GSoap: SOAP and XML Web services for Qt apps

Introduction

The gSOAP toolkit is an open source C and C++ software development toolkit for SOAP/XML Web services and generic (non-SOAP) C/C++ XML data bindings. The toolkit analyzes WSDLs and XML schemes (separately or as a combined set) and maps the XML scheme types and the SOAP messaging protocols to easy-to-use and efficient C and C++ code. The toolkit is mature, supports Symbian as well as Maemo/MeeGo; it supports many industry-standard protocols (SOAP 1.1/1.2, WSDL 1.1...) and transports (HTTP/S, TCP, UDP (SOAP-over-UDP), MIME (SwA), DIME - streaming, MTOM - streaming, HTTP1.0/1.1, IPv4, IPv6, RSS, XML-RPC, WS-Addressing).

Goal

Goal of this article is to show how to create an application which makes use of web services using gSOAP. This app will be able to get the NOK (Nokia) stock quote.

Steps

- First of all, you need to download the scheme of the web service to use: in this example the scheme has been downloaded from webservicex.net
- The WSDL scheme has been saved in the file stockquote.wsdl
- The wsdl2h tool has been used to generate a .h file from the WSDL one: *wsdl2h stockquote.wsdl*
- The header file is then processed by the soapcpp2 tool to generate proxies and service objects: *soapcpp2 -I/usr/include/gsoap/ stockquote.h*

```
gnuton@joshua:/tmp/SOAP2$ soapcpp2 -I/usr/include/gsoap/ stockquote.h

**  The gSOAP Stub and Skeleton Compiler for C and C++ 2.7.9l
**  Copyright (C) 2000-2007, Robert van Engelen, Genivia Inc.
**  All Rights Reserved. This product is provided "as is", without any warranty.
**  The gSOAP compiler is released under one of the following three licenses:
**  GPL, the gSOAP public license, or the commercial license by Genivia Inc.

Saving soapStub.h
Saving soapH.h
Saving soapC.cpp
Saving soapClient.cpp
Saving soapClientLib.cpp
Saving soapServer.cpp
Saving soapServerLib.cpp
Using ns2 service name: StockQuoteSoap
Using ns2 service style: document
Using ns2 service encoding: literal
Using ns2 service location: http://www.webservicex.net/stockquote.asmx
Using ns2 schema namespace: http://www.webserviceX.NET/StockQuoteSoap
Saving soapStockQuoteSoapProxy.h client proxy
Saving soapStockQuoteSoapObject.h server object
Saving StockQuoteSoap.GetQuote.req.xml sample SOAP/XML request
Saving StockQuoteSoap.GetQuote.res.xml sample SOAP/XML response
Saving StockQuoteSoap.nsmap namespace mapping table
Using ns3 service name: StockQuoteSoap12
Using ns3 service style: document
Using ns3 service encoding: literal
Using ns3 service location: http://www.webservicex.net/stockquote.asmx
Using ns3 schema namespace: http://www.webserviceX.NET/StockQuoteSoap12
Saving soapStockQuoteSoap12Proxy.h client proxy
Saving soapStockQuoteSoap12Object.h server object
Saving StockQuoteSoap12.GetQuote.req.xml sample SOAP/XML request
Saving StockQuoteSoap12.GetQuote.res.xml sample SOAP/XML response
Saving StockQuoteSoap12.nsmap namespace mapping table

Compilation successful
```

- From this step, the Nokia Qt SDK can be used to develop our client
- To keep the example minimal and clean, we used the Qt console template to create this example.
- Since gSoap tools generate many files, they have been put in the soap/ dir, avoiding mixing them with the Qt files written by us;

```
gnuton@joshua:~/ARTICOLI-WIKI/2Q-2010/7.GSOAP/soap2$ ls gsoap/
soapC.cpp soapServer.cpp soapStockQuoteSoapObject.h StockQuoteSoap12.GetQuote.req.xml StockQuoteSoap.GetQuote.res.xml
soapClient.cpp soapServerLib.cpp soapStockQuoteSoapProxy.h StockQuoteSoap12.GetQuote.res.xml StockQuoteSoap.nsmap
soapClientLib.cpp soapStockQuoteSoap12Object.h soapStub.h StockQuoteSoap12.nsmap stockquote.wsdl
soapH.h soapStockQuoteSoap12Proxy.h stockquote.h StockQuoteSoap.GetQuote.req.xml
```

- The most interesting file among those generated is the proxy class in the file gsoap/soapStockQuoteSoapProxy.h which looks like this:

```
class StockQuoteSoap
{   public:
        /// Constructor allocates soap engine context, sets default endpoint URL, and sets namespace mapping table
        StockQuoteSoap();

        /// Destructor frees deserialized data and soap engine context
        virtual ~StockQuoteSoap();

        /// Invoke 'GetQuote' of service 'StockQuoteSoap' and return error code (or SOAP_OK)
        virtual int __ns2__GetQuote(_ns1__GetQuote *ns1__GetQuote, _ns1__GetQuoteResponse *ns1__GetQuoteResponse);
};
```

- you can now modify the main.cpp file as follows:

```
#include <QtCore/QCoreApplication>
#include "gsoap/soapStockQuoteSoapProxy.h"
```


(C) Copyright Nokia 2013. All rights reserved.

```
#include "gsoap/StockQuoteSoap.nsmap"

#include <QDebug>

int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);

    StockQuoteSoap soapObj;

    _ns1__GetQuote *quote = new _ns1__GetQuote;
    _ns1__GetQuoteResponse *response = new _ns1__GetQuoteResponse;

    std::string s = "NOK";
    quote->symbol = &s;
    soapObj.__ns2__GetQuote(quote, response);
    qDebug() << "RESP" << response->GetQuoteResult->c_str();
    return a.exec();
}
```

- And to make the compilation working you have to add the required files to the Qt project file:

```
QT       += core
QT       -= gui

TARGET = soap
CONFIG   += console
CONFIG   -= app_bundle

CONFIG += link_pkgconfig
PKGCONFIG += gsoap++

TEMPLATE = app


SOURCES += main.cpp \
    gsoap/soapC.cpp \
    gsoap/soapClient.cpp

HEADERS += \
    gsoap/soapStockQuoteSoapProxy.h \
    gsoap/soapH.h \
    gsoap/soapStub.h \
    gsoap/soapStockQuoteSoapObject.h

OTHER_FILES += \
    gsoap/StockQuoteSoap.nsmap
```

- AND NOW THE EXAMPLE IS READY TO BE BUILT. If the web service is up, the application will print out the Nokia stock quote.

Code

Click Media:soap.zip to download the example.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.