Stručný obsah

Část III
Framework Rails pod drobnohledem

Obsah

Část II
Budování aplikace

Předmluva ke čtvrtému vydání

Když mě Dave požádal o spoluautorství na třetí edici této knihy, byl jsem nadšený. Konec konců, framework Rails jsem se naučil z prvního výtisku první edice této knihy. Dave a já máme rovněž mnoho společného. Přestože preferuje editor Emacs a systém Mac OS X, zatímco já dávám přednost spíše editoru Vim a systému Ubuntu, oba sdílíme lásku k příkazovému řádku a rádi si špiníme prsty kódem – před ponořením do těžké teorie začínáme hmatatelnými příklady.

Od publikování třetí edice (a ve skutečnosti také od první, druhé a třetí edice) se mnoho změnilo. Framework Rails se nachází v procesu značeného refaktorování, které se velkou měrou týká jeho útrob. Řada prvků, které používaly předchozí příklady, byla nejdříve zamítnutá a následně odstraněná. Došlo k přidání nových prvků a nabytí zkušeností ohledně toho, jaké jsou nejlepší postupy pro používání frameworku Rails. Framework Rails nyní funguje také na jazyku Ruby 1.9, přičemž každý z příkladů byl testovaný Ruby 1.8.7 a Ruby 1.9.2.

Kromě toho framework Rails explodoval z oblíbeného frameworku na aktivní a živý ekosystém, dovršený řadou oblíbených zásuvných modulů a hlubokou integrací do nástrojů třetích stran. V rámci tohoto procesu se framework Rails stal hlavním proudem, a přitahuje tak k sobě rozmanitější skupiny vývojářů.

To vše vedlo k novému uspořádání této knihy. Řada nováčků neměla potěšení seznámit se s jazykem Ruby, a proto se tato část přesunula z přílohy do kapitoly v části I. V části I procházíme krok za krokem sestavování skutečné aplikace, která byla aktualizovaná a dopilovaná takovým způsobem, aby se soustředila na současné nejlepší postupy.

Avšak největších změn doznala poslední část: není již praktické věnovat se celému ekosystému frameworku Rails vzhledem k jeho šíři a rychlosti změn, a proto se tato část nyní soustředí na poskytování celkového pohledu na tuto krajinu, díky čemuž budete vědět, co máte hledat a kde nejdete zásuvné moduly a související nástroje pro řešení běžných potřeb, které jdou daleko nad rámec toho, co obsahuje samotný framework.

Tato kniha se zkrátka musela opět přizpůsobit.

Sam Ruby
Březen 2011

Poděkování

Asi si říkáte, že vytvoření nové edice knihy je snadné – vždyť, veškerý text je již hotový. Jedná se vlastně jen o doladění nějakého kódu tadyhle a drobné změny formulace onde a je hotovo.

Je obtížné přesně to vyjádřit, ale máme dojem, že tvorba každé edice této knihy vyžaduje přibližně stejné úsilí jako první edice. Framework Rails se neustále vyvíjí a stejně tak se musí vyvíjet i tato kniha. Části aplikace Depot se několikrát přepisovaly a aktualizovalo se celé vyprávění. Důraz na architekturu REST a vyvarování se prvků, které se staly zastaralé, vede k opakovaným změnám struktury knihy, neboť co bylo dříve žhavé, je nyní jen vlažné.

Tato kniha by proto neexistovala bez masivní pomoci od komunit kolem jazyka Ruby a frameworku Rails. Začněme řadou neuvěřitelně ochotných oficiálních recenzentů pracovních verzí této knihy:

- Jeremy Anderson, Ken Coar, Jeff Cohen, Joel Clermont, Geoff Drake, Pavan Gorakavi, Michael Jurewitz, Mikel Lindsaar, Paul Rayner, Martijn Reuvers, Doug Rhoten, Gary Sherman, Davanum Srinivas, Stefan Turalski a José Valim

Kromě toho byla každá edice této knihy vydaná ve verzi beta: byly rozeslané rané verze ve formátu PDF, které měli lidé komentovat online. A skutečně je komentovali: jen pro tuto edici bylo odesláno přes 800 návrhů a upozornění na chyby. Drtivá většina vedla ke změnám v knize, která je díky tomu mnohem užitečnější. Přestože díky patří všem za podporu pro program beta verze této knihy a za přispění tak nesmírně cennou odezvou, řada přispěvatelů šla nad rámec svých povinností:

- Manuel E. Vidaurre Arenas, Seth Arnold, Will Bowlin, Andy Brice, Jason Catena, Victor Marius Costan, David Hadley, Jason Holloway, David Kapp, Trung LE, Kristian Riiber Mandrup, mltsy, Steve Nicholson, Jim Puls, Johnathan Ritzi, Leonel S, Kim Shrier, Don Smith, Joe Straitiff a Martin Zoller

A nakonec, hlavní tým odpovědný za framework Rails byl neuvěřitelně ochotný při odpovídání na dotazy, při kontrole úryvků kódu a při opravování chyb. Velké „díky“ patří následujícím:

- Scott Barron (htonl), Jamis Buck (minam), Thomas Fuchs (madrobby), Jeremy Kemper (bitsweat), Yehuda Katz (wycats), Michael Koziarski (nzkoz), Marcel Molina Jr, (noradio), Rick Olson (technoweenie), Nicholas Seckar (Ulysses), Sam Stephenson (sam), Tobias Lütke (xal), José Valim (josevalim) a Florian Weber (csshsh)

Sam Ruby
Březen 2011
rubys@intertwingly.net

Úvod

Ruby on Rails je framework, který usnadňuje vývoj, nasazení a údržbu webových aplikací. Během měsíců, které následovaly po jeho počátečním vydání, se framework Rails stal z neznámé hračky celosvětovým fenoménem, a co je ještě důležitější, stal se volbou pro implementaci pestré palety takzvaných aplikací pro Web 2.0.

Jak je to možné?

Framework Rails prostě působí dobře

Tak předně, velký počet vývojářů byl frustrovaný technologiemi, které používali pro vytváření webových aplikací. Nezáleželo ani tak na tom, zda používali platformu Java, PHP nebo .NET – stále totiž narůstal pocit, že práce, kterou dělají, je příliš obtížná. A pak se náhle objevil framework Rails, který byl mnohem jednodušší.

Jenže samotná snadnost není vše. Bavíme se o profesionálních vývojářích vytvářejících skutečné weby. Tito vývojáři chtěli mít pocit, že aplikace, které vyvíjejí, obstojí v testu času – že navrhují a implementují pomocí moderních, profesionálních technik. Sáhli proto po frameworku Rails a zjistili, že se nejedná jen o nástroj pro splácání webů.

Například *všechny* aplikace na frameworku Rails jsou implementované pomocí architektury MVC (Model-View-Controller – model, pohled, řadič). Vývojáři na platformě Java jsou zvyklí na frameworky, jako jsou Tapestry a Struts, které jsou založené na architektuře MVC. Avšak framework Rails posouvá architekturu MVC ještě dále: při vývoji na frameworku Rails začínáte s fungující aplikací, existuje místo pro každý kousek kódu a všechny části aplikace komunikují standardním způsobem.

Profesionální programátoři píší testy, což je oblast, o kterou se opět postará framework Rails. Veškeré aplikace na frameworku Rails mají vestavěnou podporu pro testování. Při přidávání dalších funkcí framework Rails automaticky vytváří testovací kostry pro tyto funkce. Framework navíc usnadňuje testování aplikací, a v důsledku toho bývají aplikace na frameworku Rails otestované.

Aplikace na frameworku Rails jsou napsané v jazyce Ruby, což je moderní, objektově orientovaný skriptovací jazyk. Jazyk Ruby je stručný, aniž by byl nesrozumitelně strohý – myšlenky můžete v jazyce Ruby vyjádřit přirozeným a jasným způsobem. To vede k programům, které se snadnou píšou a (což je stejně důležité) snadno se čtou i o několik měsíců později.

Framework Rails bere jazyk Ruby až na samotnou hranici rozšiřuje jej novými způsoby, které usnadňují programátorům život. Díky tomu jsou naše programy kratší a čitelnější. Kromě toho nám umožňuje provádět v kódové bázi úkoly, které by se jinak musely dělat v externích konfiguračních souborech. Tím lze ještě snadněji vysledovat, co se děje. Následující kód definuje třídu modelu pro projekt. Prozatím si s podrobnostmi nelamte hlavu, ale zamyslete se spíše nad tím, kolik je na několika málo řádcích kódu vyjádřeno informací.

```
class Project < ActiveRecord::Base
  belongs_to :portfolio
  has_one :project_manager
  has_many :milestones
  has_many :deliverables, :through => :milestones

  validates :name, :description, :presence => true
  validates :non_disclosure_agreement, :acceptance => true
  validates :short_name, :uniqueness => true
end
```

Existují dva filozofické základy, které udržují kód využívající framework Rails krátký a čitelný: DRY a konvence nad konfiguraci. DRY znamená „neopakuj se“ (don't repeat yourself): každá část znalosti v systému by měla být vyjádřena právě na jednom místě. Framework Rails využívá sílu jazyka Ruby k realizaci tohoto přístupu. V aplikaci na frameworku Rails najdete jen velmi málo duplikací. To, co potřebujete říci, řeknete na jediném místě (které vám často doporučí konvence architektury MVC), a pak jdete dál. Pro programátory zvyklé na jiné webové frameworky, kde by jednoduchá změna schématu mohla znamenat půl tuctu nebo i více změn v kódu, to bylo jako procitnutí ze zlého snu.

Konvence nad konfigurací je také naprosto zásadní. Znamená to, že framework Rails nabízí rozumné výchozí hodnoty pro téměř každý aspekt splétání vaší aplikace. Při dodržování konvencí můžete psát aplikace na frameworku Rails s menším množstvím kódu než u typické webové aplikace v jazyce Java využívající konfiguraci ve formátu XML. Potřebujete-li tyto konvence přepsat, je to s frameworkem Rails také velice snadné.

Vývojáři přecházející k frameworku Rails zjistili ještě něco. Framework Rails nehraje s novými de facto webovými standardy na schovávanou, ale pomáhá je definovat. A kromě toho usnadňuje vývojářům integrovat do kódu takové prvky, jako jsou ajaxová nebo RESTful rozhraní, protože jejich podpora je v něm přímo vestavěná. (A pokud architekturu Ajax nebo REST neznáte, ničeho se neobávejte – v pozdější části knihy si vše vysvětlíme.)

Vývojáři si dělají starosti také s nasazením. Zjistili, že s frameworkem Rails můžete provádět nasazení následných vydání vaší aplikace na libovolný počet serverů jediným příkazem (a zase je stejně snadno odrolovat zpět, pokud by se vydání ukázalo jako méně ideální).

Framework Rails byl extrahován z reálné, komerční aplikace. Ukazuje se, že nejlepší způsob, jak vytvořit framework, je najít ústřední témata určité aplikace a poté je uzavřít do generického založení kódu. Při vývoji aplikace na frameworku Rails tedy začínáte s hotovou polovinou skutečně dobré aplikace.

S frameworkem Rails je ale spojeno ještě něco – něco, co lze jen těžko popsat. Prostě tak nějak působí dobře. Samozřejmě nám v tomto musíte věřit, dokud sami nenapíšete nějaké aplikace na frameworku Rails (což by mělo být přibližně v rámci následujících čtyřiceti pěti minut). A právě o tom je tato kniha.

Framework Rails je agilní

Tato kniha nese název *Ruby on Rails Průvodce agilním vývojem webů*. Možná vás tedy překvapí, že zde nejsou explicitní části věnované aplikování agilních postupů X, Y a Z na programování s frameworkem Rails.

Důvod je prostý a spletitý zároveň. Agilita je totiž součástí konstrukce frameworku Rails.

Podívejme se na hodnoty vyjádřené v Agilním manifestu jako skupina čtyř preferencí (viz *http://agilemanifesto.org/* – Dave Thomas byl jedním ze sedmnácti autorů tohoto dokumentu):

- jedinci a interakce nad procesy a nástroji,
- fungující software nad ucelenou dokumentací,
- spolupráce zákazníka při jednání o smlouvě,
- reagování na změnu při dodržování plánu.

Framework Rails je celý o jedincích a interakcích. Nejsou zde žádné masivní sady nástrojů, žádné komplexní konfigurace a žádné komplikované procesy. Je tu jen malá skupina vývojářů, jejich oblíbené editory a kousky kódu jazyka Ruby. To vede k transparentnosti – to, co vývojáři udělají, se ihned odrazí v tom, co zákazník vidí. Jedná se o skutečně interaktivní proces.

Framework Rails neodsuzuje dokumentaci, avšak vytvoření dokumentace ve formátu HTML pro celou kódovou bázi činí neskonale jednoduché. Na druhou stranu není vývojový proces frameworku Rails řízený dokumenty. V srdci projektu na frameworku Rails tudíž nenajdete 500stránkovou specifikaci, ale spíše skupinu uživatelů a vývojářů společně zkoumajících svou potřebu a možné způsoby, jak na tuto potřebu odpovědět. Najdete framework, který dodává software již v rané fázi vývojového cyklu. Tento software může mít hrubé okraje, uživatele tak ale mohou získat zběžný pohled na to, co jim dodáte.

Tímto způsobem vede framework Rails ke spolupráci se zákazníkem. Když totiž zákazníci uvidí, jak rychle dokáže projekt na frameworku Rails reagovat na změnu, začnou

věřit tomu, že daný tým dokáže dodat to, co potřebují, a ne jen to, co bylo požadováno. Konfrontace se tedy nahradí schůzkami na téma „co kdyby?“.

To vše je spjato s myšlenkou být schopen reagovat na změnu. Silný, možná až chorobný způsob, jakým framework Rails ctí princip DRY, znamená, že změny v aplikacích na frameworku Rails mají na manším rozsahu kódu, než by stejné změny měly v jiných frameworcích. A protože aplikace na frameworku Rails jsou napsané v jazyce Ruby, kde lze koncepce vyjádřit přesně a stručně, změny jsou obvykle lokální a snadno se píší. Hluboký důraz na testování jednotek a na funkční testování společně s podporou pro testovací přípravky a kostry během testování dává vývojářům záchrannou síť nezbytnou při provádění těchto změn. S dobrou sadou testů jsou změny méně stresující.

Místo neustálé snahy svázat procesy frameworku Rails s agilními principy jsme se rozhodli nechat framework, aby mluvil sám za sebe. Snažte se během pročítání cvičebních kapitol představit si sami sebe, jak tímto způsobem vyvíjíte webové aplikace: pracujete bok po boku se svými zákazníky a společně určujete priority a řešení problémů. Když si pak budete pročítat pokročilejší koncepce, jež následují v části III, všimněte si, jak můžete díky podkladové struktuře frameworku Rails splnit potřeby zákazníka rychleji a méně okázalým způsobem.

Poslední poznámka o agilitě a frameworku Rails: i když je to asi neprofesionální, představte si, jak bude takové programování zábavné.

Komu je kniha určena

Tato kniha je pro programátory, kteří chtějí sestavovat a nasazovat webově založené aplikace. To zahrnuje programátory aplikací, kteří jsou ve frameworku Rails (a třeba i v jazyce Ruby) noví, a také ty, kteří znají základy, ale touží po hlubším seznámení s frameworkem Rails.

Předpokládáme nějakou znalost jazyků HTML, CSS (Cascading Style Sheets – kaskádové šablony stylů) a JavaScript, jinými slovy schopnost prohlížet zdrojové kódy webových stránek. Vůbec nemusíte být odborníky v těchto oblastech. Nejvýše se očekává kopírování a vkládání materiálu z knihy, který je k dispozici také ke stažení.

Jak knihu číst

První část této knihy zajistí, abyste byli připraveni. Po jejím přečtení budete mít za sebou úvod do jazyka Ruby, setkání se samotným frameworkem Rails, nainstalování jazyka Ruby i frameworku Rails a ověření této instalace pomocí jednoduchého příkladu.

V další části budete skrze delší příklad procházet koncepce stojící za frameworkem Rails – vybudujeme jednoduchý e-shop. Nebudete se ale postupně zabývat jednotlivými komponentami frameworku Rails („zde je kapitola o modelech, zde je kapitola

o pohledech“ a tak dále). Tyto komponenty jsou navržené tak, aby pracovaly společně, a každá kapitola v této části se týká určité sady souvisejících úkolů, jež zahrnují několik z těchto společně pracujících komponent.

Většina lidí ráda sestavuje aplikaci dle postupu v této knize. Nechcete-li všechno psát, můžete veškeré zdrojové kódy stáhnout na webu této knihy (*http://knihy.cpress.cz/k1958*).

Třetí část knihy zkoumá celý ekosystém frameworku Rails. Začíná funkcemi a možnostmi frameworku Rails, s nimiž jste se již dříve setkali. Poté se věnuje několika klíčovým závislostem, jež využívá framework Rails a které přímo přispívají k jeho celkové funkčnosti. Nakonec přijde na řadu přehled řady oblíbených zásuvných modulů, jež framework Rails rozšiřují a činí z něj otevřený ekosystém.

V průběhu čtení knihy se budete setkávat s několika konvencemi.

Živý kód

- Většina uváděných úryvků kódu pochází z kompletních, funkčních příkladů, které si můžete stáhnout. Pro lepší orientaci se před kódem, který se nachází v balíčku se zdrojovými kódy ke stažení, nachází takovýto proužek:

Soubor: work/demo1/app/controllers/say_controller.rb

```
class SayController < ApplicationController
  def hello
  end

  def goodbye
  end
end
```

- Ten obsahuje cestu k danému kódu uvnitř balíčku se zdrojovými kódy. V některých případech, kdy je nutné upravit stávající soubor, budou měněné řádky zvýrazněné. Ve výše uvedeném kódu jsou takto zvýrazněné dva řádky.

Tipy pro jazyk Ruby

Názvy v jazyce Ruby ➢ 67

- Přestože je nutné znát jazyk Ruby, abyste mohli psát aplikace na frameworku Rails, uvědomujeme si, že řada čtenářů se bude učit jazyk Ruby i framework Rails současně. Velmi stručné seznámení s jazykem Ruby najdete v kapitole 4, Úvod do jazyka Ruby. Při prvním použití konstruktů specifických pro jazyk Ruby budeme používat křížové odkazy do této kapitoly. Například tento odstavec bezdůvodně používá `:name`, což je symbol jazyka Ruby. Na okraji tedy uvidíte značku signalizující, že symboly jsou vysvětlované na straně 67.

David říká...

Občas můžete narazit na panel nadepsaný „David říká…" V něm vám David Heinemeier Hansson podává zajímavé informace o určitém aspektu frameworku Rails – zdůvodnění, triky, doporučení a další. Jedná se o člověka, který framework Rails vymyslel. Tyto části byste si měli rozhodně přečíst, chcete-li se stát odborníky na framework Rails.

Honza se ptá...

Honza, hypotetický vývojář, se někdy objeví s otázkou o tématu, o němž v daném textu hovoříme. V takto nadepsaném panelu pak hledáme odpovědi na tyto otázky.

Tato kniha nemá být referenční příručkou pro framework Rails. Naše zkušenost je taková, že referenční příručky nepředstavují způsob, jakým si většina lidí osvojuje nové znalosti. Místo toho si ukážeme většinu modulů a spoustu jejich metod, ať už pomocí příkladu, nebo popisem v textu, v kontextu jejich praktického použití a vzájemné spolupráce.

Nenajdete zde ani stovky stránek výpisů s rozhraním API. Máme pro to dobrý důvod – tuto dokumentaci získáte při každé instalaci frameworku Rails, kde bude navíc aktuálnější než materiál v této knize. Pokud nainstalujete framework Rails pomocí balíčkovacího systému RubyGems (což doporučujeme), tak jednoduše spusťte dokumentační server balíčků (pomocí příkazu `gem server`) a nasměrujte svůj prohlížeč na adresu *http://localhost:8808*, kde si můžete prohlédnout veškerá rozhraní API frameworku Rails. Na straně 297 najdete návod pro sestavení další dokumentace a průvodců.

Kromě toho uvidíte, že vám pomůže i samotný framework Rails, který generuje odpovědi, jež jasně identifikují jakoukoli případnou chybu, a také výpisy trasování, které vám řeknou nejen místo výskytu chyby, ale také to, jak jste se k němu dostali. Příklad takové odpovědi uvidíte na obrázku 10.3. Potřebujete-li další informace, nakoukněte do podkapitoly Iterace E2: Ošetřování chyb (str. 153), kde se dozvíte, jak vkládat příkazy pro protokolování.

Pokud byste se skutečně zasekli, pak vám může pomoci množství zdrojů na Internetu. Kromě zmíněných výpisů kódu máte k dispozici diskuzní fórum (*http://forums.pragprog.com/forums/148*), kde můžete pokládat otázky a sdílet své zkušenosti, stránku pro errata (*http://www.pragprog.com/titles/rails4/errata*), kde můžete nahlásit chyby, a wikiweb (*http://www.pragprog.com/wikis/wiki/RailsPlayTime*), kde můžete diskutovat o cvičeních uvedených v této knize.

Tyto zdroje jsou sdílenými zdroji. Nebojte se tedy přispívat do diskuzních fór a wikiwebu nejen otázkami a problémy, ale také návrhy a odpověďmi k otázkám druhých.

Pusťme se tedy do toho! První krokem je nainstalování jazyka Ruby a frameworku Rails a ověření instalace jednoduchou ukázkou.

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu přeložilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

redakce PC literatury
Computer Press
Spielberk Office Centre
Holandská 3
639 00 Brno

nebo

sefredaktor.pc@cpress.cz

Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Zdrojové kódy ke knize

Z adresy *http://knihy.cpress.cz/K1958* si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nedá. Pokud v některé z našich knih najdete chybu, ať už chybu v textu nebo v kódu, budeme rádi, pokud nám ji nahlásíte. Ostatní uživatele tak můžete ušetřit frustrace a pomoci nám zlepšit následující vydání této knihy.

Veškerá existující errata zobrazíte na adrese *http://knihy.cpress.cz/K1958* po klepnutí na odkaz Soubory ke stažení.