ISAPI-Interface Example

ISAPI Extension „Hello World“

Da ich ein herangehen ohne Objektorientierung als wenig sinnvoll erachte, hier nur ein kleines Beispiel zur Anwendung der Interface-Datei, das Simpelste ist immer „Hello World“, es folgt also das Beispiel (läuft natürlich auf dem IIS und dem Apache).

Die folgenden Schritte werden ausgeführt, wenn der Webserver einen Request (Anfrage) erhält, die einer ISAPI Extension zugeordnet werden kann

1. Der Webserver lädt die DLL, falls sie sich nicht schon im Speicher befindet. Wenn die DLL geladen wurde, ruft Windows automatisch die DLL Haupt-Routine auf, die das Application-Object erzeugt. Der IIS ruft dann die Versionsabfrage (GetExtensionVersion) der Extension auf, nun ist der Webserver auf Anfragen vorbereitet.
2. Der IIS macht nur minimale vorarbeiten für einen Request (Abfrage), er erzeugt einen Extension Control Block und füllt ihn mit allen Werten, die für den Prozess erforderlich sind.
3. Der Extension Control Block wird der ISAPI-Application durch Aufruf der Funktion HttpExtensionProc zur Verfügung gestellt.
4. Die ISAPI Extension bearbeitet den Request (Abfrage).
5. Die ISAPI Extension informiert den Webserver, dass die Bearbeitung abgeschlossen ist durch Beendigung des Prozesses und Rückgabe eines Statuscodes.
6. Der Webserver räumt den durch die Abfrage des belegten Speicherbereichs auf.
7. Falls die ISAPI Extension nicht länger benötigt wird, z.B. wenn längere Zeit keine Abfrage auf den Webservice eingeht, beendet der Webserver über den Aufruf der Funktion TerminateExtension die ISAPI-Application.

helloworld.lpr Pascal (2 kByte) 24.05.2012 22:03

```
library helloworld;
{$ifdef fpc}
  {$mode objfpc}{$H+}
{$endif}

uses SysUtils, Classes, ISAPI;

function GetExtensionVersion(var pVer: THSE_VERSION_INFO): BOOL; stdcall;
begin
  pVer.dwExtensionVersion := LONG((WORD(HSE_VERSION_MINOR)) or
((DWORD(WORD(HSE_VERSION_MAJOR))) shl 16));
  pVer.lpszExtensionDesc := 'Simple ISAPI DLL' + #0;
  result := true;
end;

function TerminateExtension(dwFlags: DWORD): BOOL; stdcall;
begin
  // This is so that the Apache web server will know what "True" really is
  Integer(result) := 1;
```

```
end;

function HttpExtensionProc(var pECB: TEXTENSION_CONTROL_BLOCK): DWORD;
stdcall;
var
  HTTPStatusCode: integer;
  dwLen: Cardinal;
  sResponse: string;
begin
  HTTPStatusCode := 200;
  result := HSE_STATUS_ERROR;
  sResponse := '<?xml version="1.0" encoding="utf-8"?>'#13#10 +
               '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">'#13#10 +
               '<html xmlns="http://www.w3.org/1999/xhtml" lang="de"
xml:lang="de">'#13#10 +
               '  <head>'#13#10 +
               '    <meta http-equiv="content-type" content="text/html;
charset=utf-8" />'#13#10 +
               '    <meta http-equiv="content-language" content="de"
/>'#13#10 +
               '    <title>Hallo Welt</title>'#13#10 +
               '  </head>'#13#10 +
               '  <body>'#13#10 +
               '    <p>Hallo Welt!</p>'#13#10 +
               '  </body>'#13#10 +
               '</html>';
  sResponse := 'HTTP/1.1 '+ IntToStr(HTTPStatusCode) + #13#10 +
               'Content-Type: text/html' + #13#10 +
               'Content-Length: ' + Format('%d', [Length(sResponse)]) +
#13#10 +
               'Content:'#13#10#13#10 + sResponse;

  pECB.dwHTTPStatusCode := HTTPStatusCode;
  dwLen := Length(sResponse);
  pECB.WriteClient(pECB.ConnID, Pointer(sResponse), dwLen, 0);
  Result := HSE_STATUS_SUCCESS;
end;

exports GetExtensionVersion, HttpExtensionProc, TerminateExtension;

end.
```

Autor: Udo Schmal, veröffentlicht: 30.05.2012, letzte Änderung: 29.08.2024

© Copyright 2024 Udo Schmal