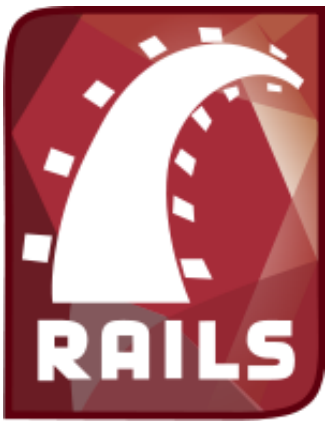
# Ruby on Rails

From Wikipedia, the free encyclopedia

**Ruby on Rails**, or simply **Rails**, is an open source web application framework written in Ruby. Rails is a full-stack framework that emphasizes the use of well-known software engineering patterns and paradigms, including convention over configuration (CoC), don't repeat yourself (DRY), the active record pattern, and model–view–controller (MVC).[3]

**Ruby on Rails**




| | |
|---|---|
| **Original author(s)** | David Heinemeier Hansson |
| **Developer(s)** | Rails Core Team (http://www.rubyonrails.org/core) |
| **Initial release** | 13 December 2005 |
| **Stable release** | 4.2.1 / March 19, 2015[1] |
| **Preview release** | 4.2.0.rc1 / November 28, 2014[2] |
| **Development status** | Active |
| **Written in** | Ruby |
| **Operating system** | Cross-platform |
| **Type** | Web application framework |
| **License** | Expat License |
| **Website** | rubyonrails.org (http://rubyonrails.org) |

## Contents



## History

David Heinemeier Hansson extracted Ruby on Rails from his work on Basecamp, a project management tool by 37signals (now a web application company).[4] Hansson first released Rails as open source in July 2004, but did not share commit rights to the project until February 2005.[5] In August 2006, the framework reached a milestone when Apple announced that it would ship Ruby on Rails with Mac OS X v10.5 "Leopard",[6] which was released in October 2007.

Rails version 2.3 was released on March 15, 2009 with major new developments in templates, engines, Rack and nested model forms. Templates enable the developer to generate a skeleton application with custom gems and configurations. Engines give developers the ability to reuse application pieces complete with routes, view paths and models. The Rack web server interface and Metal allow one to write optimized pieces of code that route around ActionController.[7]

On December 23, 2008, Merb, another web application framework, was launched, and Ruby on Rails announced it would work with the Merb project to bring "the best ideas of Merb" into Rails 3, ending the "unnecessary duplication" across both communities.[8] Merb was merged with Rails as part of the Rails 3.0 release.[9][10]

Rails 3.1 was released on August 31, 2011, featuring Reversible Database Migrations, Asset Pipeline, Streaming, jQuery as default JavaScript library and newly introduced CoffeeScript and Sass into the stack.[11]

Rails 3.2 was released on January 20, 2012 with a faster development mode and routing engine (also known as Journey engine), Automatic Query Explain and Tagged Logging.[12] Rails 3.2.x is the last version that supports Ruby 1.8.7.[13] Rails 3.2.12 supports Ruby 2.0[14]

Rails 4.0 was released on June 25, 2013, introducing Russian Doll Caching, Turbolinks, Live Streaming as well as making Active Resource, Active Record Observer and other components optional by splitting them as gems.[15]

Rails 4.1 was released on April 8, 2014, introducing Spring, Variants, Enums, Mailer previews, and secrets.yml.[16]

Rails 4.2 was released on December 19, 2014, introducing Active Job, asynchronous emails, Adequate Record, Web Console, and foreign keys.[1]

**Version history**

| Version | Date |
|---|---|
| 1.0[17] | December 13, 2005 |
| 1.2[18] | January 19, 2007 |
| 2.0[19] | December 7, 2007 |
| 2.1[20] | June 1, 2008 |
| 2.2[21] | November 21, 2008 |
| 2.3[22] | March 16, 2009 |
| 3.0[23] | August 29, 2010 |
| 3.1[24] | August 31, 2011 |
| 3.2[25] | January 20, 2012 |
| 4.0[26] | June 25, 2013 |
| 4.1[27] | April 8, 2014 |
| 4.2[1] | December 19, 2014 |

# Technical overview

Like many web frameworks, Ruby on Rails uses the model–view–controller (MVC) pattern to organize application programming.

In a default configuration, a *model* in the Ruby on Rails framework maps to a table in a database, and to a Ruby file. For example, a model class, *User,* will usually be defined in the file, 'user.rb', in the app/models directory, and linked to the table, 'users', in the database. While developers are free to ignore this convention and choose differing names for their models, files, and database table, this is not common practice and is usually discouraged in accordance with the "convention-over-configuration" philosophy.

A *controller* is a client-side component of Rails that responds to external requests from the web server to the application, by determining which view file to render. The controller may also have to query one or more models directly for information and pass these on to the view. A controller may provide one or more actions. In Ruby on Rails, an action is typically a basic unit that describes how to respond to a specific external web-browser request. Also note that the controller/action will be accessible for external web requests only if a corresponding route is mapped to it. Rails encourages developers to use RESTful routes, which include actions such as: create, new, edit, update, destroy, show, and index. These mappings of incoming requests/routes to controller actions can be easily set up in the routes configuration file.

A *view* in the default configuration of Rails is an erb file, which is evaluated and converted to HTML at run-time. Alternatively, many other templating systems can be used for views.

Ruby on Rails includes tools that make common development tasks easier "out-of-the-box", such as scaffolding that can automatically construct some of the models and views needed for a basic website.[28] Also included are WEBrick, a simple Ruby web server that is distributed with Ruby, and Rake, a build system, distributed as a gem. Together with Ruby on Rails, these tools provide a basic development environment.

Ruby on Rails is most commonly not connected to the Internet directly, but through some front-end web server. Mongrel was generally preferred over WEBrick in the early days, but it can also run on Lighttpd, Apache, Cherokee, Hiawatha, nginx (either as a module — Phusion Passenger for example — or via CGI, FastCGI or mod_ruby), and many others. From 2008 onwards, Passenger replaced Mongrel as the most-used web server for Ruby on Rails.[29] Ruby is also supported natively on the IBM i.[30]

Ruby on Rails is also noteworthy for its extensive use of the JavaScript libraries, Prototype and Script.aculo.us, for scripting Ajax actions.[31] Ruby on Rails initially utilized lightweight SOAP for web services; this was later replaced by RESTful web services. Ruby on Rails 3.0 uses a technique called Unobtrusive JavaScript to separate the functionality (or logic) from the structure of the web page. jQuery is fully supported as a replacement for Prototype and is the default JavaScript library in Rails 3.1, reflecting an industry-wide move towards jQuery. Additionally, CoffeeScript was introduced in Rails 3.1 as the default Javascript language.

Since version 2.0, Ruby on Rails offers both HTML and XML as standard output formats. The latter is the facility for RESTful web services.

Rails 3.1 introduced Sass as standard CSS templating.

By default, the server uses Embedded Ruby in the HTML views, with files having an html.erb extension. Rails supports swapping-in alternative templating languages, such as HAML and Mustache.

Ruby on Rails 3.0 has been designed to work with Ruby 1.8.7, Ruby 1.9.2, and JRuby 1.5.2+; earlier versions are not supported.[32]

Rails 3.2 series is the last series to support Ruby 1.8.7.

### Framework structure

Ruby on Rails is separated into various packages, namely ActiveRecord (an object-relational mapping system for database access), ActiveResource (provides web services), ActionPack, ActiveSupport and ActionMailer. Prior to version 2.0, Ruby on Rails also included the Action Web Service package that is now replaced by Active Resource. Apart from standard packages, developers can make plugins to extend existing packages. Rails 3.2 deprecates the old plugins Rails 2-3-stable style in which plugins are to be placed under vendor/plugins, in favor of packaged gems.[33]

### Deployment

Ruby on Rails is often installed using RubyGems, a package manager[34] which is included with current versions of Ruby. Many free Unix-like systems also support installation of Ruby on Rails and its dependencies through their native package management system.

Ruby on Rails is typically deployed with a database server such as MySQL or PostgreSQL, and a web server such as Apache running the Phusion Passenger module.

## Philosophy and design

Ruby on Rails is intended to emphasize *Convention over Configuration* (CoC), and the *Don't Repeat Yourself* (DRY) principle.

"Convention over Configuration" means a developer only needs to specify unconventional aspects of the application. For example, if there is a class *Sale* in the model, the corresponding table in the database is called *sales* by default. It is only if one deviates from this convention, such as calling the table "products sold", that the developer needs to write code regarding these names. Generally, Ruby on Rails conventions lead to less code and less repetition.[35]

"Don't repeat yourself" means that information is located in a single, unambiguous place. For example, using the ActiveRecord module of Rails, the developer does not need to specify database column names in class definitions. Instead, Ruby on Rails can retrieve this information from the database based on the class name.

"Fat models, skinny controllers" means that most of the application logic should be placed within the model while leaving the controller as light as possible.

## Trademarks

In March 2007, David Heinemeier Hansson filed three Ruby on Rails-related trademark applications to the USPTO. These applications regard the phrase "RUBY ON RAILS",[36] the word "RAILS",[37] and the official Rails logo.[38] As a consequence, in the summer of 2007, Hansson denied permission to Apress to use the Ruby on Rails logo on the cover of a new Ruby on Rails book written by some authoritative community members. The episode gave rise to a polite protest in the Ruby on Rails community.[39][40] In response to this criticism, Hansson replied:

> I only grant promotional use [of the Rails logo] for products I'm directly involved with. Such as books that I've been part of the development process for or conferences where I have a say in the execution. I would most definitely seek to enforce all the trademarks of Rails.[39]

# Reception

## Scalability

Rails running on Matz's Ruby Interpreter (the *de facto* reference interpreter for Ruby) had been criticized for issues with scalability.[41] These critics often mentioned various Twitter outages in 2007 and 2008, which spurred Twitter's partial transition to Scala (which runs on the Java Virtual Machine) for their queueing system and other middleware.[42][43] The user interface aspects of the site continued to run Ruby on Rails[44] until 2011 when it was replaced due to concerns over performance [45]

In 2011, Gartner Research noted that despite criticisms and comparisons to Java, many high-profile consumer web firms are using Ruby on Rails to build agile, scalable web applications. Some of the largest sites running Ruby on Rails include GitHub, Yammer, Scribd, Shopify, Hulu, and Basecamp.[46] As of May 2014, it is estimated that more than 600,000 web sites are running Ruby on Rails.[47]

## Security

On September 24, 2013, a session cookie persistence security flaw was reported in Ruby on Rails. In a default configuration, the entire session hash is stored within a session cookie known as CookieStore, allowing any authenticated session possessing the session cookie to log in as the target user at any time in the future. As a workaround, administrators are advised to configure cookies to be stored on the server using mechanisms such as ActiveRecordStore. [48]

# References


1. rafaelfranca. "[ANN] Rails 4.2.1 and 4.1.10 have been released!" (http://weblog.rubyonrails.org/2015/3/19/Rails-4-2-1-and-4-1-10-have-been-released/). *Ruby on Rails*. Retrieved 27 Mar 2015.
2. "Rails 4.2.0.rc1 has been released!" (http://weblog.rubyonrails.org/2014/11/28/Rails-4-9-0-rc1-has-been-released/).
3. "Getting Started with Rails: What Is Rails?" (http://guides.rubyonrails.org/getting_started.html#what-is-rails-questionmark). *Ruby on Rails Guides*. Retrieved 10 August 2014.
4. Grimmer, Lenz (February 2006). "Interview with David Heinemeier Hansson from Ruby on Rails" (https://web.archive.org/web/20130225091835/http://dev.mysql.com/tech-resources/interviews/david-heinemeier-hansson-rails.html). MySQL AB. Retrieved 2008-06-08.
5. "Rails core team profiles." (http://www.rubyonrails.org/core). Retrieved 2008-07-15.
6. Hansson, David (August 7, 2006). "Ruby on Rails will ship with OS X 10.5 (Leopard)" (http://weblog.rubyonrails.org/2006/8/7/ruby-on-rails-will-ship-with-os-x-10-5-leopard). Retrieved 2008-06-08.

7. Hansson, David (March 16, 2009). "Rails 2.3: Templates, Engines, Rack, Metal, much more!" (http://weblog.rubyonrails.org/2009/3/16/rails-2-3-templates-engines-rack-metal-much-more).
8. "The day Merb joined Rails" (http://rubyonrails.org/merb). 2008-12-27.
9. Ruby on Rails 3.0 Release Notes (http://edgeguides.rubyonrails.org/3_0_release_notes.html)
10. "Ruby on Rails 3.0 goes modular" (http://www.sdtimes.com/link/34122). sdtimes.com. 2010-02-10. Retrieved 2010-08-06.
11. "Ruby on Rails 3.1 Release Notes" (http://guides.rubyonrails.org/3_1_release_notes.html). 2012-09-01.
12. "Ruby on Rails 3.2 Release Notes" (http://guides.rubyonrails.org/3_2_release_notes.html). 2012-09-01.
13. "Rails/master is now 4.0.0.beta" (http://weblog.rubyonrails.org/2011/12/20/rails-master-is-now-4-0-0-beta/). 2012-09-01.
14. Rails 3.2.x is now compatible with Ruby 2.0.0 by sikachu · Pull Request #9406 · rails/rails · GitHub (https://github.com/rails/rails/pull/9406). Github.com. Retrieved on 2014-05-30.
15. Rails 4 is finally released (http://weblog.rubyonrails.org/2013/6/25/Rails-4-0-final/)
16. Rails 4.1.0: Spring, Variants, Enums, Mailer previews, secrets.yml (http://weblog.rubyonrails.org/2014/4/8/Rails-4-1/)
17. "Rails 1.0: Party like it's one oh oh!" (http://weblog.rubyonrails.org/2005/12/13/rails-1-0-party-like-its-one-oh-oh/). Riding Rails. Retrieved June 9, 2010.
18. "Rails 1.2: REST admiration, HTTP lovefest, and UTF-8 celebrations" (http://weblog.rubyonrails.org/2007/1/18/rails-1-2-rest-admiration-http-lovefest-and-utf-8-celebrations). Riding Rails. Retrieved June 9, 2010.
19. "Rails 2.0: It's done!" (http://weblog.rubyonrails.org/2007/12/7/rails-2-0-it-s-done). Riding Rails. Retrieved June 9, 2010.
20. Rails 2.1: Time zones, dirty, caching, gem dependencies, caching, etc (http://weblog.rubyonrails.org/2008/5/31/rails-2-1-time-zones-dirty-caching-gem-dependencies-caching-etc). Riding Rails. Retrieved June 9, 2010.
21. "Rails 2.2: i18n, HTTP validators, thread safety, JRuby/1.9 compatibility, docs" (http://weblog.rubyonrails.org/2008/11/21/rails-2-2-i18n-http-validators-thread-safety-jruby-1-9-compatibility-docs). Riding Rails. Retrieved June 9, 2010.
22. "Rails 2.3: Templates, Engines, Rack, Metal, much more!" (http://weblog.rubyonrails.org/2009/3/16/rails-2-3-templates-engines-rack-metal-much-more). Riding Rails. Retrieved June 9, 2010.
23. "Rails 3.0: It's ready!" (http://weblog.rubyonrails.org/2010/8/29/rails-3-0-it-s-done). Riding Rails. Retrieved August 30, 2010.
24. [1] (https://github.com/rails/rails/commit/505e84599aff6abf719484636b0515e1ce2e2220) www.github.com
25. [2] (https://github.com/rails/rails/commit/f36dcaf488b4357a52f43e3912628428956d351f) www.github.com
26. [3] (http://blog.wyeworks.com/2012/10/29/rails-4-in-30-minutes/) blog.wyeworks.com
27. [4] (http://weblog.rubyonrails.org/2014/4/8/Rails-4-1/) weblog.rubyonrails.org
28. There were quite a few changes in the 2.0 release, including the way that Ruby on Rails generates scaffolding code. (http://fairleads.blogspot.com/2007/12/rails-20-and-scaffolding-step-by-step.html)
29. Official deployment instructions suggests use of Passenger (http://rubyonrails.org/deploy)
30. Maxcer, Chris. "PowerRuby Brings Enterprise Ruby on Rails Support to IBM i" (http://iprodeveloper.com/application-development/powerruby-brings-enterprise-ruby-rails-support-ibm-i). iPro Developer. Retrieved 9 May 2014.
31. Ruby on Rails includes the Prototype JavaScript framework and the Scriptaculous JavaScript controls and visual effects library. (http://api.rubyonrails.org/classes/ActionView/Helpers/JavaScriptHelper.html)

32. "Rails 3.0: It's ready!" (http://weblog.rubyonrails.org/2010/8/29/rails-3-0-it-s-done). rubyonrails.org. Retrieved 2010-08-30. "Rails 3.0 has been designed to work with Ruby 1.8.7, Ruby 1.9.2, and JRuby 1.5.2+."
33. "Rails 3.2.0.rc2 has been released!" (http://weblog.rubyonrails.org/2012/1/4/rails-3-2-0-rc2-has-been-released/). 2012-09-01.
34. "Ruby on Rails: Download" (http://rubyonrails.org/download). RubyonRails.org.
35. "Getting Started with Rails" (http://guides.rubyonrails.org/getting_started.html). Rubyonrails.org. Retrieved 2014-03-10.
36. ""Ruby on Rails" Trademark Status" (http://tarr.uspto.gov/servlet/tarr?regser=serial&entry=77119206). USPTO. Retrieved 2007-08-01.
37. ""Rails" Trademark Status" (http://tarr.uspto.gov/servlet/tarr?regser=serial&entry=77119210). USPTO. Retrieved 2007-08-01.
38. "Rails Logo Trademark Status" (http://tarr.uspto.gov/servlet/tarr?regser=serial&entry=77119208). USPTO. Retrieved 2007-08-01.
39. Forde, Pete (2007-07-23). "Beginning Rails: From Novice to Professional" (http://rethink.unspace.ca/2007/7/23/beginning-rails-from-novice-to-professional). Retrieved 2007-08-01.
40. Cooper, Peter (2007-07-24). "David Heinemeier Hansson says No to Use of Rails Logo" (http://www.rubyinside.com/david-heinemeier-hansson-says-no-to-use-of-rails-logo-567.html). Retrieved 2007-08-01.
41. "5 Question Interview with Twitter Developer Alex Payne" (https://web.archive.org/web/20090423002030/http://www.radicalbehavior.com/5-question-interview-with-twitter-developer-alex-payne/?). radicalbehavior.com. 2007-03-29. Retrieved 2014-11-04. "*By various metrics Twitter is the biggest Rails site on the net right now. Running on Rails has forced us to deal with scaling issues - issues that any growing site eventually contends with – far sooner than I think we would on another framework.*"
42. Steve Jenson, Alex Payne, and Robey Pointer interview (2009-04-03). "Twitter on Scala" (http://www.artima.com/scalazine/articles/twitter_on_scala.html). artima.com. Retrieved 2009-07-18. "*We had a Ruby-based queuing system that we used for communicating between the Rails front ends and the daemons, and we ended up replacing that with one written in Scala. The Ruby one actually worked pretty decently in a normal steady state, but the startup time and the crash behavior were undesirable.*"
43. "Twitter jilts Ruby for Scala" (http://www.theregister.co.uk/2009/04/01/twitter_on_scala/). theregister.co.uk. 2009-04-01. Retrieved 2009-07-18. "*By the end of this year, Payne said, Twitter hopes to have its entire middleware infrastructure and its APIs ported to the new language. Ruby will remain, but only on the front end. "We're still happy with Rails for building user facing features... performance-wise, it's fine for people clicking around web pages. It's the heavy lifting, asynchronous processing type of stuff that we've moved away from."*"
44. ryan king (2009-09-25). "Twitter on Ruby" (http://blog.evanweaver.com/articles/2009/09/24/ree/). evan weaver. Retrieved 2009-09-29. "*We use Scala for a few things at Twitter, but the majority of the site is Ruby.*"
45. "Twitter Search is Now 3x Faster" (https://blog.twitter.com/2011/twitter-search-now-3x-faster). Twitter. Retrieved June 6, 2014..
46. "Here's Why Ruby On Rails Is Hot" (http://articles.businessinsider.com/2011-05-11/tech/30035869_1_ruby-rails-custom-software). Business Insider. Retrieved February 10, 2012.

47. "Ruby on Rails Usage Trends" (http://trends.builtwith.com/framework/Ruby-on-Rails). BuiltWith Trends. Retrieved May 4, 2014.
48. Brook, Chris (2013-09-25). "Security Issue in Ruby on Rails Could Expose Cookies" (http://threatpost.com/security-issue-in-ruby-on-rails-could-expose-cookies/102413). *Threat Post* (The Kaspersky Lab Security News Service). Retrieved 2014-06-04.

# Bibliography

- Ruby, Sam; Thomas, Dave; Hansson, David (March 28, 2009). "Agile Web Development with Rails" (http://www.pragprog.com/titles/rails3/agile-web-development-with-rails-third-edition) (Third ed.). Pragmatic Bookshelf. p. 850. ISBN 1-934356-16-6.
- Laurent, Simon St.; Dumbill, Edd (November 28, 2008). "Learning Rails" (http://oreilly.com/catalog/9780596518776/) (First ed.). O'Reilly Media. p. 442. ISBN 0-596-51877-3.
- Lenz, Patrick (May 1, 2008). "Simply Rails 2" (http://www.sitepoint.com/books/rails2/) (Second ed.). SitePoint. p. 450. ISBN 0-9804552-0-0.
- Tate, Bruce; Hibbs, Curt (August 22, 2006). "Ruby on Rails: Up and Running" (http://oreilly.com/catalog/9780596101329/) (First ed.). O'Reilly Media. p. 182. ISBN 0-596-10132-5.
- Holzner Ph.D., Steve (November 29, 2006). "Beginning Ruby on Rails" (http://www.wrox.com/WileyCDA/WroxTitle/Beginning-Ruby-on-Rails.productCd-0470069155.html) (First ed.). Wrox. p. 380. ISBN 0-470-06915-5.
- Allan Hardy, Jeffrey; Carneiro Jr, Cloves; Catlin, Hampton (July 20, 2007). "Beginning Ruby on Rails E-Commerce: From Novice to Professional" (http://www.apress.com/book/view/9781590597361) (First ed.). Wrox. p. 361. ISBN 1-59059-686-2.
- Clark, Mike (May 15, 2008). "Advanced Rails Recipes" (http://www.pragprog.com/titles/fr_arr/advanced-rails-recipes) (First ed.). Pragmatic Bookshelf. p. 464. ISBN 0-9787392-2-1.

# External links

- Ruby on Rails official website (http://www.rubyonrails.org/) – rubyonrails.org
- Railscasts.com (http://railscasts.com/) – these screencasts are short and focus on Ruby on Rails technique
- Rails source code (https://github.com/rails/rails)

Wikibooks has a book on the topic of: ***Ruby on Rails***

Retrieved from "http://en.wikipedia.org/w/index.php?title=Ruby_on_Rails&oldid=654380669"

Categories: 2004 introductions | Free computer libraries | Free software programmed in Ruby | Web 2.0 | Web application frameworks | Software using the MIT license

---

- This page was last modified on 31 March 2015, at 18:59.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.