

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Security;
using System.Xml;
using System.IO;
using Microsoft.Win32;
using System.Security.Cryptography;

namespace DemoBookNSF
{
    public partial class Form1 : Form
    {
        String server = "flexdomino/flex2domino";
        String dbname = "flex/flexdemoacl.nsf";
        String ansicht = "BooksFlat";
        String EndPunkt = "http://www.flexdomino.net:81/soapgateq.nsf";
        String user = "";
        String pwd = "";
        String nXMLDoc;
        String[,] dataList;
        XmlDocument xmldoc = new XmlDocument();
        Boolean sessionbasiert = true;
        Boolean neu;

        public Form1()
        {
            InitializeComponent();
        }
        private bool SubKeyExist(string Subkey)
        {
            // Check if a Subkey exist
            RegistryKey myKey = Registry.CurrentUser.OpenSubKey(Subkey);
            if (myKey == null)
                return false;
            else
                return true;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void label2_Click(object sender, EventArgs e)
        {
        }

        private void button7_Click(object sender, EventArgs e)
        {
            panel4.Visible = false;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            panel4.Visible = false;
            panel6.Visible = false;
            panel6.Left = 21;
            this.Width = 741;
            button6.Enabled = false;
            label8.Visible = false;
            String sessionbasierttemp;
            // Reg Prüfen wenn neu dann nach defaults Fragen!!!
            if (SubKeyExist("SOFTWARE\\qkom\\soapDemoACL"))
            {
                RegistryKey myKey = Registry.CurrentUser.OpenSubKey("SOFTWARE\\qkom\\
soapDemoACL");
```



```

        textBox1.Text=(string)myKey.GetValue("EndPunkt");
        textBox2.Text=(string)myKey.GetValue("NotesUser"); //Notes User
        textBox3.Text=Decrypt((string)myKey.GetValue("NotesPWD"),"SOAPDEMO2010"); //
/Notes PWD
        textBox4.Text=(string)myKey.GetValue("NotesServer"); //notes Server
        textBox5.Text=(string)myKey.GetValue("DatenbankNSF"); // FlexDemoACL DB
        sessionbasierttemp = (string)myKey.GetValue("Session");
        EndPunkt = textBox1.Text;
        user = textBox2.Text;
        pwd = textBox3.Text;
        server = textBox4.Text;
        dbname = textBox5.Text;
        //checkBox1.Checked = sessionbasiert;
        if (sessionbasierttemp.Equals("True"))
            checkBox1.Checked = true;
        else
            checkBox1.Checked = false;
        GridLaden();
    }
    else
    {
        textBox1.Text = "http://www.flexdomino.net:81/soapgateq.nsf"; //EndPunkt
        textBox2.Text="online demo"; //Notes User
        textBox3.Text="demo"; //Notes PWD
        textBox4.Text = "flexdomino/flex2domino"; //notes Server
        textBox5.Text = "flex/flexdemoacl.nsf"; // FlexDemoACL DB
        checkBox1.Checked = true;
        sessionbasiert = checkBox1.Checked;
        EndPunkt = textBox1.Text;
        user = textBox2.Text;
        pwd = textBox3.Text;
        server = textBox4.Text;
        dbname = textBox5.Text;
        panel4.Visible=true;
        button7.Enabled = false;
    }
}
private void GridLaden()
{
    String[] daten;
    int Spaltenzahl;
    String[] COLSTR = { "0", "1", "2", "3", "4", "5" };
    int saetze;
    int zzi;

    Spaltenzahl = 7;
    int rowCount;
    rowCount = dataGridView1.RowCount - 1;
    if (rowCount > 0)
    {
        for (int i = 0; i < rowCount; i++)
        {
            dataGridView1.Rows.Remove(dataGridView1.CurrentRow);
        }
    }

    try
    {
        if (checkBox1.Checked==true)
        {
            MyWebService dienst = new MyWebService();
            byte[] bcred = Encoding.ASCII.GetBytes(user+":"+pwd);
            string b64cred = Convert.ToBase64String(bcred);
            dienst.Url = EndPunkt + "/DominoUtilityWebServicesCOMPLEX?wsdl";
            dienst.SetRequestHeader("Authorization", "Basic " + b64cred);
            daten = dienst.DBCOLUMNX(server, dbname, ansicht, ref COLSTR, "~", true
, false, true);
        }
        else
        {
            com.qkom.ws.DominoUtilityWebServices dienst = new com.qkom.ws.
DominoUtilityWebServices();

```

```

        dienst.Url = Endpunkt + "/DominoUtilityWebServicesCOMPLEX?wsdl";
        dienst.Credentials = new System.Net.NetworkCredential(user, pwd, "");
        daten = dienst.DBCOLUMNX(server, dbname, ansicht, ref COLSTR, "~", true
, false, true);
    }
    label8.Text = Convert.ToString(daten.Length);
    saetze = daten.Length / Spaltenzahl;
    zz = 0;
    dataList = new String[saetze, Spaltenzahl];
    nXMLDoc = "<?xml version=\"1.0\" standalone=\"yes\" ?> \r\n";
    nXMLDoc = nXMLDoc + "<books>\r\n";
    for (int ll = 0; ll < saetze; ll++)
    {
        nXMLDoc = nXMLDoc + "<zeile>\r\n";

        for (int sp = 0; sp < Spaltenzahl; sp++)
        {
            label8.Text = Convert.ToString(ll) + " " + Convert.ToString(sp);
            if (zz < daten.Length)
            {
                richTextBox1.AppendText("[ll " + Convert.ToString(ll) + "][sp " +
Convert.ToString(sp) + "] " + daten[zz] + "\r\n");
                dataList[ll, sp] = daten[zz].ToString();
            }
            zz++;
        }
        nXMLDoc = nXMLDoc + " <Genre>" + xmlencode(dataList[ll, 0].ToString()) + "
</Genre>\r\n";
        nXMLDoc = nXMLDoc + " <Year>" + xmlencode(dataList[ll, 1].ToString()) + "
</Year>\r\n";
        nXMLDoc = nXMLDoc + " <Author>" + xmlencode(dataList[ll, 2].ToString()) + "
</Author>\r\n";
        nXMLDoc = nXMLDoc + " <Title>" + xmlencode(dataList[ll, 3].ToString()) + "
</Title>\r\n";
        nXMLDoc = nXMLDoc + " <Price>" + dataList[ll, 4].ToString() + "</Price>\r\
n";
        nXMLDoc = nXMLDoc + " <Modified>" + dataList[ll, 5].ToString() + "</
Modified>\r\n";
        nXMLDoc = nXMLDoc + " <NoteID>" + dataList[ll, 6].ToString() + "</NoteID>\
r\n";
        nXMLDoc = nXMLDoc + "</zeile>\r\n";
    }
    nXMLDoc = nXMLDoc + "</books>";
}
catch (Exception e)
{
    MessageBox.Show("Fehler beim Laden der Daten\r\n" + e.Message);
}
    richTextBox1.Text = nXMLDoc;
    //WriteFile(@"C:\temp\test.xml", nXMLDoc);
    System.IO.StringReader reader = new System.IO.StringReader(nXMLDoc);
    try
    {
        xmldoc.DataSet.ReadXml(reader);
        DataSet ds = new DataSet();
        ds = xmldoc.DataSet;
        try
        {
            dataGridView1.DataSource = ds.DefaultViewManager;
            dataGridView1.DataMember = "zeile";
            dataGridView1.Refresh();
        }
        catch
        {
            label3.Text = "Keine Daten.";
        }
    }
    catch (Exception e)
    {
        WriteFile(@"C:\temp\test.xml", nXMLDoc);
        MessageBox.Show("Fehler beim Laden der Daten\r\n" + e.Message);
    }
}
}

```

```
private void button3_Click(object sender, EventArgs e)
{
    GridLaden();
}
private string xmlencode(string zeile)
{
    zeile = zeile.Replace("&", "&amp;");
    zeile = zeile.Replace("\"", "&quot;");
    zeile = zeile.Replace("'", "&apos;");
    zeile = zeile.Replace("<", "&lt;");
    zeile = zeile.Replace(">", "&gt;");
    return (zeile);
}
private void WriteFile(String Datei, String Zeilen)
{
    StreamWriter myfile = new StreamWriter(Datei);
    myfile.Write(Zeilen);
    myfile.Close();
}
private void button6_Click(object sender, EventArgs e)
{
    panel6.Visible = true;
}

private void button9_Click(object sender, EventArgs e)
{
    panel6.Visible = false;
    textBox6.Text = "";
    textBox7.Text = "";
    textBox8.Text = "";
    textBox9.Text = "";
    richTextBox1.Text = "";
    richTextBox2.Text = "";
}

private void button5_Click(object sender, EventArgs e)
{
    panel6.Visible = true;
    neu = true;
    richTextBox1.Text = "";
    textBox6.Text = "";
    textBox7.Text = "";
    textBox8.Text = "";
    textBox9.Text = "";
    textBox11.Text = "";
    textBox10.Text = "";
    richTextBox2.Text = "";
}

private void dataGridView1_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
{
    //hier werden die Details geladen
    String gewaehlt;
    panel6.Visible = true;
    panel6.Left = 12;
    gewaehlt = dataGridView1.Rows[e.RowIndex].Cells["NoteID"].Value.ToString();
    label8.Text = gewaehlt;
    booklesen(gewaehlt);
    textBox11.Text = gewaehlt;
}
private void booklesen(String noteid)
{
    String[] daten;
    String[] felder = { "Book_Title", "Book_Author", "Book_Year", "Book_Genre",
"Book_Price", "Book_Summary" };
    if (checkBox1.Checked)
    {
        MyWebService dienst = new MyWebService();
        byte[] bcred = Encoding.ASCII.GetBytes(user+": "+pwd);
        string b64cred = Convert.ToBase64String(bcred);
        dienst.SetRequestHeader("Authorization", "Basic " + b64cred);
        dienst.Url = Endpunkt + "/DominoUtilityWebServicesComplex";
    }
}
```

```

        daten = dienst.DBREADDOCFIELDS(server, dbname, "BooksFlat", noteid, ref
felder, "~");
    }
    else
    {
        com.qkom.ws.DominoUtilityWebServices dienst = new com.qkom.ws.
DominoUtilityWebServices();
        dienst.Url = EndPunkt + "/DominoUtilityWebServicesComplex";
        dienst.Credentials = new System.Net.NetworkCredential(user, pwd);
        daten = dienst.DBREADDOCFIELDS(server, dbname, "BooksFlat", noteid, ref
felder, "~");
    }

    richTextBox1.Text = daten[0].ToString();
    textBox6.Text = daten[1].ToString();
    textBox7.Text = daten[2].ToString();
    textBox8.Text = daten[3].ToString();
    textBox9.Text = daten[4].ToString();
    richTextBox2.Text = daten[5].ToString();
}

private void button10_Click(object sender, EventArgs e)
{
    //Speichern

    //TODO Komma Problem

    String[] Fields = { "Book_Title", "Book_Author", "Book_Year", "Book_Genre",
"Book_Price", "Book_Summary" };
    String[] Types = { "1280", "1280", "1280", "1280", "768", "1" };
    String[] values = new String[6];
    values[0]= richTextBox1.Text;
    values[1]= textBox6.Text;
    values[2]= textBox7.Text;
    values[3]= textBox8.Text;
    values[4]= kommatausch(textBox9.Text);
    values[5]= richTextBox2.Text;
    if (checkBox1.Checked)
    {
        MyWebService dienst = new MyWebService();
        byte[] bcred = Encoding.ASCII.GetBytes(user + ":" + pwd);
        string b64cred = Convert.ToBase64String(bcred);
        dienst.SetRequestHeader("Authorization", "Basic " + b64cred);
        dienst.Url = EndPunkt + "/DominoUtilityWebServicesComplex";
        if (neu == true)
        {
            dienst.DBREADDOCFIELDS(server, dbname, "", ref Fields, ref Types, ref
values, "~", "Book", true, "");
            neu = false;
        }
        else
        {
            dienst.DBREADDOCFIELDS(server, dbname, textBox11.Text, ref Fields, ref
Types, ref values, "~", "Book", true, "");
        }
    }
    else
    {
        com.qkom.ws.DominoUtilityWebServices dienst = new com.qkom.ws.
DominoUtilityWebServices();
        dienst.Url = EndPunkt + "/DominoUtilityWebServicesComplex";
        dienst.Credentials = new System.Net.NetworkCredential(user, pwd);
        if (neu == true)
        {
            dienst.DBREADDOCFIELDS(server, dbname, "", ref Fields, ref Types, ref
values, "~", "Book", true, "");
            neu = false;
        }
        else
        {
            dienst.DBREADDOCFIELDS(server, dbname, textBox11.Text, ref Fields, ref
Types, ref values, "~", "Book", true, "");
        }
    }
}

```

```

    }

    textBox6.Text = "";
    textBox7.Text = "";
    textBox8.Text = "";
    textBox9.Text = "";
    richTextBox1.Text = "";
    richTextBox2.Text = "";
    panel6.Visible = false;
    GridLaden();
}

private void button2_Click(object sender, EventArgs e)
{
    button7.Enabled = true;
    button8.Enabled = true;
    panel4.Visible = true;
    panel4.Left = 12;
}

private void button4_Click(object sender, EventArgs e)
{
    String gewaehlt;
    panel6.Visible = true;
    panel6.Left = 12;
    gewaehlt = dataGridView1.CurrentRow.Cells["NoteID"].Value.ToString();
    label8.Text = gewaehlt;
    booklesen(gewaehlt);
    textBox11.Text = gewaehlt;
}

private void button8_Click(object sender, EventArgs e)
{
    //in reg Speichern
    RegistryKey newKey = Registry.CurrentUser.CreateSubKey("SOFTWARE\\qkom\\
soapDemoACL");
    newKey.SetValue("EndPunkt", textBox1.Text);
    newKey.SetValue("NotesUser", textBox2.Text);
    newKey.SetValue("NotesPWD", Encrypt(textBox3.Text, "SOAPDEMO2010"));
    newKey.SetValue("NotesServer", textBox4.Text);
    newKey.SetValue("DatenbankNSF", textBox5.Text);
    newKey.SetValue("Session", checkBox1.Checked);
    panel4.Visible = false;
    EndPunkt = textBox1.Text;
    user = textBox2.Text;
    pwd = textBox3.Text;
    server = textBox4.Text;
    dbname = textBox5.Text;
    sessionbasiert = checkBox1.Checked;
    GridLaden();
}

private String kommatausch(String wert)
{
    // komma in Punkt wechseln
    wert = wert.Replace(",", ".");
    return wert;
}

public static byte[] Encrypt(byte[] clearData, byte[] Key, byte[] IV)
{
    MemoryStream ms = new MemoryStream();
    Rijndael alg = Rijndael.Create();
    alg.Key = Key;
    alg.IV = IV;
    CryptoStream cs = new CryptoStream(ms, alg.CreateEncryptor(), CryptoStreamMode.
Write);
    cs.Write(clearData, 0, clearData.Length);
    cs.Close();
    byte[] encryptedData = ms.ToArray();
    return encryptedData;
}

public static string Encrypt(string clearText, string Password)

```

```

    {
        byte[] clearBytes = System.Text.Encoding.Unicode.GetBytes(clearText);
        PasswordDeriveBytes pdb = new PasswordDeriveBytes(Password, new byte[] { 0x49, 0x76,
0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
        byte[] encryptedData = Encrypt(clearBytes, pdb.GetBytes(32), pdb.GetBytes(16));
        return Convert.ToBase64String(encryptedData);
    }
    public static byte[] Decrypt(byte[] cipherData, byte[] Key, byte[] IV)
    {
        MemoryStream ms = new MemoryStream();
        Rijndael alg = Rijndael.Create();
        alg.Key = Key;
        alg.IV = IV;
        CryptoStream cs = new CryptoStream(ms, alg.CreateDecryptor(), CryptoStreamMode.
Write);
        cs.Write(cipherData, 0, cipherData.Length);
        cs.Close();
        byte[] decryptedData = ms.ToArray();
        return decryptedData;
    }
    public static string Decrypt(string cipherText, string Password)
    {
        byte[] cipherBytes = Convert.FromBase64String(cipherText);
        PasswordDeriveBytes pdb = new PasswordDeriveBytes(Password, new byte[] { 0x49, 0x76,
0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
        byte[] decryptedData = Decrypt(cipherBytes, pdb.GetBytes(32), pdb.GetBytes(16));
        ;
        return System.Text.Encoding.Unicode.GetString(decryptedData);
    }
}
public class MyWebService : com.qkom.ws.DominoUtilityWebServices
{
    private String m_HeaderName;
    private String m_HeaderValue;

    //-----
    // GetWebRequest
    //
    // Called by the SOAP client class library
    //-----
    protected override WebRequest GetWebRequest(Uri uri)
    {
        // call the base class to get the underlying WebRequest object
        HttpWebRequest req = (HttpWebRequest)base.GetWebRequest(uri);

        if (null != this.m_HeaderName)
        {
            // set the header
            req.Headers.Add(this.m_HeaderName, this.m_HeaderValue);
        }

        // return the WebRequest to the caller
        return (WebRequest)req;
    }

    //-----
    // SetRequestHeader
    //
    // Sets the header name and value that GetWebRequest will add to the
    // underlying request used by the SOAP client when making the
    // we method call
    //-----
    public void SetRequestHeader(String headerName, String headerValue)
    {
        this.m_HeaderName = headerName;
        this.m_HeaderValue = headerValue;
    }
}
}

```