




# Lightweight Apps with Sinatra

Aadesh Shrestha  
@aadeshere1  
TECHRISE



# Sinatra

---

Sinatra is a DSL for quickly creating web applications in Ruby with minimal effort:

---

# Why use Sinatra ?

---

Exposed Simplicity Instead of  
Hidden Complexity

---

# Why use Sinatra ?

---

## Flexible

- You can choose how you do things
  - It's not opinionated
-

# Why use Sinatra ?

Makes building web apps as simple

# Why use Sinatra ?

---

## Get Started Quickly

- Write single file and get your app up & running

# Why use Sinatra ?



## Helpful Community



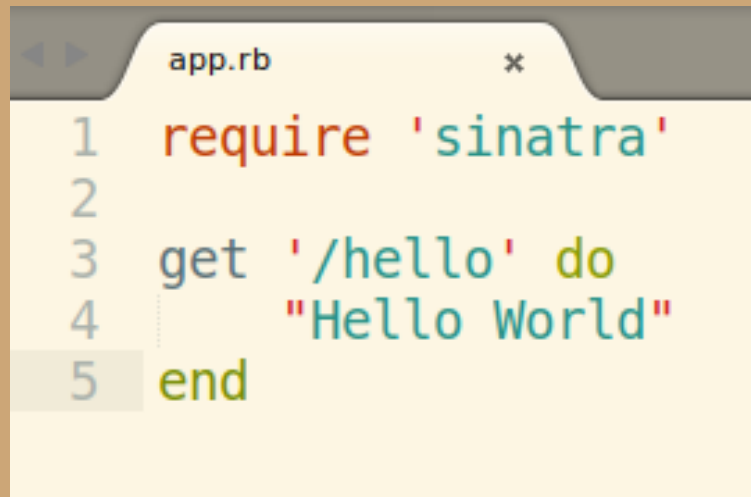
# Usefulness of Sinatra

- 
1. Testing out Ideas
  2. Prototyping Website
  3. APIs
  4. Putting existing program on web
-





## First Program

A screenshot of a code editor window titled 'app.rb'. The editor contains five lines of Ruby code. Line 1: 'require 'sinatra''. Line 2: an empty line. Line 3: 'get '/hello' do'. Line 4: ' "Hello World"'. Line 5: 'end'. The code is color-coded: 'require' is red, 'sinatra' is blue, 'get' is blue, 'do' is green, 'end' is green, and the string 'Hello World' is blue. The line numbers 1 through 5 are on the left side of the editor.

```
1 require 'sinatra'  
2  
3 get '/hello' do  
4   "Hello World"  
5 end
```

ruby app.rb

## Anatomy of Route Handler

```
1  require 'sinatra'
2
3  get('/:name' do
4      name = params[:name]
5      "Hello #{name}"
6  end
```

---

#1      require sinatra

#3      get                      -> http  
verb

#4      /:name -> named  
parameter

#5      last line is sent to  
browser

---

## Different HTTP verbs

1. GET
2. POST
3. PUT
4. PATCH
5. DELETE

## Inline Views

---

**Allows to write your views in the same file as the routes handler.**

**An actual application can be created in a single file**

---

```
1 require 'sinatra'
2
3 get '/:name' do
4   @name = params[:name]
5   erb :greet
6 end
7 END
8 @@greet
9 <h1>Greetings <%= @name %></h1>
```

---

#4 instance variables are available in views

#5 templating engine / name of view

#7 end of ruby code

#8 view names start with @@

#9 view code

---

## External Views

---

**Allows to write your views in the separate file inside 'views' directory**

**'layout.erb' renders view files**

---

## External Views

Code will appear in

`<%= yield %>`



## Views

**You can mix Inline and External Views**

## Some Useful Methods

```
1  error do
2    |   @error = request.env['sinatra_error']
3    |   erb :error
4  end
5
6  not_found do
7    |   "Page not found"
8  end
9
```

# Configuration

```
1 configure do
2   set :site_title, "Lightweight Apps with Sinatra"
3 end
```

## Configuration

---

Settings can be referred from  
routes handlers and views

`settings.set_title`

---

## Helper Methods

Methods that can be used in  
route handlers & views.

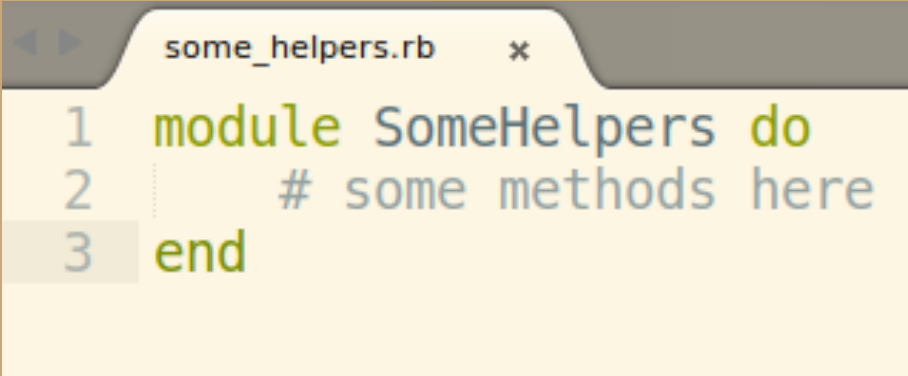
Inline

---

```
1 helpers do
2   # some methods here
3 end
```

---

External



A code editor window titled 'some\_helpers.rb' with a close button. It contains three lines of Ruby code: a module definition, a comment, and the end of the module. The first line is highlighted with a light gray background.

```
1 module SomeHelpers do
2   # some methods here
3 end
```

```
require './some_helpers.rb'
helpers SomeHelpers
```

## Databases

---

## ORM

Mongoid  
DataMapper  
ActiveRecord

---



## Mongoid (Mongo DB)

---

```
1 require 'sinatra'
2 require 'mongoid'
3
4 Mongoid.load! 'mongoid.config'
```

---

**sinatra-contrib**

---

**gem 'sinatra-contrib'**

```
1 require 'sinatra'
2 require 'sinatra/reloader' if development?
```

---





**Thank You  
for your time.**