/demo/test/esb\_cli获得esb的wsdl,zhengyq提供的有bug,需要自己裁剪一下;wsdl2h -c -o esb.h esb.wsdl 如果只是支持 soap1.1的话，就将esb.h中的

#import "soap12.h"

注释掉；

soapcpp2 -C -c esb.h然后执行maker.sh 启动多个客户端程序;rs.sh 启动模拟段服务程序;rsp\_json.data json模拟数据;rsp\_xml.data xml模拟数据;如果需要重新生成客户端代码,需要在下面这个文件中添加宏的防重复声明;sDynSvcSoap11Binding.nsmap