Date
24/02/2017

This rule matches anything (.*) and captures it ((..)), then rewrits it to append ".html". In other words, if /foo/bar was the requested URL, it will be handled as if /foo/bar.html had been requested. See (www.regular-expressions.info) for more information about regular expression matching, capturing and replacements.

Another often encountered rule is this:

```
RewriteRule (.*) index.php?url=$1
```

This, again, matches anything and rewrits it to the file index.php with the originally requested URL appended in the url query parameter i.e. for any and all requests coming in, the file index.php is executed and this file will have access to the original request in $_GET['url'], so it can do anything it wants with it.

What mod_rewrite does not do

mod-rewrite does not magically make all your URLs "preety". This is a common misunderstanding. If you have this link in your website:

```
<a href="/my/ugly/link.php?is=not&amp;very=preety">
```

There is nothing mod-rewrite can do to make that pretty. In order to make this a ~~preety~~ pretty link. You have to:

1. Change the link to a pretty link:

```
<a href="/my/pretty/link">
```

Date
24/02/2017

2. Use mod_rewrite on the server to handle the request to the URL /my/pretty/link using any one of the methods described above.

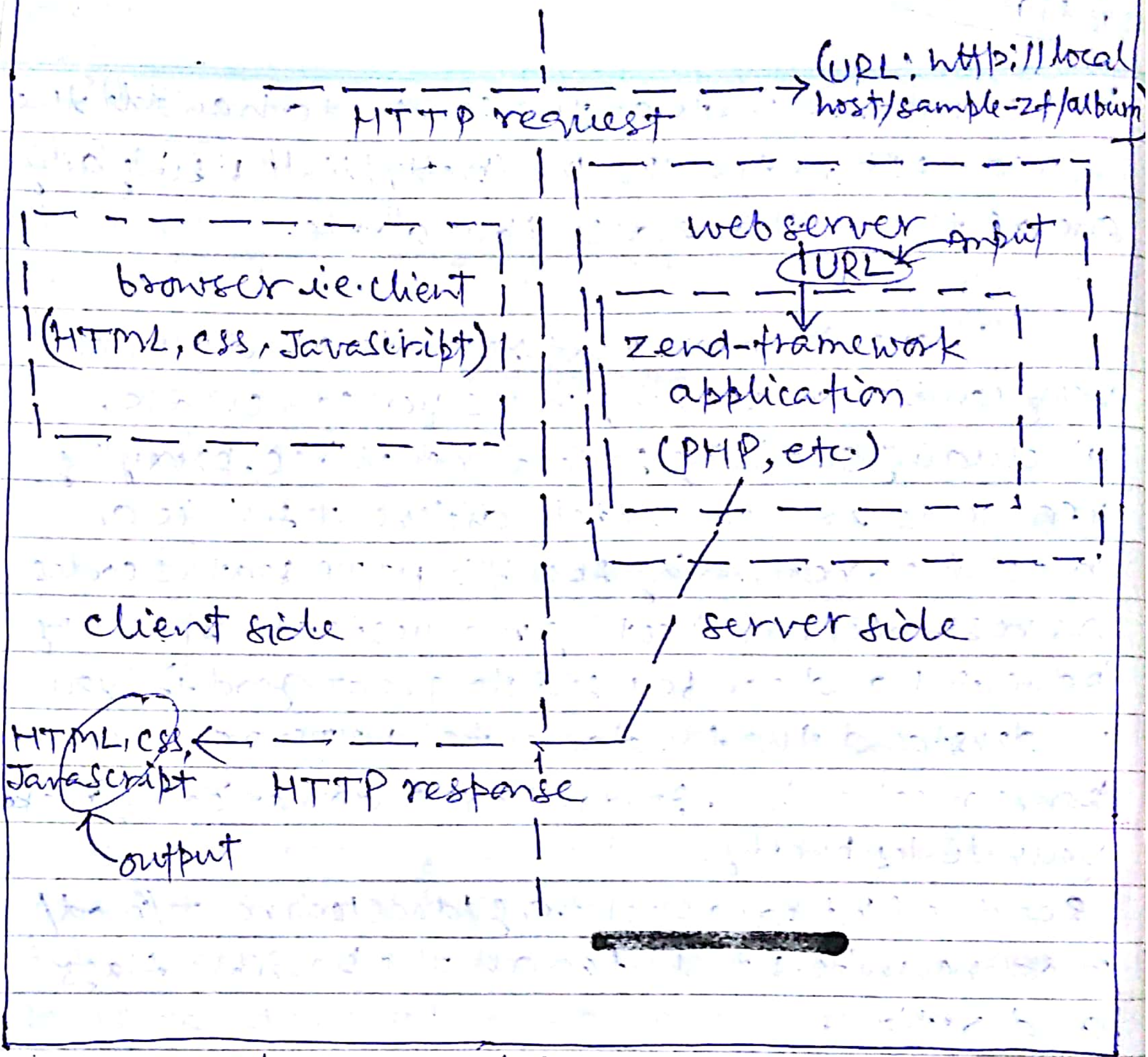
There is a lot mod_rewrite can do. and very complex matching rules you can create, including chaining several rewrites, proxying requests to a completely different service or machine, returning specific HTTP status codes as responses, redirecting requests etc. It is very powerful and can be used to great good if you understand the fundamental HTTP request-response mechanism. It does not automatically make your links pretty.

See the (httpd.apache.org/docs/current/mod/mod_rewrite.html) for all the possible flags and options.

Reference: stackoverflow.com/questions/20563772/reference-mod-rewrite-url-rewriting-and-pretty-links-explained

More rewrite example ⇒ see the 2nd answer on this stackoverflow page

Next page ⇒ web server waking Chart


HTTP request
(URL: http://local host/sample-zf/album)
web server
input
URL
zend-framework application (PHP, etc.)
browser i.e. client (HTML, css, Javascript)
client side
server side
HTML, css, Javascript
HTTP response
output

[redacted]

web server working chart

[redacted]

Date
25/02/2017

PHP: Pass by reference vs. Pass by value

```
<?php
function pass_by_value($param){
    ~~push_array~~($param, 3,4);
    array_push
}
$ar = array(1,2,3);
pass_by_value($ar);
foreach ($ar as $elem) {
    print "$elem ";
}
```

Output:

The code above prints 1 2 3. This is because the array is passed as value.

```
function pass_by_reference(&$param){
    array_push($param,4,5);
}
$ar = array(1,2,3);
pass_by_reference($ar);
foreach($ar as $elem) {
    print "$elem ";
}
```

Output!

The code above prints 1 2 3 4 5. This is because the array is passed as reference, meaning that the function pass_by_reference() does not manipulate a copy of the variable passed, but the actual variable itself.

In order to make a variable be passed by reference, It must be declared with a preceeding ampersand (&) in the function's declaration.

Date
25/02/2017

int sleep(int $seconds)

Delays/Halt the program execution for the given number of seconds. Return zero on success and FALSE on error.

Example:

```
// current time
echo date('H:i:s').'<br>';
// sleep for 10 secands
~~echo~~ sleep(10);
//wake up!
echo date('H:i:s');
```

output explanation is very important

Output (This example will produce output after 30 sec.

```
05:31:23
05:31:33
```

⇒ Both will be displayed exactly after 10 secands. It will not that firstly 05:31:23 displays & then after 10 secands 05:31:33 will be displayed.

As we know how a web server work. The web server firstly collect the HTTP response by executing the whole PHP code on server side in one go & then send the HTTP response to the client (side) in form of HTML, CSS; JavaScript
(browser)

where firstly javascript execute/run & then HTML ~~interpretation~~ interpretation starts/loads by browser.