

Erstellen eines Apache Modules

Beispiel Module mod_hello.so

Falls Sie einen Apache-Webserver (den gibt es auch als portable Version in XAMPP Paket www.apachefriends.org/de/index.html (<https://www.apachefriends.org/de/index.html>)) verwenden wollen, dann existiert unter Free Pascal schon eine Möglichkeit zum erstellen von Apache-Modulen.

Hier nun ein sehr einfaches Beispiel:

 [hello_mod.pas](https://code.hello_mod.pas?mode=download) (/code/hello_mod.pas?mode=download) Pascal (3,56 kByte) 17.03.2013 20:04

```
library mod_hello;

// The mode must be objfpc on this unit because the unix code uses
// some extensions introduced on Free Pascal

//insert into httpd.conf (Windows: xampp\apache\conf\ Linux:
/opt/lampp/etc/)
// LoadModule hello_module modules/mod_hello.so
// <Location /pmod>
//   SetHandler hello-handler
// </Location>

{$ifdef fpc}
    {$mode objfpc}{$H+}
{$endif}

{$IFDEF WIN32}
    {$DEFINE WINDOWS}
{$ENDIF}

{$define Apache2_2}

uses SysUtils, httpd {$ifndef Apache1_3}, apr{$endif};

var
    hello_module: module; public name 'hello_module';
    default_module_ptr: Pmodule;

const
    MODULE_NAME = 'mod_hello.so';

// Free Pascal only supports exporting variables on Windows
exports
    hello_module name 'hello_module';

// Handles apache requests
function DefaultHandler(r: Prequest_rec): Integer; cdecl;
var
    RequestedHandler: string;
begin
```

```

RequestedHandler := r^.handler;

// We decline to handle a request if hello-handler is not the value of r-
>handler
if not SameText(RequestedHandler, 'hello-handler') then
begin
    Result := DECLINED;
    Exit;
end;

// The following line just prints a message to the errorlog
// ap_log_error(MODULE_NAME, 54, APLOG_NOERRNO or APLOG_NOTICE,
//     {$ifndef Apache1_3}0,{$endif} r^.server,
//     'mod_hello: %s', [PChar('Before content is output')]);

// We set the content type before doing anything else
{$ifdef Apache1_3}
    r^.content_type := 'text/html';
//     ap_send_http_header(r);
{$else}
    ap_set_content_type(r, 'text/html');
{$endif}

// If the request is for a header only, and not a request for the whole
content,
// then return OK now. We don't have to do anything else.
if (r^.header_only <> 0) then
begin
    Result := OK;
    Exit;
end;

// Now we just print the contents of the document using the ap_rputs and
// ap_rprintf functions. More information about the use of these can be
found
// in http_protocol.inc }
ap_rputs('<HTML>' + LineEnding, r);
ap_rputs('<HEAD>' + LineEnding, r);
ap_rputs('<TITLE>Hello There</TITLE>' + LineEnding, r);
ap_rputs('</HEAD>' + LineEnding, r);
ap_rputs('<BODY BGCOLOR="#FFFFFF">' + LineEnding, r);
ap_rputs('<H1>Hello world</H1>' + LineEnding, r);
ap_rputs('This is the first Apache Module working with the new binding
from Free Pascal' + LineEnding, r);
// ap_rprintf(r, '<br>A sample line generated by ap_rprintf<br>' +
LineEnding, []);
ap_rputs('</BODY></HTML>' + LineEnding, r);

// We can either return OK or DECLINED at this point. If we return
// * OK, then no other modules will attempt to process this request

```

```

    Result := OK;
end;

// Registers the hooks
{$ifdef apache1_3}

procedure hw_init(s: PServer_rec; p: PPool); cdecl;
begin
end;

var
    hw_handlers: array[0..0] of handler_rec = (
        (content_type: 'hw_app'; handler: @DefaultHandler)
    );

{$else}

procedure RegisterHooks(p: Papr_pool_t); cdecl;
begin
    ap_hook_handler(@DefaultHandler, nil, nil, APR_HOOK_MIDDLE);
end;

{$endif}

// Library initialization code

{$ifdef WINDOWS}{$R *.res}{$endif}

begin
    default_module_ptr := @hello_module;
    FillChar(default_module_ptr^, SizeOf(default_module_ptr^), 0);

    {$ifdef apache1_3}
        STANDARD_MODULE_STUFF(hello_module);

        with hello_module do
            begin
                name := MODULE_NAME;
                init := @hw_init;
                handlers := hw_handlers;
            end;
    {$else}
        STANDARD20_MODULE_STUFF(hello_module);

        with hello_module do
            begin
                name := MODULE_NAME;
                register_hooks := @RegisterHooks;
            end;
    {$endif}

```

end.

Apache Configuration

In der Datei httpd.conf (Windows: xampp\apache\conf\ Linux: /opt/lampp/etc/)

```
LoadModule hello_module modules/mod_hello.so
<Location /pmod>
    SetHandler hello-handler
</Location>
```

Autor: [Udo Schmal \(#udo.schmal\)](#), veröffentlicht: 25.02.2013, letzte Änderung: 06.09.2023

© Copyright 2023 Udo Schmal

(/)