Моему сыну: Амину Башар Абдул-Джаваду. Я всегда буду любить тебя.

Краткое содержание

ЧАСТЬ 1 ■ ■ ■ Groovy в примерах

ЧАСТЬ 2 ■ ■ ■ Grails в примерах

Содержание

ЧАСТЬ 2 Grails в примерах

Об авторе

БАШАР АБДУЛ-ДЖАВАД является ведущим специалистом по программному обеспечению компании Video Monitoring Services (VMS, `http://vmsinfo.com`), которая предоставляет решения в области мониторинга новостей и рекламы. В данной должности Башар перенёс все новые проекты компании с Java и Tapestry на Groovy и Grails. В Нью-Йорке, Аризоне и Ченнае (Индия) Башар обучал разработчиков VMS использованию Groovy и Grails, а также мышлению в терминах Groovy, а не Java. И по сей день Башар всё ещё выступает там на еженедельных обучающих семинарах по темам, относящимся к Groovy, Grails и динамическим языкам.

Получив степень магистра компьютерных наук университета штата Мэн, Башар переехал в солнечный Тусон на работу в Университет штата Аризона в качестве ведущего разработчика Гидрологической информационной системы Аризоны (Arizona Hydrologic Information System, AHIS). AHIS была построена на основе Struts. С растущим разочарованием от излишней сложности Struts и недостатков Java Башар начал поиски более простого и динамического языка, а также фреймворка, который бы функционировал на базе виртуальной машины Java. Вот тогда он и открыл для себя Groovy и Grails.

Когда Башар пришёл в VMS, он принёс с собой и увлечённость Groovy и Grails. В VMS использовали сложный Web-фреймворк: Tapestry. Башар сделал своей целью переход компании на Groovy и Grails, убедив её менеджмент в том, что после нескольких лет нарастающей сложности Tapestry разработчики будут рады работать с Groovy и Grails, да и эффективнее это по крайней мере вдвое.

В дополнение к степени магистра Башар имеет степень бакалавра компьютерных наук Иорданского университета. Также Башар обладает сертификатами Sun-certified Java 1.4 Programmer и Java 1.4 Web Components Developer.

О техническом обозревателе

ДЕЙВ КЛЕЙН является разработчиком компании Contegix, которая специализируется на предоставлении управляемых Интернет-инфраструктур, основанных на Linux, Mac OS X, Java EE и Grails. Дейв вовлечён в разработку корпоративных приложений в течение последних 15 лет. Он работал в качестве разработчика, архитектора, руководителя проекта (не волнуйтесь, он выздоровел), наставника и инструктора. Дейв появляется в различных группах пользователей и на национальных конференциях. Также он является основателем Capital Java User Group в городе Мэдисон штата Висконсин.

Дейв считает себя кочующим программистом. Поработав в Калифорнии, Миннесоте, Техасе и Висконсине, добрался до Миссури. Сейчас он живёт в городе Портидж штата Висконсин со своей женой и 13 будущими консультантами. С размышлениями Дейва по поводу Groovy и Grails можно ознакомиться в его блоге `http://dave-klein.blogspot.com`.

Благодарности

Прежде всего я хотел бы поблагодарить мою семью (маму, папу, брата-близнеца, сестру, её мужа, мою маленькую племянницу и дядю в Чикаго) за их постоянную поддержку, любовь, мудрость, советы, терпение и заботу. Всем, чему я научился в этой жизни, я обязан моим родителям. Без них я бы никогда не был тем, кем являюсь сейчас.

Особенная благодарность моей любимой девушке Лесли за её нескончаемую поддержку во время написания этой книги. При том времени, что уходит на написание книги, я никогда не забуду её понимания и поддержки на протяжении всей работы. Встреча с ней – большая удача в моей жизни.

В Apress я хотел бы поблагодарить Стива Энглина, ведущего редактора по новым поступлениям, за веру в меня и мои способности при написании книги Groovy и Grails. Тома Велша, редактора по развитию, за конструктивную критику моего английского. Дейва Клейна, технического обозревателя, за понимание и советы. Кайли Джонстон, ведущего руководителя проекта, за быстрые напоминания о постоянных задержках материала. Без неё эта книга никогда бы не была опубликована вовремя. Также хотел бы поблагодарить Шарон Вилки (корректор) и Келли Гюнтер (выпускающий редактор). Все, с кем я работал в Apress, были очень дружелюбны, деликатны и преданы своей работе.

Также я хотел бы поблагодарить моих коллег из VMS. Скотта Сигала, моего менеджера, за его одобрение Groovy и Grails и предоставление мне возможности использовать их в работе. Джерри Лью, директор по информационным технологиям, за то, что прислушался к рекомендациям Скотта по Groovy и Grails. И Криса Тилмана за чтение корректуры первых трёх глав этой книги и понимание.

Наконец, я должен поблагодарить талантливых в Groovy и Grails людей: Дирка Кёнига, Эндрю Гловера, Пола Кинга, Гийома Лафоре и Джона Скита – авторов *Groovy in Action*, очень полезного руководства при написании этой книги, и Грэма Роше – основателя Grails и автора *The Definitive Guide to Grails*. Спасибо большое за столь изумительный Web-фреймворк. Также я хотел бы поблагодарить очень активное сообщество Groovy и Grails. Ваша помощь в виде почтовых рассылок оценена по достоинству.

Введение

Платформа Java топчется на месте довольно долгое время. При этом язык программирования Java начинает устаревать. Сейчас для Java-разработчиков наступает время к переходу в сторону мышления в терминах динамических языков. Groovy является одним из самых лучших динамических языков доступных на платформе Java. После нескольких лет работы с Groovy я твёрдо убедился, что всем Java-разработчикам следует по крайней мере познакомиться с Groovy. Количество кода, которое можно сократить с помощью динамического языка подобного Groovy, действительно поражает, особенно, работая с коллекциями и файлами. Именно по этой причине я решил написать эту книгу. Я хочу поделиться с Java-разработчиками огромным выигрышем в эффективности, которая достигается с помощью Groovy.

Динамические языки подобные Groovy делают реальностью такие фреймворки как Grails. Grails – это глоток свежего воздуха для Java-разработчиков и это одна из основных причин моей заинтересованности динамическими языками. Я помню первые дни разработки на Java с использованием Struts и Tapestry. Но я по ним не скучаю. Лично мне эти фреймворки всегда казались излишне сложными. Я просто не мог определить объёмы конфигурационного и шаблонного кода, необходимые для того, чтобы хоть что-то сделать. Это не те фреймворки, о которых стоит думать. Фреймворки должны делать вещи проще и позволять сосредоточиться на логике. Это то, что делает Grails. Grails является осмысленной технологией, что для меня является первым, что я ищу в новой технологии. Grails является столь простым и одновременно мощным фреймворком, что нельзя не удивляться, почему о нём никто не подумал раньше.

Одна из сильных сторон Groovy и Grails – это то, что виртуальная машина Java является для них «родной». Учитывая вездесущность Java в наше время, было бы глупо предлагать Java-разработчикам отказаться от всей Java-инфраструктуры, APIs, библиотек и фреймворков и начать всё сначала. По этой причине Groovy и Grails обязаны быть успешными в мире корпоративных приложений, где платформа Java успешно обосновалась. Их идеальная интеграция с Java – огромное подспорье в распространении. В одной из организаций мы обсуждали, следует ли использовать Ruby on Rails или Groovy и Grails. К концу дня победу одержали Groovy и Grails. Их отличное взаимодействие с Java и лёгкость изучения для Java-разработчиков стали ключевыми факторами, повлиявшими на окончательное решение.

В этой книге у меня две цели. Во-первых, на практике научить Groovy и Grails с нуля. Во-вторых, предоставить практические решения частых проблем в Groovy и Grails. Я хотел бы, чтобы Вы могли взять книгу и, выбрав интересующую проблему, быстро найти

подходящее решение. Вы никогда не найдёте подробных теоретических выкладок о том, как же всё устроено изнутри, а непосредственно небольшие фрагменты кода для решения проблемы.

Я надеюсь, Вы получите такое же удовольствие от прочтения книги, как я при её написании. С Groovy и Grails действительно удобно работать. Я не припомню, чтобы получал от работы с технологией такое удовольствие, как от работы с Groovy и Grails.

Для кого эта книга

Эта книга нацелена напрямую Java-разработчикам. Она не подразумевает предварительного знакомства с Groovy и Grails. И будет преподносить материал в виде вопросов и ответов. У разработчиков, не использовавших Java, но знакомых с динамическими языками (такими как Ruby, PHP или Python), также не должно возникнуть проблем при чтении книги, хотя Java-разработчики, вероятно, извлекут из неё больше пользы.

Grails – это больше, чем Web-фреймворк. Это множество приложений, которые собирают вместе другие технологии, а именно: Hibernate, Spring и SiteMesh. Хотя по этим технологиям и не требуется предварительных знаний, читателям, которые уже знакомы с ними, будет легче понять соответствующие советы, чем тем, которые их никогда не использовали.

Как организована эта книга

Эта книга разделена на 16 глав с помощью подхода в виде вопросов и ответов.

Я всегда был поклонник книг в виде советов. Они раскрывают суть без напрасной траты времени читателя. Именно это делает данная книга. Она состоит из двух частей: Groovy и Grails. Часть, посвящённая Groovy, включает первые девять глав.

Глава 1 представляет краткое введение в Groovy, показывает варианты его использования и по шагам объясняет, как загрузить и установить Groovy на компьютере.

Глава 2 в основном предназначена для того, чтобы облегчить Java-разработчикам переход от синтаксиса Java к Groovy. Показаны главные сходства и отличия.

Глава 3 обсуждает типы данных и управляющие структуры. Типы данных Groovy включают простые типы данных и коллекции. Управляющие структуры состоят из циклических структур и условных структур.

Глава 4 показывает Groovy с объектно-ориентированной стороны. Groovy – это полностью объектно-ориентированный язык, поэтому Java-разработчики будут чувствовать себя как дома.

Глава 5 касается темы, которая, возможно, наиболее трудна для понимания Java-разработчиками: замыканий. На примерах эта глава пытается раскрыть загадку замыканий и показать варианты их использования.