

## Outline

- [What's Rails?](#)
- [Model layer](#)
- [View layer](#)
- [Controller layer](#)
- [Frameworks and libraries](#)
- [Getting Started](#)
- [Contributing](#)
- [License](#)

## Modules

- [AbstractController](#)
- [ActionCable](#)
- [ActionController](#)
- [ActionDispatch](#)
- [ActionMailbox](#)
- [ActionMailer](#)
- [ActionText](#)
- [ActionView](#)
- [ActiveJob](#)
- [ActiveModel](#)
- [ActiveRecord](#)
- [ActiveStorage](#)
- [ActiveSupport](#)
- [Arel](#)
- [Mail](#)
- [Mime](#)
- [Minitest](#)
- [Rails](#)

## Core Extensions

- [Array](#)
- [Benchmark](#)
- [BigDecimal](#)
- [Class](#)
- [Date](#)
- [DateTime](#)
- [Delegator](#)
- [Digest](#)
- [ERB](#)
- [Enumerable](#)
- [Exception](#)
- [FalseClass](#)
- [File](#)
- [Float](#)
- [Hash](#)
- [IO](#)
- [Integer](#)
- [Kernel](#)
- [LoadError](#)
- [Method](#)
- [Module](#)
- [NameError](#)
- [NilClass](#)
- [Numeric](#)
- [Object](#)
- [Pathname](#)
- [Process](#)
- [Range](#)
- [Regexp](#)
- [SecureRandom](#)
- [Singleton](#)
- [String](#)
- [Symbol](#)
- [Thread](#)
- [Time](#)
- [TrueClass](#)
- [URI](#)
- [UnboundMethod](#)

# Welcome to Rails

## What's Rails?

Rails is a web-application framework that includes everything needed to create database-backed web applications according to the [Model-View-Controller \(MVC\)](#) pattern.

Understanding the MVC pattern is key to understanding Rails. MVC divides your application into three layers: Model, View, and Controller, each with a specific responsibility.

## Model layer

The ***Model layer*** represents the domain model (such as Account, Product, Person, Post, etc.) and encapsulates the business logic specific to your application. In Rails, database-backed model classes are derived from `ActiveRecord::Base`. [Active Record](#) allows you to present the data from database rows as objects and embellish these data objects with business logic methods. Although most Rails models are backed by a database, models can also be ordinary Ruby classes, or Ruby classes that implement a set of interfaces as provided by the [Active Model](#) module.

## View layer

The ***View layer*** is composed of “templates” that are responsible for providing appropriate representations of your application's resources. Templates can come in a variety of formats, but most view templates are HTML with embedded Ruby code (ERB files). Views are typically rendered to generate a controller response or to generate the body of an email. In Rails, View generation is handled by [Action View](#).

## Controller layer

The ***Controller layer*** is responsible for handling incoming HTTP requests and providing a suitable response. Usually, this means returning HTML, but Rails controllers can also generate XML, JSON, PDFs, mobile-specific views, and more. Controllers load and manipulate models, and render view templates in order to generate the appropriate HTTP response. In Rails, incoming requests are routed by Action Dispatch to an appropriate controller, and controller classes are derived from `ActionController::Base`. Action Dispatch and Action Controller are bundled together in [Action Pack](#).

## Frameworks and libraries

[Active Record](#), [Active Model](#), [Action Pack](#), and [Action View](#) can each be used independently outside Rails.

In addition to that, Rails also comes with:

- [Action Mailer](#), a library to generate and send emails
- [Action Mailbox](#), a library to receive emails within a Rails application
- [Active Job](#), a framework for declaring jobs and making them run on a variety of queuing backends
- [Action Cable](#), a framework to integrate WebSockets with a Rails application
- [Active Storage](#), a library to attach cloud and local files to Rails applications
- [Action Text](#), a library to handle rich text content
- [Active Support](#), a collection of utility classes and standard library extensions that are useful for Rails, and may also be used independently outside Rails

## Getting Started

1. Install Rails at the command prompt if you haven't yet:

```
$ gem install rails
```

2. At the command prompt, create a new Rails application:

```
$ rails new myapp
```

where "myapp" is the application name.

3. Change directory to `myapp` and start the web server:

```
$ cd myapp
$ bin/rails server
```

Run with `--help` or `-h` for options.

4. Go to `http://localhost:3000` and you'll see the Rails bootscreen with your Rails and Ruby versions.
5. Follow the guidelines to start developing your application. You may find the following resources handy:
  - [Getting Started with Rails](#)
  - [Ruby on Rails Guides](#)
  - [The API Documentation](#)

## Contributing

We encourage you to contribute to Ruby on Rails! Please check out the [Contributing to Ruby on Rails guide](#) for guidelines about how to proceed. [Join us!](#)

Trying to report a possible security vulnerability in Rails? Please check out our [security policy](#) for guidelines about how to proceed.

Everyone interacting in Rails and its sub-projects' codebases, issue trackers, chat rooms, and mailing lists is expected to follow the Rails [code of conduct](#).

## License

Ruby on Rails is released under the [MIT License](#).