

```
using System;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Xml;
using System.Xml.Linq;
using System.Xml.Serialization;
using System.IO;
using System.Collections;

namespace MessageLogger
{
    /// <summary>
    /// The main webservice class. This handles all login and message posting
    /// requests to the web server.
    /// </summary>
    [WebService(Namespace = "http://virus-box/MLogger",
        Description="Main MLogger Web Service")]
    public class MService : System.Web.Services.WebService
    {
        /// <summary>
        /// Default constructor, does nothing.
        /// </summary>
        public MService()
        {
        }

        /// <summary>
        /// Validates a user's login and returns a new session ID for that user
        /// </summary>
        /// <param name="userName">The user's login name</param>
        /// <param name="password">The user's password</param>
        /// <returns>0 if invalid</returns>
        [WebMethod(Description="Message Logger User Login")]
        public long UserLogin(String userName, String password)
        {
            // Create new database object
            spExecution sp = new spExecution();
            sp.SQLConnect();

            // Check if the user name & password are valid
            if (sp.ValidateUser(userName, password) == "1")
            {
                sp.SQLDisconnect(); // Disconnect from SQL
                return NewSessionID(userName);
            }
            else
            {
                sp.SQLDisconnect(); // Disconnect from SQL
                return 0L;
            }
        }

        /// <summary>
        /// Posts a new message to the database
        /// </summary>
        /// <param name="sessionID">The user's current session ID</param>
        /// <param name="title">The title of the message</param>
        /// <param name="message">The message text</param>
        /// <returns>The message ID for the new message</returns>
        private int PostNewMessage(long sessionID, String title, String message)
        {
            int messageID;

            // Create new database object
            spExecution sp = new spExecution();
            sp.SQLConnect();
        }
    }
}
```

```

        // Post new message
        messageID = sp.PostNewMessage(sessionID, title, message);

        sp.SQLDisconnect(); // disconnect from the database

        return messageID;
    }

    /// <summary>
    /// Adds a media attachment to a message post
    /// </summary>
    /// <param name="sessionID">User's current session ID</param>
    /// <param name="messageID">The message ID to which the media will be attached</
param>
    /// <param name="dataType">The type of media being attached</param>
    /// <param name="data">The base64 media data to be written to the drive</param>
    private void PostMsgAttachment(long sessionID, int messageID, String dataType, String
data)
    {
        String fileName = "";

        // Create new database object
        spExecution sp = new spExecution();
        sp.SQLConnect();

        // Set up the base directory for where the data will be stored
        String path = @"C:\Inetpub\MessageLogger\message-media\" + sp.GetUserFromSession
(sessionID) + @"\" + messageID + @"\";
        Directory.CreateDirectory(path);

        // Setup the file name
        System.Globalization.CultureInfo ci = System.Globalization.CultureInfo.
InstalledUITCulture;
        fileName += System.DateTime.Now.ToString("yyyy_mm_dd-hh_mm_ss_fff", ci);

        // Disconnect and reconnect to clear the SQL command
        sp.SQLDisconnect();
        sp.SQLConnect();

        // Post the data to the database
        sp.PostMsgAttachment(sessionID, messageID, dataType, path + fileName);
        sp.SQLDisconnect();

        // Convert the base64 string to a byte array
        byte[] dataBytes = Convert.FromBase64String(data);

        // Create the appropriate filename
        if (dataType == "jpeg-image")
        {
            fileName += ".jpg";
        }
        if (dataType == "png-image")
        {
            fileName += ".jpg";
        }
        else if (dataType == "gif-image")
        {
            fileName += ".gif";
        }
        else if (dataType == "mpeg-video")
        {
            fileName += ".mpg";
        }
        else if (dataType == "wmv-video")
        {
            fileName += ".wmv";
        }
        else if (dataType == "wav-sound")
        {

```

```

        fileName += ".wav";
    }

    // Write the file to the drive
    FileStream fs = new FileStream(path + fileName, FileMode.Create, FileAccess.
Write);
    BinaryWriter writer = new BinaryWriter(fs);

    try
    {
        for (int i = 0; i < dataBytes.Length; i++)
        {
            writer.Write(dataBytes[i]);
        }
    }
    finally
    {
        writer.Close();
        fs.Close();
    }
}

/// <summary>
/// Parses input XML and creates a new message and message attachment post.
/// </summary>
/// <param name="input">An XML SOAP message</param>
/// <returns>0 for invalid session ID, 1 for success, -1 for unspecified error</
returns>
[WebMethod(Description="Posts a message to a user's account")]
public int PostMessage([XmlAnyElement]XmlElement input)
{
    long sessionID = 0;
    int messageID = 0;
    String title = "";
    String message = "";
    String dataType = "";
    String data = "";

    IEnumerator ienum = input.GetEnumerator();
    XmlNode currentNode;

    ienum.Reset();

    while (ienum.MoveNext())
    {
        currentNode = (XmlNode)ienum.Current;

        if (currentNode.Name == "sessionID")
        {
            long.TryParse(currentNode.InnerText, out sessionID);

            // Return -1 if the session ID is invalid
            if (!IsValidSessionID(sessionID))
            {
                return 0;
            }
        }
        else if (currentNode.Name == "title")
        {
            title = currentNode.InnerText;
        }
        else if (currentNode.Name == "message") // Here we post the new message
        {
            message = currentNode.InnerText;
            messageID = PostNewMessage(sessionID, title, message);
        }
        else if (currentNode.Name == "dataType")
        {
            dataType = currentNode.InnerText;

```

```

    }
    else if (currentNode.Name == "data") // Here we post the new message
    {
        data = currentNode.InnerText;
        PostMsgAttachment(sessionID, messageID, dataType, data);
    }
}

return 1;
}

/// <summary>
/// Returns a new session ID for the user
/// </summary>
/// <param name="userName">The user login in</param>
/// <returns>The user's new session ID</returns>
private long NewSessionID(String userName)
{
    long sessionID;

    // Create new database object
    spExecution sp = new spExecution();
    sp.SQLConnect();

    // Create new session ID
    sessionID = sp.NewSessionID(userName);

    sp.SQLDisconnect(); // Disconnect from SQL

    return sessionID;
}

/// <summary>
/// Validates a current session ID
/// </summary>
/// <param name="sessionID">The user's session ID</param>
/// <returns>TRUE if valid, FALSE if not</returns>
private bool isValidSessionID(long sessionID)
{
    // Create new database connection object
    spExecution sp = new spExecution();
    sp.SQLConnect();

    if (sp.ValidateSessionID(sessionID))
    {
        sp.SQLDisconnect(); // Disconnect from SQL
        return true;
    }
    else
    {
        sp.SQLDisconnect(); // Disconnect from SQL
        return false;
    }
}
}
}

```