**Работа с .htaccess**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **[Как делать все преобразования за один 301 redirect](https://sites.google.com/a/aweb.ua/wiki/tech-department/programmer/tehniceskie-resenia/rabota-s-htaccess/1-redirect)**  Директива RewriteEngine on   |  |  | | --- | --- | | Описание: | Включает или выключает работу механизма преобразования | | Синтаксис: | RewriteEngine on|off | | Значение по умолчанию: | RewriteEngine off | | Модуль: | mod\_rewrite |   Директива RewriteEngine включает или выключает работу механизма преобразований. Если она установлена в положение off этот модуль не работает. Используйте эту директиву для выключения этого модуля вместо простого комментирования директив RewriteRule!  **Внедрение изменений только для нашего IP**  По аналогии с заключением в условный оператор if($\_SERVER['REMOTE\_ADDR'] == '82.144.208.73' || $\_SERVER['REMOTE\_ADDR'] == '194.187.128.182') { ... } в PHP стоит поступать и изменяя файл .htaccess.  Перед каждым правилом RewriteRule пишем RewriteCond %{REMOTE\_ADDR} ^(82\.144\.208\.73|194\.187\.128\.182)$ и, соответственно, с отрицанием, если планируем отменить имеющееся правило.  Например, нам нужно выключить правило old\_rewrite\_cond и дописать своё new\_rewrite\_cond, но не хотим ломать сайт, который к этому ещё не подготовлен:  RewriteCond %{REMOTE\_ADDR} !^(82\.144\.208\.73|194\.187\.128\.182)$  RewriteRule old\_rewrite\_cond  RewriteCond %{REMOTE\_ADDR} ^(82\.144\.208\.73|194\.187\.128\.182)$  RewriteRule new\_rewrite\_cond  **301 редирект с адреса без www на адрес с www**  RewriteCond %{HTTP\_HOST} !^site\.com [NC]  RewriteRule (.\*) http://site.com/$1 [R=301,QSA,L]  RewriteCond %{HTTP\_HOST} !^www\.(.+)$ [NC]  RewriteRule ^(.\*)$ http://www\.%{HTTP\_HOST}/$1 [R=301,L]  **301 редирект с www на без www**  RewriteCond %{HTTP\_HOST} ^www\.site\.com [NC]  RewriteRule (.\*) http://site.com/$1 [R=301,QSA,L]  RewriteCond %{HTTP\_HOST} ^www\.(.\*) [NC] RewriteRule ^(.\*)$ http://%1/$1 [R=301,L]  **301 редирект с index.php или index.html на /**  RewriteCond %{THE\_REQUEST} ^GET\ .\*/index\.(php|html)\ HTTP  RewriteRule ^(.\*)index\.(php|html)$ /$1 [R=301,L]  **Директива RewriteCond**   |  |  | | --- | --- | | Описание: | Определяет условие при котором происходит преобразование | | Синтаксис: | RewriteCond TestString CondPattern | | Значение по умолчанию: | None | | Модуль: | mod\_rewrite |   Директива RewriteCond определяет условия для какого-либо правила. Перед директивой RewriteRule располагаются одна или несколько директив RewriteCond. Следующее за ними правило преобразования используется только тогда, когда URI соответствует условиям этой директивы и также условиям этих дополительных директив.  **Пример:** RewriteCond %{REQUEST\_URI} !^/specialfolder  RewriteRule ^([^!].\*) /index.php?path=$1  **301 редирект со страниц без / на страницы с /**  RewriteBase / RewriteCond %{REQUEST\_FILENAME} !-f RewriteCond %{REQUEST\_FILENAME} !-d RewriteCond %{REQUEST\_URI} !-f RewriteCond %{REQUEST\_URI} !/$ RewriteCond %{REQUEST\_URI} !.html$ RewriteRule (.+) $1/ [R=301,L]  **Ссылки:**  [Модуль mod\_rewrite](http://www.google.com/url?q=http%3A%2F%2Fwww.shtogrin.com%2Flibrary%2Fweb%2Fmod_rewrite%2F&sa=D&sntz=1&usg=AFrqEzf5ZRX_NJ1w2QVpnhR7_mIM8G06oQ)  **301 редирект с заглавных символов в УРЛ ([A-Z]) на строчные ([a-z]) средствами .htaccess**  # Solution for shared hostings that does not support RewriteMap  # Rewrite uppercase letter URL to lowercase  # Skip next two rules if lowercasing in progress (Remove this rule if case-conversion plug-in below is removed)  RewriteCond %{ENV:qLow} ^yes$ [NC]  RewriteRule . - [S=2]  # Prevent recursion and over-writing of myURI and myQS  RewriteCond %{ENV:qRed} ^yes$ [NC]  RewriteRule .? - [L]  # Get the client-requested full URI and full query string  RewriteCond %{THE\_REQUEST} ^[A-Z]{3,9}\ (/[^?]\*)(\?[^\ ]\*)?\ HTTP/  RewriteRule .? - [E=myURI:%1,E=myQS:%2]  # Uppercase to lowercase conversion plug-in (This section, along with the first noted rule above,  # may be removed if not needed or wanted) Skip next 28 rules if no uppercase letters in URL  RewriteCond %{ENV:myURI} ![A-Z]  RewriteRule .? - [S=28]  # Else swap them out, one at a time  RewriteCond %{ENV:myURI} ^([^A]\*)A(.\*)$  RewriteRule . - [E=myURI:%1a%2]  RewriteCond %{ENV:myURI} ^([^B]\*)B(.\*)$  RewriteRule . - [E=myURI:%1b%2]  RewriteCond %{ENV:myURI} ^([^C]\*)C(.\*)$  RewriteRule . - [E=myURI:%1c%2]  RewriteCond %{ENV:myURI} ^([^D]\*)D(.\*)$  RewriteRule . - [E=myURI:%1d%2]  RewriteCond %{ENV:myURI} ^([^E]\*)E(.\*)$  RewriteRule . - [E=myURI:%1e%2]  RewriteCond %{ENV:myURI} ^([^F]\*)F(.\*)$  RewriteRule . - [E=myURI:%1f%2]  RewriteCond %{ENV:myURI} ^([^G]\*)G(.\*)$  RewriteRule . - [E=myURI:%1g%2]  RewriteCond %{ENV:myURI} ^([^H]\*)H(.\*)$  RewriteRule . - [E=myURI:%1h%2]  RewriteCond %{ENV:myURI} ^([^I]\*)I(.\*)$  RewriteRule . - [E=myURI:%1i%2]  RewriteCond %{ENV:myURI} ^([^J]\*)J(.\*)$  RewriteRule . - [E=myURI:%1j%2]  RewriteCond %{ENV:myURI} ^([^K]\*)K(.\*)$  RewriteRule . - [E=myURI:%1k%2]  RewriteCond %{ENV:myURI} ^([^L]\*)L(.\*)$  RewriteRule . - [E=myURI:%1l%2]  RewriteCond %{ENV:myURI} ^([^M]\*)M(.\*)$  RewriteRule . - [E=myURI:%1m%2]  RewriteCond %{ENV:myURI} ^([^N]\*)N(.\*)$  RewriteRule . - [E=myURI:%1n%2]  RewriteCond %{ENV:myURI} ^([^O]\*)O(.\*)$  RewriteRule . - [E=myURI:%1o%2]  RewriteCond %{ENV:myURI} ^([^P]\*)P(.\*)$  RewriteRule . - [E=myURI:%1p%2]  RewriteCond %{ENV:myURI} ^([^Q]\*)Q(.\*)$  RewriteRule . - [E=myURI:%1q%2]  RewriteCond %{ENV:myURI} ^([^R]\*)R(.\*)$  RewriteRule . - [E=myURI:%1r%2]  RewriteCond %{ENV:myURI} ^([^S]\*)S(.\*)$  RewriteRule . - [E=myURI:%1s%2]  RewriteCond %{ENV:myURI} ^([^T]\*)T(.\*)$  RewriteRule . - [E=myURI:%1t%2]  RewriteCond %{ENV:myURI} ^([^U]\*)U(.\*)$  RewriteRule . - [E=myURI:%1u%2]  RewriteCond %{ENV:myURI} ^([^V]\*)V(.\*)$  RewriteRule . - [E=myURI:%1v%2]  RewriteCond %{ENV:myURI} ^([^W]\*)W(.\*)$  RewriteRule . - [E=myURI:%1w%2]  RewriteCond %{ENV:myURI} ^([^X]\*)X(.\*)$  RewriteRule . - [E=myURI:%1x%2]  RewriteCond %{ENV:myURI} ^([^Y]\*)Y(.\*)$  RewriteRule . - [E=myURI:%1y%2]  RewriteCond %{ENV:myURI} ^([^Z]\*)Z(.\*)$  RewriteRule . - [E=myURI:%1z%2]  # Set lowercasing-in-progress flag  RewriteRule . - [E=qLow:yes]  # If any uppercase characters remain, re-start mod\_rewrite processing from the beginning  RewriteCond %{ENV:myURI} [A-Z]  RewriteRule . - [N]  # If any characters were lowercased, set redirect required flag and reset lowercasing-in-progress  # flag (S=28 from above lands here)  RewriteCond %{ENV:qLow} ^yes$ [NC]  RewriteRule . - [E=qRed:yes,E=qLow:done]  # End Uppercase to lowercase conversion plug-in  RewriteCond %{ENV:qRed} ^yes$ [NC]  RewriteRule .? http://%{HTTP\_HOST}%{ENV:myURI}%{ENV:myQS} [R=301,L] |

**Одно единственное перенаправление**

|  |
| --- |
| Часто ставится задача привести все ссылки на сайте к одному формату, что уже может включать в себя несколько регулярных замен в **URI**, замену **HOST** для всех псевдонимов домена, и, даже, преобразования **GET**-запроса. А после ещё и сделать несколько регулярных и нерегулярных перенаправлений. В итоге некоторые запросы на сервер превращаются в череду перенаправлений, а, [поговаривают](http://productforums.google.com/d/msg/webmasters/t6yDhHJiVcI/2KPtNRBgtacJ), что googlebot этого не любит. Да и серверу было бы куда легче обработать один запрос, а пользователю — быстрее получить. К счастью, синтаксис **.htaccess** позволяет создавать свои переменные и, даже, предусматривает кривую, но возможность их сравнения. Благодаря чему можно сначала прогнать запрос через все условные замены, а потом, если он изменился, перенаправить всего только раз. И выглядит это следующим образом:  ### Обязательная начальная часть {  # Разбираем запрос на три переменные: myHOST, myURI, myQS, которые вместе и являются ссылкой в адресной строке  # curURI и curQS, копирующие myURI и myQS, соответственно, определяются для удобства сравнения в конце  # Так же запоминаем метод, чтобы писать исключения для POST-запросов  **RewriteCond** %{THE\_REQUEST} ^([A-Z]{3,9})\ (/[^?]\*)(\?([^\ ]\*))?\ HTTP/  **RewriteRule** .? - [E=method:%1,E=myURI:%2,E=curURI:%2,E=myQS:%4,E=curQS:%4]  **RewriteCond** %{HTTP\_HOST} ^(.\*)$  **RewriteRule** .? - [E=myHOST:%1]  ### } Обязательная начальная часть  ### Модули обработки {  # Из HOST убираем "www.", например:  **RewriteCond** %{ENV:myHOST} ^www\.(.\*)$  **RewriteRule** .? - [E=myHOST:%1]  # Из URI убираем "/" на конце, например:  **RewriteCond** %{ENV:myURI} ^((/[^/]+)+)/$  **RewriteRule** .? - [E=myURI:%1]  # Из QS убираем "page=1", например:  **RewriteCond** %{ENV:myQS} ^(([^&]+&)\*)page=1(&([^&]+(&[^&]+)\*))?$  **RewriteRule** .? - [E=myQS:%1%4]  ### } Модули обработки  ### Обязательная конечная часть {  ###  # Сравниваем исходное обращение c обработанным нами обращением, и, в случае различия, перенаправляем:  **RewriteCond** %{HTTP\_HOST}%{ENV:curURI}?%{ENV:curQS}^%{ENV:myHOST}%{ENV:myURI}?%{ENV:myQS} !^([^^]+)\^\1$  **RewriteRule** .? http://%{ENV:myHOST}%{ENV:myURI}?%{ENV:myQS} [R=301,L]  ### } Обязательная конечная часть ###  Думаю, из примера ясно, как это работает. Так же удобно отключить сразу все наши фильтры для исключительных случаев. Например, для файлов и панели администратора. Для этого нужно дописать все условия только перед сравнением строк:  **RewriteCond** %{REQUEST\_FILENAME} !-f **RewriteCond** %{REQUEST\_URI} !^/administrator/ **RewriteCond** %{ENV:method} !^POST$  **RewriteCond** %{HTTP\_HOST}%{ENV:curURI}?%{ENV:curQS}^%{ENV:myHOST}%{ENV:myURI}?%{ENV:myQS} !^([^^]+)\^\1$  **RewriteRule** .? http://%{ENV:myHOST}%{ENV:myURI}?%{ENV:myQS} [R=301,L]  Теперь наиболее популярные модули преобразования **URL**:  Формат /page.html  # Убираем "/" на конце  **RewriteCond** %{ENV:myURI} ^((/[^/]+)+)/$  **RewriteRule** .? - [E=myURI:%1]  # Заменяем расширение на "html"  **RewriteCond** %{ENV:myURI} ^(/([^/]+/)\*[^/.]+(\.[^/.]+)\*)\.+[^/.]\*$  **RewriteRule** .? - [E=myURI:%1.html]  # Добавляем расширение "html"  **RewriteCond** %{ENV:myURI} ^((/([^/]+/)\*[^/.]+))$  **RewriteRule** .? - [E=myURI:%1.html]  Формат /page/  # Заменяем расширение на "/"  **RewriteCond** %{ENV:myURI} ^(/([^/]+/)\*[^/.]+(\.[^/.]+)\*)\.+[^/.]\*$  **RewriteRule** .? - [E=myURI:%1/]  # Добавляем "/" на конце  **RewriteCond** %{ENV:myURI} ^((/([^/]+/)\*[^/.]+))$  **RewriteRule** .? - [E=myURI:%1/]  Формат /page  # Убираем "/" на конце  **RewriteCond** %{ENV:myURI} ^((/[^/]+)+)/$  **RewriteRule** .? - [E=myURI:%1]  # Убираем расширение  **RewriteCond** %{ENV:myURI} ^(/([^/]+/)\*[^/.]+(\.[^/.]+)\*)\.+[^/.]\*$  **RewriteRule** .? - [E=myURI:%1]  Единственная главная страничка  # Убираем "index" с любым расширением  **RewriteCond** %{ENV:myURI} ^/index\.[^./]\*$  **RewriteRule** .? - [E=myURI:/]  Единственный домен  # Просто определяем наш домен  **RewriteRule** .? - [E=myHOST:domain.com]  Удаление переменной "page=1"  # Удаляем "page=1" на конце  **RewriteCond** %{ENV:myQS} ^(([^&]+&)\*)page=1(&([^&]+(&[^&]+)\*))?$  **RewriteRule** .? - [E=myQS:%1%4]  # Удаляем "&" на конце  **RewriteCond** %{ENV:myQS} ^(([^&]+&)\*[^&]+)?&$  **RewriteRule** .? - [E=myQS:%1]  Удаление "index.php/" из начала  # Убираем "index" с любым расширением  **RewriteCond** %{ENV:myURI} ^/index\.php((/[^/]+)+)$  **RewriteRule** .? - [E=myURI:%1]  Удаление "index.php" с конца  # Убираем "index" с любым расширением  **RewriteCond** %{ENV:myURI} ^/(([^/]+/)\*)index\.php$  **RewriteRule** .? - [E=myURI:/%1]  И, напоследок, приведу примеры реальных вставок в **.htaccess** такого кода со внедрённым плагином понижения регистра с моими комментариями (он же прикреплён файлом к статье):  **RewriteEngine** On  # Aweb begin  # Пропускаем следующие два правила, если циклическая замена (как понижение регистра, либо сокращение слешей подряд до одного)  # в процессе, чтобы переменные не сбросились. Эти две строки нужны только при наличии таких замен (все они с флагом N)  **RewriteCond** %{ENV:defined} ^yes$ [NC]  **RewriteRule** . - [S=2]  # Разбираем запрос на три переменные: myHOST, myURI, myQS, которые вместе и являются ссылкой в адресной строке  # curURI и curQS, копирующие myURI и myQS, соответственно, определяются для удобства сравнения в конце # Так же запоминаем метод, чтобы писать исключения для POST-запросов  **RewriteCond** %{THE\_REQUEST} ^([A-Z]{3,9})\ (/[^?]\*)(\?([^\ ]\*))?\ HTTP/  **RewriteRule** .? - [E=method:%1,E=myURI:%2,E=curURI:%2,E=myQS:%4,E=curQS:%4]  **RewriteCond** %{HTTP\_HOST} ^(.\*)$  **RewriteRule** .? - [E=myHOST:%1]  # Пропускаем следующие два правила, если нет слешей идущих подряд  **RewriteCond** %{ENV:myURI} !//  **RewriteRule** .? - [S=2]  # Заменяем два или более слеша, идущих подряд, на один  **RewriteCond** %{ENV:myURI} ^((/[^/]+)\*)/{2,}([^/].\*)?$  **RewriteRule** .? - [E=myURI:%1/%3]  # Если ещё остались слеши идущие подряд, перезапускаем mod\_rewrite с пометкой "в процессе"  **RewriteCond** %{ENV:myURI} //  **RewriteRule** .? - [E=defined:yes,N]  # Отсюда уже имеем myURI без повторяющихся слешей  # Пропускаем следующие 28 правил, если нет букв в верхнем регистре  **RewriteCond** %{REQUEST\_FILENAME} -f [OR]  **RewriteCond** %{ENV:myURI} ![A-Z]  **RewriteRule** .? - [S=27]  # Понижаем регистр всех 26 букв английского алфавита, каждой по одной  **RewriteCond** %{ENV:myURI} ^([^A]\*)A(.\*)$  **RewriteRule** . - [E=myURI:%1a%2]  **RewriteCond** %{ENV:myURI} ^([^B]\*)B(.\*)$  **RewriteRule** . - [E=myURI:%1b%2]  **RewriteCond** %{ENV:myURI} ^([^C]\*)C(.\*)$  **RewriteRule** . - [E=myURI:%1c%2]  **RewriteCond** %{ENV:myURI} ^([^D]\*)D(.\*)$  **RewriteRule** . - [E=myURI:%1d%2]  **RewriteCond** %{ENV:myURI} ^([^E]\*)E(.\*)$  **RewriteRule** . - [E=myURI:%1e%2]  **RewriteCond** %{ENV:myURI} ^([^F]\*)F(.\*)$  **RewriteRule** . - [E=myURI:%1f%2]  **RewriteCond** %{ENV:myURI} ^([^G]\*)G(.\*)$  **RewriteRule** . - [E=myURI:%1g%2]  **RewriteCond** %{ENV:myURI} ^([^H]\*)H(.\*)$  **RewriteRule** . - [E=myURI:%1h%2]  **RewriteCond** %{ENV:myURI} ^([^I]\*)I(.\*)$  **RewriteRule** . - [E=myURI:%1i%2]  **RewriteCond** %{ENV:myURI} ^([^J]\*)J(.\*)$  **RewriteRule** . - [E=myURI:%1j%2]  **RewriteCond** %{ENV:myURI} ^([^K]\*)K(.\*)$  **RewriteRule** . - [E=myURI:%1k%2]  **RewriteCond** %{ENV:myURI} ^([^L]\*)L(.\*)$  **RewriteRule** . - [E=myURI:%1l%2]  **RewriteCond** %{ENV:myURI} ^([^M]\*)M(.\*)$  **RewriteRule** . - [E=myURI:%1m%2]  **RewriteCond** %{ENV:myURI} ^([^N]\*)N(.\*)$  **RewriteRule** . - [E=myURI:%1n%2]  **RewriteCond** %{ENV:myURI} ^([^O]\*)O(.\*)$  **RewriteRule** . - [E=myURI:%1o%2]  **RewriteCond** %{ENV:myURI} ^([^P]\*)P(.\*)$  **RewriteRule** . - [E=myURI:%1p%2]  **RewriteCond** %{ENV:myURI} ^([^Q]\*)Q(.\*)$  **RewriteRule** . - [E=myURI:%1q%2]  **RewriteCond** %{ENV:myURI} ^([^R]\*)R(.\*)$  **RewriteRule** . - [E=myURI:%1r%2]  **RewriteCond** %{ENV:myURI} ^([^S]\*)S(.\*)$  **RewriteRule** . - [E=myURI:%1s%2]  **RewriteCond** %{ENV:myURI} ^([^T]\*)T(.\*)$  **RewriteRule** . - [E=myURI:%1t%2]  **RewriteCond** %{ENV:myURI} ^([^U]\*)U(.\*)$  **RewriteRule** . - [E=myURI:%1u%2]  **RewriteCond** %{ENV:myURI} ^([^V]\*)V(.\*)$  **RewriteRule** . - [E=myURI:%1v%2]  **RewriteCond** %{ENV:myURI} ^([^W]\*)W(.\*)$  **RewriteRule** . - [E=myURI:%1w%2]  **RewriteCond** %{ENV:myURI} ^([^X]\*)X(.\*)$  **RewriteRule** . - [E=myURI:%1x%2]  **RewriteCond** %{ENV:myURI} ^([^Y]\*)Y(.\*)$  **RewriteRule** . - [E=myURI:%1y%2]  **RewriteCond** %{ENV:myURI} ^([^Z]\*)Z(.\*)$  **RewriteRule** . - [E=myURI:%1z%2]  # Если ещё остались буквы верхнего регистра, перезапускаем mod\_rewrite с пометкой "в процессе"  **RewriteCond** %{ENV:myURI} [A-Z]  **RewriteRule** . - [E=defined:yes,N]  # Отсюда уже имеем myURI полностью в нижнем регистре  # Единственный домен  **RewriteRule** .? - [E=myHOST:www.ecosigara.com.ua]  # Пропускаем форматирование URI (следующие 3 правила) в случае обращения к файлу, к панели администратора либо POST-запроса  **RewriteCond** %{REQUEST\_FILENAME} -f [OR]  **RewriteCond** %{ENV:curURI} ^/administrator($|/) [OR]  **RewriteCond** %{ENV:method} ^POST$  **RewriteRule** .? - [S=3]  # Убираем "/" на конце  **RewriteCond** %{ENV:myURI} ^((/[^/]+)+)/$  **RewriteRule** .? - [E=myURI:%1]  # Заменяем расширение на "html"  **RewriteCond** %{ENV:myURI} ^(/([^/]+/)\*[^/.]+(\.[^/.]+)\*)\.+[^/.]\*$  **RewriteRule** .? - [E=myURI:%1.html]  # Добавляем расширение "html"  **RewriteCond** %{ENV:myURI} ^((/([^/]+/)\*[^/.]+))$  **RewriteRule** .? - [E=myURI:%1.html]  # Убираем "index.php/" в начале  **RewriteCond** %{ENV:myURI} ^/index\.php((/[^/]+)+)$  **RewriteRule** .? - [E=myURI:%1]  # Единственная главная страничка  **RewriteCond** %{ENV:myURI} ^/shop$ [OR]  **RewriteCond** %{ENV:myURI} ^/index\.[^./]\*$  **RewriteRule** .? - [E=myURI:/]  # Убираем переменную a\_aid с числом 13 порядка 16-ричной системы на главной страничке (в данном случае получается дубль)  **RewriteCond** %{ENV:myURI} ^/$  **RewriteCond** %{ENV:myQS} ^a\_aid=[0-9a-f]{13}$  **RewriteRule** .? - [E=myQS:]  # Статическое перенаправление со старой страницы  **RewriteCond** %{ENV:myURI} ^/primenimost\.html$  **RewriteRule** .? - [E=myURI:/cartridges.html]  # Сравниваем исходное обращение c обработанным нами обращением, и, в случае различия, перенаправляем:  **RewriteCond** %{HTTP\_HOST}%{ENV:curURI}?%{ENV:curQS}^%{ENV:myHOST}%{ENV:myURI}?%{ENV:myQS} !^([^^]+)\^\1$  **RewriteRule** .? http://%{ENV:myHOST}%{ENV:myURI}?%{ENV:myQS} [R=301,L]  # Aweb end |