```csharp
using System.IO;
using System.Net;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using System;
using System.Text;


class iSMSAPI
{
    /// <summary>
    /// Used to send message
    /// </summary>
    /// <param name="from">The sender ID. Max is 11 characters</param>
    /// <param name="to">The recipient(s). Use (,) to separate multiple recipients</param>
    /// <param name="message">The content of message to send </param>
    /// <param name="username">Your iSMS username</param>
    /// <param name="password">Your iSMS Password</param>
    /// <returns>Status of message submission to the SMS gateway</returns>
    public bool sendMessage(string from, string to, string message, string username, string
    password)
    {

        bool sucess = false;

        //use comma (,) to separate multiple recipients. remember that the numbers has to be in
        international format
        //example to = 23327612806,2335487899,233244587896,233266000751,233277014525
        WebRequest request;
        HttpWebResponse response;
        string responseMess = "";
        string apiparams = "username={0}&password={1}&from={2}&to={3}&message={4}";
        string URL = "http://isms.wigalsolutions.com/ismsweb/sendmsg/";
        apiparams = string.Format(apiparams, username, password, from, to, message);
        try
        {

            request = HttpWebRequest.Create(URL);
            request.Method = "POST";
            request.ContentType = "application/x-www-form-urlencoded";
            var data = Encoding.ASCII.GetBytes(apiparams);

            request.ContentLength = data.Length;
            Stream dataStream = request.GetRequestStream();
            dataStream.Write(data, 0, data.Length);
            dataStream.Close();
            request.Timeout = 5000;

            System.Net.ServicePointManager.ServerCertificateValidationCallback =
                delegate(object Certsender, X509Certificate certificate, X509Chain chain,
                SslPolicyErrors sslPolicyErrors) { return true; };

            response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream());
```

```csharp
            responseMess = reader.ReadToEnd();
            response.Close();
            if (responseMess.StartsWith("SUCCESS"))
                sucess = true;
            else
                sucess = false;

        }
        catch (Exception ex) { sucess = false; }



        return sucess;
    }


    public string getBalance(string username, string password)
    {
        WebRequest request;
        HttpWebResponse response;
        string responseMess = "";
        string apiparams = "username={0}&password={1}&query={2}";
        string URL = "http://isms.wigalsolutions.com/ismsweb/Query/";

        apiparams = string.Format(apiparams, username, password, "balance");

        request = HttpWebRequest.Create(URL + "?" + apiparams);
        request.Timeout = 10000;
        try
        {
            System.Net.ServicePointManager.ServerCertificateValidationCallback =
                delegate(object Certsender, X509Certificate certificate, X509Chain chain,
                SslPolicyErrors sslPolicyErrors) { return true; };

            response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream());
            responseMess = reader.ReadToEnd();
            response.Close();



        }
        catch (Exception ex) { responseMess = ""; }
        finally
        {

        }
        return responseMess;
    }
}
```