

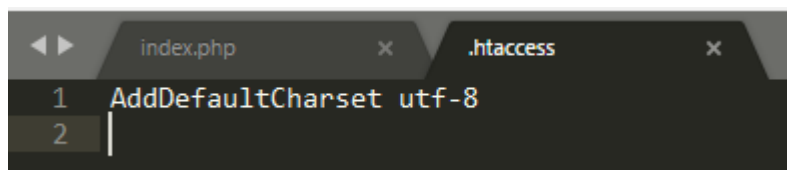
# Файл конфигурации сервера .htaccess

**.htaccess** - конфигурационный файл веб-сервера Apache, позволяющий управлять работой веб-сервера и настройками сайта с помощью различных параметров (директив) без изменения основного конфигурационного файла веб-сервера.

Директивы файла **.htaccess** действуют для каталога, в котором размещён такой файл, и для всех его подкаталогов. Если вы желаете с помощью **.htaccess** изменить настройки для сайта в целом, его следует размещать в корневом каталоге сайта.

**Внимание!** При сохранении файла **.htaccess** необходимо использовать кодировку UTF-8 без BOM.

В практике 3 мы использовали файл **.htaccess** для настройки кодировки в которой сервер отдаёт страницы.



```
index.php x .htaccess x
1 AddDefaultCharset utf-8
2 |
```

Директива AddDefaultCharset устанавливает кодировку, в которой сервер отдаёт страницы браузеру пользователя. Это никак не заменяет необходимость устанавливать кодировку файла скрипта, html страницы, кодировку для работы с БД и др.

## Задача 1. (Самостоятельно)

*Самостоятельно создайте файл **.htaccess** в папке localhost/www, сохраните его с расширением UTF-8 без BOM и установите кодировку по умолчанию utf-8.*

## Директивы .htaccess

# - используется для создания комментариев в файле **.htaccess**

## Простая переадресация

**Redirect** – используется для создания простого перенаправления.

Синтаксис:

**Redirect [status] URI\_LOCAL URL\_REDIRECT**

**status:** необязательное поле, определяет код возврата. Допустимые значения:

- permanent (301 — документ перемещен постоянно)
- temp (302 — документ перемещен временно)
- seeother (303 — смотрите другой)
- gone (410 — убран)

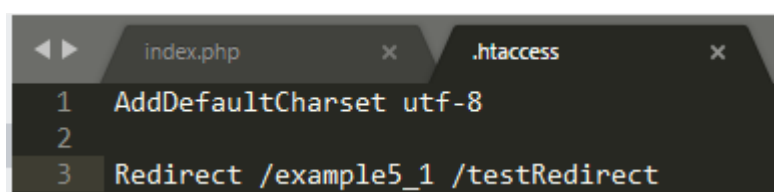
**URI\_LOCAL:** локальная часть URL запрашиваемого документа.

**URL\_REDIRECT:** URL, куда должно быть выполнено перенаправление.

```
3 Redirect / http://www.example.com/
4 # http://www.example.com - URL, на который мы перенаправляем
  запросы
```

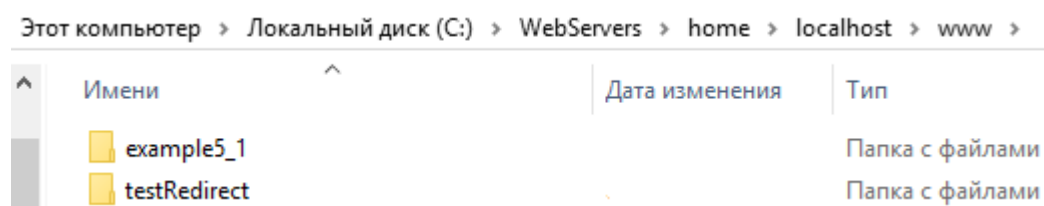
В примере все запросы к текущему сайту будут перенаправлены на сайт **http://www.example.com**. Т.е. данная директива поменяет запрос **localhost/example5\_1** на запрос следующего вида: **http://www.example.com/example5\_1/**.

Эта же директива позволяет настраивать переадресацию для отдельных страниц вашего сайта:



```
index.php x .htaccess x
1 AddDefaultCharset utf-8
2
3 Redirect /example5_1 /testRedirect
```

Указанная директива переадресует страницу **localhost/example5\_1** на страницу **localhost/testRedirect**:



## Задача 2. (Самостоятельно)

*Создать правила в файле **.htaccess** согласно которым страницы **localhost/example5\_1/help** и **localhost/example5\_1/contacts** переадресовывались бы на главную страницу.*

## **Директивы сложного перенаправления mod\_rewrite**

Напомним основные элементы из которых строится регулярное выражение:

- Круглые скобки **()** используются для выделения групп символов. В дальнейшем к ним можно обращаться по номеру.
- Символ **^** обозначает начало строки.
- Символ **\$** обозначает конец строки.
- Символ **.** обозначает любой символ.
- Символ **|** обозначает альтернативу. Например, выражения **"A|B"** означают **"А или В"**.
- Символ **?** ставится после символа (группы), который может как присутствовать, так и отсутствовать.
- Символ **\*** ставится после символа (группы), который может отсутствовать или присутствовать неограниченное число раз подряд.
- Символ **+** действует аналогично символу **\*** с той лишь разницей, что предшествующий ему символ обязательно должен присутствовать хотя бы один раз.
- Квадратные скобки **[]** используются для перечисления допустимых символов.
- Квадратные скобки **[^]** используются для перечисления недопустимых символов.
- Символ **\** ставится перед спецсимволами, если они нужны в своем первоизданном виде, т.е. осуществляет экранирование.

Включение mod\_rewrite осуществляется директивой:

**RewriteEngine on**

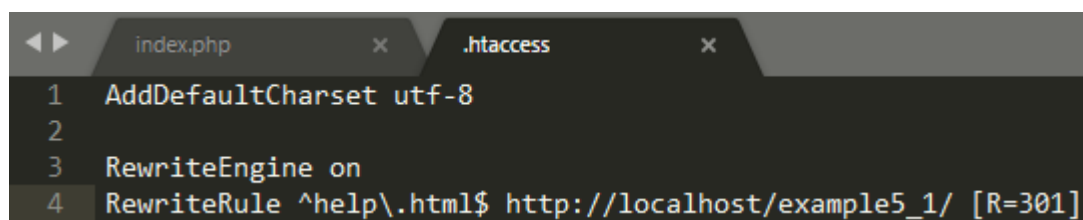
Если при использовании на внешнем хостинге вы получаете ошибку 404, то вероятно на вашем хостинге не используется `mod_rewrite`. В данном случае надо обратиться к администратору хостинга.

Переадресация задается следующей командой:

### **RewriteRule Pattern Substitution [Optional Flags]**

- **Pattern** - регулярное выражение шаблона. Если URL соответствует шаблону, то правило выполняется. Иначе правило пропускается.
- **Substitution** - новый URL, который будет использоваться вместо соответствующего шаблону адреса.
- **[Optional Flags]** - один или несколько флагов, которые определяют поведение правила.

**Пример:**



```
<> index.php x .htaccess x
1 AddDefaultCharset utf-8
2
3 RewriteEngine on
4 RewriteRule ^help\.html$ http://localhost/example5_1/ [R=301]
```

В данном примере запрос **localhost/help.html** перенаправляется на страницу **localhost/example5\_1/**. Данное перенаправление будет внешним, т.е. отобразится в адресной строке.

В файл `.htaccess` можно добавлять много правил `RewriteRule`. Модуль `mod_rewrite` проходит все правила каждый раз при запросе, обрабатывая соответствующие адресу URL. Если правило изменяет запрашиваемый URL на новый адрес, то новый URL используется дальше при проходе по файлу `.htaccess`, и может соответствовать другому правилу `RewriteRule`, размещающемуся далее в файле. (Если нужно изменить такое поведение, то надо использовать флаг `L` ("последнее правило").)

Некоторые возможные флаги:

**redirect|R [=code]'**

(вызывает редирект)

Префикс в Подстановке вида `http://thishost[:thisport]/` (создающий новый URL из какого-либо URI) запускает внешний редирект (перенаправление). Если

нет никакого кода, в подстановке ответ будет со HTTP статусом 302 (ВРЕМЕННО ПЕРЕМЕЩЕН). Для остановки процесса преобразования вам также нужно написать флаг 'L'.

**'forbidden|F [=code]'**

(делает URL запрещенным)

Это делает текущий URL запрещённым, например, клиенту немедленно отправляется ответ с HTTP статусом 403 (ЗАПРЕЩЕНО). Используйте этот флаг в сочетании с соответствующими RewriteConds для блокирования URL по некоторым критериям.

**'gone|G [=code]'**

(делает URL «мёртвым»)

Этот флаг делает текущий URL «мертвым», т.е., немедленно отправляется HTTP ответ со статусом 410 (GONE). Используйте этот флаг для маркировки «мертвыми» несуществующие более страницы.

**'last|L [=code]'**

(последнее правило)

Остановить процесс преобразования на этом месте и не применять больше никаких правил преобразований. Используйте этот флаг для того, чтобы не преобразовывать текущий URL другими, следующими за этим, правилами преобразований.

### **Задача 3. (Самостоятельно)**

*Решить задачу 2 с помощью сложного перенаправления mod\_rewrite.*

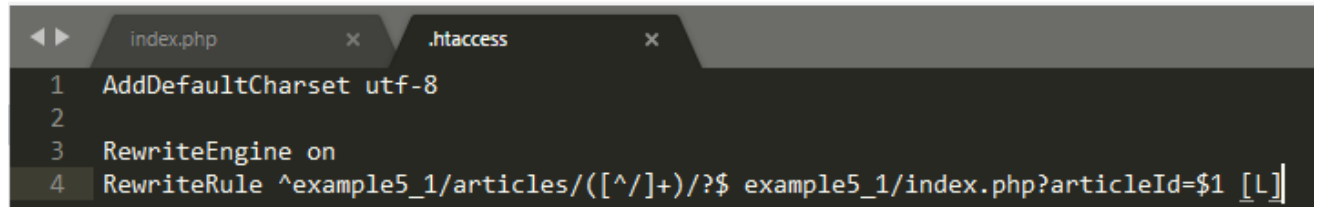
Рассмотрим пример, который позволяет использовать понятные адреса вместо передачи get параметров. Предположим скрипт, отображающий содержимое статьи, требует наличия get параметра articleId. Типичный вызов этого скрипта будет выглядеть так:

**localhost/example5\_1/index.php?articleId=123**

Представим его в более простом виде:

**localhost/example5\_1/articles/123**

Теперь напишем правило преобразования простого вида URL в требуемый формат:



```
index.php x .htaccess x
1 AddDefaultCharset utf-8
2
3 RewriteEngine on
4 RewriteRule ^example5_1/articles/([^/]+)/?$ example5_1/index.php?articleId=$1 [L]
```

Разберем это выражение:

**RewriteRule ^example5\_1/articles/([^/]+)/?\$ example5\_1/index.php?articleId=\$1 [L]**

Регулярное выражение в шаблоне означает следующее: ищем строку начинающуюся (^) с **example5\_1/articles/** далее следует группа символов не содержащая символ « / » за которыми может следовать слэш «/?», за которым следует символ окончания URL \$. Обратите внимание, что часть выражения сгруппирована круглыми скобками, к выражению, которое соответствует этим скобкам, в следующем параметре директивы **RewriteRule** можно обратиться по номеру (\$1).

#### Задача 4. (Самостоятельно)

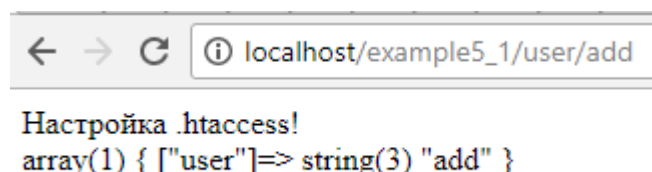
*Настроить переадресацию с следующего вида:*

**localhost/example5\_1/action/value** на

**http://localhost/example5\_1/index.php?action=value**

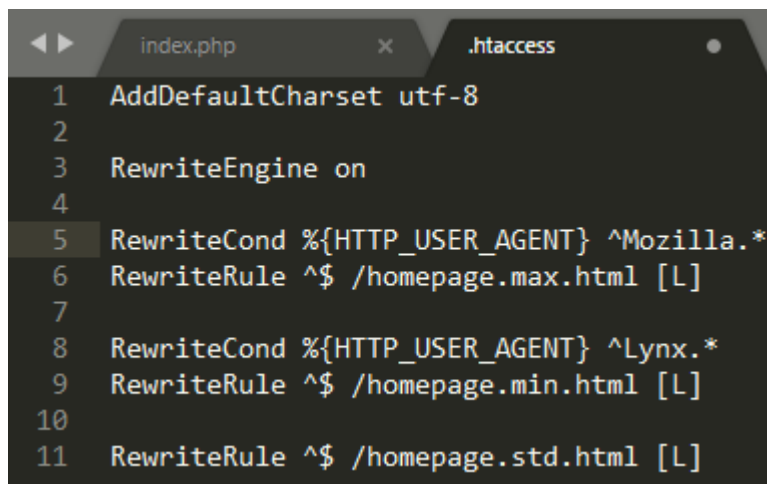
*Пример: http://localhost/example5\_1/user/add переходит в:*

**http://localhost/example5\_1/index.php?user=add**



Директива **RewriteCond** определяет условие, при котором происходит преобразование.

Пример.



```
1 AddDefaultCharset utf-8
2
3 RewriteEngine on
4
5 RewriteCond %{HTTP_USER_AGENT} ^Mozilla.*
6 RewriteRule ^$ /homepage.max.html [L]
7
8 RewriteCond %{HTTP_USER_AGENT} ^Lynx.*
9 RewriteRule ^$ /homepage.min.html [L]
10
11 RewriteRule ^$ /homepage.std.html [L]
```

Выдает различные главные страницы какого-либо сайта, согласно «User-Agent:» заголовку запроса.

Некоторые из доступных переменных %{NAME\_OF\_VARIABLE}:

**HTTP\_USER\_AGENT**

Содержит информацию о типе и версии браузера и операционной системы посетителя.

**HTTP\_REFERER**

Приводится адрес страницы, с которой посетитель пришёл на данную страницу.

**HTTP\_COOKIE**

Список COOKIE, передаваемых браузером.

**HTTP\_HOST**

Адрес сервера, например, beget.com .

**REMOTE\_ADDR**

IP-адрес посетителя.

**REMOTE\_HOST**

Адрес посетителя в нормальной форме — например, rt99.net.ru .

**REQUEST\_METHOD**

Позволяет определить тип запроса (GET или POST). Должен обязательно анализироваться, т.к. определяет дальнейший способ обработки информации.

**SCRIPT\_FILENAME**

Полный путь к веб-странице на сервере.

## DOCUMENT\_ROOT

Содержит путь к корневой директории сервера.

## SERVER\_NAME

Адрес сервера, например, kondr.beget.com

Синтаксис:

### **RewriteCond** *Анализируемая\_строка* *Условие*

Анализируемая строка просматривается на поиск соответствия Условию, которое является perl-совместимым регулярным выражением с некоторыми дополнениями:

Вы можете предварять строку шаблона префиксом '!' для указания несоответствия шаблону.

'<Условие' (лексически меньше)

'>Условие' (лексически больше)

'=Условие' (лексически равно)

'-d' (является ли каталогом)

'-f' (является ли обычным файлом)

'-s' (является ли обычным файлом с ненулевым размером)

'-l' (является ли символической ссылкой)

'-F' (проверка существования файла через подзапрос)

'-U' (проверка существования URL через подзапрос)

### **Пример:**

```
RewriteCond %{REMOTE_HOST} ^host1.* [OR]
```

```
RewriteCond %{REMOTE_HOST} ^host2.* [OR]
```

```
RewriteCond %{REMOTE_HOST} ^host3.*
```

```
RewriteRule ^(.*)$ ^mysubdomain_public_html/$1
```

Данный пример реализует перенаправление запросов на поддомены в отдельные каталоги.



## Индексные страницы

Иногда нужно сделать так, чтобы в случае отсутствия в каталоге файла, который показывается по умолчанию, листинг, то есть список файлов в каталоге, не выдавался. В этом случае добавим в .htaccess такую строчку:

```
# Запрет выдачи листинга пустого каталога
```

### Options –Indexes

По умолчанию индексная (главная) страница может называться index.html index.php, index.phtml или index.shtml. Указать в качестве индексной страницы другие файлы можно с помощью директивы DirectoryIndex.

**DirectoryIndex index.html index.shtml index.pl index.cgi index.php**

### Задача 5. (Самостоятельно)

*В качестве главной страницы вашего сайта установить файл **menu.php**. Проверить работоспособность директивы.*

Источники:

[https://beget.com/ru/articles/htaccess#mod\\_rewrite](https://beget.com/ru/articles/htaccess#mod_rewrite)

<https://blog.ukrnames.com/hosting/predstavlyaem-vashemu-vnimaniyu-podborku-iz-24-pravil-dlya-htaccess>

<http://site-on.net/optimization/4-sef-url-mod-rewrite>

[https://www.nic.ru/help/fajl-htaccess-nastrojka-perenapravlenij-i-upravlenie-konfiguraciej-veb-servera-\(staraya-panel6-uprav\\_316.html](https://www.nic.ru/help/fajl-htaccess-nastrojka-perenapravlenij-i-upravlenie-konfiguraciej-veb-servera-(staraya-panel6-uprav_316.html)