

Cheat Sheet for comprehensive Ruby on Rails

Comprehensive Ruby on Rails Cheat Sheet

1. Getting Started

Installation

- **Ruby:** ``rbenv install 3.0.0``
- **Rails:** ``gem install rails``
- **Create a new Rails app:** ``rails new my_app``
- **Start the server:** ``rails server`` or ``rails s``

Directory Structure

- **app/:** Core application code (models, views, controllers, helpers)
 - **config/:** Application configuration files
 - **db/:** Database schema and migrations
 - **public/:** Static files (images, stylesheets, JavaScript)
 - **test/:** Test files
-

2. Rails Commands

Basic Commands

- **Generate a new resource:** ``rails generate scaffold Post title:string body:text``
- **Run migrations:** ``rails db:migrate``
- **Rollback migrations:** ``rails db:rollback``
- **Create a new controller:** ``rails generate controller Posts``
- **Create a new model:** ``rails generate model Post title:string body:text``
- **Run tests:** ``rails test``
- **Install dependencies:** ``bundle install``

Advanced Commands

- **Generate a migration:** `rails generate migration AddPublishedToPosts published:boolean`
 - **Destroy a generated resource:** `rails destroy scaffold Post`
 - **Open Rails console:** `rails console` or `rails c`
 - **Open database console:** `rails dbconsole` or `rails db`
-

3. Routing

Basic Routing

- **Root route:** `root 'posts#index'`
- **Resourceful routes:** `resources :posts`
- **Nested routes:** `resources :posts do resources :comments end`
- **Custom routes:** `get 'posts/recent', to: 'posts#recent'`

Route Helpers

- **Path helpers:** `posts_path`, `new_post_path`, `edit_post_path(post)`
- **URL helpers:** `posts_url`, `new_post_url`, `edit_post_url(post)`

Route Constraints

```
get 'posts/:id', to: 'posts#show', constraints: { id: /\d+/ }
```

4. Controllers

Basic Structure

```
class PostsController < ApplicationController
  def index
    @posts = Post.all
  end

  def show
    @post = Post.find(params[:id])
  end

  def new
```

```
      @post = Post.new
    end

    def create
      @post = Post.new(post_params)
      if @post.save
        redirect_to @post
      else
        render 'new'
      end
    end

    private

    def post_params
      params.require(:post).permit(:title, :body)
    end
  end
end
```

Filters

- **Before action:** `before_action :set_post, only: [:show, :edit, :update, :destroy]`
- **After action:** `after_action :log_action, only: [:create, :update]`

Strong Parameters

```
def post_params
  params.require(:post).permit(:title, :body)
end
```

5. Models

Basic Structure

```
class Post < ApplicationRecord
  validates :title, presence: true
  has_many :comments
end
```

Validations

- **Presence:** `validates :title, presence: true`
- **Length:** `validates :body, length: { minimum: 10 }`

- **Uniqueness:** `validates :title, uniqueness: true`
- **Numericality:** `validates :price, numericality: { greater_than: 0 }`

Callbacks

- **Before save:** `before_save :normalize_title`
- **After create:** `after_create :send_notification`
- **Around update:** `around_update :log_update`

Scopes

```
scope :published, -> { where(published: true) }  
scope :recent, -> { order(created_at: :desc) }
```

6. Views

ERB Templates

```
<h1><%= @post.title %></h1>  
<p><%= @post.body %></p>
```

Form Helpers

```
<%= form_with model: @post do |form| %>  
  <%= form.label :title %>  
  <%= form.text_field :title %>  
  
  <%= form.label :body %>  
  <%= form.text_area :body %>  
  
  <%= form.submit %>  
<% end %>
```

Partials

```
<%= render 'shared/errors', object: @post %>
```

Layouts

```
<!DOCTYPE html>  
<html>  
<head>
```

```
<title>My App</title>
<%= stylesheet_link_tag 'application', media: 'all' %>
<%= javascript_include_tag 'application' %>
<%= csrf_meta_tags %>
</head>
<body>
  <%= yield %>
</body>
</html>
```

7. Migrations

Creating Tables

```
class CreatePosts < ActiveRecord::Migration[6.1]
  def change
    create_table :posts do |t|
      t.string :title
      t.text :body
      t.boolean :published, default: false
      t.timestamps
    end
  end
end
```

Modifying Tables

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

Dropping Tables

```
class DropPosts < ActiveRecord::Migration[6.1]
  def change
    drop_table :posts
  end
end
```

8. Associations

One-to-Many

```
class Post < ApplicationRecord
  has_many :comments
end

class Comment < ApplicationRecord
  belongs_to :post
end
```

Many-to-Many

```
class User < ApplicationRecord
  has_many :posts
  has_many :comments, through: :posts
end

class Post < ApplicationRecord
  belongs_to :user
  has_many :comments
end

class Comment < ApplicationRecord
  belongs_to :post
end
```

Polymorphic

```
class Picture < ApplicationRecord
  belongs_to :imageable, polymorphic: true
end

class Employee < ApplicationRecord
  has_many :pictures, as: :imageable
end

class Product < ApplicationRecord
  has_many :pictures, as: :imageable
end
```

9. Validations

Common Validations

- **Presence:** `validates :title, presence: true`
- **Length:** `validates :body, length: { minimum: 10 }`
- **Uniqueness:** `validates :title, uniqueness: true`
- **Numericality:** `validates :price, numericality: { greater_than: 0 }`

Custom Validations

```
validate :title_must_be_capitalized

def title_must_be_capitalized
  errors.add(:title, 'must be capitalized') unless title ==
  title.capitalize
end
```

10. Callbacks

Common Callbacks

- **Before save:** `before_save :normalize_title`
- **After create:** `after_create :send_notification`
- **Around update:** `around_update :log_update`

Example

```
class Post < ApplicationRecord
  before_save :normalize_title

  private

  def normalize_title
    self.title = title.downcase.capitalize
  end
end
```

11. Scopes

Basic Scopes

```
class Post < ApplicationRecord
  scope :published, -> { where(published: true) }
  scope :recent, -> { order(created_at: :desc) }
end
```

Chaining Scopes

```
Post.published.recent
```

12. Helpers

Custom Helpers

```
module PostsHelper
  def format_date(date)
    date.strftime('%B %d, %Y')
  end
end
```

Using Helpers in Views

```
<%= format_date(@post.created_at) %>
```

13. Testing

RSpec Setup

- **Install RSpec:** `gem install rspec-rails`
- **Generate RSpec files:** `rails generate rspec:install`

Writing Tests

```
require 'rails_helper'

RSpec.describe Post, type: :model do
  it 'is valid with a title and body' do
    post = Post.new(title: 'Test Post', body: 'This is a test post')
    expect(post).to be_valid
  end
end
```



```
end

it 'is invalid without a title' do
  post = Post.new(body: 'This is a test post')
  expect(post).to_not be_valid
end
end
```

Running Tests

- **Run all tests:** `rspec`
 - **Run specific test file:** `rspec spec/models/post_spec.rb`
-

14. Deployment

Heroku

- **Install Heroku CLI:** `brew install heroku/brew/heroku`
- **Login to Heroku:** `heroku login`
- **Create a new app:** `heroku create`
- **Deploy:** `git push heroku main`
- **Migrate database:** `heroku run rails db:migrate`

Capistrano

- **Install Capistrano:** `gem install capistrano`
 - **Configure Capistrano:** `cap install`
 - **Deploy:** `cap production deploy`
-

15. Advanced Topics

Active Job

```
class SendNotificationJob < ApplicationJob
  queue_as :default

  def perform(user)
    UserMailer.welcome_email(user).deliver_later
  end
end
```

Action Cable

```
class ChatChannel < ApplicationCable::Channel
  def subscribed
    stream_from "chat_#{params[:room]}"
  end

  def unsubscribed
    # Any cleanup needed when channel is unsubscribed
  end

  def send_message(data)
    ActionCable.server.broadcast("chat_#{params[:room]}", message:
data['message'])
  end
end
```

Active Storage

```
class User < ApplicationRecord
  has_one_attached :avatar
end
```

I18n

```
# config/locales/en.yml
en:
  hello: "Hello world"

# app/views/posts/index.html.erb
<%= t('hello') %>
```

This cheat sheet provides a comprehensive overview of essential Ruby on Rails concepts, commands, and best practices. Use it as a quick reference to build robust and efficient Rails applications.

By Ahmed Baheeg Khorshid

ver 1.0