

Friday, February 10, 12

## Not Frank Sinatra

DSL for writing web applications in Ruby

EdgeCase}

# It's this easy

require 'sinatra'

get '/' do
 'Hello World!'
end

EdgeCase}

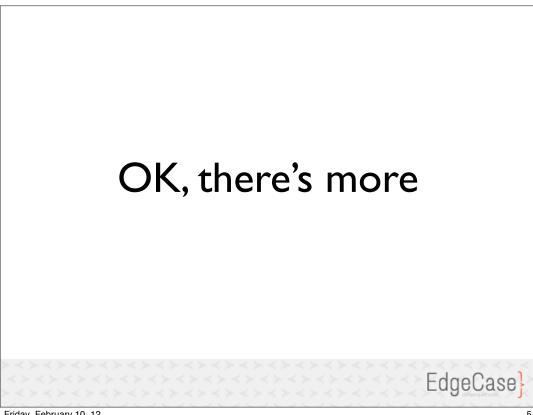
Friday, February 10, 12

3

## BOOM!

WEBSERVER!

EdgeCase}



Friday, February 10, 12

### Routes

EdgeCase}

# HTTP method with URL matcher

EdgeCase}

Friday, February 10, 12

## **HTTP GET**

get '/' do
 # show something
end

EdgeCase}

## HTTP POST

post '/' do
 # create something
end

EdgeCase}

Friday, February 10, 12

Э

## HTTP PUT

```
put '/' do
    # replace something
end
```

EdgeCase}

## HTTP DELETE

delete '/' do
 # remove something
end

EdgeCase}

Friday, February 10, 12

11

## named parameters

```
get '/:name' do
    "Hello #{params[:name]}"
end
```

EdgeCase}

# matching regular expressions

```
get %r{/hello/([\w]+)} do
    "Hello #{params[:captures].first}"
end
```

EdgeCase}

Friday, February 10, 12

13

## optional parameters

```
get '/posts.?:format' do
  # GET /posts
  # GET /posts.json
  # GET /posts.xml
end
```

EdgeCase}



Friday, February 10, 12

# served from ./public

EdgeCase}

# or manually

set :public\_folder, File.dirname(\_\_FILE\_\_) + '/static'

EdgeCase}

Friday, February 10, 12

17

# Views / Templates

EdgeCase}

builder rdoc
erb

liquid
coffeescript

sass haml

textile markdown

EdgeCase

Friday, February 10, 12

# named templates

```
get '/' do
    # render views/index.erb
    erb :index
end
```

EdgeCase}

```
get '/' do
    # render views/index.haml
    haml :index
end
```

EdgeCase}

Friday, February 10, 12

21

## custom layouts

```
get '/' do
  haml :index, :layout => post
end
```

EdgeCase}

# embedded templates

```
get '/:name' do
  erb 'Hello <%= params[:name] %>'
end
```

EdgeCase}

Friday, February 10, 12

23

#### **Filters**

EdgeCase}

#### before filters

before '/protected/\*' do
 authenticate!
end

EdgeCase}

Friday, February 10, 12

25

## after filters

after do
 puts response.status
end

EdgeCase}

#### Sessions

EdgeCase}

Friday, February 10, 12

```
enable :sessions

get '/' do
    session[:answer] = 42
end

get '/answer' do
    session[:answer].to_s
end

EdgeCase}
```

# Error Handling

EdgeCase}

Friday, February 10, 12

20

## raise Sinatra::NotFound

not\_found do
 "This page cannot be found."
end

EdgeCase}

#### error

```
error do
    'There was an error : ' + env['sinatra.error'].name
end
```

EdgeCase}

Friday, February 10, 12

3

#### error codes

```
error 403 do
'Access Verboten!'
end

error 400..500 do
'Uh oh!'
end
```

EdgeCase}

#### custom errors

error MyCustomError do

'There was an error : ' + env['sinatra.error'].name end

EdgeCase}

Friday, February 10, 12

33

# Modular Applications

EdgeCase}

# switching to Modular app

```
# my_app.rb
require 'sinatra/base'

class MyApp < Sinatra::Base
  # ... app code ...

# start server if executed directly
run! if app_file == $0
end</pre>
```

EdgeCase}

Friday, February 10, 12

35

# config.ru

EdgeCase}

# config.ru with classic app

# config.ru
require './app'
run Sinatra::Application

EdgeCase}

Friday, February 10, 12

37

# config.ru with modular app

# config.ru
require './my\_app'
run MyApp

EdgeCase}

## Resources

• www.sinatrarb.com/

EdgeCase}