

Mehrere Domänen an einem Webspaces

Franz Fiala

Sehr viele Mitglieder von ClubComputer betreiben den kostenlosen Webspaces mit einer kostenlosen Clubdomäne. Damit eine Domäne im Domäne auch Content zeigen kann, benötigt sie ein „Paket“, welches die Ressourcen zur Verfügung stellt, die man für die Publikation von Content benötigt. Bei der Gratis-Domäne ist ein 250MB-Gratis-Paket dabei.

Der Normalfall ist:

1 Paket = 1 Domäne = 1 Inhalt

Der Domänenname kann entweder ein kostenloser Name vom Club sein, wie zum Beispiel `name.clubcomputer.at` oder ein reservierter Name wie zum Beispiel `name.at` oder `name.com`.

n Pakete = n Domänen = n Inhalte

Wer mehrere Domänen betreibt, benötigt normalerweise mehrere Pakete. Jeder Domäne ist ein Paket zugeordnet. Um von einem dieser Webs auf Inhalte des anderen Webs zugreifen zu können, benötigt man für die Links absolute Adressen. Ein direkter Zugriff auf die anderen Ressourcen mit einem relativen Pfad ist nicht möglich (obwohl die Dateien am Server eng benachbart sind).

1 Paket = n Domänen = 1 Inhalt

Oft werden gleichzeitig mehrere Namen reserviert, die denselben Inhalt publizieren. Das kann zum Beispiel sein, wenn man sich den Namen für eine zukünftige Erweiterung reserviert oder ein Produkt unter verschiedenen Namen gesucht werden kann, wie zum Beispiel `hp.com` und `hewlett-packard.com`.

Egal, welche Schreibweise man in der Browserzeile eingibt, man sieht immer dieselben Dateien. Man nennt das auch einen Alias-Namen.

1 Paket = n Domänen = n Inhalte

Wenn sich aber der Content für diese Domänen nur wenig unterscheidet, und man zum Beispiel für jede dieser Domänen eine abweichende Startseite zeigen will, benötigt man ein Verfahren, mit dem man—je nach Inhalt der Adresszeile im Browser—eine verschiedene Startseite anspricht.

Man kann daher auch ein einziges Paket betreiben und dennoch mehrere verschiedene Inhalte adressieren.

Man benötigt dazu ein Skript, das die Aufgabe übernimmt, je nach Aufruf, einen anderen Inhalt zu zeigen. Jede der verfügbaren Serversprachen ist dafür geeignet (PHP, ASP.NET). ASP.NET hat aber den Vorteil, dass die Skripts kompiliert werden und daher schnell sehr ablaufen.

default.aspx

Man definiert eine Datei `default.aspx` und lässt den Programmteil dieser Datei erkennen, welche Domäne in der Adresszeile angegeben wurde. In Abhängigkeit davon lässt man das Skript verschiedene Inhalte anzeigen. Die Datei `default.aspx` selbst enthält gar keinen Inhalt.

In der Datei `default.aspx` werden Script-Elemente und Html-Elemente vermischt. Html-Elemente werden angezeigt, Script-Elemente werden ausgeführt.

Beim ersten Aufruf der Seite wird dieses Script kompiliert und bei folgenden Aufrufen wird nur mehr der kompilierte Code ausgeführt. Der erste Aufruf ist daher etwas langsamer, die folgenden Aufrufe sind schnell.

Bei Fehlern im Programm werden diese Fehler angezeigt aber nur, wenn das Web so konfiguriert wird. Dazu muss man die Datei `web.config` modifizieren.

Im Kodebeispiel unten werden unter jeweils in `DomainAliasStartDocuments` ein Domänenname und das dazugehörige Startdokument in einem `string`-Array definiert. Nach Ausführung des Skripts verzweigt das Programm mit `Response.Redirect` zu den verschiedenen Seiten.

web.config

In jedem Web unserer Clubserver wird bei manchen Einstellungen (zum Beispiel bei Änderungen des Startdokuments) die Datei `web.config` (nicht löschen!) angelegt. Im Beispiel unten sieht man die erlaubten Startdokumente.

Man sieht aber auch weitere Einträge. Zum Beispiel wird mit

```
directoryBrowse enabled="false"
```

verhindert, dass ein Besucher Verzeichnisse durchsuchen kann. Die Zeile

```
customErrors mode="Off"
```

bewirkt, dass bei Auftreten von Fehlern genaue Angaben zu dem Fehler ausgegeben werden.

Weitergehendes Skript (umseitig)

In dem weitergehenden Skript (nächste Seite) kann man über die Variable `mode` zwischen drei Betriebsarten wählen:

REDIRECT: funktioniert wie das Skript unten.

Für die beiden neuen Betriebsarten gibt es in `DomainAliasStartDocuments` einen weiteren Parameter, der den Titel der Seite festlegt.

FRAME: die aufgerufene Seite wird in einem Frame eingebettet. Vorteil: die Adresszeile verändert sich nicht, solange das Frame nicht verlassen wird.

INCLUDE: der Inhalt der aufgerufenen Seite (alles zwischen `<body>` und `</body>`) wird in der aufrufenden Seite `default.aspx` eingebettet.

default.aspx = alias.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" %>
<!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 runat="server">
<title>Alias-Redirect</title>
<script runat="server" language="C#">
string ServerName;
string DefaultDomain = "paron.at";
string[,] DomainAliasStartDocuments = new string[,]
{
new string[] { "paron.at", "aindex.htm" },
new string[] { "rondo-parfums.at", "parfum" },
new string[] { "parfomage.at", "parfomage" },
new string[] { "convention-cosmetics.at", "convention" }
};
void Page_Load()
{
ServerName = Request.ServerVariables["SERVER_NAME"];
ServerName = ServerName.ToLower();
for (int i = 0; i < DomainAliasStartDocuments.Length; i++)
{
if (ServerName.Contains(DomainAliasStartDocuments[i][0]))
Response.Redirect(
"http://"
+ ServerName + "/"
+ DomainAliasStartDocuments[i][1]);
}
Response.Redirect("http://" + DefaultDomain);
}
</script>
</head>
<body></body>
</html>
```

web.config

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
<system.web>
<customErrors mode="Off" />
</system.web>
<system.webServer>
<directoryBrowse enabled="false" />
<defaultDocument>
<files>
<clear />
<add value="default.aspx" />
<add value="index.htm" />
<add value="default.htm" />
<add value="default.asp" />
</files>
</defaultDocument>
```

Die Datei `web.config` wird vom System automatisch angelegt, wenn die Grundeinstellungen verändert werden. Zum Beispiel dann, wenn die Startdokumente verändert worden sind. Sie enthält Einstellungen für den Webserver. Ergänzt werden die Einstellungen durch die Zeile

```
<customErrors mode="Off" />
```

Diese Zeile bewirkt, dass eventuelle Syntax-Fehler im Programm mit einer konkreten Fehlermeldung ausgegeben werden. Fehlt diese Zeile, dann sieht man einen allgemeinen Hinweis für den Enduser, so etwa wie „Entschuldigen Sie bitte die Störung.“

aliasInclude.aspx

```

<%@ Page Language="C#" AutoEventWireup="True" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
<title><asp:Literal ID="Literal Title" runat="server"></asp:Literal></title>
<script runat="server" language="C#">
enum MODE { REDIRECT, FRAME, INCLUDE };
MODE mode = MODE.INCLUDE;
const bool USEFRAME = true;
string ServerName;
string DefaultDomain = "paron.at";
string[] [] DomainAliasStartDocuments = new string[] []
{
    new string[] { "localhost", "", "Das ist nur ein Versuch" },
    new string[] { "paron.at", "aindex.htm", "Homepage von Parfumerie Rondo" },
    new string[] { "rondo-parfums.at", "parfum", "Homepage von Rondo-Parfums" },
    new string[] { "parfomage.at", "parfomage", "Homepage von Parfomage" },
    new string[] { "convention-cosmetics.at", "convention", "Homepage von CC" }
};
void Page_Load()
{
    ServerName = Request.ServerVariables["SERVER_NAME"];
    ServerName = ServerName.ToLower();
    for (int i = 0; i < DomainAliasStartDocuments.Length; i++)
    {
        if (ServerName.Contains(DomainAliasStartDocuments[i][0]))
        {
            string Url = "http://" + ServerName
                + "/" + DomainAliasStartDocuments[i][1];
            switch (mode)
            {
                case MODE.FRAME:
                    Literal Title.Text = DomainAliasStartDocuments[i][2];
                    Literal Content.Text =
                        "<frameset>\r\n"
                        + "<frame src='" + Url + "'>\r\n"
                        + "</frameset>\r\n"
                        + "<noframes>\r\n"
                        + ReadFile(Url)
                        + "\r\n</noframes>\r\n";
                    break;
                case MODE.INCLUDE:
                    Literal Title.Text = DomainAliasStartDocuments[i][2];
                    Literal Content.Text =
                        "<body>\r\n" + ReadFile(Url) + "\r\n</body>\r\n";
                    break;
                default:
                case MODE.REDIRECT:
                    Response.Redirect(Url);
                    break;
            }
        }
    }
}
string ReadFile(string Path)
{
    WebRequest wReq = WebRequest.Create(Path);
    WebResponse wResp;
    try
    {
        wResp = wReq.GetResponse();
    }
    catch
    {
        return "Datei " + Path + " nicht gefunden";
    }
    Stream respStream = wResp.GetResponseStream();
    StreamReader reader = new StreamReader(respStream, Encoding.Default);
    String respHTML = reader.ReadToEnd();
    respStream.Close();
    if (respHTML.Contains("<body>"))
    {
        respHTML = respHTML.Substring(respHTML.IndexOf("<body>") + 6);
    }
    if (respHTML.Contains("</body>"))
    {
        respHTML = respHTML.Substring(0, respHTML.IndexOf("</body>") - 1);
    }
    return respHTML;
}
</script>
</head>
<asp:literal id="Literal_Content" runat="server"></asp:literal>
</html>

```

Amateurfunk, Internet und moderne Software

Rainer Kirsch

Für viele scheint in Zeiten von Handy und Internet der Amateurfunk obsolet zu sein.

Besondere Umstände wie die Lawinenkatastrophe von Galtür (1) lassen erkennen, dass Funkamateure im Notfall auch bei Ausfall oder Überlastung konventioneller Nachrichtenverbindungen Informations-Brücken spannen können.

Ein Projekt-Arbeit zum Thema „Notfunk als Unterstützung des BOS-Funknetzes in besonderen Lagen“ (2) schildert die Geschichte des Amateurfunks, skizziert kurz die verwendete Technik und die heutige Bedeutung in Notsituationen.



A.R.E.N.A. - die Organisation des Notfunks in Österreich wird durch die folgende Wiki Seite beschrieben: (3)

Die WWW-Seiten der österreichischen Funkamateure sind unter folgender URL zu finden: (4)

Gesucht werden nach wie vor Damen und Herren die sich zum konkreten Mitmachen begeistern lassen. Der Bogen der möglichen Tätigkeiten reicht weit von der klassischen Nachrichtentechnik, Antennen, Maßnahmen und Hilfsmittel zum autarken Betrieb, bis zur Kombination mit moderner EDV. Gerade für die Aktualisierung von Software-Paketen (insbesondere auch unter dem Betriebssystem Linux) würden zusätzliche helfende Hände gebraucht.

DI Rainer Kirsch, oe1kis@oevsv.at



Peter, OE3BPS aus der LWZ Tulln AOEC2010

- (1) http://de.wikipedia.org/wiki/Lawinenkatastrophe_von_Galt%C3%BCr
- (2) <http://www.darc.de/uploads/media/Projektarbeit-Notfunk.pdf>
- (3) <http://wiki.oevsv.at/index.php/Kategorie:NOTFUNK>
- (4) <http://www.oevsv.at/>

