

Cheat Sheet for comprehensive Ruby on Rails

Rails Basics

Creating a New Rails Application

```
rails new my_app
rails new my_app --database=postgresql
rails new my_app --skip-test-unit
```

Running the Rails Server

```
rails server
rails s
```

Generating Resources

```
rails generate model User name:string email:string
rails generate controller Users index show
rails generate scaffold Post title:string content:text
```

Running Migrations

```
rails db:migrate
rails db:rollback
rails db:migrate:status
```

Models

Associations

- **One-to-Many:** `has_many :posts` and `belongs_to :user`
- **Many-to-Many:** `has_and_belongs_to_many :tags`
- **Polymorphic:** `has_many :pictures, as: :imageable`

Validations

```
class User < ApplicationRecord
  validates :name, presence: true
  validates :email, uniqueness: true
  validates :age, numericality: { greater_than: 18 }
end
```

Callbacks

```
class User < ApplicationRecord
  before_save :normalize_name
  after_create :send_welcome_email

  private
  def normalize_name
    self.name = name.downcase.titleize
  end

  def send_welcome_email
    # Logic to send email
  end
end
```

Controllers

Basic Structure

```
class UsersController < ApplicationController
  def index
    @users = User.all
  end

  def show
    @user = User.find(params[:id])
  end
end
```

Strong Parameters

```
class UsersController < ApplicationController
  def create
    @user = User.new(user_params)
    if @user.save
      redirect_to @user
    else
      render 'new'
    end
  end

  private
  def user_params
    params.require(:user).permit(:name, :email)
  end
end
```

Filters

```
class ApplicationController < ActionController::Base
  before_action :authenticate_user!
  after_action :log_action

  private
  def authenticate_user!
    # Authentication logic
  end

  def log_action
    # Logging logic
  end
end
```

Views

ERB Basics

```
<h1><%= @user.name %></h1>
<p><%= @user.email %></p>
```

Form Helpers

```
<%= form_with model: @user do |form| %>
  <%= form.label :name %>
  <%= form.text_field :name %>

  <%= form.label :email %>
  <%= form.email_field :email %>

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

Partials

```
<%= render 'shared/header' %>
```

Routing

Basic Routes

```
Rails.application.routes.draw do
  get 'users', to: 'users#index'
  get 'users/:id', to: 'users#show', as: 'user'
```

```
resources :posts
end
```

Nested Resources

```
Rails.application.routes.draw do
  resources :users do
    resources :posts
  end
end
```

Custom Routes

```
Rails.application.routes.draw do
  get 'about', to: 'pages#about'
  root 'pages#home'
end
```

Active Record

Querying

```
User.where(name: 'John')
User.find_by(email: 'john@example.com')
User.order(created_at: :desc)
User.limit(10)
```

Aggregations

```
User.count
User.average(:age)
User.sum(:age)
```

Updating Records

```
user = User.find(1)
user.update(name: 'John Doe')
user.save
```

Helpers

URL Helpers

```
users_path
user_path(@user)
edit_user_path(@user)
```

Form Helpers

```
form_with(model: @user, local: true) do |form|
  form.text_field :name
  form.submit
end
```

Testing

RSpec

```
describe User do
  it "is valid with a name and email" do
    user = User.new(name: 'John', email: 'john@example.com')
    expect(user).to be_valid
  end
end
```

Capbara

```
describe "the signin process", type: :feature do
  before :each do
    User.create(email: 'user@example.com', password: 'password')
  end

  it "signs me in" do
    visit '/sessions/new'
    within("#session") do
      fill_in 'Email', with: 'user@example.com'
      fill_in 'Password', with: 'password'
    end
    click_button 'Sign in'
    expect(page).to have_content 'Success'
  end
end
```

Deployment

Capistrano

```
# config/deploy.rb
set :application, 'my_app'
set :repo_url, 'git@github.com:username/my_app.git'
set :deploy_to, '/var/www/my_app'
set :linked_files, %w{config/database.yml config/secrets.yml}
set :linked_dirs, %w{log tmp/pids tmp/cache tmp/sockets vendor/bundle
public/system}
```

Heroku

```
heroku create
git push heroku master
heroku run rails db:migrate
heroku open
```

Tips and Tricks

Debugging

- Use `byebug` for debugging in development.
- Use `Rails.logger.debug` for logging.

Performance

- Use `includes` for eager loading associations.
- Use `counter_cache` for performance in associations.

Security

- Use `has_secure_password` for password management.
- Use `attr_accessible` or strong parameters to prevent mass assignment.

Caching

- Use `Rails.cache.fetch` for caching data.
- Use `fragment_cache` for caching view fragments.

Useful Gems

Gem Name	Description
Devise	Authentication solution
CarrierWave	File uploading solution
Kaminari	Pagination solution

Sidekiq	Background job processing	
RSpec	Testing framework	
Capybara	Acceptance testing framework	
Simple Form	Form builder	
Pundit	Authorization library	

Conclusion

This cheat sheet covers the essential aspects of Ruby on Rails, from basic setup to advanced features like testing, deployment, and performance optimization. Use this as a quick reference to build robust and efficient Rails applications.

By Ahmed Baheeg Khorshid

ver 1.0