Mod_rewrite

@dataf3l 2017

About the Speaker

- Felipe
- Reach me at dataf5l@gmail.com
- Just another PHP Programmer

Eso qué es y con qué se come?

- Es un módulo de Apache que ayuda con urls limpias
- Circa 1998
- Donado inicialmente por Ralf S. Engelschall

Ralf S. Engelschall


RTFM


RTFM

Más trabajo

- **References**
- In his role as a *Software Artist* and *Hacker* Ralf S. Engelschall made the following major achievements: founder and developer at OSSP (software components)
- founder and developer at OpenSSL (cryptography toolkit)
- founder and developer at OpenPKG (software distribution)
- founder and developer at RPM5 (packaging tool)
- developer of Apache mod_ssl (HTTP security)
- developer of Apache mod_rewrite (URL manipulation)
- developer of GNU pth (multi-threading facility)
- developer of GNU shtool (development tool)
- developer of WML (webdesign tool)
- developer at FreeBSD (operating system)
- contributor at SQLite (RDBMS)
- contributor at CVSTrac (version control frontend)
- contributor at Monotone (version control system)
- contributor at jQuery (DHTML/AJAX library)
- See also the resume on his major Open Source Software community efforts for additional details.

Cool URIs don't change

- **We just reorganized our website to make it better.**
- **We have so much material that we can't keep track of what is out of date and what is confidential and what is valid and so we thought we'd better just turn the whole lot off.**
- **Well, we found we had to move the files...**
- **We used to use a cgi script for this and now we use a binary program.**
- **I didn't think URLs have to be persistent - that was URNs.**
- **We would like to, but we just don't have the right tools.**

Hall of flame -- story 2: Microsoft Netmeeting

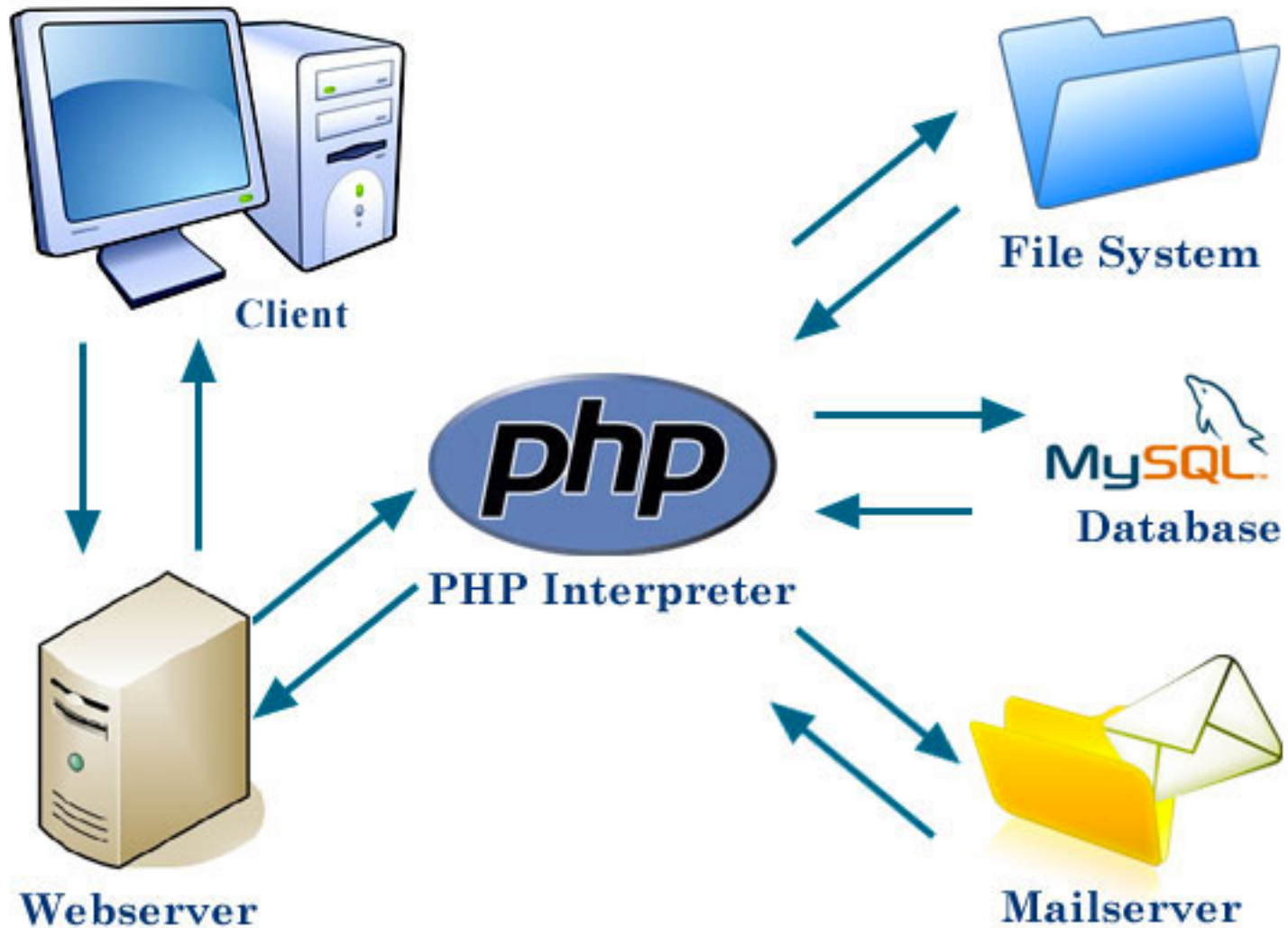
- One of the smarts which came with a growing dependency on the web was that applications could have built-in links back to the manufacturer's web site. This has been used and abused to a great extent, but - you do have to keep the URL the same. Just the other day I tried a link from Microsoft's Netmeeting 2/something client under a menu "Help/Microsoft on the Web/Free stuff" and got an Error 404 - not found response from the server. They have probably fixed it by now...

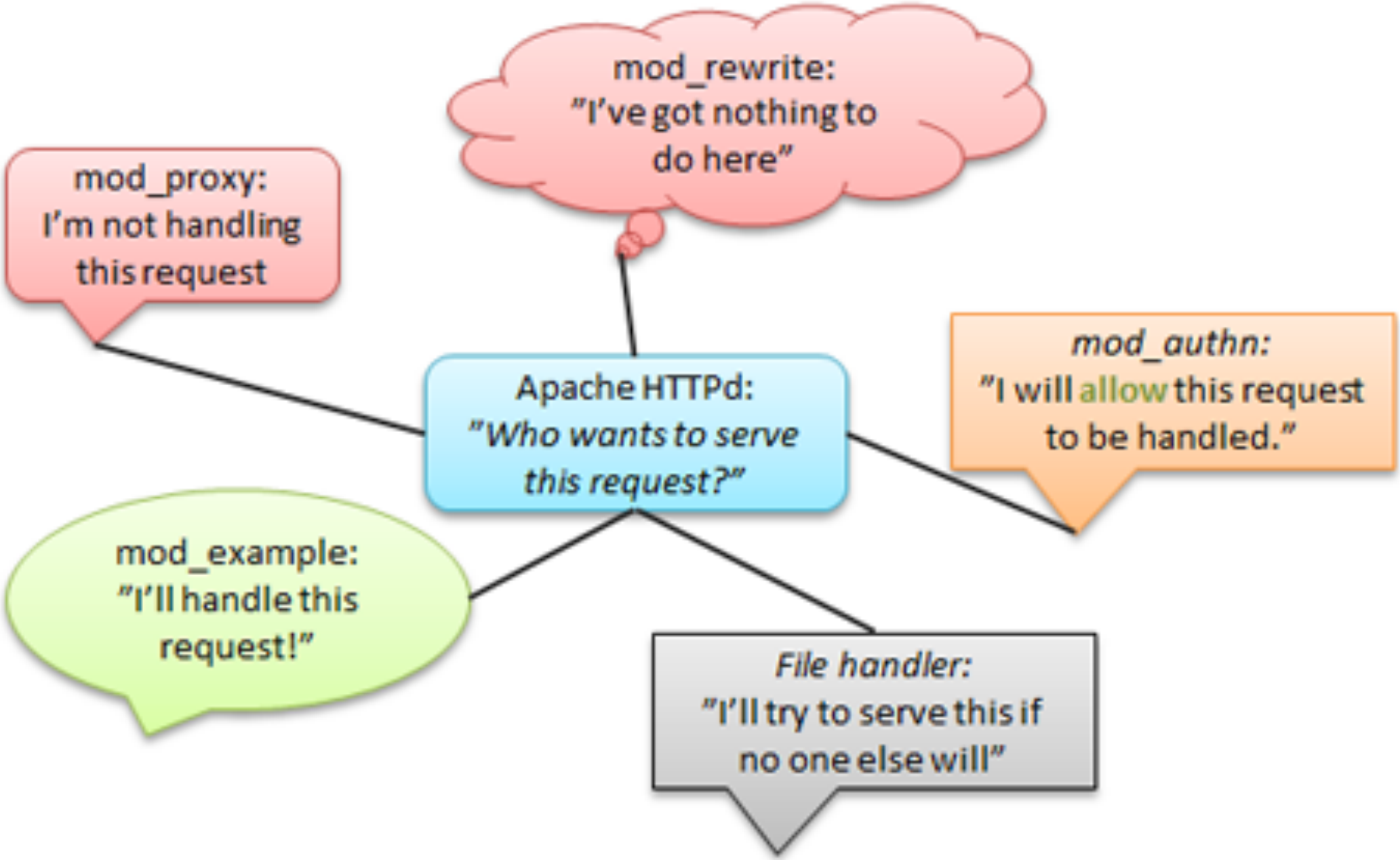
Rants

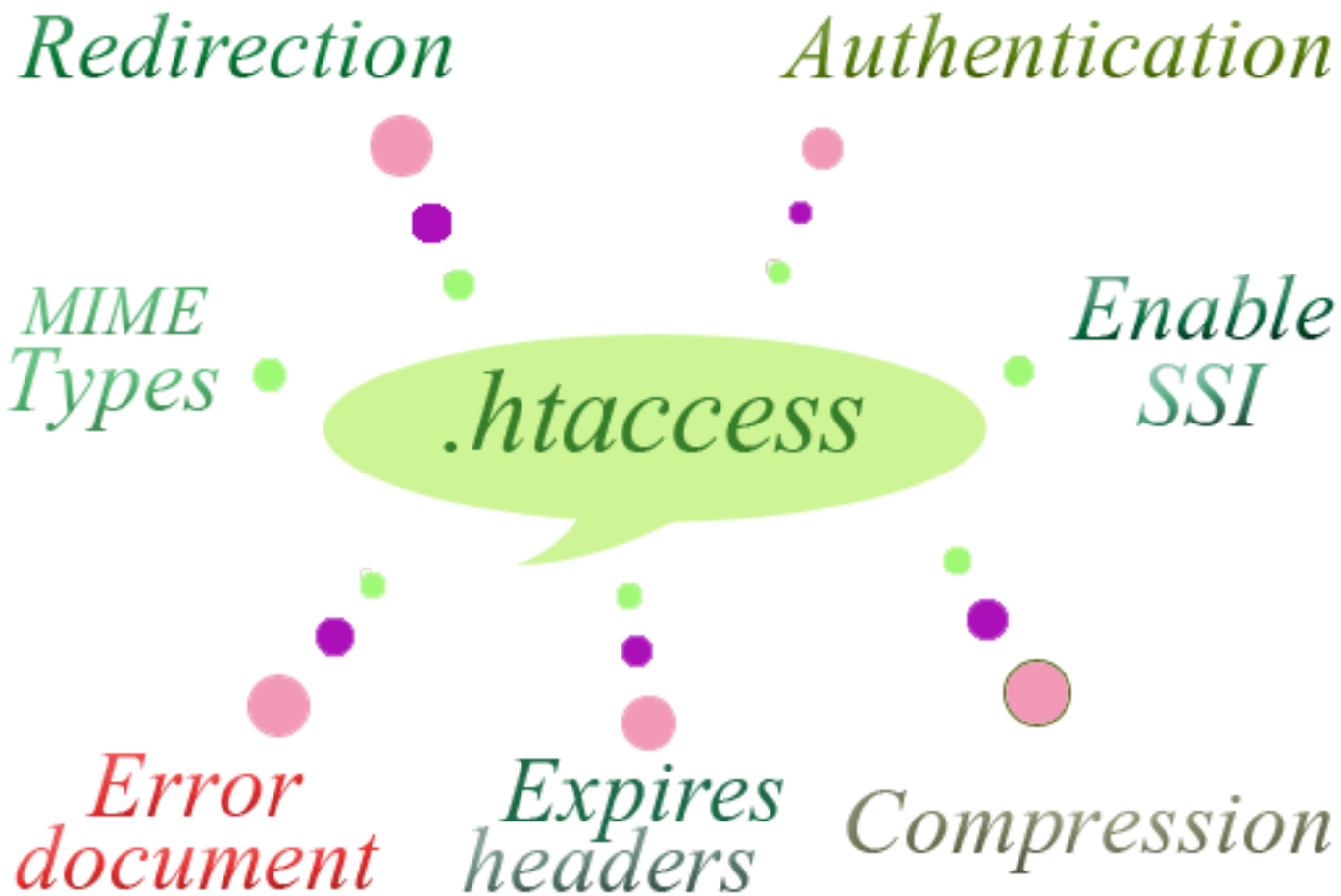
- mldk: Aargh! .htaccess and mod_rewrite can be such a pain in the ---!
- bsterzenbach: Man do I love mod_rewrite. I could work with it the rest of my life and still not master it - so powerful
- mikemackay: Still loving the total flexibility of mod_rewrite - coming to the rescue again. Often so overlooked…and easier than you might think too!
- hostpc: I hate mod_rewrite. Can’t get this dang application to work properly :(
- awanderingmind: Oh Wordpress and Apache, how thou dost vex me. Mod_rewrite be damned!
- danielishiding: Why won’t mod_rewrite work! Damn it!

Rants

- Despite the tons of examples and docs, mod_rewrite is voodoo. Damned cool voodoo, but still voodoo.” — Brian Moore


Client
File System
php
PHP Interpreter
MySQL
Database
Webserver
Mailserver


mod_rewrite:
"I've got nothing to do here"
mod_proxy:
I'm not handling this request
Apache HTTPd:
"Who wants to serve this request?"
mod_authn:
"I will allow this request to be handled."
mod_example:
"I'll handle this request!"
File handler:
"I'll try to serve this if no one else will"


Redirection
Authentication
MIME Types
.htaccess
Enable SSI
Error document
Expires headers
Compression

.htaccess

- .htaccess is a configuration file for use on web servers running the Apache Web Server software. When a .htaccess file is placed in a directory which is in turn 'loaded via the Apache Web Server', then the .htaccess file is detected and executed by the Apache Web Server software.

.htaccess

- .htaccess is a configuration file for use on web servers running the Apache Web Server software. When a .htaccess file is placed in a directory which is in turn 'loaded via the Apache Web Server', then the .htaccess file is detected and executed by the Apache Web Server software.

Basics

- http://www.pets.com/pet_care_info_07_07_2008.php
- http://www.pets.com/pet-care/

Basics

- RewriteEngine On
- # Turn on the rewriting engine
- RewriteRule ^pet-care/?$ pet_care_info_01_02_2008.php [NC,L]
- # Handle requests for "pet-care"

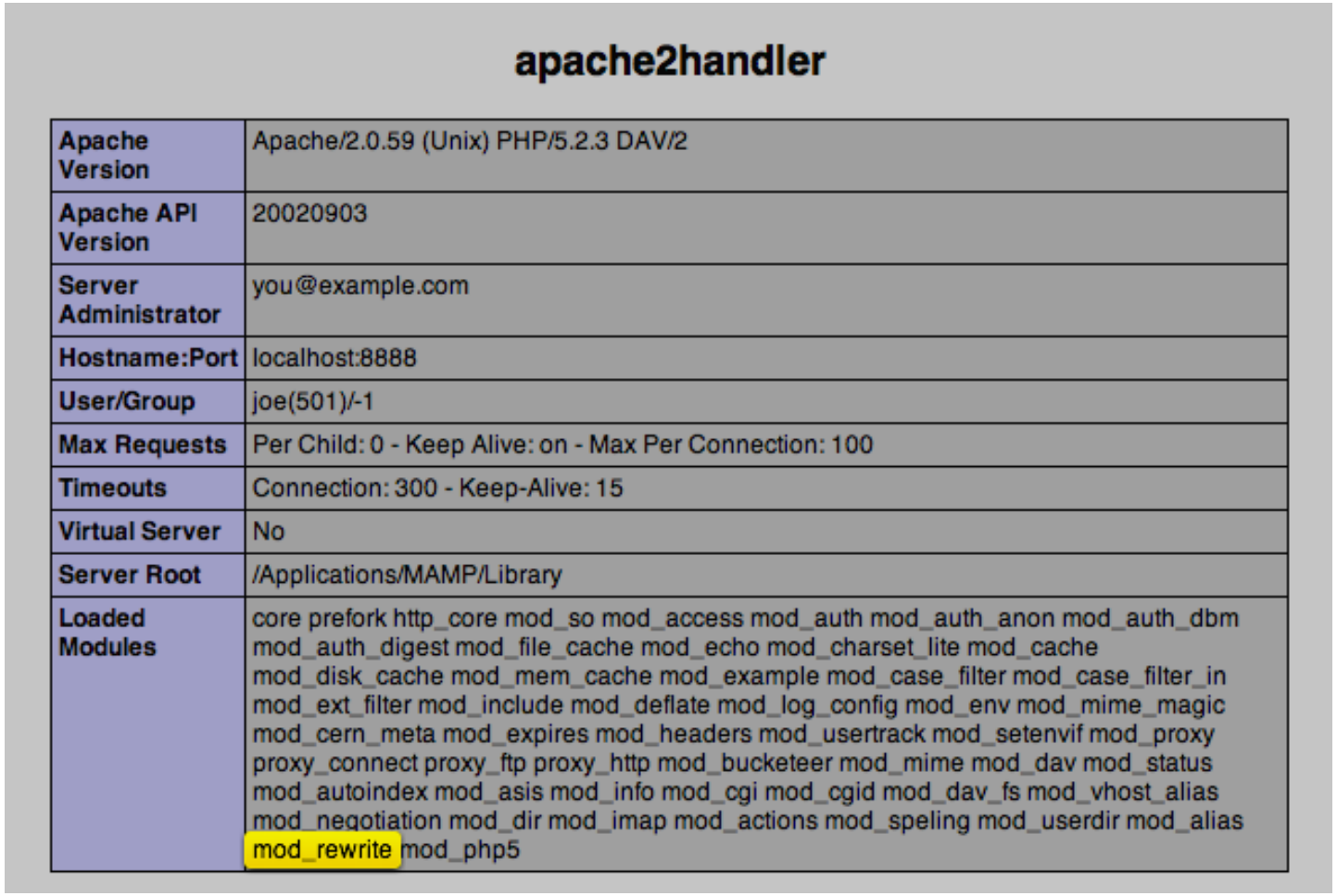
Installation

- LoadModule rewrite_module modules/mod_rewrite.so

- sudo a2enmod rewrite

Installation Testing

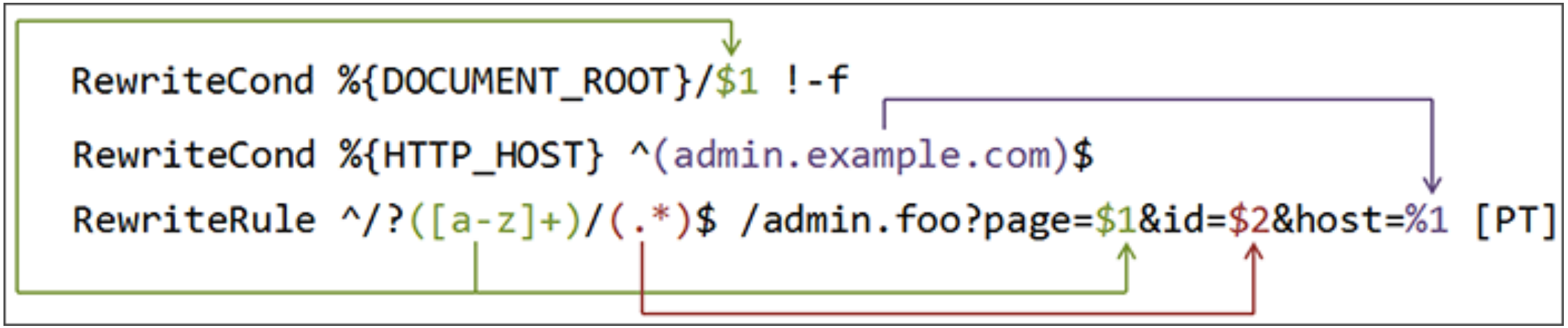
- <?php phpinfo(); ?>



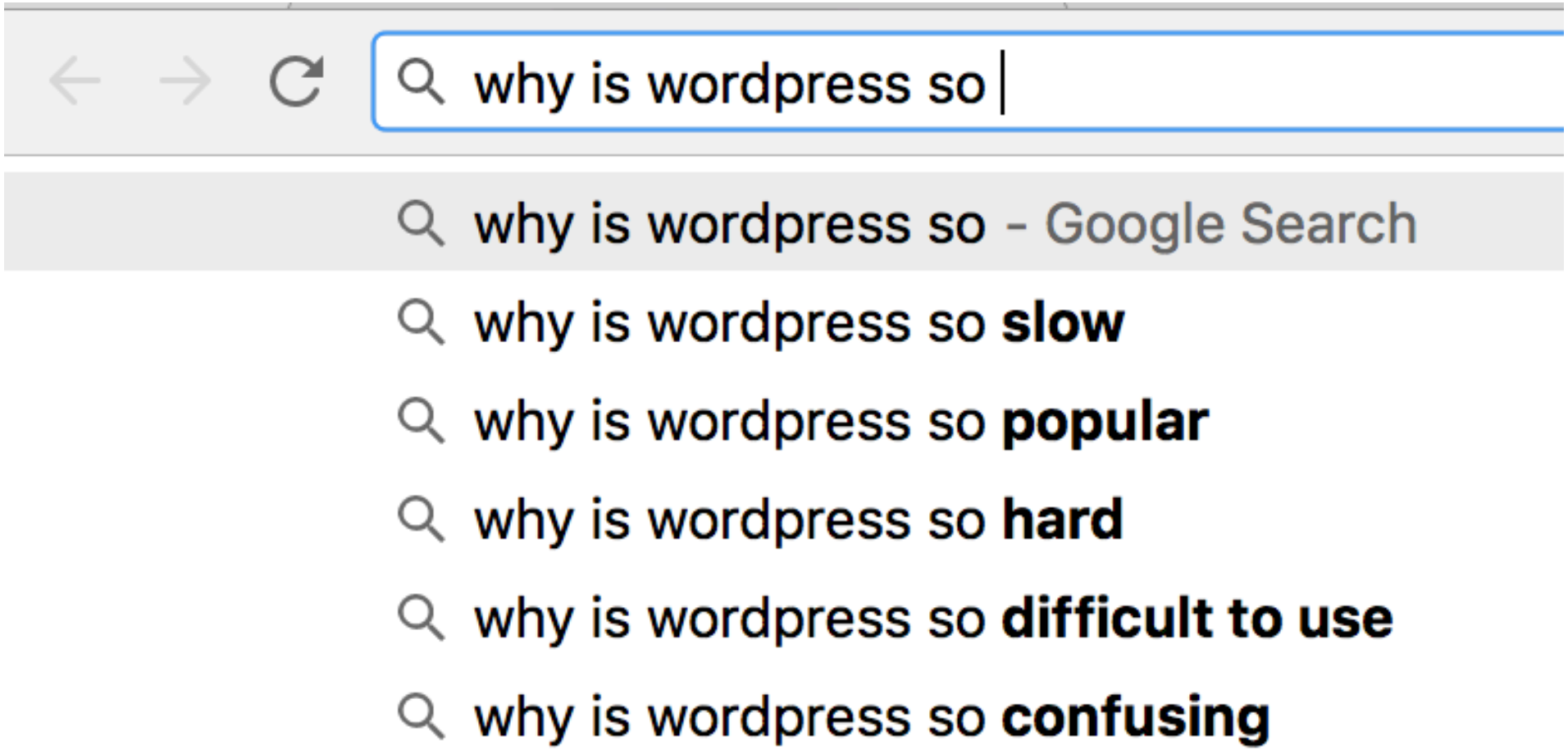
apache2handler

Apache Version	Apache/2.0.59 (Unix) PHP/5.2.3 DAV/2
Apache API Version	20020903
Server Administrator	you@example.com
Hostname:Port	localhost:8888
User/Group	joe(501)/-1
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 15
Virtual Server	No
Server Root	/Applications/MAMP/Library
Loaded Modules	core prefork http_core mod_so mod_access mod_auth mod_auth_anon mod_auth_dbm mod_auth_digest mod_file_cache mod_echo mod_charset_lite mod_cache mod_disk_cache mod_mem_cache mod_example mod_case_filter mod_case_filter_in mod_ext_filter mod_include mod_deflate mod_log_config mod_env mod_mime_magic mod_cern_meta mod_expires mod_headers mod_usertrack mod_setenvif mod_proxy proxy_connect proxy_ftp proxy_http mod_bucketeer mod_mime mod_dav mod_status mod_autoindex mod_asis mod_info mod_cgi mod_cgid mod_dav_fs mod_vhost_alias mod_negotiation mod_dir mod_imap mod_actions mod_speling mod_userdir mod_alias mod_rewrite mod_php5

Basics

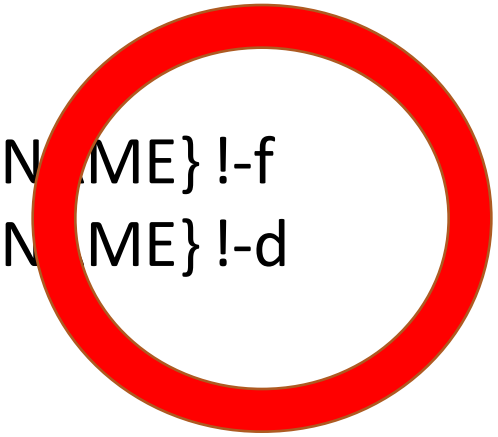

RewriteCond %{DOCUMENT_ROOT}/$1 !-f
RewriteCond %{HTTP_HOST} ^(admin.example.com)$
RewriteRule ^/?([a-z]+)/(.*)$ /admin.foo?page=$1&id=$2&host=%1 [PT]

Wordpress Example



.htaccess

- # BEGIN WordPress
- <IfModule mod_rewrite.c>
- RewriteEngine On
- RewriteBase /
- RewriteRule ^index\.php$ - [L]
- RewriteCond %{REQUEST_FILEN ME} !-f
- RewriteCond %{REQUEST_FILEN ME} !-d
- RewriteRule . /index.php [L]
- </IfModule>
- # END WordPress



Why…

.htaccess

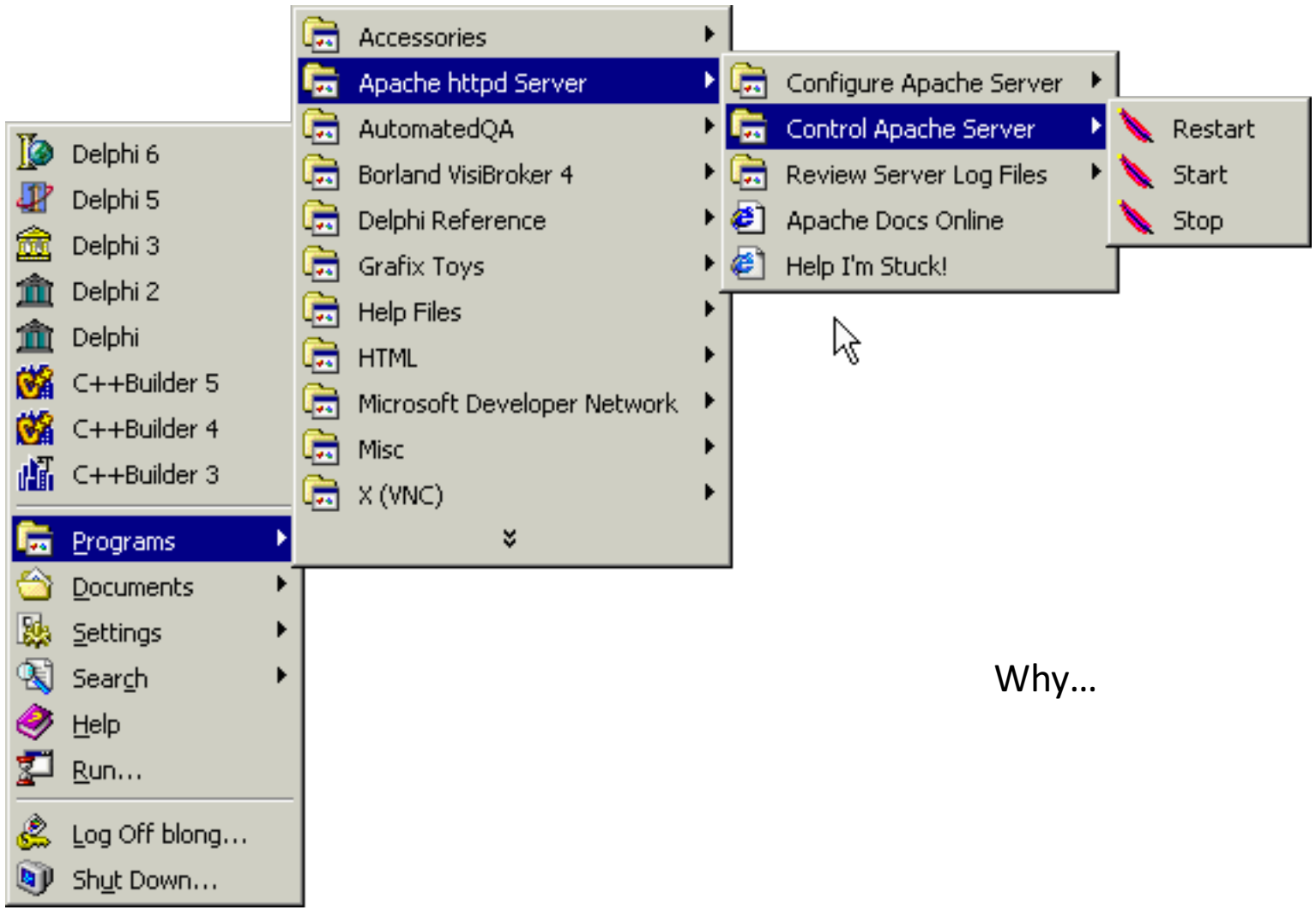
- # BEGIN WordPress
- <IfModule mod_rewrite.c>
- RewriteEngine On
- RewriteBase /
- RewriteRule ^index\.php$ - [L]
- RewriteCond %{REQUEST_FILENAME} !-f
- RewriteCond %{REQUEST_FILENAME} !-d
- RewriteRule . /index.php [L]
- </IfModule>
- # END WordPress

RewriteEngine On/Off

- Shutdown on subfolder
- Don’t take more resources than you need

RewriteEngine On/Off

- Shutdown on subfolder
- Don’t take more resources than you need


Delphi 6
Delphi 5
Delphi 3
Delphi 2
Delphi
C++Builder 5
C++Builder 4
C++Builder 3
Programs
Documents
Settings
Search
Help
Run...
Log Off blong...
Shut Down...
Accessories
Apache httpd Server
AutomatedQA
Borland VisiBroker 4
Delphi Reference
Grafix Toys
Help Files
HTML
Microsoft Developer Network
Misc
X (VNC)
Configure Apache Server
Control Apache Server
Review Server Log Files
Apache Docs Online
Help I'm Stuck!
Restart
Start
Stop

Why…

.htaccess

- # BEGIN WordPress
- <IfModule mod_rewrite.c>
- RewriteEngine On
- RewriteBase /
- RewriteRule ^index\.php$ - [L]
- RewriteCond %{REQUEST_FILENAME} !-f
- RewriteCond %{REQUEST_FILENAME} !-d
- RewriteRule . /index.php [L]
- </IfModule>
- # END WordPress

Why…

Syntax of a RewriteRule:

Regular Expression checked against the requested URI, which is the part after http://hostname

Optional:
All kinds of special actions: Define variables, Control headers, Redirect, Deny...

```
RewriteRule Pattern Substitution [Flags]
```

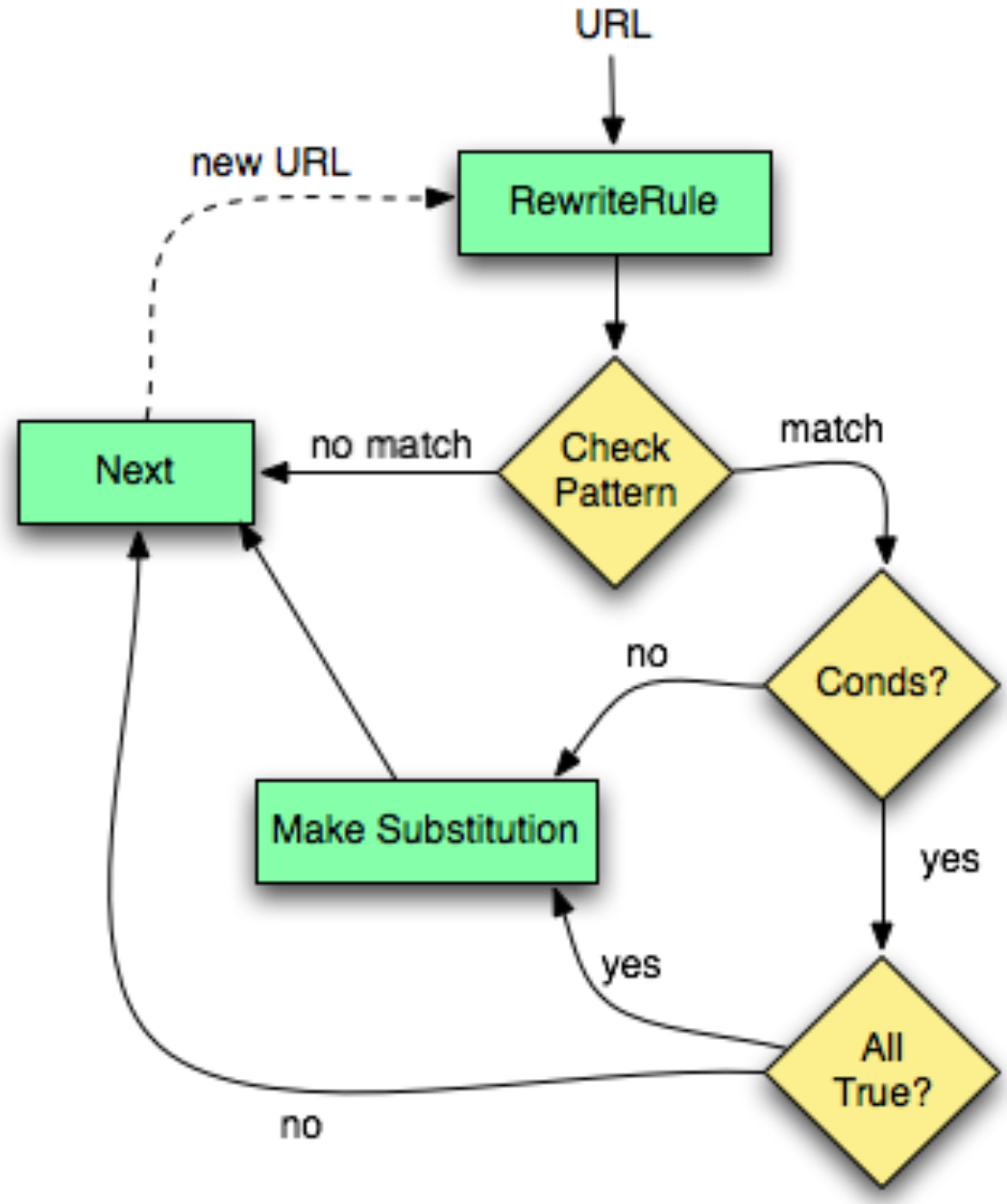
One of the following:
1. Modification to the part matched by the Pattern
2. Absolute path to a file
3. Full URL to redirect to
4. A dash "-" to do nothing

RewriteRule

```
typedef struct {
    apr_array_header_t *rewriteconds;/* the corresponding RewriteCond entries */
    char      *pattern;              /* the RegExp pattern string             */
    ap_regex_t *regexp;              /* the RegExp pattern compilation        */
    char      *output;               /* the Substitution string               */
    int        flags;                /* Flags which control the substitution  */
    char      *forced_mimetype;      /* forced MIME type of substitution      */
    char      *forced_handler;       /* forced content handler of subst.      */
    int        forced_responsecode;  /* forced HTTP response status           */
    data_item *env;                  /* added environment variables           */
    data_item *cookie;               /* added cookies                         */
    int        skip;                 /* number of next rules to skip          */
    int        maxrounds;            /* limit on number of loops with N flag  */
    char       *escapes;             /* specific backref escapes              */
} rewriterule_entry;
```

RewriteRule

```
typedef struct {
    apr_array_header_t *rewriteconds;/* the corresponding RewriteCond entries */
    char      *pattern;              /* the RegExp pattern string             */
    ap_regex_t *regexp;              /* the RegExp pattern compilation        */
    char      *output;               /* the Substitution string               */
    int        flags;                /* Flags which control the substitution  */
    char      *forced_mimetype;      /* forced MIME type of substitution      */
    char      *forced_handler;       /* forced content handler of subst.      */
    int        forced_responsecode;  /* forced HTTP response status           */
    data_item *env;                  /* added environment variables           */
    data_item *cookie;               /* added cookies                         */
    int        skip;                 /* number of next rules to skip          */
    int        maxrounds;            /* limit on number of loops with N flag  */
    char       *escapes;             /* specific backref escapes              */
} rewriterule_entry;
```


URL
new URL
RewriteRule
Next
no match
Check Pattern
match
no
Conds?
Make Substitution
yes
yes
All True?
no

```
# Enable Rewriting
RewriteEngine on

# Rewrite profile URLs
#   Input:  profile/NAME/
#   Output: profile.php?id=NAME
RewriteRule ^profile/(\w+)/?$ profile.php?id=$1

# Prevent Direct Access to profile.php
# THE_REQUEST is the whole original request,
# if the original request has "profile.php"
# send back a 403 Forbidden Warning
RewriteCond %{THE_REQUEST} profile\.php
RewriteRule ^profile\.php - [F]
```

.htaccess

- # BEGIN WordPress
- <IfModule mod_rewrite.c>
- RewriteEngine On
- RewriteBase /
- RewriteRule ^index\.php$ - [L]
- RewriteCond %{REQUEST_FILENAME} !-f
- RewriteCond %{REQUEST_FILENAME} !-d
- RewriteRule . /index.php [L]
- </IfModule>
- # END WordPress

Why…

Syntax of a RewriteCond:

Typically a Server Variable
which is of the form
%{VARIABLE}

Optional:
NC - Ignore Case
OR - Logical "or"
NV - No Vary

RewriteCond TestString Condition [Flags]

One of the following:
1. Regular Expression
2. String Comparison
3. File/Path Test

RewriteCond

```
typedef struct {
    char           *input;   /* Input string of RewriteCond   */
    char           *pattern; /* the RegExp pattern string     */
    ap_regex_t     *regexp;  /* the precompiled regexp        */
    ap_expr_info_t *expr;    /* the compiled ap_expr          */
    int             flags;   /* Flags which control the match */
    pattern_type    ptype;   /* pattern type                  */
    int             pskip;   /* back-index to display pattern */
} rewritecond_entry;
```

.htaccess

- # BEGIN WordPress
- <IfModule mod_rewrite.c>
- RewriteEngine On
- RewriteBase /
- RewriteRule ^ind
- RewriteCond %{REQUEST_FILENAME}
- RewriteCond %{REQUEST_FILENAME}
- RewriteRule .
- </IfModule>
- # END WordPress

Why…

Variables: %{VAR_NAME}

- API_VERSION
- AUTH_TYPE
- CONTENT_LENGTH
- CONTENT_TYPE
- DOCUMENT_ROOT
- GATEWAY_INTERFACE
- IS_SUBREQ
- ORIG_PATH_INFO
- ORIG_PATH_TRANSLATED
- ORIG_SCRIPT_FILENAME
- ORIG_SCRIPT_NAME

Variables: %{VAR_NAME}

- PATH
- PATH_INFO
- PHP_SELF
- QUERY_STRING
- REDIRECT_QUERY_STRING
- REDIRECT_REMOTE_USER
- REDIRECT_STATUS
- REDIRECT_URL
- REMOTE_ADDR
- REMOTE_HOST
- REMOTE_IDENT
- REMOTE_PORT
- REMOTE_USER

Variables: %{VAR_NAME}

- REQUEST_FILENAME
- REQUEST_METHOD
- REQUEST_TIME
- REQUEST_URI
- SCRIPT_FILENAME
- SCRIPT_GROUP
- SCRIPT_NAME
- SCRIPT_URI
- SCRIPT_URL
- SCRIPT_USER

Variables: %{VAR_NAME}

- SERVER_ADDR
- SERVER_ADMIN
- SERVER_NAME
- SERVER_PORT
- SERVER_PROTOCOL
- SERVER_SIGNATURE
- SERVER_SOFTWARE

Variables: %{VAR_NAME}

- THE_REQUEST
- TIME
- TIME_DAY
- TIME_HOUR
- TIME_MIN
- TIME_MON
- TIME_SEC
- TIME_WDAY
- TIME_YEAR
- TZ
- UNIQUE_ID

Variables: %{VAR_NAME}

- HTTP_ACCEPT
- HTTP_ACCEPT_CHARSET
- HTTP_ACCEPT_ENCODING
- HTTP_ACCEPT_LANGUAGE
- HTTP_CACHE_CONTROL
- HTTP_CONNECTION
- HTTP_COOKIE
- HTTP_FORWARDED
- HTTP_HOST
- HTTP_KEEP_ALIVE
- HTTP_PROXY_CONNECTION
- HTTP_REFERER
- HTTP_USER_AGENT

Variables: %{VAR_NAME}

- HTTPS
- SSL_CIPHER
- SSL_CIPHER_ALGKEYSIZE
- SSL_CIPHER_EXPORT
- SSL_CIPHER_USEKEYSIZE
- SSL_CLIENT_VERIFY
- SSL_PROTOCOL

Variable Handling

```
case 11:
        switch (var[8]) {
        case 'A':
          if (!strcmp(var, "SERVER_NAME"))  {
            result = ap_get_server_name_for_url(r);
          }
          break;

        case 'D':
          if (*var == 'R' && !strcmp(var, "REMOTE_ADDR")) {
            result = r->useragent_ip;
          }
          else if (!strcmp(var, "SERVER_ADDR")) {
            result = r->connection->local_ip;
          }
          break;
```

```
case 'E':
        if (*var == 'H' && !strcmp(var, "HTTP_ACCEPT")) {
          result = lookup_header("Accept", ctx);
        }
        else if (!strcmp(var, "THE_REQUEST"))  {
          result = r->the_request;
        }
        break;

      case 'I':
        if (!strcmp(var, "API_VERSION"))  {
          return apr_psprintf(r->pool, "%d:%d",
                              MODULE_MAGIC_NUMBER_MAJOR,
                              MODULE_MAGIC_NUMBER_MINOR);
        }
        break;
```

!-f ??

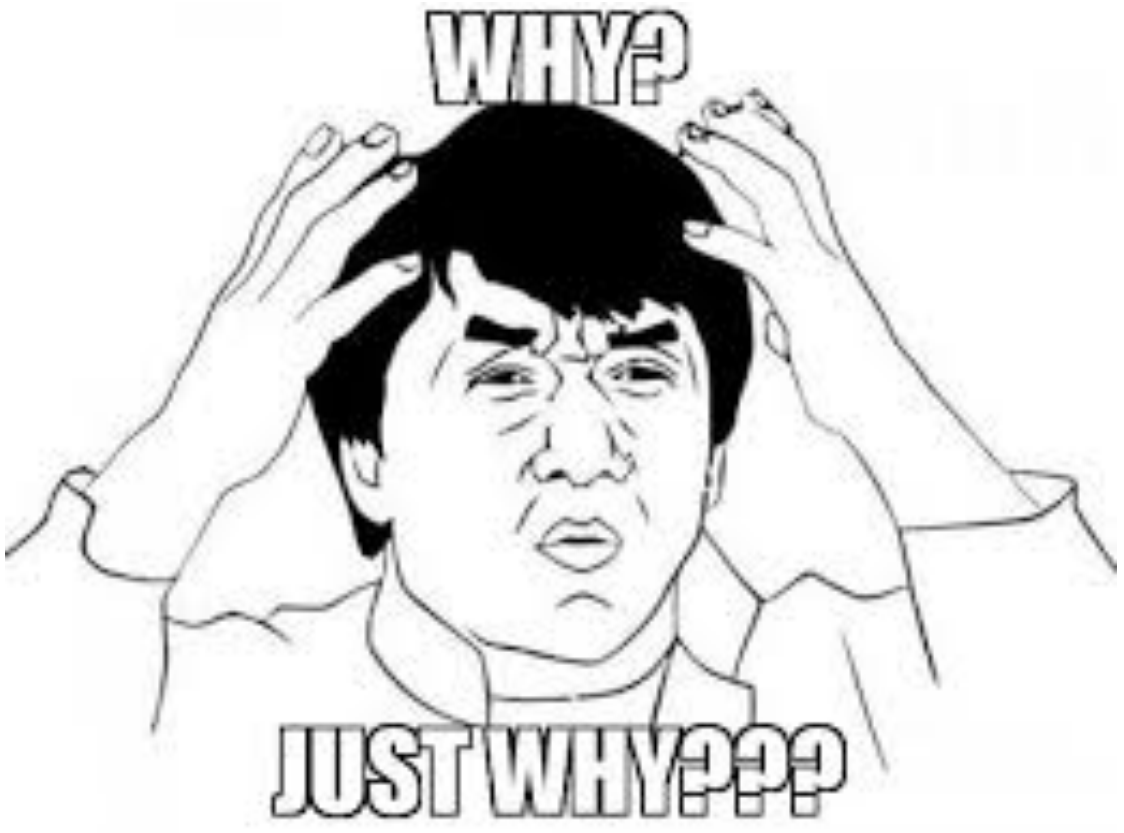
- !-f : no such file
- !-d: no such directory

!-f ??

- !-f : no such file
- !-d: no such directory

!-f ??

- !-f : no such file
- !-d: no such directory



Because bash, that’s why

- if [! -f /tmp/foo.txt];
- then
- 	echo "File not found!"
- fi

Bash File Testing

- -b filename - Block special file
  -c filename - Special character file
  -d directoryname - Check for directory Existence
  -e filename - Check for file existence, regardless of type
- (node, directory, socket, etc.)
  -f filename - Check for regular file existence not a directory
  -G filename - Check if file exists and is owned by effective group ID
  -G filename set-group-id - True if file exists and is set-group-id
  -k filename - Sticky bit
  -L filename - Symbolic link
  -O filename - True if file exists and is owned by the effective user id
  -r filename - Check if file is a readable
  -S filename - Check if file is socket
  -s filename - Check if file is nonzero size
  -u filename - Check if file set-user-id bit is set
  -w filename - Check if file is writable
  -x filename - Check if file is executable

Supported:

```
static const char *cmd_rewritecond(cmd_parms *cmd, void *in_dconf,
const char *in_str){
// . . .
                case 'f': newcond->ptype = CONDPAT_FILE_EXISTS; break;
                case 's': newcond->ptype = CONDPAT_FILE_SIZE;   break;
                case 'd': newcond->ptype = CONDPAT_FILE_DIR;    break;
                case 'x': newcond->ptype = CONDPAT_FILE_XBIT;   break;
                case 'h': newcond->ptype = CONDPAT_FILE_LINK;   break;
                case 'L': newcond->ptype = CONDPAT_FILE_LINK;   break;
                case 'l': newcond->ptype = CONDPAT_FILE_LINK;   break;
                case 'U': newcond->ptype = CONDPAT_LU_URL;      break;
                case 'F': newcond->ptype = CONDPAT_LU_FILE;     break;
```

Bash File Testing

- -b filename - Block special file
  -c filename - Special character file
  -d directoryname - Check for directory Existence
  -e filename - Check for file existence, regardless of type
- (node, directory, socket, etc.)
  -f filename - Check for regular file existence not a directory
  -G filename - Check if file exists and is owned by effective group ID
  -G filename set-group-id - True if file exists and is set-group-id
  -k filename - Sticky bit
  -L filename - Symbolic link
  -O filename - True if file exists and is owned by the effective user id
  -r filename - Check if file is a readable
  -S filename - Check if file is socket
  -s filename - Check if file is nonzero size
  -u filename - Check if file set-user-id bit is set
  -w filename - Check if file is writable
  -x filename - Check if file is executable

.htaccess

- # BEGIN WordPress
- <IfModule mod_rewrite.c>
- RewriteEngine On
- RewriteBase /
- RewriteRule ^index\.php$ - [L]
- RewriteCond %{REQUEST_FILENAME} !-f
- RewriteCond %{REQUEST_FILENAME} !-d
- RewriteRule . /index.php [L]
- </IfModule>
- # END WordPress

Why…

.htaccess

- # BEGIN WordPress
- <IfModule mod_rewrite.c>
- RewriteEngine On
- RewriteBase /
- RewriteRule ^index\.php$ - [L]
- RewriteCond %{REQUEST_FILENAME} !-f
- RewriteCond %{REQUEST_FILENAME} !-d
- RewriteRule . /index.php [L]
- </IfModule>
- # END WordPress

Why…

Rewrite Flags

- #define CONDFLAG_NONE (1<<0)
- #define CONDFLAG_NOCASE (1<<1)
- #define CONDFLAG_NOTMATCH (1<<2)
- #define CONDFLAG_ORNEXT (1<<3)
- #define CONDFLAG_NOVARY (1<<4)

- #define RULEFLAG_NONE (1<<0)
- #define RULEFLAG_FORCEREDIRECT (1<<1)
- #define RULEFLAG_LASTRULE (1<<2)
- #define RULEFLAG_NEWROUND (1<<3)
- #define RULEFLAG_CHAIN (1<<4)
- #define RULEFLAG_IGNOREONSUBREQ (1<<5)
- #define RULEFLAG_NOTMATCH (1<<6)
- #define RULEFLAG_PROXY (1<<7)
- #define RULEFLAG_PASSTHROUGH (1<<8)
- #define RULEFLAG_QSAPPEND (1<<9)
- #define RULEFLAG_NOCASE (1<<10)
- #define RULEFLAG_NOESCAPE (1<<11)
- #define RULEFLAG_NOSUB (1<<12)
- #define RULEFLAG_STATUS (1<<13)
- #define RULEFLAG_ESCAPEBACKREF (1<<14)
- #define RULEFLAG_DISCARDPATHINFO (1<<15)
- #define RULEFLAG_QSDISCARD (1<<16)
- #define RULEFLAG_END (1<<17)
- #define RULEFLAG_ESCAPENOPLUS (1<<18)
- #define RULEFLAG_QSLAST (1<<19)

Rewrite Flags: F|forbidden

- **RewriteRule** "\.exe" "-" [F]

Rewrite Flags: L|last

- **RewriteBase** "/"
- **RewriteCond** "%{REQUEST_URI}" "!=/index.php"
- **RewriteRule** "^(.*)" "/index.php?req=$1" [L,PT]

- The [L] flag causes mod_rewrite to stop processing the rule set. In most contexts, this means that if the rule matches, no further rules will be processed.

Rewrite Flags: N|next

- **RewriteRule** "(.*)A(.*)" "$1B$2" [N]

- # Be willing to replace 1 character in each pass of the loop
  RewriteRule "(.+)[><;]$" "$1" [N=64000]
- # ... or, give up if after 10 loops
- **RewriteRule** "(.+)[><;]$" "$1" [N=10]

Rewrite Flags: NC|nocase

- **RewriteRule** "(.*\.(jpg|gif|png))$" "http://images.example.com$1" [P,NC]
-

Rewrite Flags: P|proxy

- **RewriteRule** "/(.*)\.(jpg|gif|png)$" "http://images.example.com/$1.$2" [P]

- Use of the [P] flag implies [L] - that is, the request is immediately pushed through the proxy, and any following rules will not be considered.

Rewrite Flags: QSA|qsappend

- **RewriteRule** "/pages/(.+)" "/page.php?page=$1" [QSA]

- With the [QSA] flag, a request for /pages/123?one=two will be mapped to /page.php?page=123&one=two. Without the [QSA] flag, that same request will be mapped to /page.php?page=123 - that is, the existing query string will be discarded.

Rewrite Flags: R|redirect

- *Any* valid HTTP response status code may be specified, using the syntax [R=305], with a 302 status code being used by default if none is specified. The status code specified need not necessarily be a redirect (3xx) status code. However, if a status code is outside the redirect range (300-399) then the substitution string is dropped entirely, and rewriting is stopped as if the L were used.

- You will almost always want to use [R] in conjunction with [L] (that is, use [R,L]) because on its own, the [R] flag prepends http://thishost[:thisport]to the URI, but then passes this on to the next rule in the ruleset, which can often result in 'Invalid URI in request' warnings.
-

Rewrite Flags: S|skip

- # Is the request for a non-existent file?
- **RewriteCond** "%{REQUEST_FILENAME}" "!-f"
- **RewriteCond** "%{REQUEST_FILENAME}" "!-d"
- # If so, skip these two RewriteRules
- **RewriteRule** ".?" "-" [S=2]

- **RewriteRule** "(.*\.gif)" "images.php?$1"
- **RewriteRule** "(.*\.html)" "docs.php?$1"

Rewrite Flags: T|type

- # Serve .pl files as plain text
- **RewriteRule** "\.pl$" "-" [T=text/plain]

Rewrite Flags: E|env

- [E=VAR:VAL]
- [E=!VAR]

- **RewriteRule** "\.(png|gif|jpg)$" "-" [E=image:1]
- **CustomLog** "logs/access_log" combined env=!image

- Note that this same effect can be obtained using SetEnvIf. This technique is offered as an example, not as a recommendation.

Usage Examples

Mod_rewrite: Canonical URLs

- On some webservers there are more than one URL for a resource. Usually there are canonical URLs (which should be actually used and distributed) and those which are just shortcuts, internal ones, etc. Independent of which URL the user supplied with the request he should finally see the canonical one only.

Mod_rewrite Canonical URLs

- Solution:We do an external HTTP redirect for all non-canonical URLs to fix them in the location view of the Browser and for all subsequent requests. In the example ruleset below we replace /~user by the canonical /u/user and fix a missing trailing slash for /u/user.

- `RewriteRule ^/~([^/]+)/?(.*) /u/$1/$2 [R]`
- `RewriteRule ^/([uge])/([^/]+)$ /$1/$2/ [R]`

Mod_rewrite: Canonical Hostnames

- The goal of this rule is to force the use of a particular hostname, in preference to other hostnames which may be used to reach the same site. For example, if you wish to force the use of **www.example.com** instead of **example.com**, you might use a variant of the following recipe.

Mod_rewrite: Canonical Hostnames

```
# For sites running on a port other than 80
RewriteCond %{HTTP_HOST}   !^www\.example\.com [NC]
RewriteCond %{HTTP_HOST}   !^$
RewriteCond %{SERVER_PORT} !^80$
RewriteRule ^/(.*)         http://www.example.com:%{SERVER_PORT}/$1 [L,R]

# And for a site running on port 80
RewriteCond %{HTTP_HOST}   !^www\.example\.com [NC]
RewriteCond %{HTTP_HOST}   !^$
RewriteRule ^/(.*)         http://www.example.com/$1 [L,R]
```

Forbid Hotlinking

- # Give Hotlinkers a 403 Forbidden warning.
- RewriteEngine on
- RewriteCond %{HTTP_REFERER} !^http://example\.net/?.*$ [NC]
- RewriteCond %{HTTP_REFERER} !^http://example\.com/?.*$ [NC]
- RewriteRule \.(gif|jpe?g|png|bmp)$ - [F,NC]

Give Hotlinkers a Warning Image

- # Redirect Hotlinkers to "warning.png"
- RewriteEngine on
- RewriteCond %{HTTP_REFERER} !^http://example\.net/?.*$
- RewriteCond %{HTTP_REFERER} !^http://example\.com/?.*$ [NC]
- RewriteRule \.(gif|jpe?g|png|bmp)$ http://example.com/warning.png [R,NC]

Custom 404

- # Generic 404 to show the "custom_404.html" page
- # If the requested page is not a file or directory
- # Silent Redirect: the user's URL bar is unchanged.
- RewriteEngine on
- RewriteCond %{REQUEST_FILENAME} !-f
- RewriteCond %{REQUEST_FILENAME} !-d
- RewriteRule .* custom_404.html [L]

Custom 404

- # Generic 404 to show the "custom_404.html" page
- # If the requested page is not a file or directory
- # Silent Redirect: the user's URL bar is unchanged.
- RewriteEngine on
- RewriteCond %{REQUEST_FILENAME} !-f
- RewriteCond %{REQUEST_FILENAME} !-d
- RewriteRule .* custom_404.html [L]

Final look

- # BEGIN WordPress
- <IfModule mod_rewrite.c>
- RewriteEngine On
- RewriteBase /
- RewriteRule ^index\.php$ - [L]
- RewriteCond %{REQUEST_FILENAME} !-f
- RewriteCond %{REQUEST_FILENAME} !-d
- RewriteRule . /index.php [L]
- </IfModule>
- # END WordPress

- On "The Limits of My Language Mean the Limits of My World"
    - Ludwig Wittgenstein

Thanks to our sponsors :D