RUBY ON RAILS ДЛЯ НАЧИНАЮЩИХ

ИЗУЧАЕМ РАЗРАБОТКУ ВЕБ-ПРИЛОЖЕНИЙ НА ОСНОВЕ RAILS

Майкл Хартл

ДМК ПРЕСС
ИЗДАТЕЛЬСТВО

УДК 004.438Ruby on Rails
ББК 32.973.22
Х22

Хартл М.
Х22 Ruby on Rails для начинающих / пер. с анг. А. Разуваева. – М.: ДМК Пресс, 2017. – 572 с.: ил.

ISBN 978-5-97060-429-8

Ruby on Rails – один из наиболее популярных фреймворков для разработки веб-приложений, но его изучение и использование – не самая простая задача. Эта книга поможет вам решить ее независимо от того, имеете ли вы опыт веб-разработки вообще и Rails в частности. Известный автор и ведущий разработчик Rails Майкл Хартл познакомит вас с Rails на примере разработки трех приложений. Автор рассказывает не только о Rails, но также описывает основы Ruby, HTML, CSS и SQL, которые пригодятся вам при разработке своих веб-приложений. Начиная обсуждение каждой новой методики, Хартл доходчиво объясняет, как она помогает решать практические задачи, а затем демонстрирует ее применение в программном коде, достаточно простом и понятном.

Издание предназначено для всех программистов, желающих изучить Ruby on Rails.

УДК 004.438Ruby on Rails
ББК 32.973.22

ISBN 978-0-13-407770-3 (анг.)
ISBN 978-5-97060-429-8 (рус.)

Содержание

Глава 1

От нуля к развертыванию

Добро пожаловать в «Ruby on Rails для начинающих: Изучаем разработку веб-приложений на основе Rails». Цель данной книги – научить вас разрабатывать собственные веб-приложения с использованием популярного фреймворка Ruby on Rails. Если эта тема для вас в новинку, «Ruby on Rails для начинающих» даст вам полное представление о разработке веб-приложений, включая основополагающие знания о Ruby, Rails, HTML и CSS, базах данных, управлении версиями, тестировании и развертывании – достаточно для начала вашей карьеры веб-разработчика или предпринимателя в сфере компьютерных технологий. С другой стороны, если вы уже знакомы с веб-разработкой, эта книга поможет вам освоить основы фреймворка Rails, включая MVC и REST, генераторы, миграции, маршрутизацию и встроенный язык Ruby. В любом случае, когда вы закончите чтение этой книги, вы будете готовы извлечь максимум пользы из более продвинутых книг, блогов и скринкастов (видеоуроков), являющихся частью обширной образовательной экосистемы[1].

В данной книге используется комплексный подход к веб-разработке: в процессе обучения мы напишем три примера приложений с возрастающей сложностью; в разделе 1.3 мы начнем создание простенького приложения, затем, в главе 2, перейдем к более продвинутому и, наконец, в главах с 3 по 12 полностью сосредоточимся на большом учебном примере. Как можно догадаться, примеры приложений в этой книге не привязаны к какой-либо категории веб-сайтов; хотя заключительный пример будет иметь значительное сходство с весьма популярной социальной сетью Twitter (которая, по совпадению, изначально была написана на Rails). Основное внимание здесь будет уделяться общим принципам разработки, поэтому полученные знания станут для вас надежным фундаментом, вне зависимости от того, какие виды приложений вы будете создавать.

Я часто слышу вопрос: «Какими знаниями нужно обладать для изучения веб-разработки с этой книгой?» Как мы увидим в разделе 1.1.1, веб-разработка – сложная тема, особенно для полных новичков. Хотя изначально книга предназначалась

---

[1] Самую свежую версию книги (на англ. языке. – *Прим. перев.*) можно найти на сайте http://www.railstutorial.org/. Если вы читаете печатную версию книги, сверьте версию своего экземпляра с онлайн-версией на http://www.railstutorial.org/book.

для читателей с некоторым опытом программирования, она нашла множество читателей среди начинающих разработчиков. Учитывая это, в данном (третьем) издании значительно снижен порог входа в разработку с Ruby on Rails (см. блок 1.1).

Блок 1.1 ❖ Снижение порога входа

С целью снижения порога входа для начинающих разработчиков в третьем издании сделано следующее:

- использована стандартная среда разработки, размещенная в облаке (раздел 1.2), что позволяет обойти множество проблем, связанных с установкой и настройкой новой системы;
- использован «предустановленный стек» Rails, включающий встроенный фреймворк тестирования MiniTest;
- уменьшено количество внешних зависимостей (RSpec, Cucumber, Capybara, Factory Girl);
- использован облегченный и более гибкий подход к тестированию;
- исключены сложные в настройке варианты (Spork, RubyTest);
- меньше внимания уделяется особенностям, свойственным конкретной версии Rails, и больше – общим принципам веб-разработки.

Я надеюсь, что эти изменения сделают третье издание книги доступным для еще более широкой аудитории, чем предыдущие.

В этой первой главе мы начнем изучение фреймворка Ruby on Rails с установки необходимого программного обеспечения и настройки рабочего окружения (раздел 1.2). Затем создадим первое Rails-приложение с названием `hello_app`. В книге «Ruby on Rails для начинающих» особое значение придается хорошим привычкам в программировании, поэтому сразу после создания нового Rails-проекта мы поместим его в систему управления версиями Git (раздел 1.4). И, верите вы в это или нет, в этой главе мы даже развернем наше первое приложение в Сети (раздел 1.5).

В главе 2 мы создадим второй проект для демонстрации основ работы Rails-приложения. Для быстроты в этом *мини-приложении* (с названием `toy_app`) мы применим прием «скаффолдинга» (scaffolding, см. блок 1.2) для генерации кода. Поскольку код получится одновременно и сложным, и уродливым, мы оставим его в стороне и сосредоточимся на взаимодействии с приложением через его идентификатор *URI* (который часто называют адресом *URL*)[1], с использованием веб-браузера.

Остальная часть книги описывает разработку единственного большого учебного приложения (с названием `sample_app`), весь код которого будет написан с нуля. При этом мы будем использовать фиктивные объекты, приемы разработки через тестирование (Test-Driven Development, TDD) и интеграционные тесты. В главе 3 мы сначала создадим статические страницы, а затем добавим в них немного ди-

[1] Аббревиатура URI расшифровывается как Uniform Resource Identifier (универсальный идентификатор ресурсов), а аббревиатура URL – как Uniform Resource Locator (универсальный указатель ресурсов). На практике URL обычно соответствует тому, что «находится в адресной строке браузера».

намического контента. В главе 4 мы совершим небольшое турне по языку Ruby, лежащему в основе Rails. Затем, в главах с 5 по 10, завершим базовую часть приложения, создав макет сайта, модель данных пользователя, а также систему регистрации и аутентификации (включая активацию учетной записи и сброс пароля). Наконец, в главах 11 и 12 мы добавим поддержку микроблоггинга и немного социальных функций, чтобы закончить действующий пример сайта.

Блок 1.2 ❖ Скаффолдинг: быстрее, проще, заманчивее

С самого начала фреймворк Rails привлек к себе внимание знаменитым видеороликом от Дэвида Хейнмейера Ханссона (David Heinemeier Hansson) – создателя Rails, демонстрирующим создание микроблога за 15 минут. Этот и другие подобные видеоролики – отличный способ познакомиться с возможностями Rails, и я рекомендую посмотреть их. Но предупреждаю: они совершают свой удивительный пятнадцатиминутный подвиг, используя функцию под названием «скаффолдинг» (scaffolding), которая в значительной степени опирается на код, волшебным образом сгенерированный Rails-командой `generate scaffold`.

Работая над книгой по Ruby on Rails, мне приходилось бороться с искушением положиться на эту функцию, потому что это быстрее, проще, заманчивее. Но сложность и огромный объем кода, производимый генератором, могли стать непреодолимым препятствием для начинающего Rails-разработчика; вы, вероятно, сможете использовать его, но почти наверняка не сможете понять. Увлекшись скаффолдингом, вы рискуете превратиться в виртуозного генератора сценариев с весьма поверхностным знанием Rails.

В этой книге мы будем придерживаться (почти) полярно противоположного подхода: хотя в главе 2 создадим мини-приложение, сгенерировав код автоматически, основой книги все же является приложение, которое мы начнем писать в главе 3. На каждом этапе его разработки мы будем писать небольшие куски кода, достаточно простые для понимания, но все же требующие некоторых усилий для их усвоения. Результатом станет более глубокое понимание Rails, которое, в свою очередь, даст вам хорошую базу для написания практически любых типов веб-приложений.

1.1. Введение

«Ruby on Rails» (или просто «Rails») – это фреймворк для разработки веб-приложений на языке программирования Ruby. Со времен своего дебюта в 2004 году Ruby on Rails довольно быстро стал одним из самых мощных и популярных инструментов создания динамических веб-приложений. Rails используется множеством различных компаний: Airbnb, Basecamp, Disney, GitHub, Hulu, Kickstarter, Shopify, Twitter и Yellow Pages. Помимо этого, существует множество компаний, занимающихся разработкой веб-приложений и специализирующихся на Rails, таких как ENTP, thoughtbot, Pivotal Labs, Hashrocket и HappyFunCorp, плюс бесчисленное множество независимых консультантов, преподавателей и индивидуальных разработчиков.

Что же делает Rails таким замечательным? Во-первых, Ruby on Rails – это открытый исходный код, доступный на условиях лицензии MIT, и, как следствие, его можно загружать и использовать совершенно бесплатно. Rails также обязан своим успехом изящному и компактному дизайну. Используя податливость языка Ruby, лежащего в его основе, Rails фактически определяет предметно-ориентированный язык для разработки веб-приложений. В результате множество часто встречающихся задач веб-программирования, таких как динамическое создание разметки HTML, определение моделей данных и маршрутизация URL, легко решаются в Rails, а конечный код приложений получается кратким и читаемым.

Rails также быстро адаптируется к новым тенденциям в веб-технологиях. Например, Rails одним из первых полностью реализовал архитектурный стиль REST структурирования веб-приложений (мы будем изучать его на всем протяжении книги). И когда в других фреймворках появляются новые успешные приемы, создатель Rails, Дэвид Хейнмейер Ханссон (David Heinemeier Hansson), и рабочая группа Rails не стесняются использовать чужой опыт. Пожалуй, наиболее драматичным примером является слияние Rails с конкурирующим веб-фреймворком Merb, благодаря которому Rails получил модульный дизайн, стабильный API и улучшенную производительность.

Наконец, вокруг Rails сплотилось необычайно увлеченное и разнообразное сообщество: сотни разработчиков, представительные конференции, огромное количество гемов (пакетов, законченных решений конкретных задач, таких как постраничный вывод и выгрузка изображений), богатый набор информативных блогов и рог изобилия форумов и IRC-каналов. Большое количество Rails-программистов также облегчает работу с (неизбежными) ошибками, возникающими в процессе разработки: алгоритм «Ищи в Google сообщение об ошибке» практически всегда помогает найти подходящую статью в блоге или сообщение на форуме.

1.1.1. Необходимая квалификация

Формально эта книга не требует некоторого набора необходимых знаний. Она содержит учебные сведения не только о фреймворке Rails, но также о лежащем в его основе языке Ruby, встроенном фреймворке тестирования (MiniTest), командной строке Unix, HTML, CSS, немного JavaScript и даже чуть-чуть SQL. Это очень объемный материал, и потому для работы с книгой желательно иметь некоторый опыт использования HTML и программирования. Однако данную книгу использовало так много новичков для изучения веб-разработки с нуля, что даже если у вас мало опыта, я все же советую попробовать. Если содержимое учебника покажется сложным, вы всегда можете сделать шаг назад и начать с одного из ресурсов, перечисленных ниже. Другая возможная стратегия (рекомендованная многими читателями): прочитать книгу дважды; возможно, вы удивитесь, как много узнали в первый раз (и насколько проще читать по второму кругу).

Стоит ли вначале изучить Ruby? Ответ зависит от вашего личного стиля обучения и опыта в программировании. Если вы предпочитаете изучать все систематически, с самых основ, или прежде никогда не программировали, вам, возмож-

но, стоит начать с изучения Ruby, и в этом случае я рекомендую книгу «Учись программировать» Криса Пайна (Chris Pine)[1] и «Начало Ruby» Питера Купера (Peter Cooper)[2]. С другой стороны, множество начинающих Rails-разработчиков имеет своей целью создание веб-приложений и, скорее, предпочтет не корпеть над толстенной книгой, чтобы написать одну-единственную веб-страницу. В этом случае я рекомендую поработать с коротким интерактивным учебником «Try Ruby»[3], чтобы получить общее представление о Ruby. Если и после этого книга окажется для вас слишком сложной, попробуйте начать с «Learn Ruby on Rails» Дэниэла Кехо (Daniel Kehoe)[4] или «One Month Rails»[5] – обе книги ориентированы на начинающих, в отличие от данной книги.

Независимо от того, где вы начнете, к концу этой книги вы будете готовы к чтению других, более продвинутых ресурсов по Rails. Вот некоторые из тех, которые я могу рекомендовать:

- Code School[6]: хорошие интерактивные онлайн-курсы по программированию;
- Turing School of Software & Design[7]: очные, 27-недельные курсы в Денвере, Колорадо. Читатели этой книги могут получить скидку в $500, воспользовавшись промокодом RAILSTUTORIAL500;
- Tealef Academy[8]: хорошая образовательная онлайн-площадка, посвященная Rails-разработке (включая продвинутый материал);
- Thinkful[9]: онлайн-курс в паре с профессиональным инженером, работа по учебному плану, основанному на конкретном проекте;
- RailsCastsот[10] Райана Бейтса (Ryan Bates): отличные (в основном бесплатные) видеоуроки по Rails;
- RailsApps[11]: большой выбор подробных проектов и учебников, посвященных разным темам;
- Rails Guides[12]: актуальные руководства по Rails.

1.1.2. Типографские соглашения

Соглашения в этой книге главным образом очевидны. В этом разделе я упомяну лишь те, которые таковыми не являются.

---

1 http://www.shokhirev.com/mikhail/ruby/ltp/title.html. – *Прим. перев.*
2 http://www.amazon.com/gp/product/1430223634.
3 http://tryruby.org/.
4 http://learn-rails.com/learn-ruby-on-rails.html.
5 http://mbsy.co/7Zdc7.
6 https://my.getambassador.com/.
7 https://www.turing.io/friends/tutorial.
8 https://launchschool.com/.
9 https://www.thinkful.com/a/railstutorial.
10 http://railscasts.com/.
11 https://tutorials.railsapps.org/.
12 http://rusrails.ru/.

В этой книге присутствует множество примеров применения командной строки. Для простоты все они используют приглашение в стиле Unix (знак доллара):

```
$ echo "hello, world"
hello, world
```

Как отмечено в разделе 1.2, я рекомендую пользователям всех операционных систем (особенно Windows) использовать облачную среду разработки (раздел 1.2.1), в которой есть встроенная командная строка Unix (Linux). Это особенно полезно, потому что в Rails имеется множество команд, которые можно запустить из командной строки. Например, в разделе 1.3.2 мы запустим локальный веб-сервер командой **rails server**:

```
$ rails server
```

По аналогии с приглашением командной строки, здесь используется соглашение о разделителях каталогов в стиле Unix (то есть прямого слеша /). Например, конфигурационный файл **production.rb** учебного приложения будет представлен как:

```
config/environments/production.rb
```

Этот путь к файлу следует интерпретировать как относительный, начинающийся в корневом каталоге приложения, абсолютный путь к которому зависит от системы; в облачной интегрированной среде разработки (IDE) (раздел 1.2.1) этот путь выглядит так:

```
/home/ubuntu/workspace/sample_app/
```

То есть полный путь к `production.rb` имеет вид:

```
/home/ubuntu/workspace/sample_app/config/environments/production.rb
```

Для простоты я часто буду опускать путь к корневому каталогу приложения и писать просто: **config/environments/production.rb**.

В книге нередко демонстрируется вывод разных программ (команд, системы управления версиями, программ на Ruby и т. д.). Из-за неисчислимого количества небольших различий между компьютерными системами вывод, который вы увидите, возможно, не всегда в точности совпадет с тем, что показан в тексте, но это не повод для беспокойства. Некоторые команды могут вызывать ошибки, связанные с особенностями вашей системы, поэтому, чтобы не решать сизифову задачу документирования всех таких погрешностей, я отсылаю вас к алгоритму «ищи сообщение об ошибке в Google», который, между прочим, является хорошей практикой для программирования. Если вы столкнетесь с проблемами в процессе обучения, я советую обратиться к ресурсам, список которых приводится в справочном разделе[1].

[1] https://www.railstutorial.org/#help.

Поскольку в книге (помимо всего прочего) рассматривается вопрос тестирования Rails-приложений, часто бывает полезно знать, что данный кусок кода приводит к неудаче (обозначается красным цветом) или, наоборот, к успеху тестирования (обозначается зеленым цветом). Для удобства программный код тестов будет отмечаться словами **КРАСНЫЙ** и **ЗЕЛЕНЫЙ**.

К каждой главе прилагаются упражнения, выполнение которых необязательно, но рекомендуется. Чтобы сделать основной текст независимым от упражнений, решения обычно не включаются в последующие листинги кода. В редких случаях, когда решение упражнения используется в дальнейшем, оно будет приведено в основном тексте.

Наконец, для удобства в книге применяются два соглашения, облегчающие понимание большого количества примеров кода. Во-первых, в некоторых листингах выделены одна или несколько строк, как показано ниже:

```
class User < ActiveRecord::Base
  validates :name, presence: true
  validates :email, presence: true
end
```

Такое выделение обычно указывает на наиболее важный новый код в данном примере и часто (хотя и не всегда) отражает разницу между этим и предыдущим листингом. Во-вторых, для краткости и простоты во многих листингах в книге используются вертикальные точки, например:

```
class User < ActiveRecord::Base
  .
  .
  .
  has_secure_password
end
```

Эти точки обозначают пропущенный код, и их не нужно копировать буквально.

1.2. За работу

Установка Ruby, Rails и всего сопутствующего программного обеспечения поддержки может привести в отчаяние даже опытных Rails-разработчиков. Проблема осложняется большим разнообразием окружений: разные операционные системы, версии, предпочтения в выборе текстовых редакторов и интегрированных сред разработки (IDE) и т. д. Пользователи, уже установившие среду разработки, могут сразу переходить к настройкам, а новым пользователям (как указано в блоке 1.1) я предлагаю избежать проблем с установкой и настройкой за счет использования *облачной интегрированной среды разработки*. Облачная IDE запускается в обычном браузере и, следовательно, одинаково хорошо работает на любой платформе, что особенно полезно для операционных систем (например, Windows), в которых разработка с Rails исторически была сложным занятием. Если, несмотря на все

предполагаемые трудности, вы по-прежнему хотите создать локальную среду разработки, я рекомендую следовать инструкциям на InstallRails.com[1].

1.2.1. Среда разработки

Принимая во внимание множество своеобразных настроек и предпочтений, число вариантов окружений разработки может равняться числу Rails-программистов. Чтобы избежать этих сложностей, я адаптировал данную книгу под превосходную облачную среду разработки Cloud9. В частности, мне было очень приятно сотрудничать с Cloud9, чтобы предложить вам среду разработки, ориентированную на потребности именно этого издания учебника. Получившееся рабочее пространство Cloud9 предоставляется предварительно настроенным, со всем программным обеспечением (включая Ruby, RubyGems, Git), необходимым для профессиональной разработки с Rails. (Отдельно мы будем устанавливать только сам фреймворк Rails, но это сделано намеренно (раздел 1.2.2).) Облачная IDE также включает три основных компонента, необходимых для разработки веб-приложений: текстовый редактор, навигатор файловой системы и терминал командной строки (рис. 1.1). Кроме того, ее текстовый редактор поддерживает глобальный поиск «Найти в файлах», который я считаю необходимым для навигации по любым большим

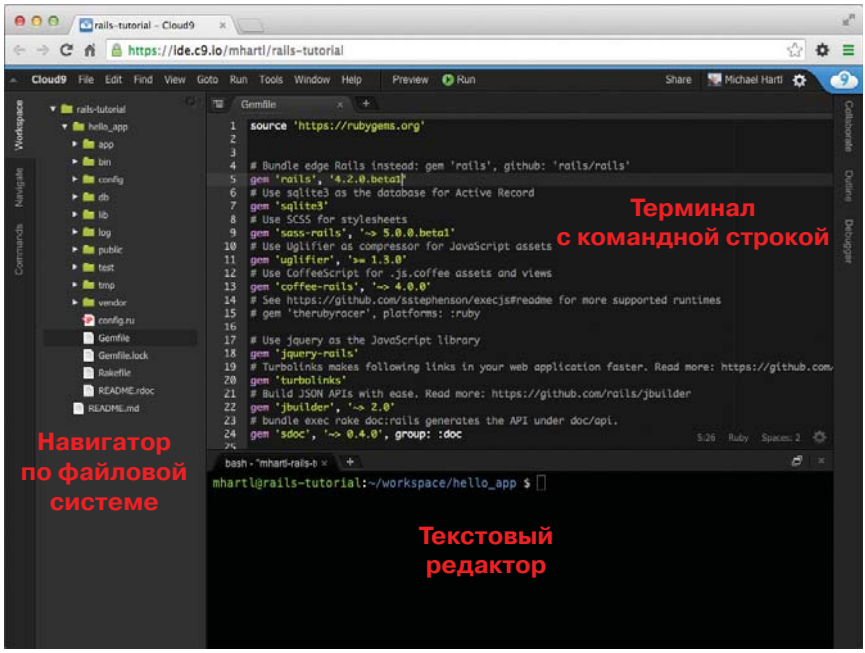

Рис. 1.1 ❖ Анатомия облачной IDE

[1] http://installrails.com/.

проектам Ruby или Rails[1]. И наконец, если вы решите использовать не только облачную IDE (я могу лишь приветствовать изучение других инструментов), на ее примере вы получите великолепное введение в основные возможности текстовых редакторов и других инструментов разработки.

Ниже приводится пошаговая инструкция, описывающая, как начать работу с облачной средой разработки:

- создайте бесплатную учетную запись в Cloud9[2];
- щелкните на ссылке **Goto your Dashboard** (Перейти к панели управления);
- щелкните на ссылке **Create New Workspace** (Создать новое рабочее пространство);
- как показано на рис. 1.2, создайте рабочее пространство `rails-tutorial` (не `rails_tutorial`), установите в настройках флажок **Private to the people I invite** (Только для приглашенных) и выберите значок **RailsTutorial** (не значок **Ruby on Rails**);

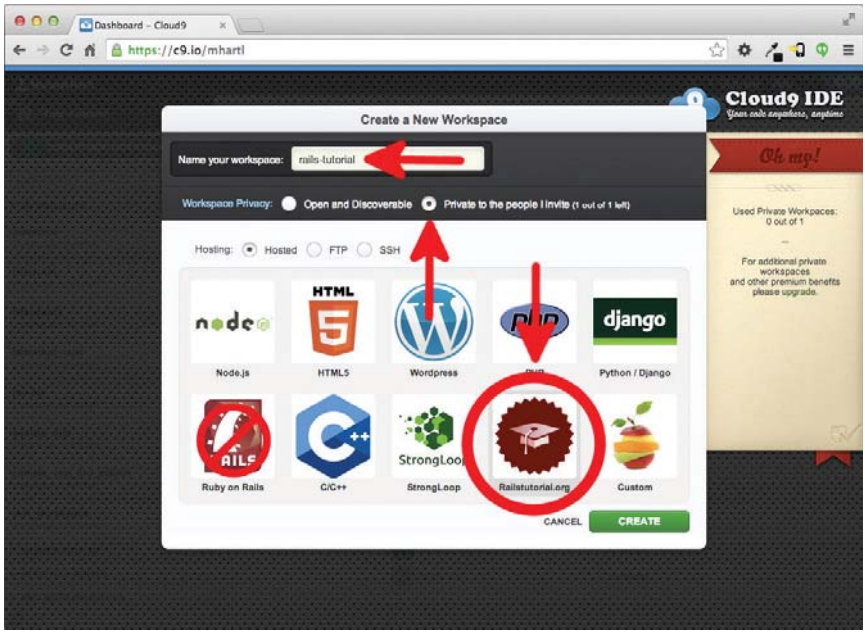

Рис. 1.2 ❖ Создание нового рабочего пространства в Cloud9

- щелкните на ссылке **Create** (Создать);
- после того как Cloud9 закончит подготовку рабочего пространства, выберите его и щелкните на ссылке **Start editing** (Начать редактирование).

---

[1] Например, чтобы найти определение функции `foo`, можно запустить глобальный поиск по фразе `def foo`.

[2] https://c9.io/web/sign-up/free.

Так как использование двух пробелов для отступов считается практически универсальным соглашением в Ruby, я также рекомендую изменить настройки редактора (по умолчанию используются четыре пробела). Как показано на рис. 1.3, щелкните на значке с изображением шестеренки в правом верхнем углу, выберите пункт **Code Editor (Ace)** (Редактор кода (Ace)) и измените параметр **Soft Tabs** (Мягкая табуляция). (Обратите внимание, что настройки применяются мгновенно и нет необходимости нажимать кнопку **Save** (Сохранить).)

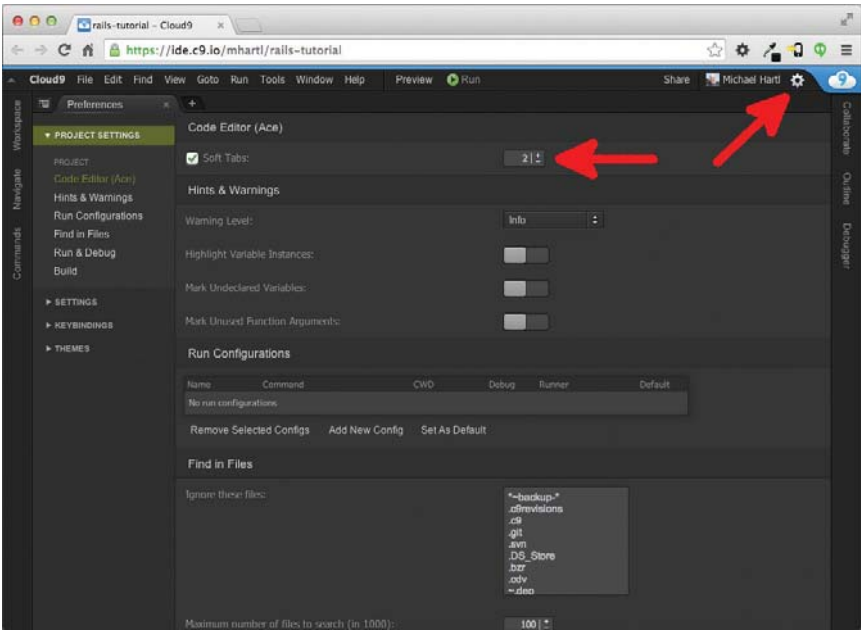

Рис. 1.3 ❖ Настройка использования двух пробелов для отступа в Cloud9

1.2.2. Установка Rails

Среда разработки, представленная в разделе 1.2.1, включает все ПО, необходимое для начала, кроме самого фреймворка Rails[1]. Для установки воспользуемся командой `gem` диспетчера пакетов *Ruby Gems*. Введите в командной строке терминала команду, как показано в листинге 1.1. (Если вы работаете в локальной системе, необходимо использовать обычное окно терминала; если в облачной IDE – область командной строки, как на рис. 1.1.)

Листинг 1.1 ❖ Установка Rails с определенным номером версии

```
$ gem install rails -v 4.2.0
```

[1] На данный момент в Cloud9 используется более старая версия Rails, и она несовместима с настоящей книгой, поэтому так важно установить его самостоятельно.

Параметр -v гарантирует установку определенной версии Rails. Это очень важно для получения результатов, совместимых с этой книгой.

1.3. Первое приложение

Следуя давним традициям в программировании, нашей первой целью будет создание программы, которая выводит фразу «hello, world» («Привет, мир!»). В частности, мы создадим простое приложение, которое отобразит строку «hello, world!» на веб-странице: в среде разработки на компьютере (раздел 1.3.4) и в Интернете (раздел 1.5).

Практически все приложения Rails начинаются одинаково – с команды `rails new`. Эта удобная команда создает скелет приложения Rails в выбранном каталоге. Тем, кто не стал использовать Cloud9 IDE, рекомендованную в разделе 1.2.1, сначала необходимо создать каталог `workspace` для проекта, если он еще не создан (листинг 1.2), и перейти в него. (В листинге 1.2 показаны команды `cd` и `mkdir`; если вы с ними пока не знакомы, прочитайте блок 1.3.)

Листинг 1.2 ❖ Создание каталога workspace для проекта (не нужно в облачной IDE)

```
$ cd                # Перейти в домашний каталог.
$ mkdir workspace   # Создать каталог workspace.
$ cd workspace/     # Перейти в каталог workspace.
```

Блок 1.3 ❖ Краткий курс командной строки Unix

Для читателей, пришедших из Windows или (в меньшей, но все еще значительной степени) из Macintosh OS X, командная строка Unix может выглядеть непривычной. К счастью, если вы используете рекомендованную облачную среду разработки, вы автоматически получаете доступ к командной строке Unix (Linux) в стандартной оболочке интерфейса, известной как Bash[1].

Основная идея командной строки проста: вводя короткие команды, пользователь может выполнить множество операций, таких как создание каталогов (`mkdir`), перемещение и копирование файлов (`mv` и `cp`), навигация в файловой системе за счет смены каталогов (`cd`). Командная строка может показаться примитивной тем, кто знаком в основном с графическим интерфейсом, внешность обманчива: это один из наиболее мощных инструментов в арсенале разработчика. Действительно, крайне редко удается увидеть рабочий стол опытного разработчика, на котором не было бы открыто несколько окон терминала с командной оболочкой.

Вообще говоря, это очень глубокая тема, но для следования за примерами в этой книге понадобится лишь несколько наиболее часто используемых команд, перечисленных в табл. 1.1. Для более глубокого изучения командной строки Unix читайте «Conquering the Command Line»[2] (Покоряя командную строку) Марка Бейтса (доступны бесплатная онлайн-версия и электронная книга с видеоуроками[3]).

---

1 https://ru.wikipedia.org/wiki/Bash.

2 http://conqueringthecommandline.com/book.

3 http://conqueringthecommandline.com/#pricing.