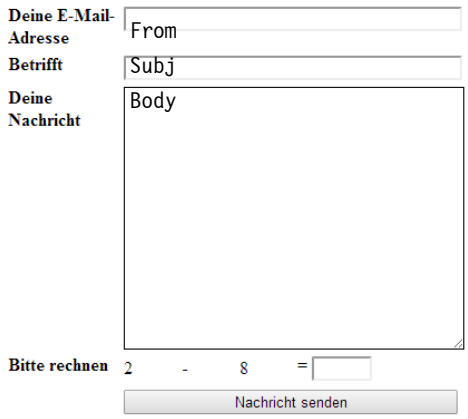
# ASPX-Sendmail

*Franz Fiala*

Es kommt immer wieder vor, dass man im Rahmen einer Homepage statt eine E-Mail-Adresse anzugeben, ein Formular absenden möchte, denn das kann ein Besucher der Seite auch in einem Internet-Cafe. Das folgende C#-Programm erledigt diese Aufgabe.

Das Programm definiert die Felder **From, Subj** und **Body.** Als Spammerschutz muss eine Rechenaufgabe gelöst werden.




Das Programm besteht aus der einzigen Datei **mail.aspx** und muss vor der ersten Benutzung angepasst werden. Man benötigt für den Versand eine Mailbox und kann dafür auch eine neue Mailbox anlegen, die ausschließlich für den Empfang dieser Kontaktdaten reserviert ist.

Unbedingt erforderlich sind folgende Variablen:

```
string Mailserver = "";
string strTo = "";
string Username = "";
string Passwort = "";
```

Der Mailserver ist bei Verwendung am Club-Mail-Server `mailenable.ccc` die eigene Domäne mit vorangestelltem `mail`. Auch eine Clubadresse `name@clubcomputer.at` kann zur Absendung verwendete warden, dann ist der Mailserver `mail.clubcomputer.at`.

`strTo` ist die Mailadresse an die das Formular die Daten abliefern soll. Der Username ist im Falle von Mailenable die vollständige E-Mail-Adresse der verwendeten Mailbox. Passwort nicht zu einfach wählen. Die anderen Variablen sind optional.

## Zugriffsschutz

Damit die Datei **mail.aspx** nicht von fremden Websites aufgerufen warden kann, sind grundsätzlich nur lokale Zugriffe (vom selben Rechner) erlaubt. Damit aber für Testzwecke ein externer Rechner Zugriffsrechte bekommt, können über `strIPsAllowed` eine oder mehrere IP-Adressen definiert warden, von denen zugegriffen werden darf.

## Post- und Get-Zugriff

Normalerweise erfolgt die Eingabe der Kontaktdaten über das Formular (Post-Daten). Man kann das Programm aber auch durch Parameterübergabe über die Kommandozeile verweden und damit für den Benutzer unsichtbar eine Mail versenden. Zum Beispiel beim Besuch einer Seite oder beim Anklicken eines Links. Man gibt dann als Link ein:

```
http://domain/mail.aspx?
from=name@domain
&subj=betrifft-text
&body=nachrichten-text
```

## mail.aspx

```
<%@ Page Language="C#" Debug="true" %>
<%@ Import Namespace="System.Net.Mail" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title>Senden einer Nachricht</title>
    <script runat="server">
        string Mailserver = "";
        string strTo = "";
        string Username = "";
        string Passwort = "";
        string[] strIPsAllowed = new string[] { "86.59.41.251" };
        string strTextOK = "Danke für die Mitteilung!";
        string strTextForbidden = "Verbotener Zugriff";
        string strTextError = "Fehler beim Versand";
        string strTextCalc = "Bitte richtig rechnen";
        string strTextInput = "Alle Felder ausfüllen";
        string strTextMailError = ": Falsches Mailformat";
        string redirectTo = "";
        bool DEBUG = false;
        string strCc = "";
        string strBcc = "";
        string strFrom = "";
        string strSubject = "";
        string strBody = "";
        bool CheckOk = false;
        void Page_Load(object sender, EventArgs e)
        {
            Panel_Message.Visible = false;
            Panel_MailDialog.Visible = false;
            if ((Mailserver.Trim() == "") || (Username.Trim() == "") || (Passwort.Trim() == "") || (strTo.Trim() == ""))
            {
                WriteLn("Fehlerhafte Konfiguration"); return;
            }
            bool RequestLocal = Request.IsLocal; if (DEBUG) WriteLn("RequestLocal", RequestLocal.ToString());
            bool IPAllowed = IsIPAllowed(); if (DEBUG) WriteLn("IPAllowed", IPAllowed.ToString());
            if (!(Request.IsLocal || IsIPAllowed()))
            {
                Panel_Message.Visible = true; Label_Message.Text = strTextForbidden; return;
            }
            if (!IsPostBack)
            {
                Panel_MailDialog.Visible = true;
                string[] op = new string[] { "+", "-", "*" }; Random rnd = new Random();
                Label_x.Text = (rnd.Next(10) + 1).ToString();
                Label_y.Text = (rnd.Next(10) + 1).ToString();
                Label_op.Text = op[rnd.Next(3)];
                return;
            }
            int x = 0; int.TryParse(Label_x.Text, out x);
            int y = 0; int.TryParse(Label_y.Text, out y);
            int z = 0; int.TryParse(TextBox_z.Text, out z);
            if (DEBUG) WriteLn(x + Label_op.Text + y + " = " + z);
            switch (Label_op.Text)
            {
                case "+":
                    if (x + y == z) CheckOk = true;
                    break;
                case "-":
                    if (x - y == z) CheckOk = true;
                    break;
                case "*":
                    if (x * y == z) CheckOk = true;
                    break;
            }
            if (CheckOk)
            {
                strFrom = GetVar("from"); strSubject = GetVar("subj"); strBody = GetVar("body");
                if ((strFrom == "") || (strSubject == "") || (strBody == ""))
                {
                    Panel_MailDialog.Visible = true;
                    Label_SendError.Text = strTextInput;
                }
                else
                {
                    Label_SendError.Text = SendMailMessage(strFrom, strTo, strBcc, strCc, strSubject, strBody);
                    if (Label_SendError.Text == "")
                    {
                        if (redirectTo == "")
                        {
                            Panel_Message.Visible = true;
                            Label_Message.Text = strTextOK;
                        }
                        else Response.Redirect(redirectTo);
                    }
                    else Panel_MailDialog.Visible = true;
                }
            }
            else
            {
                Panel_MailDialog.Visible = true;
                Label_SendError.Text = strTextCalc;
            }
        }
        bool IsIPAllowed()
        {
            string IP = Request.UserHostAddress;
            for (int i = 0; i < strIPsAllowed.Length; i++) if (strIPsAllowed[i] == IP) return true;
            return false;
        }
        void WriteLn(string text) { Write(text + "<br/>"); }
        void Write(string text) { Response.Write(text); Response.Flush(); }
        void Write(string name, string value) { Write(name + ": " + value); }
        void WriteLn(string name, string value) { WriteLn(name + ": " + value); }
        string GetVar(string param)
        {
            string value = Request.Params[param];
            if (string.IsNullOrEmpty(value)) value = "";
            if (DEBUG) WriteLn(param, value);
            return value;
        }
        void Button_Send_Click(object sender, EventArgs e) { }
        public string SendMailMessage(string from, string to, string bcc, string cc, string subject, string body)
        {
            try
            {
                MailMessage mMailMessage = new MailMessage();
                try { mMailMessage.To.Add(new MailAddress(to)); } catch { return "To" + strTextMailError; };
                try { mMailMessage.From = new MailAddress(from); } catch { return "From" + strTextMailError; };
                if (!string.IsNullOrEmpty(bcc))
                    try { mMailMessage.Bcc.Add(new MailAddress(bcc)); } catch { return "BCC" + strTextMailError; }
                if (!string.IsNullOrEmpty(cc))
                    try { mMailMessage.CC.Add(new MailAddress(cc)); } catch { return "CC" + strTextMailError; }
                mMailMessage.Subject = subject;
                mMailMessage.Body = body;
                mMailMessage.IsBodyHtml = false;
                mMailMessage.Priority = MailPriority.Normal;
                SmtpClient mSmtpClient = new SmtpClient(Mailserver);
                mSmtpClient.Credentials = new System.Net.NetworkCredential(Username, Passwort);
                mSmtpClient.Send(mMailMessage);
                return "";
            }
            catch (Exception ex)
            {
                Exception ex2 = ex;
                string errorMessage = string.Empty;
                // while (ex2 != null)
                // {
                errorMessage += ex2.ToString();
                //  ex2 = ex2.InnerException;
                // }
                return errorMessage;
            }
        }
    </script>
</head>
```

## Konfektionierung

Eine Homegpage hat üblicherweise ein einheitliches Aussehen. Das Programm **mail.aspx** ist aber ganz roh und ohne Layout-Elemente und enthält auch keinen sonstigen Inhalt wie er üblicherweise auf einer Webseite verwendet wird.

Man entwirft also zunächst die Seite, die späterdas Kontaktformular enthalten soll, zum Beispiel **kontakt.html** mit dem eigenen Html-Designer, zum Beispiel Expression Web und lässt den Bereich des Formulars zunächst frei.

### Integration als iFrame

Die einfachste Integration erfolgt über ein `iframe`. Dazu genügt es, im Dokument **kontakt.html** an der gewünschten Stelle einzufügen:

```
<iframe width="450" height ="500" src="mail.aspx"
scrolling="no" frameborder="0"></iframe>
```

Damit die Formatierung der Formular-Felder korrekt erfolgt, ist es auch zusätzlich nötig, in der Datei **mail.aspx** im head-Abschnitt auf den Style-Sheet der Seite **kontakt.html** zu verweisen, also zum Beispiel

```
<link rel="stylesheet" type="text/css"
href="mystyle.css" media="screen">
```

### In bestehende Webseite integrieren

Wer kein iFrame verwenden will, kann auch die Datei **kontakt.html** editieren, und zwar so:

Man benennt die Seite in **kontakt.aspx** um und fügt folgende Kopfzeilen ein

```
<%@ Page Language="C#" Debug="true" %>
<%@ Import Namespace="System.Net.Mail" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
```

Innerhalb des Header-Tag fügt man den folgenden Abschnitt aus **mail.aspx** ein.

```
<script runat="server">
...
</script>
```

Unmittelbar nach dem body-Tag fügt man ein

```
<form id="form1" runat="server">
```

Und vor dem /body-Tag

```
</form>
```

Den Rest zwischen

```
<asp:Panel ID="Panel_Message" runat="server">
   <asp:Label ID="Label_Message" runat="server"></asp:Label>
   </asp:Panel>
<asp:Panel ID="Panel_MailDialog" runat="server">
…
</asp:Panel>
```

fügt man an der für das Formular vorgesehenen Stelle ein.

Dann muss man noch den Link/die Links, die auf **kontakt.html** gezeigt haben, auf **kontakt.aspx** ändern.

## Zum Programm

Eine ASPX-Datei enthält im HTML-Teil neben den üblichen HTML-Tags (in **mail.aspx** ist es eine Tabelle, siehe Kode rechts oben) auch so genannte Webserver-Steuerelemente, die sehr ähnlich wie Html-Elemente aufgebaut sind aber Objekte mit vielen Eigenschaften und Methoden sind, über eine Id und über das spezielle Tag `runat="server"` durch ein Programm im Kopfteil angesprochen werden können (hier TextBox, Label und Button).

Das Programm im Kopfteil (vorige Seite) ist kein Skript sondern ein echtes C#-Programm. Dieses Programm wird beim ersten Aufruf am Server kompiliert und für den Anwender unsichtbar im Betriebssystem abgelegt. Beim Aufruf der Datei wird mit großem Geschwindigkeitsvorteil nur mehr das fertige Binärprogramm gerufen. Eine automatische serverseitige Neukompilierung findet immer dann statt, wenn die Datei in einer neuen Version auf den Server geladen wird.

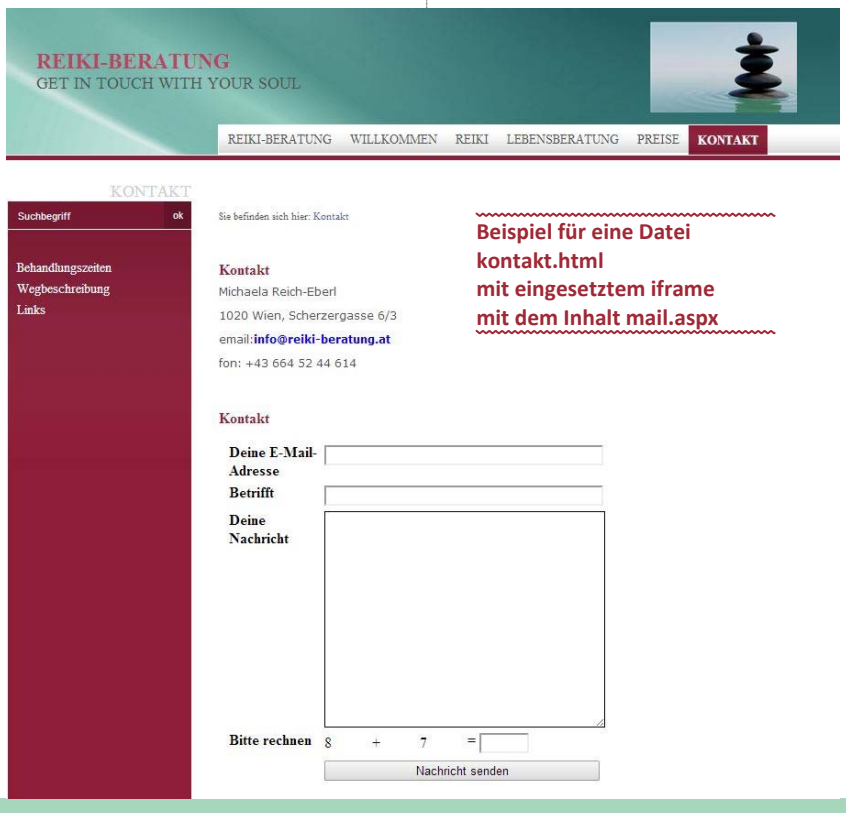
```
<body>
    <form id="form1" runat="server">
        <asp:Panel ID="Panel_Message" runat="server">
            <asp:Label ID="Label_Message" runat="server"></asp:Label>
        </asp:Panel>
        <asp:Panel ID="Panel_MailDialog" runat="server">
            <asp:Label ID="Label_SendError" runat="server"></asp:Label>
            <table>
                <tr>
                    <td valign="top" width="100">
                        <b>Deine E-Mail-Adresse</b>
                    </td>
                    <td valign="top">
                        <asp:TextBox ID="from" Width="300px" runat="server"></asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td valign="top" width="100">
                        <b>Betrifft</b>
                    </td>
                    <td valign="top">
                        <asp:TextBox ID="subj" runat="server" Width="300px"></asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td valign="top" width="100">
                        <b>Deine Nachricht</b>
                    </td>
                    <td valign="top">
                        <asp:TextBox ID="body" runat="server" Height="230px" TextMode="MultiLine"
                            Width="300px"></asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td valign="top" width="100">
                        <b>Bitte rechnen</b>
                    </td>
                    <td valign="top">
                        <asp:Label ID="Label_x" runat="server" Width="48px"></asp:Label>
                        <asp:Label ID="Label_op" runat="server" Width="48px"></asp:Label>
                        <asp:Label ID="Label_y" runat="server" Width="48px"></asp:Label>
                        =
                    <asp:TextBox ID="TextBox_z" runat="server" Width="50px"></asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td valign="top" width="100">
                        <b></b>
                    </td>
                    <td valign="top">
                        <asp:Button ID="Button_Send" Width="300px" runat="server" OnClick="Button_Send_Click"
                            Text="Nachricht senden" />
                    </td>
                </tr>
            </table>
        </asp:Panel>
    </form>
</body>
</html>
```

Das Programm prüft zunächst, ob die Konfigurations-Strings angegeben worden sind und danach wird getestet, ob der Aufruf vom lokalen Server erfolgt oder von einer erlaubten IP-Adresse. Wenn es ein Erstaufruf ist (kein PostBack, also ein nochmaliger Aufruf derselben Seite durch eine Benutzerinteraktion auf der Seite), wird die Rechenaufgabe initialisiert. Wenn es aber ein PostBack ist, wird die Lösung der Rechenaufgabe geprüft. Wenn dieser Test erfolgreich war (CheckOK), werden die Inhalte aus den Text-Boxen gelesen (`From`, `Subj`, `Body`) und nur wenn sie ausgefüllt wurden, wird die Sendung veranlasst. Das Sendeprogramm prüft das Format der beteiligten E-Mail-Adressen.

Die Darstellung besteht aus den zwei Panels (werden als `div` gerendert), die bei Programmstart zunächst als unsichtbar geschaltet werden und von denen jeweils eines, je nach Programmverlauf, sichtbar gemacht werden. Das `Panel_MailDialog` enthält das Formular, das Rechenbeispiel und den Sendebutton. Das Panel `Panel_Message` enthält nur den Text, der angezeigt wird, wenn die Mail erfolgreich gesendet wurde. In diesem Panel könnten auch mehr Informationen verpackt werden, etwa eine Zusammenfassung der soeben versendeten Mail.

Das Bild unten zeigt das Formular auf der Webseite eines Clubmitglieds.

Der Kode kann auch bei der Webversion dieser Ausgabe downgeloadet werden.




**Beispiel für eine Datei kontakt.html mit eingesetztem iframe mit dem Inhalt mail.aspx**