Лорел посвящается

ОГЛАВЛЕНИЕ

ОБ АВТОРЕ

Джош Локхарт (Josh Lockhart) – создатель Slim Framework (http://slimframework.com/), популярного микрофреймворка для PHP, обеспечивающего быструю разработку веб-приложений и программных интерфейсов. Джош также основал и до сих пор курирует «PHP The Right Way» (http://www.phptherightway.com/) – популярную в PHP-сообществе инициативу, поощряющую современные методики разработки и распространяющую качественную информацию для PHP-разработчиков по всему миру.

Джош работает программистом в New Media Campaigns (http://www.newmediacampaigns.com/), агентстве по предоставлению полного комплекса услуг веб-дизайн, разработки и маркетинга в Каррборо, Северная Каролина. Получает удовольствие от создания приложений с использованием HTML, CSS, PHP, JavaScript, Bash и различных фреймворков управления контентом.

Окончил курс Information and Library Science (http://sils.unc.edu/) в Университете Северной Каролины, в Чапел-Хилл, в 2008. В настоящее время проживает в Чапел-Хилл, штат Северная Каролина, вместе со своей замечательной женой Лорел и двумя их собаками.

Джоша можно найти в Твиттере (https://twitter.com/codeguy). Его блог доступен по адресу https://joshlockhart.com, а его проекты с открытым исходным кодом – на GitHub (https://github.com/codeguy).

ПРЕДИСЛОВИЕ

В Интернете можно найти миллионы электронных пособий по PHP. Большинство из них устарели и содержат описание отработавших свое технологий. Но Google продолжает выдавать ссылки на эти пособия, обеспечивая им бессмертие. Устаревшая информация не приносит пользы начинающим PHP-программистам, которые, основываясь на ней, создают медленные и ненадежные PHP-приложения. Я осознал эту проблему в 2013 году, и это явилось основной причиной появления инициативы «PHP The Right Way» (http://www.phptherightway.com/) по обеспечению доступа к качественной актуальной информации от авторитетных членов PHP-сообщества.

Книга «Современный PHP» служит той же цели. Эта книга не справочное руководство. Нет. Эта книга представляет собой дружеский и живой разговор между нами. Я познакомлю вас с современным языком программирования PHP. Расскажу о новейших PHP-технологиях, которыми ежедневно пользуюсь в работе и в своих проектах с открытым исходным кодом. И помогу вам начать использовать новейшие стандарты программирования, освоив которые, вы сможете распространять свои PHP-компоненты и библиотеки среди членов PHP-сообщества.

Я часто буду упоминать слово «сообщество», повторяя его снова и снова. PHP-сообщество дружелюбно, приветливо и всегда готово прийти на помощь, хотя, иногда встречаются досадные исключения. Если вам захочется узнать больше об определенной особенности, упомянутой в этой книге, обратитесь в местную группу пользователей PHP с возникшими вопросами. Я уверен, что вокруг вас найдутся PHP-разработчики, которые захотят помочь вам лучше узнать PHP. Ваша местная группа PHP-пользователей это бесценный источник, который позволит вам совершенствоваться в PHP и после завершения чтения этой книги.

Что нужно знать об этой книге

Прежде чем начать, я хочу, остановиться на том, что вы найдете в этой книге. Во-первых, я не смогу описать *все* подходы, используемые

в PHP. На это просто не хватит времени. Вместо этого, я расскажу, *как я пользуюсь PHP*. Да, это мой личный взгляд на вещи, но я использую те же методы и стандарты, что и многие другие PHP-разработчики. Все, что вы вынесете из нашей краткой беседы, вы сможете сразу же применить в своих проектах.

Во-вторых, я предполагаю, что вы знакомы с переменными, условными операторами, циклами и так далее. От вас не требуется знание языка PHP, но вы должны, по крайней мере, иметь базовое представление об этих фундаментальных понятиях программирования. Я не буду против, если вы захватите с собой кофе (я очень люблю кофе). А я принесу с собой все, что там полагается к кофе.

И в-третьих, я не настаиваю на использовании какой-либо конкретной операционной системы. Однако примеры кода написаны мной для Linux. Bash-команды, которые вы увидите в книге, я использую в Ubuntu и CentOS, но они также будут работать в OS X. Если вы пользуетесь Windows, настоятельно рекомендую установить и настроить виртуальную машину с Linux для опробования примеров кода, прилагаемых к этой книге.

Структура книги

Часть I посвящена описанию новых возможностей языка PHP, таких как пространства имен, генераторы и трейты (traits). Она познакомит вас с современным языком PHP и его особенностями, с которыми вы, возможно, до сих пор не сталкивались.

Часть II рассматривает передовые технологии, которые обязательно следует применять в своих PHP-приложениях. Вы слышали о *PSR*, но не совсем понимаете, что это такое и как им пользоваться? Вы хотите узнать, как безопасно обработать пользовательский ввод и выполнять защищенные запросы к базе данных? Тогда вам стоит ее прочесть.

Часть III содержит больше технических подробностей, чем первые две части. Она посвящена развертыванию, настройке, тестированию и профилированию PHP-приложений. Мы углубимся в методику развертывания с помощью Capistrano. Поговорим об инструментах тестировании, таких как PHPUnit и Travis CI. Обсудим настройку PHP и ее влияние на работу вашего приложения.

Приложение A содержит пошаговые инструкции по установке и настройке PHP-FPM.

Приложение B описывает процедуру создания локальной среды разработки максимально приближенной к среде действующего сервера. Мы познакомимся с Vagrant, Puppet, Chef и альтернативными им инструментами для быстрого начала работы.

Соглашения, принятые в этой книге

В этой книге приняты следующие типографские соглашения:

Курсив

Используется для обозначения новых терминов, адресов электронной почты, имен файлов и расширений имен файлов .

`Моноширинный`

Применяется для оформления листингов программ и программных элементов внутри обычного текста, таких как имена переменных и функций, типов данных, переменных окружения, инструкций и ключевых слов .

`Моноширинный жирный`

Обозначает команды или другой текст, который должен вводиться пользователем.

`Моноширинный курсив`

Обозначает текст, который должен замещаться фактическими значениями, вводимыми пользователем или определяемыми из контекста .

Так обозначается совет или рекомендация.

Так будут выделены общие примечания.

Так будут выделены предупреждения и предостережения.

Использование примеров кода

Сопроводительные материалы (примеры кода, упражнения и т. д.) можно загрузить на странице https://github.com/codeguy/modern-php.

Данная книга призвана оказать вам помощь в решении ваших задач. Вы можете свободно использовать примеры программного кода

из этой книги в своих приложениях и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части программного кода. Например, если вы разрабатываете программу и используете в ней несколько отрывков программного кода из книги, вам не нужно обращаться за разрешением. Однако в случае продажи или распространения компакт-дисков с примерами из этой книги вам необходимо получить разрешение от издательства O'Reilly. Если вы отвечаете на вопросы, цитируя данную книгу или примеры из нее, получение разрешения не требуется. Но при включении существенных объемов программного кода примеров из этой книги в вашу документацию необходимо получить разрешение издательства.

Мы приветствуем, но не требуем добавлять ссылку на первоисточник при цитировании. Под ссылкой на первоисточник мы подразумеваем указание авторов, издательства и ISBN. Например: «*Modern PHP* by Josh Lockhart (O'Reilly). Copyright 2015 Josh Lockhart, 978-1-491-90501-2.».

За получением разрешения на использование значительных объемов программного кода примеров из этой книги обращайтесь по адресу *permissions@oreilly.com*.

Как связаться с нами

С вопросами и предложениями, касающимися этой книги, обращайтесь в издательство:

O'Reilly Media, Inc.
1005 Gravenstein Highway North Sebastopol, CA 95472
800-998-9938 (США или Канада)

Список опечаток, файлы с примерами и другую дополнительную информацию вы найдете на странице книги http://bit.ly/modern_php.

Свои пожелания и вопросы технического характера отправляйте по адресу *bookquestions@oreilly.com*.

Мы в Facebook: http://facebook.com/oreilly
Мы в Twitter: http://twitter.com/oreillymedia
Мы в YouTube: http://www.youtube.com/oreillymedia

Благодарности

Это моя первая книга. Когда О'Рейли (O'Reilly) предложил мне написать книгу «Современный PHP», это сильно взволновало меня и перепугало до смерти. Первое, что я сделал, исполнил танец Уолтера Хьюстона (Walter Huston). *О'Рейли* предложил *мне* написать книгу.

Это круто!? И тогда я спросил себя, а *смогу ли я написать так много страниц?* Написание книги долгий и трудоемкий процесс.

Конечно же, я сразу ответил «да». Я знал, что смогу написать книгу «Современный PHP», потому что у меня есть семья, друзья, сослуживцы, редакторы, рецензенты, которые и помогли мне во всем. Я хочу выразить признательность и поблагодарить всех своих сторонников за неоценимую поддержку. Без них эта книга никогда бы не была написана.

Прежде всего, я хочу поблагодарить моего редактора Эллисон Макдональд (Allyson MacDonald, *@allyatoreilly*) из O'Reilly Media. Элли мила, требовательна, благосклонна и умна. Она точно знает, как и когда мягко подтолкнуть меня в правильном направлении, когда я терял нить повествования. Я не могу себе представить лучшего редактора.

Я также хочу поблагодарить моих технических рецензентов Адама Феирхолма (Adam Fairholm, *@adamfairholm*) и Эда Финклера (Ed Finkler, *@funkatron*). Адам – блестящий веб-разработчик из Newfangled (https://www.newfangled.com/) и, пожалуй, самой известной его работой является IMVDb (http://imvdb.com/) – популярная база данных музыкальных клипов. Эд хорошо известен в PHP-сообществе за его невероятные навыки PHP-разработчика, его персональный подкаст */dev/hell* (http://devhell.info/) и его заслуживающую всяческих похвал компанию Open Sourcing Mental Illness (http://funkatron.com/osmi). Адам и Эд указали мне на все неясные, нелогичные и ошибочные моменты в моих черновиках. Эта книга стала намного лучше благодаря их замечаниям. Я навсегда в долгу перед ними за их советы и здравый смысл. Если ошибки или неточности все же просочились в окончательный вариант рукописи, это, безусловно, моя оплошность.

Мои коллеги из New Media Campaigns (http://www.newmediacampaigns.com/) были для меня постоянным источником вдохновения. Джоэл, Клей, Крис, Алекс, Патрик, Эшли, Ленни, Клэр, Тодд, Паскаль, Генри и Натан – снимаю шляпу перед вами за добрые ободряющие слова.

И, самое главное, я хочу поблагодарить мою семью: Лорел, Итан, Тесса, Чарли, Лайза, Гленн и Лиз. Спасибо вам за поддержку, без которой я бы никогда не закончил эту книгу. Моей любимой жене Лорел отдельное спасибо за терпение. Спасибо тебе за кофе Caribou, когда я засиживался за рукописью допоздна. Спасибо за понимание, когда я писал в выходные дни. Спасибо за сохранение моего творческого настроя. Я люблю тебя сейчас и навсегда.

ЧАСТЬ I

Особенности языка

ГЛАВА 1. Новый PHP

Язык PHP переживает ренессанс. PHP трансформируется в современный язык сценариев с такими полезными особенностями, как пространства имен, трейты (traits), замыкания и встроенное кэширование байт-кода. Экосистема современного языка PHP также развивается. PHP-разработчики все реже используют монолитные фреймворки и все чаще небольшие специализированные компоненты. Менеджер зависимостей Composer внес революционные изменения в построение PHP-приложений. Он вывел нас из обнесенного неприступными стенами рассадника фреймворков и позволил смешивать и сочетать взаимодействующие между собой PHP-компоненты, лучше всего соответствующие нуждам PHP-приложений. Совместимость компонентов была бы невозможной без предложенных сообществом стандартов и их курирования со стороны PHP Framework Interop Group.

Книга «Современный PHP» является вашим путеводителем по новому PHP, она расскажет, как создавать и развертывать поразительные PHP-приложения, используя стандарты сообщества, лучшие методики и совместимые компоненты.

Прошлое

Прежде, чем приступать к знакомству с современным языком PHP, важно больше узнать о его происхождении. PHP – язык сценариев, интерпретируемый на стороне сервера. Это значит, что вы должны написать PHP-код, выгрузить его на веб-сервер и запустить с помощью интерпретатора. Язык PHP, как правило, используется в паре с веб-серверами Apache или nginx для поддержки динамического контента. Кроме того, язык PHP можно использовать для создания мощных приложений, запускаемых из командной строки (по аналогии с программами на bash, Ruby, Python и т. д.). Многие PHP-разработчики

не знают об этом и упускают из виду эту действительно интересную особенность. Но вы не должны входить в их число.

Вы можете прочитать официальную историю PHP на странице http://php.net/manual/ history.php.php. Я не буду повторять, что так хорошо сказано создателем PHP Расмусом Лердорфом (Rasmus Lerdorf). Я просто скажу, что PHP имеет противоречивое прошлое. Язык PHP берет начало с коллекции CGI-сцеариев, написанных Расмусом Лердорфом для отслеживания посещений его электронного резюме. Лердорф назвал свой набор CGI-сценариев «Инструментами личной домашней страницы» («Personal Home Page Tools»). Это раннее воплощение полностью отличается от сегодняшнего языка PHP, который мы знаем. Изначально инструменты PHP Лердорфа не были языком сценариев; они были инструментами поддержки элементарных переменных и автоматической интерпретации форм, основанных на встроенном синтаксисе HTML.

Между 1994 и 1998 годами, PHP претерпел множество изменений и даже несколько раз был переписан с нуля. Энди Гутманс (Andi Gutmans) и Зеев Сураски (Zeev Suraski), два программиста из Тель-Авива, объединили свои усилия с Расмусом Лердорфом, чтобы превратить PHP из небольшого набора CGI-инструментов в полноценный язык программирования с последовательным синтаксисом и базовой поддержкой объектно-ориентированного программирования. Они назвали свой конечный продукт PHP 3 и выпустили его в конце 1998 года. Новое название «PHP» не совпадало с предыдущим и представляло собой акроним от Hypertext Preprocessor (гипертекстовый процессор). PHP 3 был первой версией, которая уже походила на сегодняшний PHP. Она обеспечивала превосходную расширяемость для поддержки различных баз данных, протоколов и программных интерфейсов. Именно расширяемость PHP 3 привлекла к проекту внимание многих новых разработчиков. К концу 1998 года PHP 3 уже был установлен на ошеломляющих 10% всех веб-серверов.

Настоящее

Сегодня язык PHP быстро развивается и поддерживается десятками программистов со всего мира, входящих в состав основной команды разработчиков. Методики разработки на PHP также меняются. В прошлом программист обычно писал сценарий на PHP и выгружал его на действующий сервер по FTP, в надежде, что он заработает. Такая дикая по современным понятиям стратегия разработки использовалась из-за отсутствия средств локальной разработки.

В настоящее время, мы отказались от FTP, заменив его инструментами управления версиями. Программное обеспечение управления версиями, например Git, позволяет управлять историей развития программного кода, его ветвлением и слиянием ветвей. Локальные среды разработки стали идентичны действующим серверам, благодаря инструментам виртуализации, таким как Vagrant, и инструментам удаленного комплектования, таким как Ansible, Chef и Puppet. Мы используем специализированные PHP-компоненты с помощью менеджера зависимостей Composer. Наш PHP-код соответствует общественным стандартам PSR, курируемых PHP Framework Interop Group. Мы тщательно тестируем свой код с помощью инструментов, таких как PHPUnit. Развертываем свои приложения с помощью менеджера PHP-процессов FastCGI на веб-серверах, таких как nginx. И увеличиваем производительность приложений с помощью кэширования байт-кода.

Современный язык PHP поддерживает множество новых методов, которые могут быть вам незнакомы, они либо являются новыми для PHP либо уже имелись в старых версиях PHP и претерпели существенные изменения. Но это не должно вас волновать. Я подробно остановлюсь на каждом таком методе в этой книге.

Я очень рад, что PHP теперь имеет официальную проектную спецификацию, которой так не хватало до 2014 года.

Большинство зрелых языков программирования имеют спецификацию. Говоря простыми словами, спецификация PHP – это набор правил, определяющих, что именно значит быть языком PHP. Спецификация предназначена для разработчиков, создающих программы для анализа, интерпретации и выполнения кода PHP. Она не предназначена для разработчиков приложений и веб-сайтов на PHP.

Сара Гоулман (Sara Golemon) и Facebook анонсировали первую проектную спецификацию PHP на конференции OSCON O'Reilly в 2014 году Официальный анонс можно найти во внутреннем списке рассылки PHP (http://bit.ly/php-internals), а спецификацию PHP – на GitHub (http://bit.ly/php-langspec).

Официальная спецификация языка PHP становится еще более важной, с учетом появления нескольких конкурирующих между собой движков PHP. Оригинальным движком PHP является Zend Engine (http://www.zend.com/en/company/community/php/) – PHP-

интерпретатор, написанный на C и введенный в PHP 4. Движок Zend Engine был создан Расмусом Лердорфом, Энди Гутманом и Зеев Сураски. Сегодня Zend Engine является основным вкладом компании Zend в PHP-сообщество. Однако в настоящее время появился еще один крупный движок – HipHop Virtual Machine от Facebook. Спецификация языка гарантирует, что оба движка будут обладать базовой совместимостью.

PHP-движок – это программа для анализа, интерпретации и выполнения PHP-кода (например, Zend Engine или HipHop Virtual Machine). Его не следует путать с PHP – названием языка PHP в общем смысле.

Будущее

Движок Zend Engine развивается быстрыми темпами, обрастая новыми функциями и улучшая производительность. Я объясняю ускоренное прогрессирование движка Zend Engine появлением новых конкурентов, а именно движка *HipHop Virtual Machine* от компании Facebook и языка программирования *Hack*.

Hack является новым языком программирования, надстройкой над PHP. Он вводит статическую типизацию, новые структуры данных и дополнительные интерфейсы, сохраняя при этом, обратную совместимость с существующей динамической типизацией в PHP. Язык Hack ориентирован на разработчиков, которые ценят быстроту разработки, присущую PHP, но нуждаются в предсказуемости и стабильности статической типизации.

Мы обсудим преимущества и недостатки динамической и статической типизации в этой книге ниже. Разница между ними состоит в моменте проверки типов интерпретатором PHP. Динамические типы проверяются во время выполнения, а статические – во время компиляции. Дополнительную информацию вы найдете в главе 12.

Движок HipHop Virtual Machine (HHVM) – это интерпретатор языков PHP и Hack, выполняющий *динамическую компиляцию* (just-in-time, сокращенно JIT) для улучшения производительности приложений и экономии памяти.