

Table of Contents



- [1. What is Ruby on Rails?](#)
- [2. Install Ruby on Rails](#)
- [3. Create a New Rails Project](#)
- [4. Project Structure](#)
- [5. Generate Resources](#)
 - [Generate Model, Controller, and Views](#)
 - [Apply Migrations](#)
- [6. MVC Structure](#)
 - [Model](#)
 - [Controller](#)
 - [View](#)
- [7. Routing](#)
- [8. Create a Form](#)
- [9. CRUD Operations](#)
 - [Create \(New & Save\)](#)
 - [Read \(Index & Show\)](#)
 - [Update](#)
 - [Delete](#)
- [10. Validations](#)
- [11. Database Migrations](#)
 - [Create a Migration](#)
- [12. Associations](#)
 - [One-to-Many](#)
- [13. Partial](#)
- [14. Flash Messages](#)
- [15. Debugging](#)
- [16. Testing](#)
 - [Generate Tests](#)
 - [Run Tests](#)
- [17. Rails Console](#)
- [18. Seed Database](#)
- [19. API with Rails](#)
- [20. Useful Commands](#)

1. What is Ruby on Rails?

Ruby on Rails (RoR) is a **web application framework** written in Ruby. It follows the **Model-View-Controller (MVC)** architecture and promotes **convention over configuration**.

2. Install Ruby on Rails

```
# Install Ruby
brew install ruby          # macOS
sudo apt install ruby-full # Linux
```

```
# Install Rails
gem install rails
```

Verify Installation:

```
ruby -v
rails -v
```

3. Create a New Rails Project

```
rails new myapp
cd myapp
```

Run the development server:

```
rails server
```

Visit <http://localhost:3000> in your browser.

4. Project Structure

```
myapp/
├── app/                # Application code (MVC)
│   ├── controllers/   # Controllers (Business logic)
│   ├── models/        # Models (Data and DB logic)
│   └── views/         # Views (HTML, ERB templates)
├── config/            # Configuration files
├── db/                # Database schema and migrations
├── public/            # Static files
├── Gemfile             # Gem dependencies
└── Rakefile            # Task automation
```

5. Generate Resources

Generate Model, Controller, and Views

```
rails generate scaffold Article title:string body:text
```

- **Model:** Defines the structure and relationships
 - **Controller:** Manages requests and responses
 - **View:** Renders HTML for users
-

Apply Migrations

```
rails db:migrate
```

6. MVC Structure

Model

```
app/models/article.rb
```

```
class Article < ApplicationRecord
  validates :title, presence: true
end
```

Controller

app/controllers/articles_controller.rb

```
class ArticlesController < ApplicationController
  def index
    @articles = Article.all
  end

  def show
    @article = Article.find(params[:id])
  end
end
```

View

app/views/articles/index.html.erb

```
<h1>Articles</h1>
<%= link_to 'New Article', new_article_path %>

<% @articles.each do |article| %>
  <h2><%= article.title %></h2>
  <p><%= article.body %></p>
  <%= link_to 'Show', article_path(article) %>
<% end %>
```

7. Routing

config/routes.rb

```
Rails.application.routes.draw do
  resources :articles # RESTful routes for articles
  root "articles#index"
end
```

8. Create a Form

app/views/articles/_form.html.erb

```
<%= form_with(model: @article, local: true) do |form| %>
  <div>
    <%= form.label :title %>
    <%= form.text_field :title %>
  </div>

  <div>
    <%= form.label :body %>
    <%= form.text_area :body %>
  </div>

  <div>
    <%= form.submit %>
  </div>
<% end %>
```

9. CRUD Operations

Create (New & Save)

```
@article = Article.new(article_params)
```

```
if @article.save
  redirect_to @article
else
  render :new
end
```

Read (Index & Show)

```
@articles = Article.all
@article = Article.find(params[:id])
```

Update

```
@article = Article.find(params[:id])
if @article.update(article_params)
  redirect_to @article
else
  render :edit
end
```

Delete

```
@article.destroy
redirect_to articles_path
```

10. Validations

app/models/article.rb

```
class Article < ApplicationRecord
  validates :title, presence: true
  validates :body, length: { minimum: 10 }
```

end

11. Database Migrations

Create a Migration

```
rails generate migration AddPublishedToArticles published:boolean  
  
db/migrate/xxxx_add_published_to_articles.rb
```

```
class AddPublishedToArticles < ActiveRecord::Migration[6.1]  
  def change  
    add_column :articles, :published, :boolean, default: false  
  end  
end
```

Apply migration:

```
rails db:migrate
```

12. Associations

One-to-Many

Article has many **comments**.

```
class Article < ApplicationRecord  
  has_many :comments  
end
```

Comment belongs to an **article**.

```
class Comment < ApplicationRecord
```

```
    belongs_to :article
  end
```

13. Partial

app/views/articles/_article.html.erb

```
<h2><%= article.title %></h2>
<p><%= article.body %></p>
```

Render in views:

```
<%= render @articles %>
```

14. Flash Messages

```
flash[:notice] = "Article was successfully created"
flash[:alert] = "Something went wrong"
```

15. Debugging

```
<%= debug(params) %>
```

16. Testing

Generate Tests

```
rails generate test_unit:scaffold Article
```


Run Tests

```
rails test
```

17. Rails Console

```
rails console
```

```
Article.all
```

```
Article.create(title: "Rails Guide")
```

18. Seed Database

```
db/seeds.rb
```

```
Article.create(title: "First Article", body: "This is the body")
```

Run seeds:

```
rails db:seed
```

19. API with Rails

```
# config/routes.rb
namespace :api do
  resources :articles
end
```

```
app/controllers/api/articles_controller.rb
```

```
class Api::ArticlesController < ApplicationController
  def index
    render json: Article.all
  end
end
```

```
end  
end
```

20. Useful Commands

```
rails routes          # List all routes  
rails dbconsole       # Connect to DB  
rails generate model  # Create a model  
rails generate scaffold # Create CRUD resources  
rails destroy scaffold # Rollback  
rails db:rollback     # Undo last migration
```