



Бесплатная электронная книга

УЧУСЬ

apache

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#apache

.....	1
<b>1: apache</b> .....	<b>2</b>
.....	2
.....	2
Apache httpd.....	2
Examples.....	2
.....	2
Ubuntu.....	2
Windows.....	2
CentOS.....	3
macOS.....	3
[Ubuntu] Hello World.....	3
.....	3
HTML.....	3
-.....	4
.....	4
<b>2: .htaccess Apache</b> .....	<b>5</b>
Examples.....	5
.....	5
Force HTTPS.....	5
CORS.....	6
.....	7
301 Htaccess.....	7
<b>3: Apache</b> .....	<b>9</b>
.....	9
Examples.....	9
.....	9
PHP Development.....	10
WAMP.....	11
1) IP- vhosts 2) vhosts 3) vhosts .....	12
HTTPS .....	13

<b>4: Apache</b> .....	<b>15</b>
.....	15
Examples.....	15
/ .....	15
.....	<b>16</b>

---

# Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [apache](#)

It is an unofficial and free apache ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official apache.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# глава 1: Начало работы с apache

## замечания

В этом разделе представлен обзор того, что такое apache, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в apache и ссылки на связанные темы. Поскольку Documentation для apache является новым, вам может потребоваться создать начальные версии этих связанных тем.

## Версии

### Различные версии Apache httpd

Версия	Текущая версия	Релиз
1,3	1.3.42	1998-06-06
2,0	2.0.65	2002-04-06
2,2	2.2.32	2005-12-01
2,4	2.4.25	2012-02-21

## Examples

### Установка или настройка

Подробные инструкции по установке или настройке apache.

### Установка Ubuntu

```
sudo apt-get install apache2
```

### Установка Windows

Проверьте стек [WAMP](#) . WAMP означает Windows, Apache, MySQL, PhpMyAdmin.

# Установка CentOS

Apache 2.2 поставляется с CentOS6, тогда как 2.4 поставляется с CentOS7, для установки на любую ОС, запускать

```
yum -y install httpd
```

# Установка macOS

macOS поставляется с предустановленной Apache, однако может установить Apache через Homebrew

Если у вас уже есть встроенный Apache, он должен быть выключен первым, и все сценарии автоматической загрузки удалены.

```
$ sudo apachectl stop
$ sudo launchctl unload -w /System/Library/LaunchDaemons/org.apache.httpd.plist 2>/dev/null
$ brew install httpd24 --with-privileged-ports --with-http2
```

## [Ubuntu] Простой пример Hello World

В этом примере вы проведете настройку задней части, обслуживающей страницу Hello World HTML.

## Установка требований

Порядок имеет значение для этого шага!

- `sudo apt-get install apache2`

## Настройка HTML

Файлы Apache живут в `/var/www/html/`. Позвольте быстро добраться туда. Убедитесь, что вы в корневом каталоге сначала, `cd`, затем `cd /var/www/html/`.

Этот каталог `html` будет содержать все файлы вашего сайта. Позволяет быстро создать простой файл Hello World.

Используя ваш любимый текстовый редактор, введите следующее в

```
<!DOCTYPE html>
```

```
<html>
<head>
  <title>Hello World!</title>
</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

Сохраните этот файл как `index.html` в текущем каталоге, и вы готовы к работе!

## Посещение веб-страницы

Чтобы перейти к странице, которую вы только что создали, в вашем браузере по выбору, перейдите на `localhost` . Если это не работает, попробуйте `127.0.0.1` . Вы должны увидеть «Hello World!» как `h1` . Все готово!

### Чтобы сервер работал.

Если вы получите сообщение о том, что браузер не может подключиться к серверу, сначала проверьте, чтобы сервер был включен.

```
$ ps -aef | grep httpd
```

Вы должны увидеть несколько процессов `httpd`, если Apache запущен и работает.

Прочитайте [Начало работы с apache онлайн: https://riptutorial.com/ru/apache/topic/964/начало-работы-с-apache](https://riptutorial.com/ru/apache/topic/964/начало-работы-с-apache)

# глава 2: .htaccess файлы в Apache

## Examples

### Перезапустить двигатель

Модуль RewriteEngine в Apache используется для динамического переписывания URL-адресов и путей в зависимости от различных выражений:

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

Вышеупомянутые правила переписывают файлы PHP, чтобы больше не показывать их расширение, и поэтому index.php будет просто отображаться как голый домен (аналогично поведению, обычно наблюдаемому в index.html). Вышеупомянутое правило поставляется с WordPress.

Обратите внимание, что в Apache httpd 2.2.16 и более поздних версиях весь этот блок можно заменить одной строкой с помощью директивы FallbackResource:

```
FallbackResource /index.php
```

### Force HTTPS

.htaccess можно использовать для перенаправления вашего HTTP-сайта на HTTPS.

Вот быстрый способ, который не требует редактирования кода для вашего домена:

```
RewriteEngine On
RewriteCond %{HTTPS} =off
RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

**Предупреждение.** В приведенном выше коде предполагается, что вы можете доверять `%{HTTP_HOST}` чтобы указать на свой домен.

Если вам нужно быть уверенным, что место перенаправления - ваш домен, замените `%{HTTP_HOST}` своим доменом.

Вышеприведенный код делает следующее:

1. Включить [RewriteEngine](#) .
2. Продолжайте, если текущий запрос не использует HTTPS.
3. Переадресация HTTP 301 на `https://%{HTTP_HOST}%{REQUEST_URI}` , где
  - `%{HTTP_HOST}` - это хост, запрашиваемый браузером и
  - `%{REQUEST_URI}` - это URI, запрошенный браузером (все после домена).

**Предупреждение.** Ваше веб-приложение должно иметь возможность обрабатывать HTTPS-запросы, а Apache для вашего хоста должен быть настроен с действительным сертификатом сайта.

Обратите внимание, что значительно проще выполнять `Redirect` в `http`-хосте, чем выполнять эти множественные сравнения по запросу в файле `.htaccess`. См.

[Http://wiki.apache.org/httpd/RedirectSSL](http://wiki.apache.org/httpd/RedirectSSL) для дальнейшего обсуждения этого метода.

## Включить CORS

Чтобы включить **совместное использование ресурсов Cross-Origin ( CORS )** в Apache, вам нужно установить по крайней мере один HTTP-заголовок, который его изменит (по умолчанию используется блокировка CORS). В следующем примере мы собираемся настроить этот HTTP-заголовок внутри `.htaccess` , но его также можно установить на вашем сайте `your-site.conf` файл `your-site.conf` или файл конфигурации Apache. Независимо от того, как выглядит ваша конфигурация, вы можете установить соответствующие HTTP-заголовки в любом блоке конфигурации Apache, то есть `<VirtualHost>` , `<Directory>` , `<Location>` И `<Files>` .

Есть несколько заголовков HTTP, связанных с CORS, которые вы можете вернуть в ответ:

```
Access-Control-Allow-Origin
Access-Control-Allow-Credentials
Access-Control-Allow-Methods
Access-Control-Max-Age
Access-Control-Allow-Headers
Access-Control-Expose-Headers
```

Некоторые из вышеперечисленных требований требуются для запросов «предполетный». Некоторые HTTP-клиенты (а именно современные браузеры) выполняют запрос **перед** желаемым запросом только для того, чтобы узнать, есть ли у них разрешение на выполнение фактического запроса на сервере. См. [https://en.wikipedia.org/wiki/Cross-origin\\_resource\\_sharing](https://en.wikipedia.org/wiki/Cross-origin_resource_sharing) для получения дополнительной информации о предполетном запросе.

Главный HTTP-заголовок, который нам нужен, - `Access-Control-Allow-Origin` И ЭТО мы собираемся установить. Тем не менее, тот же принцип применяется в значительной степени для всех из них (вам просто нужно знать, что вернуть).

Следующий пример устанавливает требуемый HTTP-заголовок в блоке конфигурации `<Directory>` чтобы включить защищенный SSL-клиент Полное полное доменное имя (FQDN):

```
<Directory /path/to/your/site/>
    Header set Access-Control-Allow-Origin "https://my.CLIENT.domain"
</Directory>
```

После того, как мы установили это на **сервере**, теперь мы можем выполнить запрос от <https://my.client.domain> на наш сервер, и он должен ответить.

Примечание. Многие люди используют `Access-Control-Allow-Origin: *` который является подстановочным знаком, так как запросы из **всех** доменов должны быть приняты. Обычно это не рекомендуется, если вы не используете какой-либо публичный API или репозиторий файлов. Также обратите внимание на контекст настройки HTTP-заголовка. Возможно, вы захотите разрешить HTTP-запросы для API, но не для «ссылок на хотлинкинг» и т. Д. Вы можете настроить этот заголовок в любом месте в вашем потоке конфигурации Apache, чтобы устанавливать его **только** в определенных ситуациях. Например, следующее будет **только** установить заголовок HTTP CORS, когда запрашиваемый путь **не** является файл или каталог (подходит для публичного API, который не разрешает хотлинкинг изображения):

```
<Directory /path/to/your/site/>
    Options +FollowSymLinks
    Options +Indexes
    RewriteEngine On

    #Make sure it's not a specific file or directory that they're trying to reach
    RewriteCond %{SCRIPT_FILENAME} !-f
    RewriteCond %{SCRIPT_FILENAME} !-d
    Header set Access-Control-Allow-Origin "*"
    RewriteRule ^(.*)$ index.php/$1 [L]
</Directory>
```

## Предпосылки

Вы должны быть `mod_headers` установлены и включены: `a2enmod headers`

## 301 Перенаправление по Htaccess

Код статуса HTTP-ответа 301 Перемещается на постоянной основе используется для постоянного перенаправления URL-адресов, что означает, что текущие ссылки или записи с использованием URL-адреса, на который получен ответ, должны быть обновлены. Новый URL-адрес должен быть указан в поле «Место», включенном в ответ. Перенаправление 301 считается лучшей практикой для обновления пользователей от HTTP до HTTPS. написать этот код в файле `htaccess` для PHP-APACHE

```
Redirect 301 /oldpage/ /newpage/
```

Ниже приведен пример использования файла htaccess для перенаправления на не-www с SSL, прикрепленным к домену.

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ http://%1/$1 [R=301,L]

RewriteCond %{HTTPS} on
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ https://%1/$1 [R=301,L]

RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://example.com/$1 [R,L]
```

Прочитайте .htaccess файлы в Apache онлайн: <https://riptutorial.com/ru/apache/topic/2089/-htaccess-файлы-в-apache>

---

# глава 3: Как создать виртуальный хост в Apache

## замечания

Основная точка входа для `VirtualHost` от Apache - в документации [Apache Virtual Host](#) . Оттуда у вас есть общая документация о конфигурации виртуального хоста, а также справочная документация о `VirtualHost` и соответствующих директивах.

## Examples

### Конфигурация виртуального хоста на основе имени

Виртуальный хостинг на основе имени на Apache описан на [веб-сайте Apache](#) как таковой:

При использовании виртуального хостинга на основе имени сервер полагается на клиенту, чтобы сообщить имя хоста как часть заголовков HTTP. Используя этот метод, многие разные хосты могут использовать один и тот же IP-адрес.

Таким образом, с помощью этого метода можно разместить на одном сервере более одного веб-сайта. На `ubuntu` файлы конфигурации находятся в `/etc/apache2/sites-available`. В этом каталоге вы найдете `000-default.conf`. Это конфигурация по умолчанию, все запросы будут отправлены в этот файл конфигурации до тех пор, пока не будут настроены другие.

Чтобы настроить виртуальный хост, здесь будет использоваться **example.com** , но вы должны заменить его на **domain.com** . Скопируйте файл по умолчанию:

```
cp 000-default.conf example.com.conf
```

Файл конфигурации может иметь следующие директивы:

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com

    DocumentRoot /var/www/example.com/html

    ErrorLog /var/log/apache/logs/error.log
    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /var/log/apache/logs/access.log combined
</VirtualHost>
```

- Первая строка указывает, что все запросы на порт 80 (порт http по умолчанию) должны быть сопоставлены. Вы также можете иметь IP-адрес вместо \*, который является IP-адресом сервера.
- `ServerAdmin` - это контактные данные администратора сайта, используемые для отображения с сообщениями об ошибках HTTP.
- `ServerName` - это доменное имя веб-сайта.
- `ServerAlias` является вторичным именем веб-сайта, как правило, будет `www.domain.com`
- `DocumentRoot` - это корневая папка, загружаемая при просмотре веб-сайта.
- `ErrorLog` - это файл, в котором указаны ошибки
- `LogLevel` . - уровень ошибок, которые должны быть отправлены в журнал
- `CustomLog` - это файл, в котором `CustomLog` информация о доступе

Отредактируйте файл, заменяющий `example.com` своим доменным именем и соответствующим каталогом файлов веб-сайта.

Сохраните файл и включите сайт со следующей командой Apache:

```
sudo a2ensite example.com.conf
```

Перезагрузить апачу

```
sudo service apache2 reload
```

Еще несколько вещей, которые необходимо проверить:

- Убедитесь, что ваш DNS для вашего домена настроен для правильного IP-адреса (это может занять время для распространения)
- Убедитесь, что порт 80 открыт на брандмауэре
- Убедитесь, что ваши права на файлы настроены правильно на файлах сервера. Владение должно быть `www`-данными: права доступа к `www`-данным и каталогу должны быть `750`, а разрешения для файлов должны быть `640`.

Ваш виртуальный хост должен быть запущен и запущен! Вы можете повторить это для других веб-сайтов на том же сервере, с другим конфигурационным файлом (с использованием того же правила именования) и разными каталогами в `/var/www`.

## Виртуальный хостинг PHP Development

Это пример того, как управлять ведением журнала ошибок PHP на виртуальном узле хоста для разработки и отладки. Предположения

- Установлен модуль PHP.
- Среда разработки не предназначена для производства.

```
<VirtualHost *:80>
    ServerName example.com
    DocumentRoot /var/www/domains/example.com/html
    ErrorLog /var/www/domains/example.com/apache.error.log
    CustomLog /var/www/domains/example.com/apache.access.log common
    php_flag log_errors on
    php_flag display_errors on
    php_value error_reporting 2147483647
    php_value error_log /var/www/domains/example.com/php.error.log
</VirtualHost>
```

**Примечание** . Конфигурация виртуального хоста предназначена для разработки только потому, что `display_errors` включен, и вы не хотите, чтобы это было на производстве.

## Виртуальный хост в WAMP

### Предполагая, что вы работаете с ПК с Windows 7

**Шаг 1:** GOTO -> C:\Windows\System32\drivers\etc Где вы найдете файл с именем «hosts», скопируйте его и вставьте его в том же месте. Там будет создан файл с копиями хостов.

Теперь нам нужно внести некоторые изменения в этот файл, но если вы попытаетесь отредактировать его в любом редакторе, таком как блокнот или блокнот ++, он не позволит вам сохранить файл.

Теперь снова скопируйте тот же файл и вставьте его на свой рабочий стол, теперь вы можете легко редактировать этот файл.

Вы найдете одну или несколько записей: 127.0.0.1 localhost В этом файле. Теперь добавьте еще одну строку ниже этой строки, например: 127.0.0.1 myproject1.local Таким образом, вы определили новый поддомен «myproject1.local», который может работать вместо «localhost / myproject1».

**Шаг 2:** Хорошо, теперь пришло время определить корневой путь для доступа к этому вновь созданному домену правильно? GOTO: C:\wamp\bin\apache\Your-Apache-Version\conf\extra Здесь вы найдете файл с именем «httpd-vhosts». Откройте его в редакторе и вставьте в него следующие строки.

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy.example.com
    DocumentRoot "c:/wamp/www/myproject1/"
    ServerName myproject1.local
    ErrorLog "logs/myproject1.local-error.log"
    CustomLog "logs/myproject1.local.log" common
</VirtualHost>
```

Теперь вы почти можете получить доступ к проекту, который находится на «c:/wamp/www/myproject1 /»

### Шаг 3: GOTO: C:\wamp\bin\apache\your-Apache-Version\conf

Найдите файл с именем «**httpd.conf**» , скопируйте его и вставьте его в том же месте для обеспечения безопасности. Откройте файл в редакторе и найдите слово «**# Virtual hosts**» , ниже вы найдете строку «**Include conf / extra / httpd-vhosts.conf**». Если она прокомментирована, сделайте ее незапланированной и перезапустите службы вашего сервера **Wamp** .

Перейдите в свой веб-браузер и напишите myproject1.local, вы можете увидеть, как работает проект.

Теперь вы можете столкнуться с проблемой, что ваш localhost не будет работать, используя localhost в качестве URL-адреса. No Worries ... вставьте этот код в файл «**httpd-vhosts**» .

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy.example.com
    DocumentRoot "c:/wamp/www"
    ServerName localhost
    ErrorLog "logs/localhost-error.log"
    CustomLog "logs/localhost.log" common
</VirtualHost>
```

**Перезапустите все службы WAMP, работа будет выполнена.**

Спасибо и приветствует **Chintan Gor**

## 1) IP-адреса vhosts 2) Несколько vhosts с тем же портом 3) Определение vhosts с использованием макроса (Apache2.4)

### 1) IP-хосты

```
<VirtualHost 192.168.13.37>
    ServerName example.com
    DocumentRoot /var/www/domains/example.com/html
    ErrorLog /var/log/example.com/error.log
    CustomLog /var/log/example.com/access.log common
</VirtualHost>

<VirtualHost 192.168.47.11>
    ServerName otherurl.com
    DocumentRoot /srv/www/htdocs/otherurl.com/html
    ErrorLog /var/log/otherurl.com/error.log
    CustomLog /var/log/otherurl.com/access.log common
</VirtualHost>
```

Просто измените порт на ваш IP-адрес. Порт не имеет значения для решения, которое выбрано для vhost.

### 2) Несколько vhosts с тем же портом

Поскольку NameVirtualHost больше не требуется, вы можете просто написать несколько vhosts с тем же портом.

```
<VirtualHost *:80>
    DocumentRoot /srv/www/htdocs/otherurl.com/html
    ErrorLog /var/log/otherurl.com/error.log
    CustomLog /var/log/otherurl.com/access.log common
</VirtualHost>

<VirtualHost *:80>
    ServerName example.com
    ServerAlias ex1.com ex2.com
    DocumentRoot /var/www/domains/example.com/html
    ErrorLog /var/log/example.com/error.log
    CustomLog /var/log/example.com/access.log common
</VirtualHost>
```

Здесь применяется обратное: IP не имеет значения, но если запрос получен на порт 80, то оценивается имя, которое вы ввели. Вы назвали ex1.com, 2-й призрак получает. И если вы вызвали какой-либо другой URL-адрес (например, otherurl.com, но также example3.com), первый будет выбран. Вы можете использовать этот призрак как «резерв», если хотите.

### 3) Определение vhosts с использованием макроса (Apache2.4)

```
<Macro VHost $port $host>
    <VirtualHost *: $port>
        Servername $host
        DocumentRoot /srv/www/htdocs/$host
        ErrorLog /var/log/$host/error.log
    </VirtualHost>
</Macro>

Use VHost 80 example.com
Use VHost 443 secure_example.com
```

Создает два vhosts, один для порта 80, один для 443 и соответственно устанавливает используемые переменные.

### Запустить HTTPS с помощью виртуального хоста

Используйте «**Перенаправить**», чтобы заставить пользователей подключаться к защищенному URL-адресу.

```
<VirtualHost *:80>
    ServerName example.com
    SSLProxyEngine on
    Redirect permanent / https://secure_example.com/
</VirtualHost>
```

Остальную конфигурацию можно поместить в виртуальный хост ssl (порт 443), поскольку все перенаправлено.

```
<VirtualHost _default_:443>
  ServerName secure_example.com
  ServerAdmin webmaster@example.com
  DocumentRoot /var/www/domains/secure_example.com/html
  ErrorLog /var/log/secure_example.com/error.log
  CustomLog /var/log/secure_example.com/access.log common
  SSLEngine On
  ...
</VirtualHost>
```

Прочитайте [Как создать виртуальный хост в Apache онлайн](https://riptutorial.com/ru/apache/topic/4856/как-создать-виртуальный-хост-в-apache):

<https://riptutorial.com/ru/apache/topic/4856/как-создать-виртуальный-хост-в-apache>

---

# глава 4: Потолок Apache

## Вступление

Apache Flume - это инструмент для приема / обработки данных / данных для сбора агрегирования и транспортировки больших объемов потоковых данных, таких как файлы журналов, события (и т. Д.) Из разных источников в **централизованное хранилище данных** .

Flume - высоконадежный, распределенный и настраиваемый инструмент. Он в основном предназначен для копирования потоковых данных (данных журнала) с различных веб-серверов в HDFS .

## Examples

### Потоковые / журнальные данные

Как правило, большая часть данных, которые должны быть проанализированы, будет производиться различными источниками данных, такими как серверы приложений, сайты социальных сетей, облачные серверы и корпоративные серверы. Эти данные будут представлены в виде файлов журналов и событий.

Файл журнала. Как правило, файл журнала представляет собой файл, в котором перечислены события / действия, которые происходят в операционной системе. Например, веб-серверы перечисляют каждый запрос, сделанный на сервер в файлах журнала.

При сборе таких данных журнала мы можем получить информацию о -

производительности приложения и обнаружения различных сбоев программного и аппаратного обеспечения. поведения пользователя и получения лучшего понимания бизнеса. Традиционным методом передачи данных в систему HDFS является использование команды put. Давайте посмотрим, как использовать команду put.

Прочитайте Потолок Apache онлайн: <https://riptutorial.com/ru/apache/topic/9630/потолок-apache>

---

## кредиты

S. No	Главы	Contributors
1	Начало работы с apache	<a href="#">Community</a> , <a href="#">fab</a> , <a href="#">Flamewires</a> , <a href="#">hjpotter92</a> , <a href="#">James</a> , <a href="#">Katie</a> , <a href="#">Kuhan</a> , <a href="#">Nicholas Qiao</a> , <a href="#">Rich Bowen</a>
2	.htaccess файлы в Apache	<a href="#">Chintan Gor</a> , <a href="#">Deltik</a> , <a href="#">ezra-s</a> , <a href="#">Luke Bearl</a> , <a href="#">Rich Bowen</a> , <a href="#">SimpleAnecdote</a>
3	Как создать виртуальный хост в Apache	<a href="#">Chintan Gor</a> , <a href="#">Clutch</a> , <a href="#">fab</a> , <a href="#">Harikrishnan</a> , <a href="#">Hello Fishy</a> , <a href="#">Katie</a> , <a href="#">Olaf Dietsche</a>
4	Потолок Apache	<a href="#">Vinod Kumar</a>