

ACTIVESTATE RUBY ON RAILS FOR WEB APPLICATIONS



Ruby on Rails (RoR) is a server-side web development framework that's often been called the killer app for Ruby. Primarily used to develop database-backed web applications, it combines the ease of use of Ruby with Rails' powerful web app framework, allowing not only for fast prototyping but also the creation of web platforms of any complexity.

As an interpreted, high-level, programming language, Ruby places an emphasis on creating clear, syntactically simple, logical code that makes any development team highly productive. Typical of interpreted languages, Ruby is dynamically typed, uses garbage collection and provides just-in-time compilation whose speed has been improved three fold in recent releases. And despite the fact that Ruby was originally released in the mid-1990's, the Rails framework (released in 2005) continues to make the Ruby language relevant today, driving popular sites from Basecamp and GitHub to Shopify and AirBnB.

	Ruby	Ruby on Rails
What is it?	A language	A framework
What inspired it?	Perl and Smalltalk	Python's Django web framework
What language is it written in?	C	Ruby
Primary Use Case?	Desktop apps	Data-driven web app

SPEED, COST, FLEXIBILITY – CHOOSE THREE

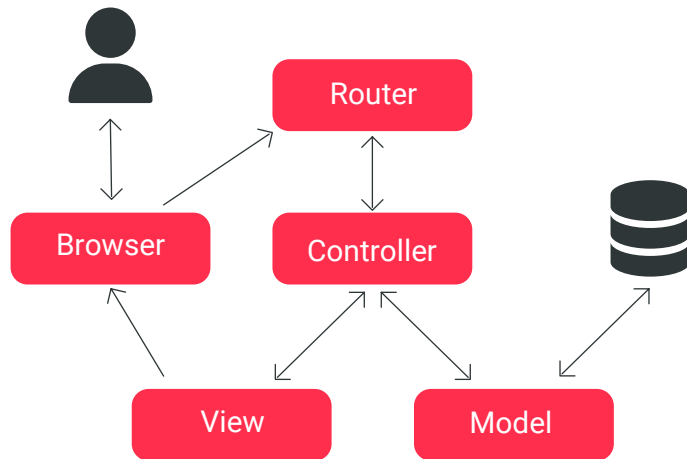
Ruby's longevity is no accident. RoR remains a top choice for web developers simply because of the framework's efficiency and the speed with which a codebase can be generated. Developer productivity benefits from the fact that RoR consists of pre-written Ruby code for things like communication, file handling, database connections, and more. By taking care of the lower level routines, it allows developers to focus on coding features that deliver value. And adoption of key concepts like Don't Repeat Yourself (DRY) and Convention over Configuration (CoC) means developers need to write fewer lines of redundant code.

Since time is money, accelerated coding speed lowers development costs. By providing default web service, web page and database structures (such as seamless database table creation) RoR not only facilitates web app creation, but maintenance, as well. And since Ruby is easily readable and largely self-documenting, developers can maximize their time spent coding.

The Rails framework evolves rapidly from release to release, adopting newer technologies and standards much faster than many comparable frameworks. As a result, it offers developers the flexibility to work with the best-suited standard/technology for their project. And because RoR supports a wide range of integrations with popular technologies and frameworks like Angular and React, you can create separate layers of your web application that use multiple, different technologies.

MVC ARCHITECTURE MADE EASY

Getting started writing a RoR web application is quick and easy thanks to the fact that everything has a pre-assigned home. RoR enforces the Model-View-Controller (MVC) web app architecture, which centralizes the business logic of the application and the rules to manipulate the data:



- **Models** contain the logic of an application, all the essential data, and high-level classes. By default, a RoR Model consists of one or more tables in a database and a Ruby file.
- **Views** are the UI that users interact with and see on their screens. Typically, RoR Views are erb files, which embed Ruby code into an HTML document.
- **Controllers** are server-side components that connect the Model and the View via routes, which allow for RESTful create, edit, update, etc actions. Mapping of incoming requests/routes to controller actions are usually set up in a routes.rb configuration file.

ACTIVESTATE RUBY ON RAILS

While open source Ruby provides many of the WebDev tools and libraries you'll need, development teams can end up wasting days on the low-value work of building, configuring and troubleshooting their development environment before they can even write their first line of code. ActiveState Ruby lets anyone on the team configure and automatically build a secure Rails environment from source code for Windows and Linux in a matter of minutes. No language or OS expertise is required to create a runtime environment that acts as a single source of truth for all your team members, ensuring they can work with a consistent, reproducible environment that eliminates "works on my machine" issues.

ActiveState is the de-facto standard for millions of developers around the world who have been using our commercially-backed, secure open source language distributions for over 20 years. With the ActiveState Platform, developers can now automatically build their own open source artifacts and environments—all without requiring language or operating system expertise.

You can try the ActiveState Platform by signing up for a free account at platform.activestate.com

ActiveState®

www.activestate.com

Toll-free in NA: 1-866.631.4581

solutions@activestate.com

©2022 ActiveState Software Inc. All rights reserved. ActiveState®, ActivePerl®, ActiveTcl®, ActivePython®, Komodo®, ActiveGo™, ActiveRuby™, ActiveNode™, ActiveLua™, and The Open Source Languages Company™ are all trademarks of Activetate.

[Get a Demo](#)

[Contact Sales](#)