

Browscap Apache Module

Ein Modul zur Nutzung der Browscap.ini auf dem Apache 2 Webserver

Hier nun meine erste Variante eines Apache Moduls zur Nutzung der Browscap.ini Datei, sicherlich stehe ich mit diesem Projekt zur Zeit noch ganz am Anfang, da es mein erstes Apache Modul ist, welches ich mit Free Pascal unter Windows entwickle, was auch unter Linux laufen soll. Also das bedeutet Cross-Kompilieren ich habe mir die benötigten Dateien aus dem Unix /lib Verzeichnis besorgt und einfach kompiliert ob das nun von Erfolg gekrönt ist weiß ich nicht da mir zur Zeit keine unterschiedlichen Linux Testumgebungen zur Verfügung stehen, darum bin ich auf Eure Mithilfe angewiesen!

Doch nun zu meinem Ansatz, ich lade die browscap.ini die im Apache Verzeichnis liegen muss, während des Serverstarts in den Speicher und erzeuge eine vorsortierte Liste der Sections. Bei einer Abfrage wird der User-Agent aus dem Request-Header gelesen und mit Hilfe der geladenen Browscap.ini die Eigenschaften ermittelt. Anschließend werden ein paar von mir als relevant befundene Eckwerte dem Request-Header hinzugefügt, wodurch sie dann allen folgenden Aktionen zur Verfügung stehen. Falls jemand von Euch eine bessere Idee hat an welcher Stelle ich sonst die Werte für eine folgende Bearbeitung z.B. durch PHP zur Verfügung stellen könnte immer heraus damit.

Schön wäre es natürlich die Werte in einer "Session"-Datei auf dem Server abzulegen und dann dem Header lediglich eine Cookie zu setzen der Auskunft über die Datei gibt.

Erforderliche Eintrag in der **httpd.conf** damit das Modul überhaupt geladen wird:

```
LoadModule browscap_module modules/mod_browscap.so
```

Aktivieren des Moduls, in einem Bereich <VirtualHost>, <Directory>, <Location>, <Files>, .htaccess (All), damit es an dieser Stelle zum Einsatz kommt.

```
<IfModule browscap_module>
    GetBrowscap On
</IfModule>
```

Die Apache Module .so **Datei gehört in das /apache/modules Verzeichnis**, hier zwei unterschiedliche Versionen, die erste Datei funktioniert in meiner lokalen UwAmp installation unter Windows, die zweite Datei ist ungetestet!

 [mod browscap-i386-win32.zip](#)

 [mod browscap-i386-linux.zip](#) (99,62 kByte) 30.12.2018 21:29 (262,86 kByte) 30.12.2018 21:29

mod_browscap.ini (Konfigurationsdatei für mod_browscap.so) auch im modules Verzeichnis ablegen:

```
[browscap]
path=C:\UwAmp\bin\apache\modules\browscap.ini
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
```

Download der browscap.ini von browscap.org (<https://browscap.org/>) die Variante der INI-Datei sollte egal sein.

In der mod_browscap.ini muss der Pfad zur browscap.ini angepasst werden, das setzen von debug auf 1 oder 0 aktiviert oder deaktiviert die Debug-Zeilen in der Apache error.log.

Die Zuordnung in der Section 'fields' steuert die Übergabe der Werte in Request-Header.

Ein Beispiel was unter PHP die von dem Modul zur Verfügung gestellten Werte ausgibt:

```
<?php
echo 'client name: '.$_SERVER['HTTP_X_CLIENT_NAME'].'<br />';
echo 'client version: '.$_SERVER['HTTP_X_CLIENT_VERSION'].'<br />';
echo 'client is mobile device:
'.$_SERVER['HTTP_X_CLIENT_IS_MOBILE_DEVICE'].'<br />';
echo 'client is tablet: '.$_SERVER['HTTP_X_CLIENT_IS_TABLET'].'<br />';
echo 'client is crawler: '.$_SERVER['HTTP_X_CLIENT_IS_CRAWLER'].'<br />';
?>
```

Die Ausgabe:

```
client name: Firefox
client version: 29.0
client is mobile device: false
client is tablet: false
client is crawler: false
```

Der Logfile Auszug:

```
[Tue May 27 01:26:20 2014] [notice] Apache/2.2.22 (Win32) PHP/5.3.25
configured -- resuming normal operations
[Tue May 27 01:26:20 2014] [notice] Server built: Jan 28 2012 11:16:39
[Tue May 27 01:26:20 2014] [notice] Parent: Created child process 2416
[Tue May 27 01:26:20 2014] [debug] mpm_winnt.c(477): Parent: Sent the
scoreboard to the child
[Tue May 27 01:26:21 2014] [debug] mod_browscap.so(146):
createDirConfig(C:/UwAmp/www/test/)
[Tue May 27 01:26:21 2014] [debug] mod_browscap.so(182):
createServerConfig(test.localhost)
[Tue May 27 01:26:21 2014] [debug] mod_browscap.so(212): cmd_browscap()
[Tue May 27 01:26:21 2014] [debug] mod_browscap.so(200):
mergeServerConfig(SVR(),SVR(test.localhost))
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(146):
createDirConfig(C:/UwAmp/www/test/)
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(182):
createServerConfig(test.localhost)
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(212): cmd_browscap()
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(200):
mergeServerConfig(SVR(),SVR(test.localhost))
[Tue May 27 01:26:22 2014] [notice] Child 2416: Child process is running
[Tue May 27 01:26:22 2014] [debug] mpm_winnt.c(398): Child 2416: Retrieved
our scoreboard from the parent.
[Tue May 27 01:26:22 2014] [info] Parent: Duplicating socket 220 and sending
it to child process 2416
[Tue May 27 01:26:22 2014] [debug] mpm_winnt.c(595): Parent: Sent 1 listeners
to child 2416
[Tue May 27 01:26:22 2014] [debug] mpm_winnt.c(554): Child 2416: retrieved 1
listeners from parent
```

```
[Tue May 27 01:26:22 2014] [notice] Child 2416: Acquired the start mutex.
[Tue May 27 01:26:22 2014] [notice] Child 2416: Starting 64 worker threads.
[Tue May 27 01:26:22 2014] [notice] Child 2416: Starting thread to listen on
port 80.
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(114): get header(User-
Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:29.0) Gecko/20100101
Firefox/29.0)
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(127): add header(X-Client-
Name: Firefox)
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(127): add header(X-Client-
Version: 29.0)
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(127): add header(X-Client-
Is-Mobile-Device: false)
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(127): add header(X-Client-
Is-Tablet: false)
[Tue May 27 01:26:22 2014] [debug] mod_browscap.so(127): add header(X-Client-
Is-Crawler: false)
```

ToDo:

1. Überprüfen der auf Cookies basierenden Variante
2. Tests besonders unter UNIX/Linux

Der Quelltext:

 [mod browscap.lpr](#) Pascal (10,53 kByte) 28.05.2014 11:06

```
//
*****
//  Title..... :  browscap Apache Modul
//
//  Type ..... :  Apache Module Project File
//  Modulname ..... :  mod_browscap.pas
//  Apache httpd.conf . :  LoadModule browscap_module "
{APACHEPATH}/modules/mod_browscap.so"
//  Context ..... :  Server, <VirtualHost>, <Directory>, <Location>,
<Files>, .htaccess (All)
//  Syntax ..... :  GetBrowscap On/Off
//  Author ..... :  Udo Schmal
//  Development Status :  27.04.2014
//  Operating System .. :  Win32/64 Linux
//  IDE ..... :  Lazarus
//  known bugs ..... :  apache double-startup
//*****

library mod_browscap;

{$mode objfpc}{$H+}
{$LIBPREFIX ''}
{$IFDEF WIN32}
    {$DEFINE WINDOWS}
{$ENDIF}
```

```

uses {$ifdef WINDOWS}windows,{$endif}
Classes, SysUtils, httpd, apr, IniFiles, browscap;

const
MODULE_NAME = 'mod_browscap.so';

// configuration specific to this module
type
config = record
loc: PChar;           // Location to which this record applies.
enabled: integer;     // Boolean: directive enabled here?
congenital: integer; // Boolean: did we inherit an enabled
end;
Pconfig = ^config;

var
browscap_module: module; // this module
browscap_module_ptr: Pmodule; // pointer to this module
cmdRec: command_rec; // define the directive specified in this module.
BrowscapIniLock: TRTLCriticalSection; // critical section for TStringList
BrowscapIni: TBrowscap; // class that handle browscap.ini access
FilePath, BrowscapIniPath: string; // path to browscap.ini
ini: TIniFile; // mod_browscap.ini
FieldList: TStringList;
serverRec: Pserver_rec = nil;
debugMode: boolean;

// export for unix & windows
{$ifdef UNIX} public name 'browscap_module'; {$endif}
{$ifdef WINDOWS} exports browscap_module name 'browscap_module'; {$endif}

// locate directory configuration record for the current request.
function locateDirectoryConfig(const r: Prequest_rec): Pconfig; cdecl;
begin
    Result := Pconfig(ap_get_module_config(r^.per_dir_config,
@browscap_module));
end;

// locate server configuration record for the specified server.
function locateServerConfig(const s: Pserver_rec): Pconfig; cdecl;
begin
    Result := Pconfig(ap_get_module_config(s^.module_config, @browscap_module));
end;

// Likewise for configuration record for the specified request.
function locateRequestConfig(const r: Prequest_rec): Pconfig; cdecl;
begin
    Result := Pconfig(ap_get_module_config(r^.request_config,
@browscap_module));
end;

```

```

// Likewise for configuration record for a connection.
function locateConnectionConfig(const c: Pconn_rec): Pconfig; cdecl;
begin
    Result := Pconfig(ap_get_module_config(c^.conn_config, @browscap_module));
end;

// This routine is called after the request has been read but before any
// other
// phases have been processed. This allows us to make decisions based upon
// the input header fields.
function postReadRequest(r: Prequest_rec): integer; cdecl;
var
    cfg: Pconfig;
    UserAgent, FileName, Ext: string;
    i: integer;
    ValueList: TStringList;
begin
    // check if request in enabled server
    // cfg := locateServerConfig(r^.server);
    // if (cfg^.enabled = 0) then
    // begin
    //     Result := DECLINED;
    //     Exit;
    // end;
    // check if request in enabled directory
    cfg := locateDirectoryConfig(r);
    if (cfg^.enabled = 0) then
    begin
        Result := DECLINED;
        Exit;
    end;
    // figure out which file is being requested, only add infos if they are
    // later used
    FileName := r^.filename;
    Ext := ExtractFileExt(FileName);
    // only add infos to a call of an php-files (dynamic content)
    if not (Ext = '') and not (Ext = '.php') then
    begin
        Result := DECLINED;
        Exit;
    end;
    // get the useragent from request header
    UserAgent := apr_table_get(r^.headers_in, 'User-Agent');
    if debugMode then
        ap_log_error(MODULE_NAME, 114, APLOG_DEBUG, 0, r^.server, PChar('get
header(User-Agent: ' + UserAgent + ')'), []);
    // get browscap values for the useragent
    ValueList := TStringList.Create;
    EnterCriticalSection(BrowscapIniLock);
    try

```

```

    BrowscapIni.GetUserAgentInfo(UserAgent, ValueList);
finally
    LeaveCriticalSection(BrowscapIniLock);
end;
// add browscap values to request header
for i:=0 to FieldList.Count-1 do
begin
    if debugMode then
        ap_log_error(MODULE_NAME, 127, APLOG_DEBUG, 0, r^.server, PChar('add
header(' + FieldList.Names[i] + ': ' +
ValueList.Values[FieldList.ValueFromIndex[i]]+ ')'), []);
        apr_table_set(r^.headers_in, PChar(FieldList.Names[i]),
PChar(ValueList.Values[FieldList.ValueFromIndex[i]]));
    end;
    ValueList.Free;
    Result := DECLINED; // go on
end;

function createDirConfig(p: Papr_pool_t; dir: PChar): Pointer; cdecl;
var
    cfg: Pconfig;
    note: PChar;
begin
    cfg := Pconfig(apr_pcalloc(p, sizeof(config)));
    cfg^.enabled := 0;
    cfg^.congenital := 0;
    note := dir;
    if note = nil then note := '';
    cfg^.loc := apr_pstrcat(p, [PChar('DIR('), note, PChar(')'), nil]);
    if debugMode then
        ap_log_error(MODULE_NAME, 146, APLOG_DEBUG, 0, serverRec,
'createDirConfig(%s)', [note]);
    Result := Pointer(cfg);
end;

function mergeDirConfig(p: Papr_pool_t; parent_config, newloc_config:
Pointer): Pointer; cdecl;
var
    merged_cfg, parent_cfg, new_cfg: Pconfig;
    note: PChar;
begin
    parent_cfg := Pconfig(parent_config);
    new_cfg := Pconfig(newloc_config);
    merged_cfg := Pconfig(apr_pcalloc(p, sizeof(config)));
    merged_cfg^.enabled := new_cfg^.enabled;
    merged_cfg^.loc := apr_pstrdup(p, new_cfg^.loc);
    merged_cfg^.congenital := (parent_cfg^.congenital or parent_cfg^.enabled);
    if debugMode then
begin
        note := apr_pstrcat(p, [parent_cfg^.loc, PChar(','), new_cfg^.loc, nil]);
        ap_log_error(MODULE_NAME, 164, APLOG_DEBUG, 0, serverRec,

```

```

'mergeDirConfig(%s)', [note]);
    end;
    Result := Pointer(merged_cfg);
end;

function createServerConfig(p: Papr_pool_t; server: Pserver_rec): Pointer;
cdecl;
var
    cfg: Pconfig;
    note: PChar;
begin
    if (serverRec=nil) then serverRec := server;
    note := server^.server_hostname;
    cfg := Pconfig(apr_pcalloc(p, sizeof(config)));
    cfg^.enabled := 0;
    cfg^.congenital := 0;
    if note = nil then note := '';
    cfg^.loc := apr_pstrcat(p, [PChar('SVR('), note, PChar(')'), nil]);
    if debugMode then
        ap_log_error(MODULE_NAME, 182, APLOG_DEBUG, 0, server,
PChar('createServerConfig(%s)'), [note]);
    Result := Pointer(cfg);
end;

function mergeServerConfig(p: Papr_pool_t; server1_config, server2_config:
Pointer): Pointer; cdecl;
var
    merged_cfg, s1_cfg, s2_cfg: Pconfig;
    note: PChar;
begin
    s1_cfg := Pconfig(server1_config); // pointer to config record
    s2_cfg := Pconfig(server2_config); // pointer to config record
    merged_cfg := Pconfig(apr_pcalloc(p, sizeof(config))); // create new record
for merged data
    merged_cfg^.enabled := s2_cfg^.enabled;
    merged_cfg^.congenital := (s1_cfg^.congenital or s1_cfg^.enabled);
    merged_cfg^.loc := apr_pstrdup(p, s2_cfg^.loc);
    if debugMode then
        begin
            note := apr_pstrcat(p, [s1_cfg^.loc, PChar(','), s2_cfg^.loc, nil]);
            ap_log_error(MODULE_NAME, 200, APLOG_DEBUG, 0, serverRec,
'mergeServerConfig(%s)', [note]);
        end;
    result := Pointer(merged_cfg);
end;

// Handler for the GetBrowsecap command, whith it's FLAG.
function cmd_browsecap(cmd: Pcmd_parms; module_config: Pointer; arg: integer):
PChar; cdecl;
var cfg: Pconfig;
begin

```

```

cfg := Pconfig(module_config); // pointer to config record
cfg^.enabled := arg;
if debugMode then
    ap_log_error(MODULE_NAME, 212, APLOG_DEBUG, 0, cmd^.server,
PChar('cmd_browscap()'), []);
    result := nil;
end;

// Registers the hook
procedure registerHooks(p: Papr_pool_t); cdecl;
begin
    ap_hook_post_read_request(@postReadRequest, nil, nil, APR_HOOK_MIDDLE);
end;

function ParentDir(const sPath: string): string;
begin
    result := copy(sPath, 1, LastDelimiter(':', '\',
copy(sPath, 1, length(sPath) - 1)));
end;

function ModuleFileName(): string;
var
    {$ifdef windows}
        Buffer: array[0..MAX_PATH] of Char;
    {$else}
        i, p: integer;
        s: string;
    {$endif}
begin
    {$ifdef windows}
        FillChar(Buffer, SizeOf(Buffer), #0);
        SetString(result, Buffer, GetModuleFileName(hInstance, Buffer,
Length(Buffer)));
        if pos('\?\'', result) = 1 then
            result := copy(result, 5, MAX_PATH);
    {$else}
        // try to find param --path-to-module=/usr/bin/module
        for i:=1 to ParamCount-1 do
            begin
                s := ParamStr(i);
                p := pos('--path-to-module=', s);
                if (p>0) then
                    begin
                        p := Length(s);
                        Delete(s, 1, p);
                    end
                else
                    s := '';
                end;
            end;
        // not found then take the path to the executable apache/bin/apache.exe
        if s='' then

```



```

    s := ParentDir(ExtractFilePath(ParamStr(0))) + 'module';
    s := s + '/' + MODULE_NAME;
{$endif}
end;

// Library initialization code
initialization
    browscap_module_ptr := @browscap_module;
    FillChar(browscap_module_ptr^, SizeOf(browscap_module_ptr^), 0);

    with cmdRec do // command record
    begin
        name := 'GetBrowscap'; // command for this module
        func := cmd_func(@cmd_browscap); // command target
        cmd_data := nil;
        req_override := OR_OPTIONS;
        args_how := FLAG; // On or Off
        errmsg := 'whether or not to get browscap info';
    end;

    STANDARD20_MODULE_STUFF(browscap_module);
    with browscap_module do
    begin
        name := MODULE_NAME;
        magic := MODULE_MAGIC_COOKIE;
        version := MODULE_MAGIC_NUMBER_MAJOR;
        minor_version := MODULE_MAGIC_NUMBER_MINOR;
        module_index := -1;
        //dynamic_load_handle; // Apache will fill it with a handle
        create_dir_config := @createDirConfig; // directory config creator
        merge_dir_config := @mergeDirConfig; // dir config merger
        create_server_config := @createServerConfig; // server config creator
        merge_server_config := @mergeServerConfig; // server config merger
        cmds := @cmdRec; // command record
        register_hooks := @registerHooks; //
    end;

    InitCriticalSection(BrowscapIniLock);
    FilePath := ExtractFilePath(ModuleFileName());
    Ini := TIniFile.Create(FilePath + ChangeFileExt(MODULE_NAME, '.ini'));
    BrowscapIniPath := Ini.ReadString('browscap', 'path', FilePath +
'browscap.ini');
    debugMode := Ini.ReadBool('browscap', 'debug', false);
    BrowscapIni := TBrowscap.Create(BrowscapIniPath);
    FieldList := TStringList.Create;
    Ini.readSectionValues('fields', FieldList);
    Ini.Free;
finalization
    FieldList.Free;
    BrowscapIni.Free;
    DoneCriticalSection(BrowscapIniLock);
end.

```

Autor: Udo Schmal, veröffentlicht: 09.05.2014, letzte Änderung: 29.08.2024

© Copyright 2024 Udo Schmal