



SETUP & BASIC APP

```
gem install sinatra
# app.rb
require 'sinatra'
get '/' do
  'Hello World!'
end
ruby app.rb # starts on port 4567
rerun app.rb # auto-reload on change
```

ROUTES

```
get '/users' do; json User.all; end
post '/users' do
  user =
  User.create(JSON.parse(request.body.read))
  status 201; json user
end
put '/users/:id' do; ... end
delete '/users/:id' do;
  User.find(params[:id]).destroy; end
```

PARAMS

```
get '/users/:id' do
  id = params[:id] # route parameter
  User.find(id).to_json
end
get '/search' do
  q = params[:q] # query string
  ?q=alice
  User.where(name: q).to_json
end
# POST body (form-encoded):
name = params[:name]
```

VIEWS

```
# views/ directory with .erb files
get '/users' do
  @users = User.all
  erb :users_index # renders
  views/users_index.erb
end
get '/page' do
  haml :page # or slim, liquid, etc
end
# Inline:
erb '<%= @name %>', locals: { name:
'Alice' }
```

BEFORE/AFTER FILTERS

```
before do
  @current_user = User.find_by(token:
request.env['HTTP_AUTHORIZATION'])
  halt 401, json(error: 'Unauthorized')
  unless @current_user
  end
  before '/admin/*' do
  halt 403 unless @current_user.admin?
  end
  after do
  headers 'Content-Type' =>
'application/json'
  end
```

SESSIONS

```
enable :sessions
set :session_secret,
'change-me-in-production'
# In route:
session[:user_id] = user.id
user_id = session[:user_id]
session.clear
# Cookie-based by default; use
rack-session-pool for server-side
```

HALT & REDIRECT

```
halt 404, 'Not found'
halt 401, json(error: 'Unauthorized')
redirect '/'
redirect '/login', 302
redirect back # to referrer
# In before filter:
halt(401, {error: 'No token'}.to_json)
unless token
```

MODULAR vs CLASSIC

```
# Classic (top-level DSL, one app):
require 'sinatra'
get '/' do; 'Hello'; end
# Modular (multiple apps, explicit run):
require 'sinatra/base'
class MyApp < Sinatra::Base
  get '/' do; 'Hello'; end
end
# config.ru: run MyApp
```

COMMON ROUTE HELPERS

```
json(data) # {Content-Type:
application/json}
status 201 # set response
status
headers 'X-Foo' => '1' # set response
headers
send_file 'file.pdf' # file download
stream { |out| out << 'chunk' } #
streaming response
request.body.read # raw request body
request.ip # client IP
address
```

COMMON GOTCHAS

Classic mode pollutes global scope
use Modular for production
enable :sessions stores in cookie
set strong secret
Sinatra halts after first matching
route
request.body.read can only be read
once rewind if needed
No built-in ORM pair with Sequel,
ActiveRecord, or ROM