

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ

УСТАНОВКА И НАСТРОЙКА ВЕБ-СЕРВЕРА АРАСНЕ

*Учебно-методическое пособие
по дисциплине
«ВЕБ-ПРОГРАММИРОВАНИЕ»*

**Набережные Челны
2018**

Галиуллин Л.А. Установка и настройка веб-сервера Apache: учебно-методическое пособие по дисциплине «Веб-программирование» [Электронный ресурс] / Казанский федеральный университет, Электронный архив, 2018.

Рассматривается установка веб-сервера Apache на компьютер на базе Windows. Представлены описание версии языка PHP, конфигурирование веб-сервера Apache. Приведены контрольные вопросы. Для студентов направлений подготовки «Информатика и вычислительная техника», «Программная инженерия».

Введение

Язык *PHP* был разработан как инструмент для решения чисто практических задач. Его создатель, *Расмус Лердорф*, хотел знать, сколько человек читают его online-резюме, и написал для этого простенькую *CGI*-оболочку на языке Perl, т.е. это был набор Perl-скриптов, предназначенных исключительно для определенной цели – сбора статистики посещений.

CGI (*Common Gateway Interface* – общий интерфейс шлюзов) является стандартом, который предназначен для создания серверных приложений, работающих по протоколу HTTP. Такие приложения (их называют шлюзами или *CGI-программами*) запускаются *сервером* в режиме реального времени. *Сервер* передает запросы пользователя *CGI-программе*, которая их обрабатывает и возвращает результат своей работы на экран пользователя. Таким образом, посетитель получает динамическую информацию, которая может изменяться в результате влияния различных факторов. Сам шлюз (*скрипт CGI*) может быть написан на различных языках программирования – Си/C++, Fortran, Perl, *TCL*, UNIX Shell, Visual Basic, Python и др.

Вскоре выяснилось, что *оболочка* обладает небольшой производительностью, и пришлось переписать ее заново, но уже на языке *Ci*. После этого *исходники* были выложены на всеобщее обозрение для исправления ошибок и дополнения. Пользователи *сервера*, где располагался *сайт* с первой версией *PHP*, заинтересовались инструментом, появились желающие его использовать. Так что скоро *PHP* превратился в самостоятельный проект, и в начале 1995 года вышла первая известная версия продукта, называвшаяся *Personal Home Page Tools* (средства для персональной домашней страницы). Средства эти были более чем скромными: анализатор кода, понимающий всего лишь несколько специальных команд, и набор утилит, полезных для создания гостевой книги, счетчика посещений, чата и т.п.

К середине 1995 года после основательной переработки появилась вторая версия продукта, названная *PHP/FI* (*Personal Home Page / Forms Interpreter* – персональная домашняя

страница/интерпретатор форм). Она включала набор базовых возможностей сегодняшнего *PHP*, возможность автоматически обрабатывать *html-формы* и *встраиваться в html-коды*. Синтаксис *PHP/FI* сильно напоминал синтаксис Perl, но был более простым.

В 1997 вышла вторая версия Си-реализации *PHP – PHP/FI 2.0*. К тому моменту *PHP* использовали уже несколько тысяч человек повсему миру, примерно с 50 тыс. доменов, что составляло около 1% всего числа доменов *Internet*. Число разработчиков *PHP* увеличилось до нескольких человек, но, несмотря на это, *PHP/FI 2.0* все еще оставался крупным проектом одного человека. Официально *PHP/FI 2.0* вышел только в ноябре 1997 года, просуществовав до этого в основном в бета-версиях. Вскоре после выхода его заменили альфа-версии *PHP 3.0*.

PHP 3.0

PHP 3.0 была первой версией, напоминающей *PHP*, каким мы знаем его сегодня. Он очень сильно отличался от *PHP/FI 2.0* и появился опять же как инструмент для решения конкретной прикладной задачи. Его создатели, *Энди Гутманс (Andi Gutmans)* и *Зив Сураски (Zeev Suraski)*, в 1997 году переписали заново код *PHP/FI*, поскольку он показался им непригодным для разработки приложения *электронной коммерции*, над которым они работали. Для того чтобы получить помощь в реализации проекта от разработчиков *PHP/FI*, *Гутманс* и *Сураски* решили объединиться с ними и объявить *PHP3* официальным преемником *PHP/FI*. После объединения разработка *PHP/FI* была полностью прекращена.

Одной из сильных сторон *PHP 3.0* была возможность расширения ядра. Именно свойство расширяемости *PHP 3.0* привлекло внимание множества разработчиков, желающих добавить свой модуль расширения. Кроме того, *PHP 3.0* предоставляла широкие возможности для взаимодействия с базами данных, различными протоколами и *API*. Немаловажным шагом к успеху оказалась разработка нового, намного более мощного и полного синтаксиса с поддержкой *ООП*. С момента

появления *PHP 3.0* изменилась не только функциональность и внутреннее устройство языка, но и его название. В аббревиатуре *PHP* больше не было упоминания о персональном использовании, *PHP* стало сокращением (рекурсивным акронимом) от *PHP: Hypertext Preprocessor*, что значит "*PHP: препроцессор гипертекста*".

К концу 1998 года число пользователей *PHP* возросло до десятков тысяч. Сотни тысяч web-сайтов сообщали о том, что они работают с использованием этого языка. Почти на 10% серверов *Internet* был установлен *PHP 3.0*.

Официально *PHP 3.0* вышел в июне 1998 года, после 9 месяцев публичного тестирования. А уже к зиме Энди Гутманс и Зив Сураскина начали переработку ядра *PHP*. В их задачи входило увеличение производительности работы сложных приложений и улучшение модульности кода, лежащего в основе *PHP*.

Новое ядро было названо "*Zend Engine*" (от имен создателей: Zeev и Andi) и впервые представлено в середине 1999 года. *PHP 4.0*, основанный на этом ядре и принесший с собой набор дополнительных функций, официально вышел в мае 2000 года, почти через два года после своего предшественника, *PHP 3.0*. Помимо улучшения производительности, *PHP 4.0* имел еще несколько ключевых нововведений, таких как поддержка сессий, буферизация вывода, более безопасные способы обработки вводимой пользователем информации и несколько новых языковых конструкций.

В настоящее время ведутся работы по улучшению *Zend Engine* и внедрению нововведений в *PHP 5.0*, первые бета-версии которого уже вышли в свет. Одно из существенных изменений произошло в объектной модели языка, ее основательно подлатали и добавили много новых возможностей. (Комментарий пользователя: Пятая версия *PHP* была выпущена разработчиками 13 июля 2004 года. Изменения включают обновление ядра *Zend (Zend Engine 2)*, что существенно увеличило эффективность интерпретатора. Введена поддержка языка разметки *XML*. Полностью переработаны функции *ООП*, которые стали во многом схожи с

моделью, используемой в *Java*. В частности, введён *деструктор*, открытые, закрытые и защищённые члены и методы, окончательные члены и методы, интерфейсы *клонирования* объектов. В последующих версиях также были введены пространства имён, замыкания и *целый* ряд достаточно серьёзных изменений, количественно и качественно сравнимых с теми, которые появились при переходе на *PHP 5.0*. Шестая версия *PHP* разрабатывалась с октября 2006 года. Было сделано множество нововведений, как, например, *исключение* из ядра регулярных выражений *POSIX* и "длинных" суперглобальных массивов, удаление директив *safe_mode*, *magic_quotes_gpc* и *register_globals* из конфигурационного файла *php.ini*. Одним из основных новшеств должна была стать *поддержка* Юникода. Однако в марте 2010 года разработка *PHP6* была признана бесперспективной из-за сложностей с поддержкой Юникода. Исходный код *PHP6* перемещён на *ветвь*, а основной линией разработки стала версия 5.4.)

Сегодня *PHP* используется сотнями тысяч разработчиков. Несколько миллионов сайтов написаны на *PHP*, что составляет более 20% доменов *Internet*.

Возможности PHP

"*PHP* может все", – заявляют его создатели. В первую очередь *PHP* используется для создания *скриптов*, работающих на стороне *сервера*, для этого его, собственно, и придумали. *PHP* способен решать те же задачи, что и любые другие *CGI - скрипты*, в том числе обрабатывать данные *html-форм*, динамически генерировать *html* страницы и т.п. Но есть и другие области, где может использоваться *PHP*. Всего выделяют три основные области применения *PHP*.

- Первая область, как уже говорилось, – это создание приложений (*скриптов*), которые исполняются на стороне *сервера*. *PHP* наиболее широко используется именно для создания такого рода *скриптов*. Для того чтобы работать таким образом, понадобится *PHP - парсер* (т.е. обработчик *php - скриптов*) и *web-сервер* для обработки *скрипта*, браузер для просмотра результатов работы *скрипта*, ну, и, конечно, какой-либо текстовый редактор для написания самого *php -*

кода. *Парсер PHP* распространяется в виде *CGI - программы* или серверного модуля. Как *установить* его и *web-сервер* на свой компьютер, мы рассмотрим немного позднее. В этом курсе мы будем обсуждать, как правило, создание именно серверных приложений, как пример использования языка *PHP*.

- Вторая область – это создание *скриптов, выполняющихся в командной строке*. То есть с помощью *PHP* можно создавать такие *скрипты*, которые будут исполняться, вне зависимости от *web-сервера* и браузера, на конкретной машине. Для такой работы потребуется лишь *парсер PHP* (в этом случае его называют *интерпретатором* командной строки (*cli, command line interpreter*)). Этот способ работы подходит, например, для *скриптов*, которые должны выполняться регулярно с помощью различных *планировщиков задач* или для решения задач простой обработки текста.
- И последняя область – это создание *GUI*-приложений (графических интерфейсов), выполняющихся на стороне клиента. В принципе это не самый лучший способ использовать *PHP*, особенно для начинающих, но если вы уже досконально изучили *PHP*, то такие возможности языка могут оказаться весьма полезны. Для применения *PHP* в этой области потребуется специальный инструмент – *PHP-GTK*, который является расширением *PHP*.

Итак, область применения *PHP* достаточно обширна и разнообразна. Тем не менее существует множество других языков программирования, способных решать похожие задачи. Почему стоит изучать *PHP*? Что это нам дает? Во-первых, *PHP* очень прост в изучении. Достаточно ознакомиться лишь с основными правилами синтаксиса и принципами его работы, и можно начинать писать собственные *программы*, причем браться за такие задачи, решение которых на другом языке требовало бы серьезной подготовки.

Во-вторых, *PHP* поддерживается почти на всех известных платформах, почти во всех операционных системах и на самых разных *серверах*. Это тоже очень важно. Вряд ли кому-то

захочется переходить, например, от работы под *Windows* к работе под *Linux* или от *сервера IIS* к *серверу Apache* только для того, чтобы изучить еще один язык программирования.

В *PHP* сочетаются две самые популярные парадигмы программирования – объектная и процедурная. В *PHP4* более полно поддерживается *процедурное программирование*, но есть возможность писать программы и в объектном стиле. Уже в первых пробных версиях *PHP5* большинство недочетов в реализации объектно-ориентированной модели языка, существующих в *PHP4*, устранены. Таким образом, можно выбрать наиболее привычный стиль работы.

Если говорить о возможностях сегодняшнего *PHP*, то они выходят далеко за рамки тех, что были реализованы в его первых версиях. С помощью *PHP* можно создавать изображения, *PDF*-файлы, флэш-ролики, в него включена поддержка большого числа современных баз данных, встроены функции для работы с текстовыми данными любых форматов, включая *XML*, и функции для работы с файловой системой. *PHP* поддерживает взаимодействие с различными сервисами посредством соответствующих протоколов, таких как *протокол управления доступом к директориям LDAP*, *протокол работы с сетевым оборудованием SNMP*, *протоколы передачи сообщений IMAP, NNTP и POP3*, *протокол передачи гипертекста HTTP* и т.д.

Обращая внимание на взаимодействие между различными языками, следует упомянуть о поддержке объектов *Java* и возможности их использования в качестве объектов *PHP*. Для доступа к удаленным объектам можно использовать расширение *CORBA*.

Для работы с текстовой информацией *PHP* унаследовал (с небольшими изменениями) механизмы работы с регулярными выражениями из языка *Perl* и *UNIX*-систем. Для обработки *XML*-документов можно использовать как стандарты *DOM* и *SAX*, так и *API* для *XSLT*-трансформаций.

Для создания приложений *электронной коммерции* существует ряд полезных функций, таких как функции осуществления платежей *Cybercash*, *CyberMUT*, *VeriSign Payflow Pro* и *CCVS*.

Установка и настройка ПО

Возможности языка мы обсудили, области применения рассмотрели, *историю* изучили. Теперь можно приступать к *установке* необходимого инструментария. Поскольку в качестве практической основы курса мы будем рассматривать задачи, решаемые с помощью технологии клиент-*сервер*, и *PHP* соответственно будет использоваться для создания *скриптов*, обрабатываемых *сервером*, нам нужно *установить web-сервер* и *интерпретатор PHP*. В качестве *web-сервера* выберем, например, *Apache*, как наиболее популярный среди *web-разработчиков*. Для просмотра результатов работы *программ* нам понадобится *web-браузер*, например *Internet Explorer*.

Установка Apache под Windows

Чтобы что-нибудь *установить*, нужно для начала иметь соответствующее программное обеспечение (ПО). Скачать ПО для *установки Apache* можно, например, с его официального сайта <http://www.apache.org> Мы скачали файл *apache_1.3.29-win32-x86-no_src.exe*. Это автоматический установщик (иначе – *wizard*) *сервера Apache* под Windows. Эта *программа* попытается почти самостоятельно (а точнее, с минимальными усилиями с вашей стороны) *установить* на компьютер какое-либо программное обеспечение, а в данном случае *сервер*. После запуска файла установщика появляется следующее окошко (рис. 1).

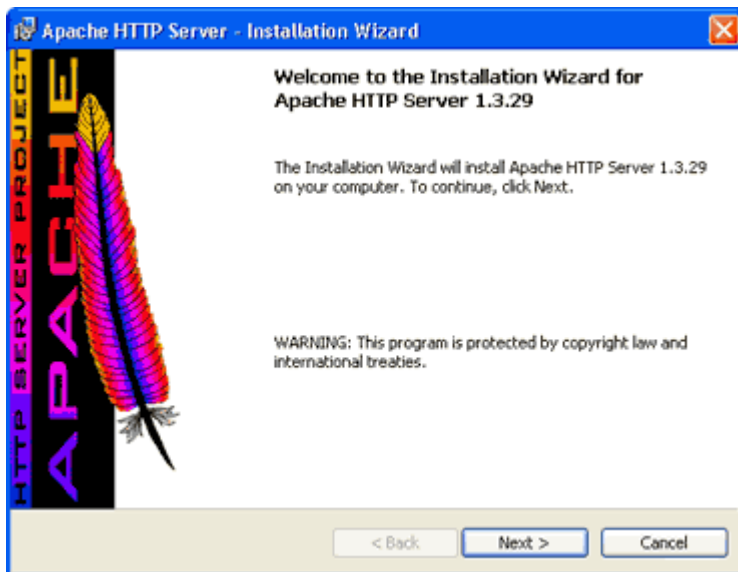


Рис. 1. Автоматическая установка сервера Apache

Чтобы *установить* HTTP-сервер Apache версии 1.3.29 на свой компьютер, нужно нажать на кнопку Next. Кстати говоря, эта *же программа* позволит изменить или удалить уже *установленный web-сервер*.

После нажатия кнопки Next *программа* предложит согласиться с условиями лицензии (рис. 2).

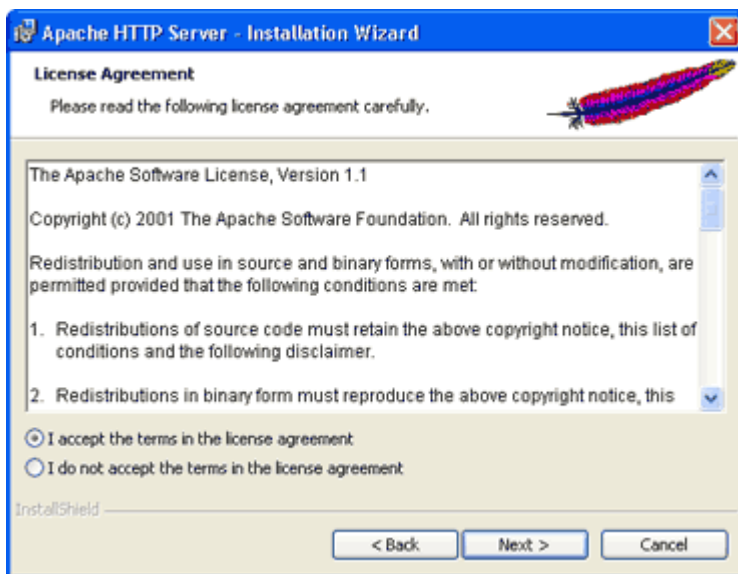


Рис. 2. Лицензионное соглашение

Следующий экран будет содержать информацию о сервере *Apache*, и в частности о его Windows-реализации (его изображение не приводим).

На следующем шаге нужно ввести имя сетевого домена, имя сервера и e-mail администратора. Программа попытается автоматически определить ваш домен и хост по настройкам компьютера (рис. 3).

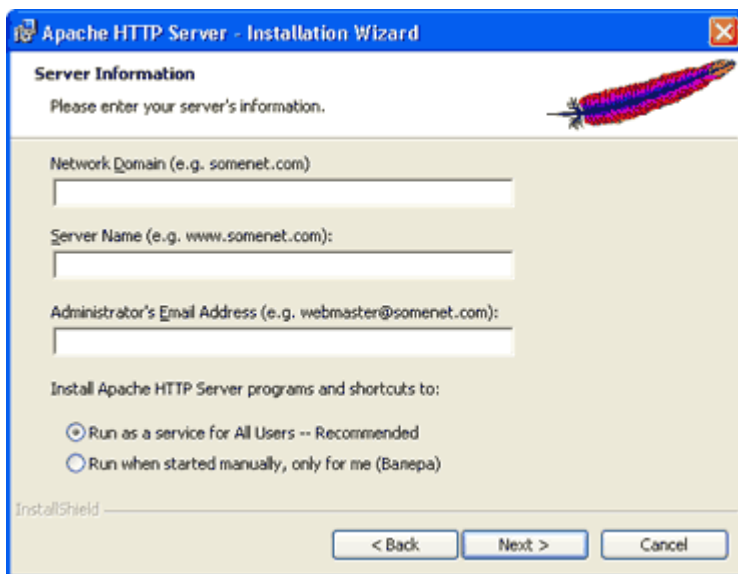


Рис. 3. Основная информация о сервере

После того как вы ввели данные в вышеприведенную форму, нужно выбрать тип *установки*: полная (устанавливаются все компоненты *сервера*) или определяемая пользователем (можно выбрать компоненты для *установки*) (рис. 4).

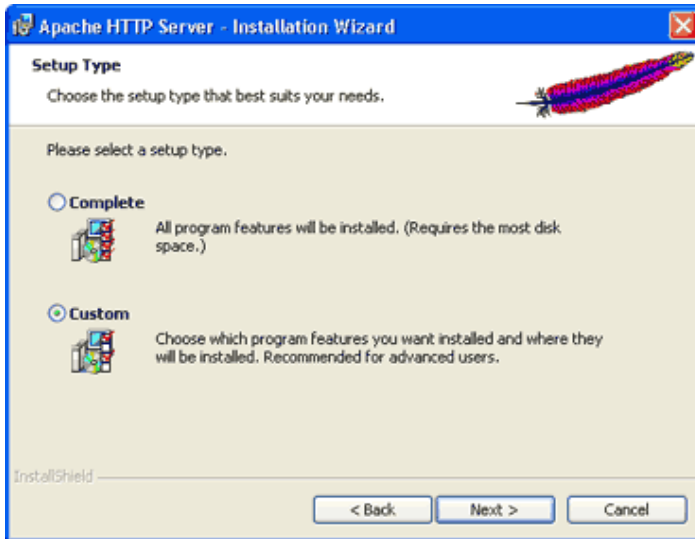


Рис. 4. Тип установки

Выбор компонентов *сервера* не очень большой – это инструменты, необходимые для работы *сервера* и документация к нему (рис. 5).

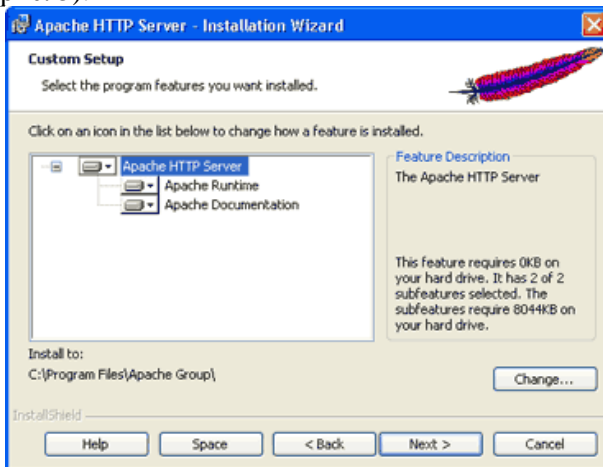


Рис. 5. Выбор компонент пользовательской установки

Мы выберем полную *установку*. Тогда на следующем шаге *программа* предложит выбрать папку, в которую будет *установлен сервер*. По умолчанию *сервер* устанавливается в папку `c:\Program Files\Apache Group\`.

Контрольные вопросы

1. Что Вы знаете о PHP?
2. Что Вы знаете о Perl?
3. Что Вы знаете о HTTP?
4. Что Вы знаете о CGI?
5. Что Вы знаете о HTML?
6. Что Вы знаете о Python?
7. Что Вы знаете об Apache?
8. Что Вы знаете об установке Apache под Windows?
9. Что Вы знаете об автоматической установке сервера Apache?
10. Что Вы знаете о типах установки Apache?

Рекомендуемые источники

1. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2017. - 416 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=336649>.
2. Гагарина Л.Г. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2015. - 400 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=389963>.
3. Голицына О. Л. Программирование на языках высокого уровня: Учебное пособие / О.Л. Голицына, И.И. Попов. - М.: Форум, 2016. - 496 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=139428>.