

Browscap ISAPI-Filter für IIS

An ISAPI-Filter to use the browscap.ini File Values on the IIS, with any programming language.

All you must have to run it:

 [browscap_filter.zip](#) (99,97 kByte) 30.12.2018 21:29

The Free Pascal source for the Filter:

 [browscap_filter.lpr](#) Pascal (5,23 kByte) 27.05.2014 14:39

```
//
*****

//  Title..... :  browscap ISAPI Filter
//
//  Modulname ..... :  browscap_filter.lpr (ISAPI-DLL)
//  Type ..... :  Lazarus Project File
//  Author ..... :  Udo Schmal
//  Development Status :  10.05.2014
//  Operating System .. :  Win32
//  IDE ..... :  Lazarus
//
*****

library browscapFilter;
{$mode objfpc}{$H+}

uses
  Windows, SysUtils, Classes, IniFiles, browscap, isapi;

var
  LoggingLock, BrowscapLock: TRTLCriticalSection; // critical section for
TStringList
  BrowscapIni: TBrowscap; // class that handle browscap.ini access
  FieldList: TStringList;
  debugMode: boolean;
  logfilePath: string;

procedure Log(sLine: string);
var
  f: textfile;
  fn: string;
begin
  if debugMode then
  begin
    EnterCriticalSection(LoggingLock);
    try
      fn := logfilePath + 'browscap_' + FormatDateTime('dd"."mm"."yyyy',
Now) + '.log';
      {$I-}
```

```

    AssignFile(f, fn);
    if FileExists(fn) then
        Append(f)
    else
        Rewrite(f);
    Writeln(f, FormatDateTime('hh:nn:ss:zzz'|'', Now) +
StringReplace(sLine, #13#10, #13#10'          | ', [rfReplaceAll]));
    CloseFile(f);
    IOResult;
{$I+}
    finally
        LeaveCriticalSection(LoggingLock);
    end;
end;
end;
end;

function GetFilterVersion(var Ver: THTTP_FILTER_VERSION): BOOL; stdcall;
begin
    Log('GetFilterVersion: ISAPI Filter is running!');
    Ver.dwFilterVersion := MakeLong(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);
    Ver.lpszFilterDesc := 'Browscap ISAPI Filter';
    Ver.dwFlags := SF_NOTIFY_PREPROC_HEADERS or SF_NOTIFY_LOG;
    result := true;
end;

function TerminateFilter(dwFlags: DWORD): BOOL; stdcall;
begin
    Log('TerminateFilter: ISAPI Filter terminating...');
    Integer(result) := 1; // This is so that the Apache web server will know
what "True" really is
end;

function HttpFilterProc(var pfc: THTTP_FILTER_CONTEXT; NotificationType:
DWORD;

                        pvNotification: pointer): DWORD; stdcall;

function OnPreprocHeaders: integer;
var PreprocHeader: PHTTP_FILTER_PREPROC_HEADERS;

    function GetHeaderValue(Name: string): string;
    var
        Buffer: array[0..4096] of char;
        lenBuffer : DWORD;
    begin
        result := '';
        try
            lenBuffer := SizeOf(Buffer);
            if PreprocHeader^.GetHeader(pfc, PChar(Name), Buffer, lenBuffer)
then
                result := Buffer;

```

```

    except
        Log('GetHeaderValue Error');
    end;
end;

procedure AddHeaderValue(Name: string; Value: string);
begin
    try
        PreprocHeader^.AddHeader(pfc, PChar(Name), PChar(Value));
    except
        Log('AddHeaderValue Error');
    end;
end;

var
    UserAgent: string;
    ValueList: TStringList;
    i: integer;
begin
    result := SF_STATUS_REQ_NEXT_NOTIFICATION;
    PreprocHeader := PHTTP_FILTER_PREPROC_HEADERS(pvNotification);
    UserAgent := GetHeaderValue('User-Agent:');
    Log('User-Agent: ' + UserAgent);
    // get browscap values for the useragent
    ValueList := TStringList.Create;
    EnterCriticalSection(BrowscapLock);
    try
        BrowscapIni.GetUserAgentInfo(UserAgent, ValueList);
    finally
        LeaveCriticalSection(BrowscapLock);
    end;
    // add browscap values to request header
    for i:=0 to FieldList.Count-1 do
    begin
        Log('add header(' + FieldList.Names[i] + ': ' +
ValueList.Values[FieldList.ValueFromIndex[i]]+ ')');
        AddHeaderValue(FieldList.Names[i]+':',
ValueList.Values[FieldList.ValueFromIndex[i]]);
    end;
    ValueList.Free;
end;

//HttpFilterProc mainline code
begin
    result := SF_STATUS_REQ_NEXT_NOTIFICATION;
    if NotificationType = SF_NOTIFY_PREPROC_HEADERS then
        result := OnPreprocHeaders
    else
        result := SF_STATUS_REQ_NEXT_NOTIFICATION;
    end;
end;

```

```

exports GetFilterVersion, HttpFilterProc, TerminateFilter;

function ModuleFileName(): string;
var Buffer: array[0..MAX_PATH] of Char;
begin
    FillChar(Buffer, SizeOf(Buffer), #0);
    SetString(result, Buffer, GetModuleFileName(hInstance, Buffer,
Length(Buffer)));
    if pos('\\?\\', result) = 1 then
        result := copy(result, 5, MAX_PATH);
end;

var
    BrowscapIniPath: string; // path to browscap.ini
    ini: TIniFile; // browscap_filter.ini
    dllPath: string;
initialization
    InitCriticalSection(LoggingLock); // logfile access
    InitCriticalSection(BrowscapLock); // browscap ini access
    dllPath := ModuleFileName();
    Ini := TIniFile.Create(ChangeFileExt(dllPath, '.ini'));
    dllPath := ExtractFilePath(dllPath);

    BrowscapIniPath := Ini.ReadString('browscap', 'path', dllPath +
'browscap.ini');
    debugMode := Ini.ReadBool('browscap', 'debug', false);
    logfilePath := Ini.ReadString('browscap', 'logpath', dllPath);
    BrowscapIni := TBrowscap.Create(BrowscapIniPath);
    FieldList := TStringList.Create;
    Ini.readSectionValues('fields', FieldList);
    Ini.Free;
    Log('initialization ISAPI Filter!');
    if IsLibrary then Log('is library');
    if IsMultiThread then Log('is multithreaded');
finalization
    Log('finalization ISAPI Filter!');
    DoneCriticalSection(BrowscapLock);
    DoneCriticalSection(LoggingLock);
end.

```

Config file for the ISAPI-Filter (browscap_filter.ini):

```

[browscap]
path=C:\inetpub\browscap.ini
logpath=C:\inetpub\logging\
debug=1

```

```

[fields]
X-Client-Name=Browser
X-Client-Version=Version

```

```
X-Client-Is-Mobile-Device=isMobileDevice
X-Client-Is-Tablet=isTablet
X-Client-Is-Crawler=Crawler
```

Logfile Lines for one request, if debug=1:

```
14:40:37:973|User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:29.0)
Gecko/20100101 Firefox/29.0
14:40:38:051|add header(X-Client-Name: Firefox)
14:40:38:051|add header(X-Client-Version: 29.0)
14:40:38:051|add header(X-Client-Is-Mobile-Device: false)
14:40:38:051|add header(X-Client-Is-Tablet: false)
14:40:38:051|add header(X-Client-Is-Crawler: false)
```

ASP (VB-Script) Example:

```
<%
Response.Write(Replace(Request.ServerVariables("ALL_RAW"), vbCrLf, "<br />")
+ "<br />")
Response.Write("Name: " + Request.ServerVariables("HTTP_X_CLIENT_NAME") + "
<br/>")
Response.Write("Version: " +
Request.ServerVariables("HTTP_X_CLIENT_VERSION") + "<br/>")
Response.Write("Mobile: " +
Request.ServerVariables("HTTP_X_CLIENT_IS_MOBILE_DEVICE") + "<br />")
Response.Write("Tablet: " +
Request.ServerVariables("HTTP_X_CLIENT_IS_TABLET") + "<br />")
Response.Write("Crawler: " +
Request.ServerVariables("HTTP_X_CLIENT_IS_CRAWLER") + "<br />")
```

Output:

```
Cache-Control: max-age=0
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: de-de,de;q=0.7,en-us;q=0.3
Cookie: ASPSESSIONIDAQQSDBSS=FBKBCEDCKLICGFBEMGGHBNFF
Host: test.localhost
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:29.0) Gecko/20100101
Firefox/29.0
DNT: 1
X-Client-Name: Firefox
X-Client-Version: 29.0
X-Client-Is-Mobile-Device: false
X-Client-Is-Tablet: false
X-Client-Is-Crawler: false

Name: Firefox
Version: 29.0
Mobile: false
Tablet: false
Crawler: false
```

Autor: Udo Schmal, veröffentlicht: 26.05.2014, letzte Änderung: 29.08.2024

© Copyright 2024 Udo Schmal