

МДК 02.02. Web-программирование. Язык PHP

**Основы разработки серверной
части Web-приложения**

Рекомендуемые источники

1. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – СПб.: Питер, 2016.
2. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. – СПб.: БХВ-Петербург, 2015.
3. Скляр Д., Трахтенберг А. PHP. Рецепты программирования. 3-е изд. – СПб.: Питер, 2015.
4. <http://www.php.su>
5. <http://php.net/manual/ru>
6. <http://phpclub.ru>
7. youtube.com: [PHP для начинающих Кудренко](#)
8. <https://coder-booster.ru/learning/php-beginners>
9. <https://www.mysql.com/>
10. <http://www.mysql.ru>

Web-приложение – это приложение, разработанное по архитектуре «клиент-сервер», использующее в качестве клиента Веб-браузер и работающее с использованием протокола HTTP на стороне Веб-сервера.

Примеры web-приложений:

- ✓ поисковые системы (например, google.ru, rambler.ru);
- ✓ видео-коллекции (например, youtube.com);
- ✓ новостные системы (например, lenta.ru);
- ✓ масс-медиа: сайты радиостанций, телеканалов;
- ✓ интернет-магазины;
- ✓ социальные сети (например: facebook.com, vk.ru).

Этапы разработки Web-приложения

1. Определение целей сайта и требований к нему.
2. Разработка дизайн-макета.
3. Верстка дизайн-макета.
4. Программирование, установка и настройка CMS.
5. Наполнение контентом и тестирование.
6. Публикация сайта в сети.
7. Продвижение и оптимизация сайта.
8. Администрирование (поддержка) сайта.

Технология «клиент-сервер» – это технология взаимодействия, в которой одна программа (**клиент**) запрашивает выполнение какой-либо совокупности действий ("запрашивает услугу"), а другая (**сервер**) ее выполняет.

Клиенты – активные программы, с которыми обычно работает пользователь сети на своих компьютерах, отправляют запросы серверам для выполнения некоторых действий.

Серверы – пассивные программы, которые ожидают запросы от клиентов, обрабатывают их, отправляют запрашиваемую информацию и ожидают следующих запросов.

Веб-сервер – это сервер, отвечающий за прием и обработку запросов от клиентов к одному или нескольким веб-сайтам на получение определенных ресурсов.

Клиентом может выступать:

- ✓ веб-браузер;
- ✓ разнообразные программы, самостоятельно обращающиеся к веб-серверам для получения обновлений или другой информации (антивирус);
- ✓ мобильный телефон, получающий доступ к ресурсам веб-сервера при помощи протокола WAP;
- ✓ другие интеллектуальные устройства или бытовая техника.

Ресурсы – это HTML-страницы, изображения, файлы, медиа-потоки или другие данные, которые необходимы клиенту.

HTTP – это протокол передачи гипертекста.

Типы серверов

- ✓ виртуальный сервер;
- ✓ виртуальный выделенный сервер;
- ✓ выделенный сервер;
- ✓ co-location.

Виртуальный сервер

Подразумевает размещение на одном физическом сервере сайтов нескольких владельцев, между которыми распределяются совместно используемые ресурсы сервера.

Виртуальный выделенный сервер

Подразумевает размещение на одном физическом сервере сайтов нескольких владельцев, при помощи специальных программ ресурсы сервера разделяются независимо между несколькими виртуальными выделенными серверами.

Выделенный сервер

Подразумевает аренду физического сервера владельцем крупного Интернет-ресурса у хостинг-провайдера. Сервер постоянно подключен к бесперебойной сети электропитания и высокоскоростным каналам передачи данных.

Co-location

Подразумевает размещение собственного сервера клиента на технической площадке хостинг-провайдера.

Функции Web-сервера

1. прием запроса от клиента (Web-браузера);
2. поиск и передача файла гипертекста или документа в браузер;
3. контроль доступа на основе имен и паролей доступа (аутентификация и авторизация пользователей);
4. ведение регистрационного журнала (создание log-файлов) обращений пользователей к ресурсам;

Функции Web-сервера

5. административное и оперативное управление сервером;
6. поддержка динамически генерируемых страниц;
7. поддержка работы с базами данных;
8. поддержка HTTPS для защищённых соединений с клиентами, т.е. кодирование сетевого трафика (технология SSL), которое обеспечивает безопасность каналов связи между клиентами и серверами с помощью шифрования данных.

Языки для разработки клиентской части Web-приложений

HTML – язык разметки гипертекста;

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

JavaScript – разработан для создания интерактивных HTML-документов. Программный код может выполняться, как на стороне клиента, так и на стороне сервера.

Языки программирования для реализации серверной части Веб-приложения

PHP – интерпретируемый язык программирования, который используется для создания динамических и интерактивных веб-сайтов.



Интерпретация – пооператорный (покомандный, построчный) анализ, обработка и выполнение исходной программы или запроса.

Python – интерпретируемый язык для скриптов различного назначения.

Синтаксис ядра Python минималистичен.

Стандартная библиотека включает большой объём полезных функций.

Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное.



Ruby – динамический, рефлексивный, интерпретируемый
высокоуровневый язык программирования



Обзор Веб-серверов

Apache – свободный веб-сервер, наиболее часто используемый в Unix-подобных операционных системах.

Достоинства:

- ✓ гибкость конфигурации,
- ✓ надежность,
- ✓ бесплатное распространение.

IIS (Microsoft)

Достоинства:

- ✓ стабильность,
- ✓ высокая скорость работы,
- ✓ возможность подключать дополнительные модули,
- ✓ простота в установке, настройке и обслуживании.

Обзор Веб-серверов

Google Web Server – веб-сервер, разработанный компанией Google, основан на Apache и используется компанией Google для организации своей веб-инфраструктуры.

Создание рабочей среды

1. **Веб-сервер (Apache)** – программное обеспечение, отвечающее за отображение документов, запрашиваемых при наборе URL-адреса в Web-браузере.

<https://httpd.apache.org>

Последняя стабильная версия Apache **2.4.29**.

2. **Сервер баз данных (MySQL).**

<https://dev.mysql.com>

Последняя стабильная версия MySQL **5.7.20**

Создание рабочей среды

- 3. Интерпретатор PHP** – для выполнения программ, написанных на языке PHP.

php.net

Доступны версии **PHP 5.6.33, PHP 7.0.27, PHP 7.1.13** и **PHP 7.2.1**.

- 4. phpMyAdmin** – набор скриптов на PHP для управления базами данных.

Позволяет наглядно работать с базами данных.

<http://www.phpmyadmin.net>

Комплекты пакетов программ (сборки)

I. WAMP – «Windows, Apache, MySQL и PHP».

1. **Wampserver** – <http://www.wampserver.com/>

Поддерживает большое количество языков, возможность установить любую версию Apache, MySQL и PHP.

Базовый комплект:

- ✓ Apache;
- ✓ PHP;
- ✓ MySQL;
- ✓ XDebug;
- ✓ PHPMyAdmin;

Комплекты пакетов программ (сборки)

2. **Denwer** (русифицирован) – <http://www.denwer.ru/>

Малый размер и хорошая «база знаний» на форумах, прост и понятен, подходит для начинающих.

3. **Open Server** (русифицирован) – <http://open-server.ru/>

Имеет привлекательный интерфейс, обладает мощными возможностями по настройке и администрированию КОМПОНЕНТОВ.

Комплекты пакетов программ (сборки)

II. LAMP – «Linux, Apache, MySQL и PHP».

Вместо PHP могут подразумеваться другие языки, такие как Perl и Python).

III. MAMP – Mac OS

IV. SAMP – Solaris

V. WASP – Windows, Apache, SQL Server и PHP

VI. WIMP – Windows, IIS, MySQL и PHP

VII. XAMPP – кроссплатформенная сборка веб-сервера,

X – любая из четырёх операционных систем, Apache, MySQL, PHP, Perl

Настройка работы сервера Apache

Apache-сервер – свободный веб-сервер.

Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основные достоинства Apache: надёжность и гибкость конфигурации.

Структура каталогов сервера Apache, их содержание и назначение

- ✓ **bin** – содержит исполняемые файлы сервера (httpd.exe и другие);
- ✓ **conf** – содержит конфигурационный файл сервера (httpd.conf) и другие файлы конфигурации;
- ✓ **error** – папка для файлов с сообщениями об ошибках;
- ✓ **icons** – содержит изображения, используемых в листингах каталогов;

Структура каталогов сервера Apache, их содержание и назначение

- ✓ **logs** – содержит журналы регистрации посещений и ошибок, позволяющие получить подробную информацию о запросах и ошибках;
- ✓ **manual** – содержит файлы документации;
- ✓ **modules** – содержит подключаемые модули.

Настройка конфигурации сервера Apache

Система конфигурации Apache основана на текстовых конфигурационных файлах.

Файл конфигурации `httpd.conf`

Файл `httpd.conf` – это основной файл конфигурации сервера, в котором содержатся директивы, влияющие на работу сервера.

После любого изменения в файле конфигурации необходимо перезагрузить сервер.

Директива представляет собой ключевое слово, за которым следует одно или несколько значений.

Директивы бывают

- ✓ простыми (изменяющие одно свойство сервера),
- ✓ объединенные в разделы (позволяют изменять сразу несколько свойств объекта).

Символ `#` означает строку комментария

Файл конфигурации .htaccess

Используется для конфигурирования отдельных каталогов

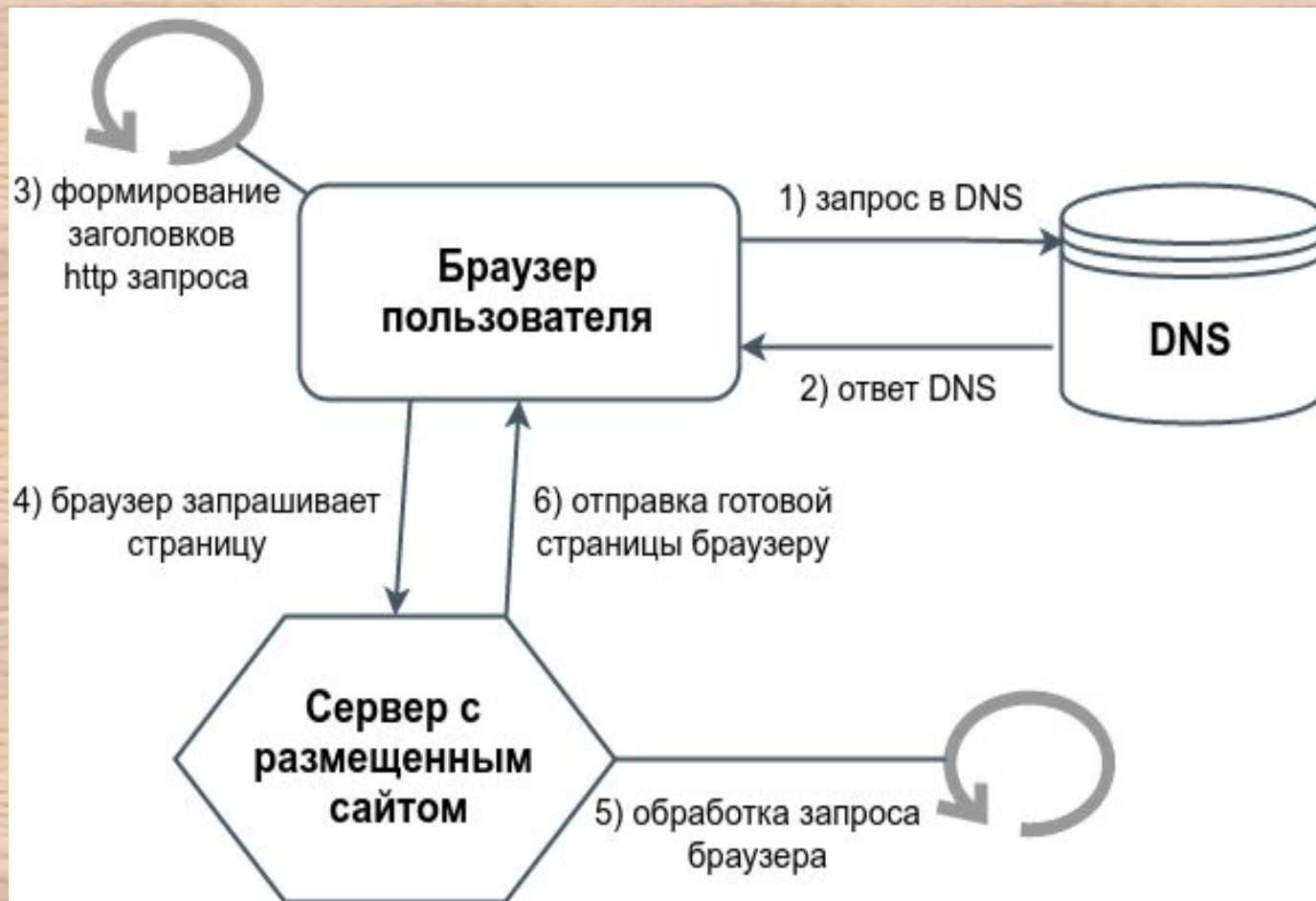
При изменении файла нет необходимости перезагружать сервер.

Файлы .htaccess анализируются при каждом запросе файла из каталога.

Настраивается, если сервер в полном распоряжении,
использование файлов `.htaccess` нужно запретить.

Файл `httpd.conf` анализируется один раз (при запуске
сервера)

Основные принципы работы сервера Apache



1. При вводе доменного имени в адресной строке браузера или при переходе по ссылке на сайте, браузер обращается к DNS.

DNS (система доменных имён) – распределенная база данных, в задачи которой входит определение IP-адреса хоста (сервера, компьютера, устройства) по его доменному имени.

2. DNS возвращает браузеру конкретный IP-адрес сервера, к которому привязан сайт.

3. Браузер формирует блок информации, именуемый HTTP-заголовками.

HTTP-заголовки — часть HTTP-запроса, которая содержит административную информацию о самом запросе, сформированная в виде ключ-значение. В заголовках содержится информация о запрашиваемой странице, IP-адресе запросившей стороны, типе её браузера и другая.

4. Браузер посылает HTTP-запрос с заголовками серверу, на котором находится сайт.

5. Веб-сервер получает запрос и обрабатывает его. Если запрошен статичный документ (изображение, медиа файл), то Apache передает его. Если же идет обращение к скрипту, то Apache запускает его на выполнение через PHP-интерпретатор, после чего выдает полученный результат.

6. После обработки запроса, сформированный ответ возвращается браузеру.