



Denne guide er oprindeligt
udgivet på Eksperten.dk

Geografisk lokalisering i ASP.NET

Denne artikel forklarer lidt om hvorfor og hvordan man laver geografisk lokalisering og viser noget kode.

Der er andre artikler for andre web sprog.

Den forudsætter lidt kendskab til generel web, ASP.NET, C# / VB.NET og ADO.NET.

Skrevet den **06. Feb 2009** af **arne_v** i kategorien **Programmering / ASP.NET** | ★★★★★

Historie:

V1.0 - 30/12/2004 - original

Hvad er geografisk lokalisering

Din side finder ud af hvor brugeren er henne. Typisk hvilket land. Men i nogen tilfælde også hvilken landsdel eller hvilken by.

Hvorfor laver man geografisk lokalisering

Det kan der være mange grunde til.

Nogle eksempler:

- 1) Man vil forbyde adgang til hele siten eller visse sider fra enkelte lande eller eventuelt alle andre lande end ens eget p.g.a. lovmæssige krav.
- 2) Man vil automatisk skifte til sider i brugerens sprog.
- 3) Man vil generere sider som viser nærmeste butikker eller nærmeste download server eller lignende.

Hvordan laver man geografisk lokalisering

Man henter adressen på client maskinen fra HTTP headeren Remote-Addr og slår op i en database som mapper fra IP adresser til geografisk lokation.

Kan det ikke gøres nemmere ?

Man kigger på HTTP headeren Accept-Language og ser hvilket sprog browseren er sat til.

I praksis er det ubrugeligt, da masser af ikke personer i ikke engelsk talende lande bruger engelske versioner

af software.

Man laver reverse DNS lookup på adressen på client maskinen og ser hvilket land domæne navnet tilhører.

I praksis er det ubrugeligt, da meget almindelige domæner som .com .org .net .biz etc. ikke nødvendigvis er USA.

Man beder brugeren selv vælge sprog eller nærmeste butik / download site.

Det virker naturligvis for disse formål. Og er absolut en simpel og nem løsning, som man bør overveje.

Løsningen kan åbenlyst ikke bruges til at blokere for uønskede med.

Geo lokaliserings databaser

Man skal som sagt have en database der mapper fra IP adresser til land.

Der eksisterer adskillige sådanne. En masse kommercielle og enkelte gratis.

Det er vigtigt at huske at den database skal løbende opdateres, da internettet ikke er statisk.

Man skal også gøre sig klart at det ikke er en 100% sikker lokalisering.

Forskellene på de kommercielle og de gratis er typisk:

- * de gratis er kun på land - de kommercielle er både land og landsdel/by
- * de gratis har 90-95% sikkerhed - de kommercielle har 95-98% sikkerhed
- * de gratis opdateres en gang om måneden - de kommercielle opdateres hver dag

Jeg kender 2 gratis:

- * <http://ip-to-country.webhosting.info/>
- * http://www.maxmind.com/app/geoip_country

Inden du starter brug af dem, så skal du nærlæse deres licens betingelser og checke om de er kompatible med dit brug.

Nedenstående kode viser hvordan de kan bruges. Du vælger hvilken af de 2 du vil bruge og retter eventuelt koden til så den passer til dit formål.

Database byg kode

GeoLoad.cs

```
using System;
using System.IO;
using System.Net;
using System.Data;
using System.Data.SqlClient;
using ICSharpCode.SharpZipLib.Zip;
```

```

namespace E_Geo
{
    public class GeoLoad
    {
        /* START CONFIGURATION */
        private const String UrlWebHostingInfo =
"http://ip-to-country.webhosting.info/downloads/ip-to-country.csv.zip";
        private const String UrlMaxMind =
"http://www.maxmind.com/download/geoip/database/GeoIPCountryCSV.zip";
        private const String WorkDir = @"C:\e";
        private const String ConnStr = "server=ARNEPC2;Integrated
Security=SSPI;database=Test";
        /* END CONFIGURATION */
        private static void Download(string urlstr, string zipfnm)
        {
            WebClient wc = new WebClient();
            wc.DownloadFile(urlstr, WorkDir + @"\" + zipfnm);
        }
        private static void Unzip(string zipfnm, string csvfnm)
        {
            Stream s = new FileStream(WorkDir + @"\" + csvfnm,
            FileMode.Create, FileAccess.Write);
            ZipInputStream zis = new ZipInputStream(new FileStream(WorkDir +
            @"\" + zipfnm, FileMode.Open, FileAccess.Read));
            ZipEntry ze = zis.GetNextEntry();
            byte[] b = new byte[100000];
            int n;
            while((n = zis.Read(b, 0, b.Length)) > 0)
            {
                s.Write(b, 0, n);
            }
            zis.Close();
            s.Close();
        }
        private static void DbImport(string csvfnm, int fromcol, int tocol,
        int countrycol, string tblnam)
        {
            SqlConnection con = new SqlConnection(ConnStr);
            con.Open();
            SqlCommand drp = new SqlCommand("DROP TABLE " + tblnam, con);
            try
            {
                drp.ExecuteNonQuery();
            }
            catch(Exception)
            {
                /* we will end here first time */
            }
            SqlCommand cre1 = new SqlCommand("CREATE TABLE " + tblnam + "(id
            INTEGER NOT NULL, ip1 NUMERIC(10,0), ip2 NUMERIC(10,0), country CHAR(2),
            PRIMARY KEY(id))", con);
            cre1.ExecuteNonQuery();
            SqlCommand cre2 = new SqlCommand("CREATE INDEX ixip1 ON " + tblnam
            + "(ip1)", con);
            cre2.ExecuteNonQuery();
        }
    }
}

```

```

        SqlCommand cre3 = new SqlCommand("CREATE INDEX ixip2 ON " + tblnam
+ "(ip2)", con);
        cre3.ExecuteNonQuery();
        SqlCommand ins = new SqlCommand("INSERT INTO " + tblnam + "
VALUES(@id,@ip1,@ip2,@country)", con);
        ins.Parameters.Add("@id", SqlDbType.Int);
        ins.Parameters.Add("@ip1", SqlDbType.Decimal);
        ins.Parameters.Add("@ip2", SqlDbType.Decimal);
        ins.Parameters.Add("@country", SqlDbType.VarChar);
        StreamReader sr = new StreamReader(WorkDir + @"\" + csvfnm);
        int n = 0;
        string line;
        while((line = sr.ReadLine()) != null)
        {
            n++;
            string[] cols = line.Split(",".ToCharArray());
            decimal fromval = decimal.Parse(cols[fromcol-1].Replace("'", '
').Trim());
            decimal toval = decimal.Parse(cols[tocol-1].Replace("'", '
').Trim());
            string code = cols[countrycol-1].Replace("'", ' ').Trim();
            ins.Parameters["@id"].Value = n;
            ins.Parameters["@ip1"].Value = fromval;
            ins.Parameters["@ip2"].Value = toval;
            ins.Parameters["@country"].Value = code;
            ins.ExecuteNonQuery();
        }
        sr.Close();
        con.Close();
    }
    private static void LoadWebHostingInfo()
    {
        Download(UrlWebHostingInfo, "whi.zip");
        Unzip("whi.zip", "whi.csv");
        DbImport("whi.csv", 1, 2, 3, "whi");
    }
    private static void LoadMaxMind()
    {
        Download(UrlMaxMind, "mm.zip");
        Unzip("mm.zip", "mm.csv");
        DbImport("mm.csv", 3, 4, 5, "mm");
    }
    public static void Main(string[] args)
    {
        LoadWebHostingInfo();
        LoadMaxMind();
    }
}
}

```

```

Imports System
Imports System.IO
Imports System.Net
Imports System.Data
Imports System.Data.SqlClient
Imports ICSharpCode.SharpZipLib.Zip

Namespace E_Geo
    Public Class GeoLoad
        Private Const UrlWebHostingInfo As String =
"http://ip-to-country.webhosting.info/downloads/ip-to-country.csv.zip"
        Private Const UrlMaxMind As String =
"http://www.maxmind.com/download/geoip/database/GeoIPCountryCSV.zip"
        Private Const WorkDir As String = "C:\e"
        Private Const ConnStr As String = "server=ARNEPC2;Integrated
Security=SSPI;database=Test"

        Private Shared Sub Download(ByVal urlstr As String, ByVal zipfnm As
String)
            Dim wc As WebClient = New WebClient
            wc.DownloadFile(urlstr, WorkDir + "\" + zipfnm)
        End Sub

        Private Shared Sub Unzip(ByVal zipfnm As String, ByVal csvfnm As
String)
            Dim s As Stream = New FileStream(WorkDir + "\" + csvfnm,
FileMode.Create, FileAccess.Write)
            Dim zis As ZipInputStream = New ZipInputStream(New
FileStream(WorkDir + "\" + zipfnm, FileMode.Open, FileAccess.Read))
            Dim ze As ZipEntry = zis.GetNextEntry
            Dim b(100000) As Byte
            Dim n As Integer
            n = zis.Read(b, 0, b.Length)
            While n > 0
                s.Write(b, 0, n)
                n = zis.Read(b, 0, b.Length)
            End While
            zis.Close
            s.Close
        End Sub

        Private Shared Sub DbImport(ByVal csvfnm As String, ByVal fromcol As
Integer, ByVal tocol As Integer, ByVal countrycol As Integer, ByVal tblnam As
String)
            Dim con As SqlConnection = New SqlConnection(ConnStr)
            con.Open
            Dim drp As SqlCommand = New SqlCommand("DROP TABLE " + tblnam,
con)
            Try
                drp.ExecuteNonQuery
            Catch e As Exception
                ' we will end here first time
            End Try
            Dim cre1 As SqlCommand = New SqlCommand("CREATE TABLE " + tblnam +
"(id INTEGER NOT NULL, ip1 NUMERIC(10,0), ip2 NUMERIC(10,0), country CHAR(2),

```

```

PRIMARY KEY(id))", con)
    cre1.ExecuteNonQuery
    Dim cre2 As SqlCommand = New SqlCommand("CREATE INDEX ixip1 ON " +
tblnam + "(ip1)", con)
    cre2.ExecuteNonQuery
    Dim cre3 As SqlCommand = New SqlCommand("CREATE INDEX ixip2 ON " +
tblnam + "(ip2)", con)
    cre3.ExecuteNonQuery
    Dim ins As SqlCommand = New SqlCommand("INSERT INTO " + tblnam + "
VALUES(@id,@ip1,@ip2,@country)", con)
    ins.Parameters.Add("@id", SqlDbType.Int)
    ins.Parameters.Add("@ip1", SqlDbType.Decimal)
    ins.Parameters.Add("@ip2", SqlDbType.Decimal)
    ins.Parameters.Add("@country", SqlDbType.VarChar)
    Dim sr As StreamReader = New StreamReader(WorkDir + "\" + csvfnm)
    Dim n As Integer = 0
    Dim line As String
    line = sr.ReadLine
    While Not (line Is Nothing)
        n = n + 1
        Dim cols As String() = line.Split(",".ToCharArray)
        Dim fromval As Decimal = Decimal.Parse(cols(fromcol -
1).Replace("'", " ").Trim)
        Dim toval As Decimal = Decimal.Parse(cols(tocol -
1).Replace("'", " ").Trim)
        Dim code As String = cols(countrycol - 1).Replace("'", "
").Trim
        ins.Parameters("@id").Value = n
        ins.Parameters("@ip1").Value = fromval
        ins.Parameters("@ip2").Value = toval
        ins.Parameters("@country").Value = code
        ins.ExecuteNonQuery
        line = sr.ReadLine
    End While
    sr.Close
    con.Close
End Sub

Private Shared Sub LoadWebHostingInfo()
    'Download(UrlWebHostingInfo, "whi.zip")
    Unzip("whi.zip", "whi.csv")
    DbImport("whi.csv", 1, 2, 3, "whi")
End Sub

Private Shared Sub LoadMaxMind()
    'Download(UrlMaxMind, "mm.zip")
    Unzip("mm.zip", "mm.csv")
    DbImport("mm.csv", 3, 4, 5, "mm")
End Sub

Public Shared Sub Main(ByVal args As String())
    LoadWebHostingInfo
    LoadMaxMind
End Sub
End Class

```

End Namespace

Bemærk at dette program gør det hele:

- downloader zip fil
- udpakker csv fil fra zip fil
- loader csv fil til databasen

Du skal kun tilrette konfigurationen i toppen.

Koden bruger SharpZipLib som kan hentes her:

<http://www.icsharpcode.net/OpenSource/SharpZipLib/Default.aspx>

Database søge kode

GeoLocate.cs

```
using System;
using System.Collections;
using System.Net;
using System.Data.SqlClient;

namespace E_Geo
{
    public class GeoLocate
    {
        /* START CONFIGURATION */
        private const String ConnStr = "server=ARNEPC2;Integrated
Security=SSPI;database=Test";
        /* END CONFIGURATION */
        private static SqlConnection con;
        private static Hashtable cache;
        static GeoLocate()
        {
            string connstrcopy = ConnStr;
            con = new SqlConnection(connstrcopy);
            con.Open();
            cache = new Hashtable();
        }
        public static string Locate(string ip, string tblnam)
        {
            string country = (string)cache[ip];
            if(country == null) {
                byte[] b = Dns.Resolve(ip).AddressList[0].GetAddressBytes();
                long ipnum = 0;
                for(int i = 0; i < 4; i++) {
                    ipnum = (ipnum << 8) | (uint)(b[i] & 0xFF);
                }
                lock(con)
                {
                    SqlCommand sel = new SqlCommand("SELECT country FROM " +
tblnam + " WHERE ip1 < " + ipnum + " AND " + ipnum + " < ip2", con);
                    SqlDataReader rdr = sel.ExecuteReader();
```

```

        if(rdr.Read())
        {
            country = (string)rdr[0];
        }
        else
        {
            country = "??";
        }
        rdr.Close();
        cache.Add(ip, country);
    }
}
return country;
}
}
}
}

```

GeoLocate.vb

```

Imports System
Imports System.Collections
Imports System.Net
Imports System.Data.SqlClient

Namespace E_Geo
    Public Class GeoLocate
        Private Const ConnStr As String = "server=ARNEPC2;Integrated
Security=SSPI;database=Test"
        Private Shared con As SqlConnection
        Private Shared cache As Hashtable

        Shared Sub New()
            Dim connstrcopy As String = ConnStr
            con = New SqlConnection(connstrcopy)
            con.Open
            cache = New Hashtable
        End Sub

        Public Shared Function Locate(ByVal ip As String, ByVal tblnam As
String) As String
            Dim country As String = CType(cache(ip), String)
            If country Is Nothing Then
                Dim b As Byte() =
                Dns.Resolve(ip).AddressList(0).GetAddressBytes
                Dim ipnum As Long = 0
                Dim i As Integer
                For i = 0 To 3
                    ipnum = (ipnum << 8) Or (b(i) And &HFF)
                Next
                SyncLock con
                    Dim sel As SqlCommand = New SqlCommand("SELECT country
FROM " + tblnam + " WHERE ip1 < " + ipnum.ToString + " AND " + ipnum.ToString

```

```

+ " < ip2", con)
        Dim rdr As SqlDataReader = sel.ExecuteReader
        If rdr.Read Then
            country = CType(rdr(0), String)
        Else
            country = "??"
        End If
        rdr.Close
        cache.Add(ip, country)
    End SyncLock
End If
Return country
End Function
End Class
End Namespace

```

test.aspx

```

<%@ Page Language="C#" %>
<%@ import namespace="E_Geo" %>
<%=GeoLocate.Locate(Request.UserHostAddress, "whi")%>
<%=GeoLocate.Locate(Request.UserHostAddress, "mm")%>

```

test.aspx

```

<%@ Page Language="VB" %>
<%@ import namespace="E_Geo" %>
<%=GeoLocate.Locate(Request.UserHostAddress, "whi")%>
<%=GeoLocate.Locate(Request.UserHostAddress, "mm")%>

```

Du skal selvfølgelig også tilrette konfigurationen her.

.dll filen og .aspx filen der bruger den skal deployes helt normalt (ligger udenfor scope af denne artikel).

De viste test.aspx er naturligvis uinteressante, men de viser hvordan man kalder.

Kommentar af wever d. 02. Dec 2008 | 1

Fin artikel, arne_v rundede 800.000 point da jeg købte adgang :)

Kommentar af tofte d. 31. Dec 2004 | 2

Belyser problemer, og kommer med en fin løsning.

Kommentar af olhansen d. 22. Feb 2007 | 3

Fin løsning

Kommentar af rasmus-madsen d. 02. Jan 2005 | 4

Jeg har nu, efter testning fundet ud af hvor præcis den egentlig var... Så stemmer lig igen, med bedre karakter ;)

Kommentar af madschristensen d. 21. Jun 2005 | 5

Fin artikel!

Kommentar af andr3as d. 31. Dec 2004 | 6

jeg ved ikke lige hvorfor rasmus-madsen skriver som han gør, men jeg synes det er en meget god artikel!