

# Dynamic routing in Ruby



janko-m

@jankomarohnic



Mail

Sequel  
ROM  
Mongoid

Ruby stdlib  
dry-inflector  
TZInfo  
Symmetric Encryption  
Time Math / as-duration

Paperclip  
CarrierWave  
Dragonfly  
Refile  
Shrine

~~ActiveRecord~~

~~ActiveSupport~~

~~ActiveStorage~~

~~ActionMailer~~

ActionPack

~~Sprockets~~

~~ActiveJob~~

~~ActionView~~

~~ActionCable~~

Sidekiq  
Resque  
Queue Classic

Tilt  
Forme

Faye  
websocket-driver

Webpack

# 1. Routing

```
# config/routes.rb

Rails.application.routes.draw do
  get "/" => "home#index"

  resources :albums do
    resources :photos
  end

  # ...
end
```

# 2. Handling

```
# app/controllers/albums.rb

class AlbumsController < ApplicationController
  # GET /albums/new
  def new
  end

  # POST /albums
  def create
  end

  # ...
end
```

**GET /photos/123/edit**



**ActionDispatch::Routing**



**Journey**



**PhotosController#edit**



## DESCRIPTION:

---

Journey is a router. It routes requests.

Note: This gem was merged on Rails 4.0 and will only receive security fixes.

## FEATURES/PROBLEMS:

---

- Designed for rails

## SYNOPSIS:

---

```
Too complex right now. :(
```

Branch: master ▾

[rails](#) / [actionpack](#) / [lib](#) / [action\\_dispatch](#) / [journey](#) /

Create new file

Upload files

Find file

History



**fatkodima** Fix array routing constraints

Latest commit 94a27cb 4 days ago

..

📁	<a href="#">gtg</a>	[Action Pack] require => require_relative	4 months ago
📁	<a href="#">nfa</a>	Enable `Layout/LeadingCommentSpace` to not allow cosmetic changes in ...	2 months ago
📁	<a href="#">nodes</a>	Write the code in a more natural way.	4 days ago
📁	<a href="#">path</a>	Fix array routing constraints	4 days ago
📁	<a href="#">router</a>	Update links to use https link instead of http [ci skip]	6 months ago
📁	<a href="#">visualizer</a>	CSS fix for the router visualizer	3 years ago
📄	<a href="#">formatter.rb</a>	[Action Pack] `rubocop -a --only Layout/EmptyLineAfterMagicComment`	8 months ago
📄	<a href="#">parser.rb</a>	[Action Pack] require => require_relative	4 months ago
📄	<a href="#">parser.y</a>	[Action Pack] require => require_relative	4 months ago
📄	<a href="#">parser_extras.rb</a>	[Action Pack] require => require_relative	4 months ago
📄	<a href="#">route.rb</a>	Use frozen string literal in actionpack/	7 months ago
📄	<a href="#">router.rb</a>	removed unnecessary returns	4 months ago
📄	<a href="#">routes.rb</a>	Use frozen string literal in actionpack/	7 months ago
📄	<a href="#">scanner.rb</a>	correct the dedup code	5 days ago
📄	<a href="#">visitors.rb</a>	[Action Pack] `rubocop -a --only Layout/EmptyLineAfterMagicComment`	8 months ago

```
1 class ActionDispatch::Journey::Parser
2   options no_result_var
3   token SLASH LITERAL SYMBOL LPAREN RPAREN DOT STAR OR
4
5   rule
6     expressions
7       : expression expressions { Cat.new(val.first, val.last) }
8       | expression            { val.first }
9       | or
10      ;
11    expression
12      : terminal
13      | group
14      | star
15      ;
16    group
17      : LPAREN expressions RPAREN { Group.new(val[1]) }
18      ;
19    or
20      : expression OR expression { Or.new([val.first, val.last]) }
21      | expression OR or { Or.new([val.first, val.last]) }
22      ;
23    star
24      : STAR { Star.new(Symbol.new(val.last)) }
25      ;
26    terminal
27      : symbol
```

# Journey



## *A Journey into the Rails router*

*Lexers, parsers, and automata*

by Andy Lindeman, ATLRUG 2015

*“Journey is a regular expression engine”*

`^/posts/[^/]+/edit$`

POSIX regex

`/posts/:id/edit`

"Journey" regex

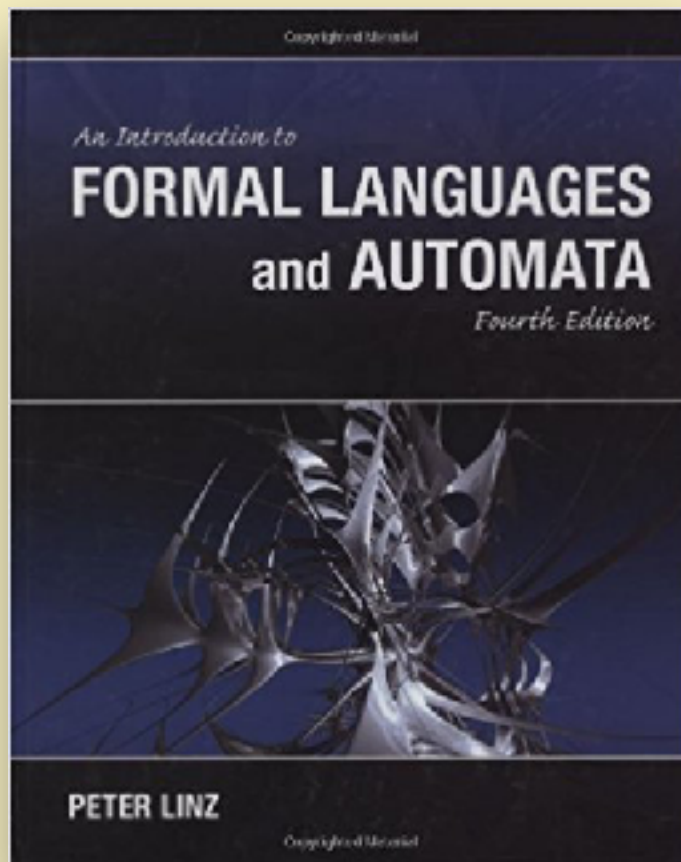


# Journey



## *A Journey into the Rails router* *Lexers, parsers, and automata*

by Andy Lindeman, ATLRUG 2015



- Lexers
- Parsers
- Grammars (with YACC)
- Tokens
- (Non)Deterministic finite automata
- Generalized transition graph (GTG)
- ...

```
# config/routes.rb
```

```
Rails.application.routes.draw do
```

```
  get "/" => "home#index"
```

```
  resources :albums do
```

```
    resources :photos
```

```
  end
```

```
# ...
```

```
end
```

```
# config/routes.rb
```

```
Rails.application.routes.draw do
```

```
  get "/" => "home#index"
```

```
  resources :albums do
```

```
    resources :photos
```

```
  end
```

```
  if request.env["HTTP_USER_AGENT"] =~ /iPhone/
```



```
    get "/iphone-app" => "iphone#index"
```

```
  end
```

```
  # ...
```

```
end
```

```
# config/routes.rb
```

```
Rails.application.routes.draw do
```

```
  get "/" => "home#index"
```

```
  resources :albums do
```

```
    resources :photos
```

```
  end
```

```
  constraints(-> (req) { req.env["HTTP_USER_AGENT"] =~ /iPhone/ }) do
```

```
    get "/iphone-app" => "iphone#index"
```

```
  end
```

```
  # ...
```

```
end
```



Sinatra

{grape}



HANAMI

# "Dynamic" routing



Cuba



*Roda*





*Roda*



```
class App < Roda
  route do |r|
    # ...
  end
end
```



```
class App < Roda
  route do |r|
    r.get "hello" do
      "Hello world!"
    end
  end
end
```

```
class App < Roda
  route do |r|
    r.get "hello" do
      "Hello world!"
    end

    r.post "answer" do
      "42"
    end
  end
end
```

```
class App < Roda
  route do |r|
    if r.path == "/hello" && r.get?
      response.write "Hello world!"
      r.halt
    end

    if r.path == "/answer" && r.post?
      response.write "42"
      r.halt
    end
  end
end
```

# Example 1

## CRUD

```
r.get "albums", String do |album_id|  
  @album = Album[album_id]  
  view :album  
end
```

```
r.put "albums", String do |album_id|  
  @album = Album[album_id]  
  @album.update(r.params["album"])  
  r.redirect  
end
```

```
r.delete "albums", String do |album_id|  
  @album = Album[album_id]  
  @album.destroy  
  r.redirect "/"  
end
```

```
r.is "albums", String do |album_id|  
  r.get do  
    @album = Album[album_id]  
    view :album  
  end  
  
  r.put do  
    @album = Album[album_id]  
    @album.update(r.params["album"])  
    r.redirect  
  end  
  
  r.delete do  
    @album = Album[album_id]  
    @album.destroy  
    r.redirect "/"  
  end  
end
```

```
r.is "albums", String do |album_id|  
  @album = Album[album_id]  
  
  r.get do  
    view :album  
  end  
  
  r.put do  
    @album.update(r.params["album"])  
    r.redirect  
  end  
  
  r.delete do  
    @album.destroy  
    r.redirect "/"  
  end  
end
```

```
class AlbumsController < ApplicationController
  before_action :find_album, only: [:show, :update]

  def show
  end

  def update
    @album.update(params[:album])
    redirect_to @album
  end

  # ...

  private

  def find_album
    @album = Album.find(params[:id])
  end
end
```



# **Example II**

## **Before Action Hell**

```
class RubygemsController < ApplicationController
  before_action :redirect_to_root, only: %i[edit update], unless: :signed_in?
  before_action :set_blacklisted_gem, only: %i[show], if: :blacklisted?
  before_action :find_rubygem, only: %i[edit update show], unless: :blacklisted?
  before_action :latest_version, only: %i[show], unless: :blacklisted?
  before_action :find_versioned_links, only: %i[show], unless: :blacklisted?
  before_action :load_gem, only: %i[edit update]
  before_action :set_page, only: :index

  def index
    # ...
  end

  def show
    # ...
  end

  def edit
    # ...
  end

  def update
    # ...
  end
end
```

```
r.on "gems" do
  r.get true do
    set_page
    # ...
  end

  r.on String do |gem_name|
    find_rubygem unless blacklisted?

    r.get true do
      set_blacklisted_gem if blacklisted?
      latest_version unless blacklisted?
      find_versioned_links unless blacklisted?
      # ...
    end

    redirect_to_root unless signed_in?
    load_gem

    r.get "edit" do
      # ...
    end

    r.put true do
      # ...
    end
  end
end
end
```

# Example III

## Authentication

```
class ApplicationController < ActionController::Base
  before_action :require_login!
```

```
  private
```

```
  def require_login!
```

```
    # ...
```

```
  end
```

```
end
```

```
class LoginController < ApplicationController
```

```
  skip_before_action :require_login!
```

```
end
```

```
class DocumentationController < ApplicationController
```

```
  skip_before_action :require_login!
```

```
end
```

```
class ArticlesController < ApplicationController
```

```
  skip_before_action :require_login!, only: [:index]
```

```
end
```

```
class App < Roda
  route do |r|
    r.on "login" do
      # ...
    end

    r.on "documentation" do
      # ...
    end

    r.get "articles" do
      # ...
    end

    require_login!

    r.on "profile" do
      # ...
    end

    # ...
  end
end
```

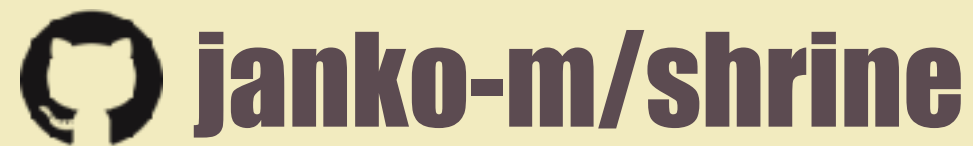


Not authenticated

Authenticated

# Example IV

## Shrine



## File Attachment toolkit for Ruby applications

```
# Rack app for direct uploads  
Shrine.upload_endpoint(:disk)
```

```
Rails.application.routes.draw do  
  # Adds `POST /upload` route  
  mount Shrine.upload_endpoint(:disk) => "/upload"  
end
```

**Q: How to authenticate direct uploads?**



```
# Rack app for direct uploads  
Shrine.upload_endpoint(:disk)
```

```
Rails.application.routes.draw do  
  # Adds `POST /upload` route  
  mount Shrine.upload_endpoint(:disk) => "/upload"  
end
```

## Devise

✗ `Devise::Controller#authenticate_user!`

```
# Rack app for direct uploads  
Shrine.upload_endpoint(:disk)
```

```
Rails.application.routes.draw do  
  authenticate(:user) do  
    mount Shrine.upload_endpoint(:disk) => "/upload"  
  end  
end
```

## Devise

✗ Devise::Controller#authenticate\_user!

✓ ActionController::Routing::Mapper#authenticate

```
# Rack app for direct uploads  
Shrine.upload_endpoint(:disk)
```

```
Rails.application.routes.draw do  
  mount Shrine.upload_endpoint(:disk) => "/upload"  
end
```

## Sorcery?

```
# Rack app for direct uploads  
Shrine.upload_endpoint(:disk)
```

```
Rails.application.routes.draw do  
  mount Shrine.upload_endpoint(:disk) => "/upload"  
end
```

## Clearance?

```
# Rack app for direct uploads  
Shrine.upload_endpoint(:disk)
```

```
Rails.application.routes.draw do  
  mount Shrine.upload_endpoint(:disk) => "/upload"  
end
```

## Authlogic?

```
class Roda < App
  route do |r|
    r.on "upload" do
      authenticate_user!

      r.run Shrine.upload_endpoint(:disk)
    end
  end
end
```

# Example V

## Matchers

```
Rails.application.routes.draw do
  resources :albums
end
```

```
class AlbumsController < ApplicationController
  before_action :set_album,
    only: [:show, :edit, :update, :destroy]

  # ...

  def set_album
    @album = Album.find(params[:id])
  rescue ActiveRecord::RecordNotFound
    render json: { error: "Record Not Found" }
  end
end
```



**GET /albums/12345**

**{ "error": "Record Not Found" }**

**GET /albums/recent**

**{ "error": "Record Not Found" }**



```
Rails.routes.application.draw do
  get    "/albums"           => "albums#index"
  get    "/albums/new"       => "albums#new"
  post   "/albums"           => "albums#create"
  get    "/albums/:id"       => "albums#show"
  get    "/albums/:id/edit"  => "albums#edit"
  put    "/albums/:id"       => "albums#update"
  delete "/albums/:id"       => "albums#destroy"
end
```

**:id** – matches any string

```
r.is "albums", String do |album_id|  
  album_id #=> "12345"  
  # ...  
end
```

```
r.is "albums", Integer do |album_id|  
  album_id #=> 12345  
  # ...  
end
```

```
Roda.plugin :symbol_matchers
Roda.symbol_matcher(
  :uuid,
  /\w{8}-\w{4}-\w{4}-\w{4}-\w{12}/)
```

```
r.is "albums", :uuid do |album_id|
  album_id #=> "44be628b-733f-415c-..."
  # ...
end
```

```
Rails.routes.application.draw do
  get "/articles/:y/:m/:d/:slug" => "articles#show"
  # ...
end
```

```
r.on "articles" do
  r.on Integer, Integer, Integer do |y, m, d|
    y #=> 2018
    m #=> 2
    d #=> 27
    # ...
  end
end
```

```
Roda.plugin :class_matchers
Roda.class_matcher(
  Date, /\d{4}\d{2}\d{2}/) do |y, m, d|
    [Date.new(y.to_i, m.to_i, d.to_i)]
  end
end
```

```
r.on "articles" do
  r.on Date do |date|
    date #=> #<Date: 2018-02-26>
    # ...
  end
end
```



# Advantages

- **Flexible**
- **Precise**
- **Simple (no automatas 😊)**
- **DRY**
- **Natural**

# Disadvantages

- No route introspection

```
r.on "articles" do
  # route: GET /articles
  r.get true do
    end

  # route: GET /articles/:id
  r.get String do |album_id|
    end
end
```



[jeremyevans/roda-route\\_list](https://github.com/jeremyevans/roda-route_list)

# Ecosystem

- **Rodauth** – authentication and account management framework
- **dry-web-roda** – lightweight web application stack
- **Autoforme** – web administrative console
- **tus-ruby-server** – app for resumable uploads