IMPORTANT APACHE EXPRESSIONS

.htaccess Regex Character Definitions

The following table showcases the different regex characters that are available to you when dealing with a .htaccess file.

Regex Character Explanation

Forces the server to ignore the text following the # on the same line. Typically used for comments

[F] Indicates Forbidden, with this the server should return a 403 forbidden error to the client

[L] The Last rule forces the server to stop processing rules in the .htaccess file

[N] Indicates Next and forces Apache to redo the rewrite process, except using the currently rewritten URL instead o f the initial URL

[G] Gone tells the server to deliever the gone status message, which is used to mark pages that no longer exist on the site.

[R] This forces Apache to initialize a redirect, this can be a permanent redirect (page has moved, 301), or a temporar y redirect (302).

[P] Indicates Proxy which tells the server to use mod_proxy to handle requests

[C] Tells the server to chain a rule with the next rule. If the rule matches for example, then the chained rules will run as well, if not, then they will not run.

[QSA] Tells the server to use the query string at the end of an expression

[NC] No Case instructs the server to treat any argument as case insensitive

[NS] The No Subrequest forces the server to skip if it is an internal sub request

[PT] Pass Through has mod_rewrite send a formatted URL back to Apache

[NE] No Escape forces the server to parse through all output ignoring escaping characters, meaning spaces in the U RL will not be replaced with %20 for example

[OR] Specifies a logical ‘OR‘ statement that evaluates two expressions

[S=x] Forces the server to skip “x” number of rules based on if a match is found, not the same as the Chain flag [C]

[a-z] Denotes a range of characters between the two characters separated by a dash

[^] Defines not within a character class, or the Start of a string of characters

[]+ Defines that any combination characters defined within the brackets is a match there can be multiple matches

[] Defines that any characters defined within the brackets is a match

[T=MIME-type] Defines the mime type, forces the target file to be that mime type

[E=variableName:newValue] Forces the server to set the environmental variable “variableName” to the value “new Value”

a{n} Defines the specific number of the preceding character to be matched

? Defines the preceding character as being optional

$ Signals the end of a regular expression

() Can be used to group characters together

^ Signals the beginning of a regular expression

. Specifies a single arbitrary character

– Signals not to perform an action

! Defines negation

+ Will match at least one preceding character

| Logical ‘OR‘ operator

* Wildcard that will match any occurrence of the preceding character

. Signals an escaped literal period

Used to escape special characters

-d Analyzes if a string exists within a directory

-f Determines if a string is a preexisting file

-s Tests for a non zero value