Kemal

Into the Deep

Serdar Doğruyol
@sdogruyol

Crystal CodeCamp SF
May 2017

Serdar Doğruyol

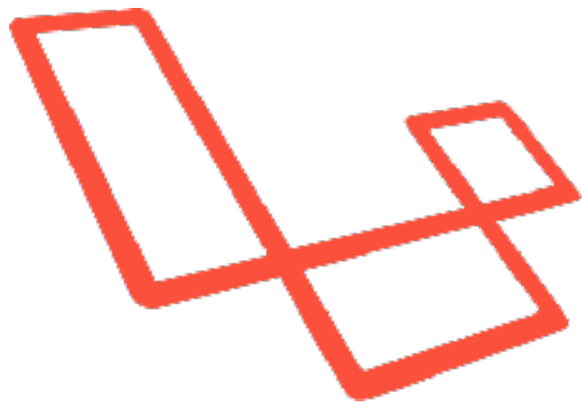
CTO - Twentify

- Rubyist, Crystal Evangelist
- Ruby Turkey, Crystal Turkey organizer
- Open Source Developer
- Polyglot(**ルビー大好き**^o^)


Twentify


bounty

Web Developers?



Flask

web development,
one drop at a time

Sinatra

Fast
Effective
Simple

ce’mal

Kemal

- HTTP verbs - RESTful
- Built-in WebSocket
- Session Engines
- Built-in Parameter Parser
- Static File Serving
- Pluggable Middlewares
- Generic View Templating

```crystal
require "kemal"

get "/" do
  "Hello from Kemal!"
end

Kemal.run
```

HTTP Verbs - REST

```
require "kemal"

get "/" do
.. show something ..
end

post "/" do
.. create something ..
end

put "/" do
.. replace something ..
end

patch "/" do
.. modify something ..
end

delete "/" do
.. annihilate something ..
end

Kemal.run
```

Form Body

```
require "kemal"

post "/users/:id" do |ctx|
  id = ctx.params.url["id"]
  name = ctx.params.body[“name”].as(String)
  age = ctx.params.body[“age”].as(Int32)
  "User #{id} Name #{name} Age #{age}"
end

Kemal.run
```

➜ ~ curl --data "name=serdar&age=27" http://localhost:3000/users/1

User 1 Name serdar Age 27

JSON Body

```
require "kemal"

post "/users/:id" do |ctx|
  id = ctx.params.url["id"]
  name = ctx.params.json[“name”].as(String)
  age = ctx.params.json[“age”].as(Int32)
  "User #{id} Name #{name} Age #{age}"
end

Kemal.run
```

➜ ~ curl -H "Content-Type: application/json" -X POST -d '{"name":"serdar","age":27}' http://localhost:3000/users/1

User 1 Name serdar Age 27

Static File Serving

```
app/
  src/
    your_app.cr
  public/
    js/
      jquery.js
      app.js
    css/
      your_app.css
    index.html
```

Serves **app/public** folder by default. JS / CSS / Image etc. It's pretty fast.

Can override like

```
require "kemal"

public_folder "assets"

Kemal.run
```

Generic View Templating

Kemal uses Kilt(ala Tilt) as the view engine.

hello.ecr

```
<h1>This is just like ERB!</h1>

<p>Let's run some Crystal Code</p>

<% some_var = "Serdar %>

<%= Process.pid %>
```

hello.slang

```
h1 This is just like SLIM!

p Let's run some Crystal code.

- some_var = "Serdar"
= Process.pid
```

You can render your view like

```
render "src/views/hello.ecr"
```

With layout

```
render "src/views/hello.ecr", "src/views/layouts/layout.ecr"
```

WebSockets

Baked-in!

```
sockets = [] of HTTP::WebSocket

ws "/" do |socket|
  sockets.push socket
  # Handle incoming message and dispatch it to all clients
  socket.on_message do |message|
    sockets.each do |a_socket|
      a_socket.send messages.to_json
    end
  end

  # Handle disconnection and clean sockets
  socket.on_close do |_|
    sockets.delete(socket)
    puts "Closing Socket: #{socket}"
  end
end
```

See it in action!

http://kemal-react-chat.herokuapp.com/

HTTP::Handler

a.k.a

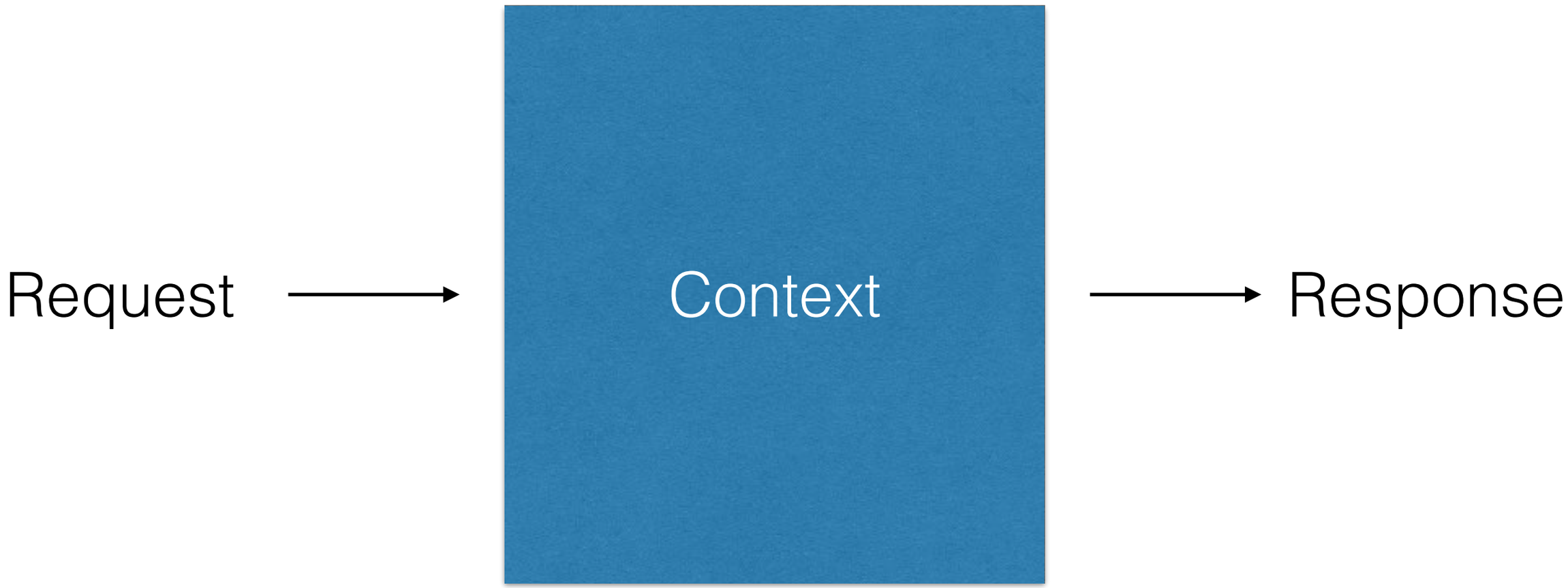
Middleware

A handler is a class which includes **HTTP::Handler** and implements the **call** method.

You can use a handler to intercept any incoming request and can modify the response.These can be used for request throttling, ip-based whitelisting, adding custom headers e.g.

```ruby
class CustomHandler
  include HTTP::Handler

  def call(context)
    puts "Doing some stuff"
    call_next(context)
  end
end
```

Handler



path, headers, params, body, status_code e.g

So how do we route?

Route

- **HTTP Verb (GET, POST, DELETE e.g)**
- **Path (/, /admin, /api/v1 e.g)**
- **Action (What to do if matched)**

```
class Kemal::Route
  getter method, path, handler

  def initialize(@method, @path : String, &handler : HTTP::Server::Context -> _)
  end
end
```

```
get "/" do
  "Hello from Kemal!"
end
```

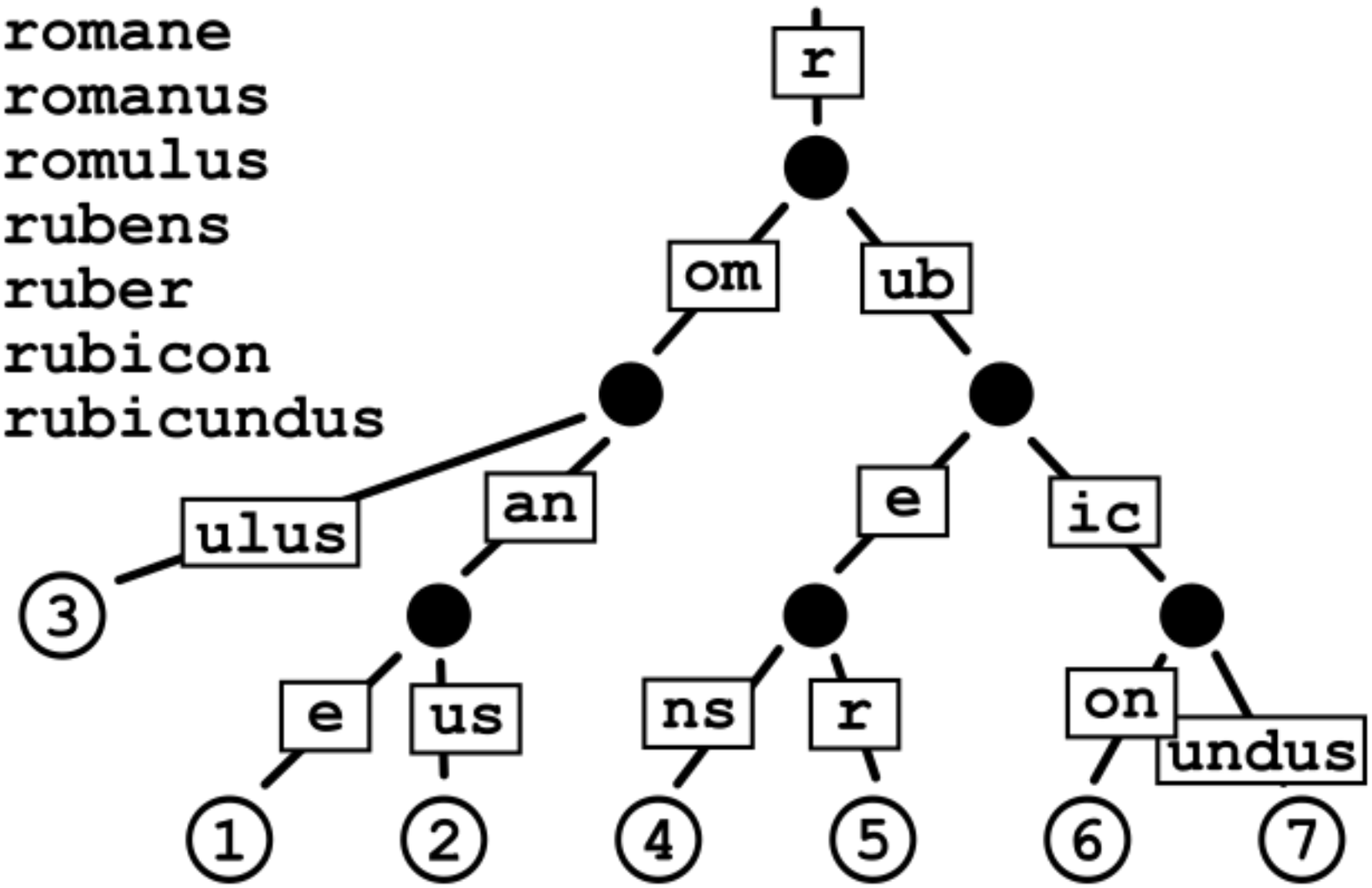
Poor Man's RouteHandler

```
class Kemal::RouteHandler
  include HTTP::Handler
  INSTANCE = new

  def initialize
    @routes = [] of Route
  end

  def call(context)
    process_request(context) if match?(request)
    call_next(context)
  end

  def match?
    # Some route matching logic
  end
end
```

Slo……w!

Radix Tree

1 romane
2 romanus
3 romulus
4 rubens
5 ruber
6 rubicon
7 rubicundus



Everything is a Handler

Request

InitHandler

LogHandler

StaticFileHandler

ExceptionHandler

Custom Handlers / Filters

Custom Handlers / Filters

Custom Handlers / Filters

RouteHandler

Response

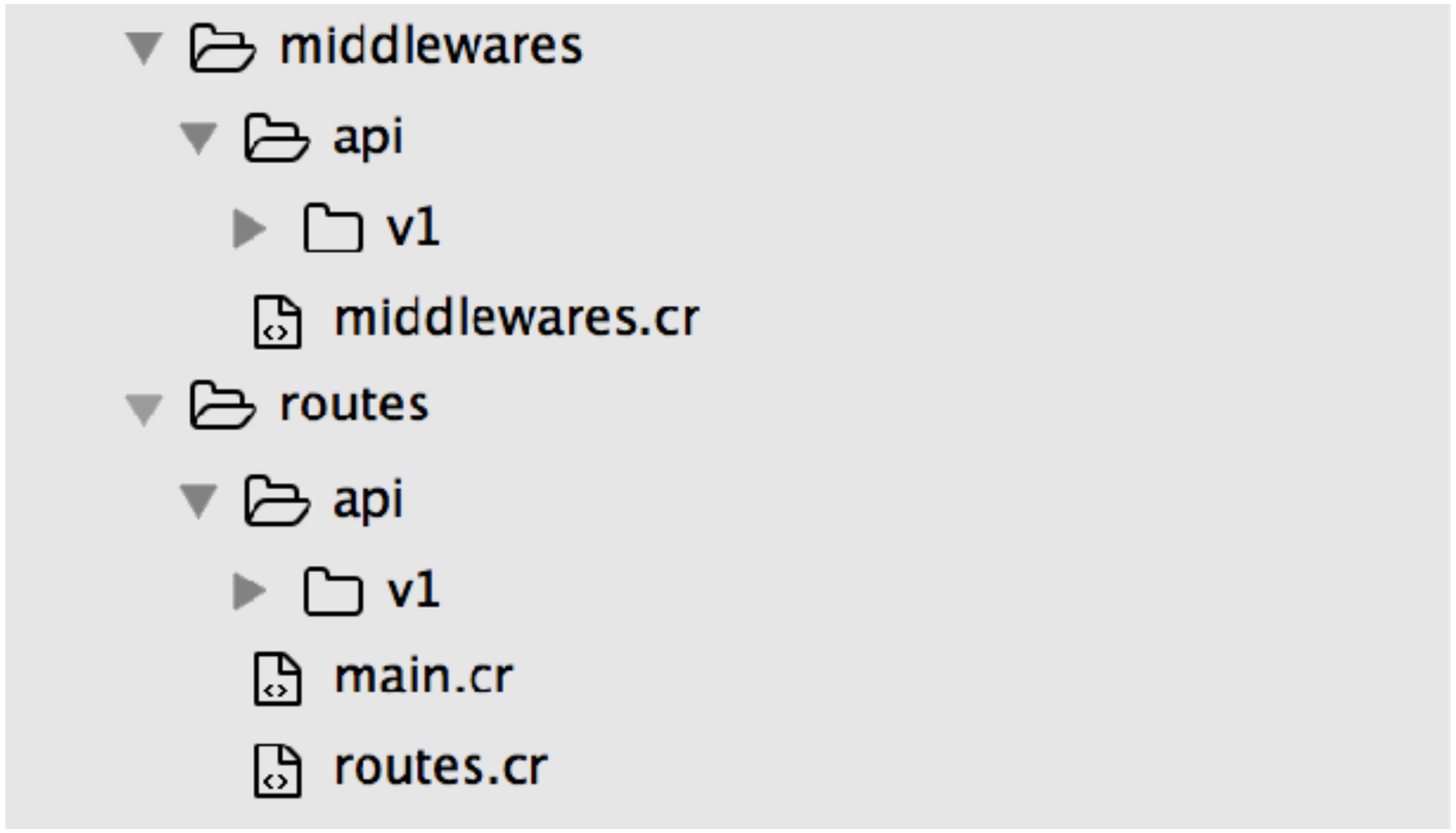
Kemal::Handler

Conditional execution

- only, only_match?
- exclude, exclude_match?

```ruby
class ApiV1OnlyHandler < Kemal::Handler
  only ["/api/v1/**"]

  def call(env)
    return call_next(env) unless only_match?(env)
    puts “Runs if namespace is /api/v1/“
  end
end
```

```
class ApiV1ExcludeHandler < Kemal::Handler
  exclude [“/api/v1/**“]

  def call(env)
    return call_next(env) if exclude_match?(env)
    puts “Runs If the path is not /api/v1/**."
  end
end
```


middlewares
api
v1
middlewares.cr
routes
api
v1
main.cr
routes.cr

How do you keep state in sync between middlewares?

Context Storage

```
add_context_storage_type(User)

class AuthenticationHandler < Kemal::Handler
  exclude ["/"]

  def call(env)
    return call_next(env) if exclude_match?(env)
    user = User.find env.params.query["user_id"].as(Int32)
    env.set("user", user)
  end

add_handler(AuthenticationHandler.new)

get "/profile" do
  user = env.get("user").as(User)
  user.to_json
end
```

Sessions

kemal-session

```ruby
Session.config do |config|
  config.cookie_name = "session_id"
  config.secret = "some_secret"
  config.gc_interval = 2.minutes # 2 minutes
end
```

Session Engines

- MemoryEngine (Default)
- FileSystemEngine
- RedisEngine

```
require "kemal"
require "kemal-session"

get "/set" do |env|
  env.session.int("number", rand(100)) # set the value of "number"
  "Random number set."
end

get "/get" do |env|
  num  = env.session.int("number") # get the value of "number"
  env.session.int?("hello") # get value or nil, like []?
  "Value of random number is #{num}."
end

Kemal.run
```

Testing

spec-kemal

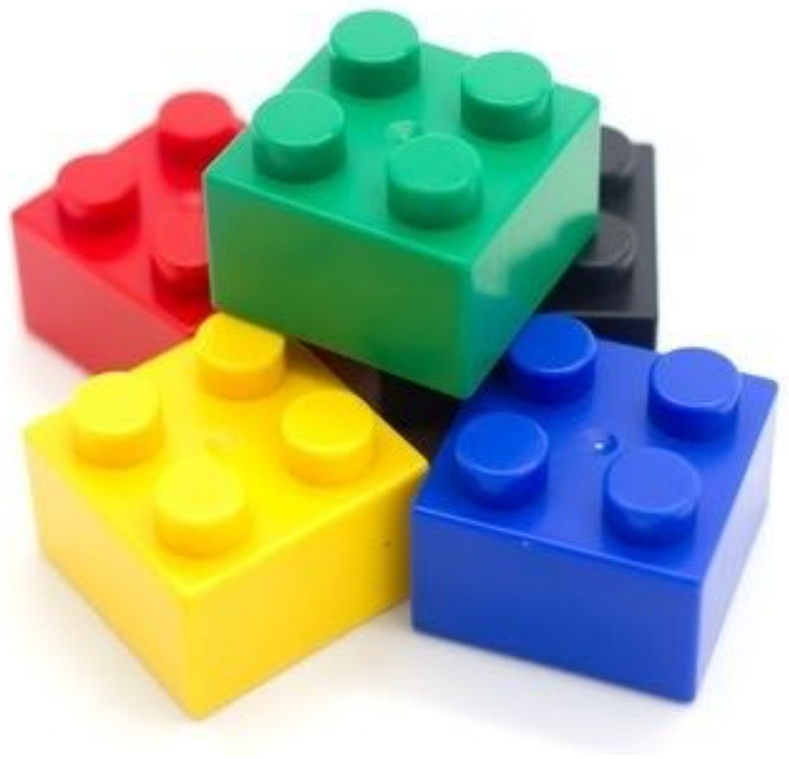
Testing

```
get "/" do
  "Hello World!"
end
```

Spec

```
# spec/your-kemal-app-spec.cr

describe "Your::Kemal::App" do

  # You can use get,post,put,patch,delete to call the corresponding route.
  it "renders /" do
    get "/"
    response.body.should eq "Hello World!"
  end

end
```

Unopinionated



Your Own Stack

- Kemal
- crystal-db
- MySQL
- Apache
- npm

ActiveRecord

#TREN

https://github.com/sdogruyol/tren

```
-- name: get_users(name : String, surname : String)

SELECT * FROM users WHERE name = '{{ name }}' AND surname =
'{{ surname }}'
```

```
require "tren"

Tren.load(“/path/to/your/file.sql")

query = get_users(“Serdar”, “Dogruyol”)
```

- Kemal
- crecto
- PostgreSQL
- Nginx
- Yarn

Deployment

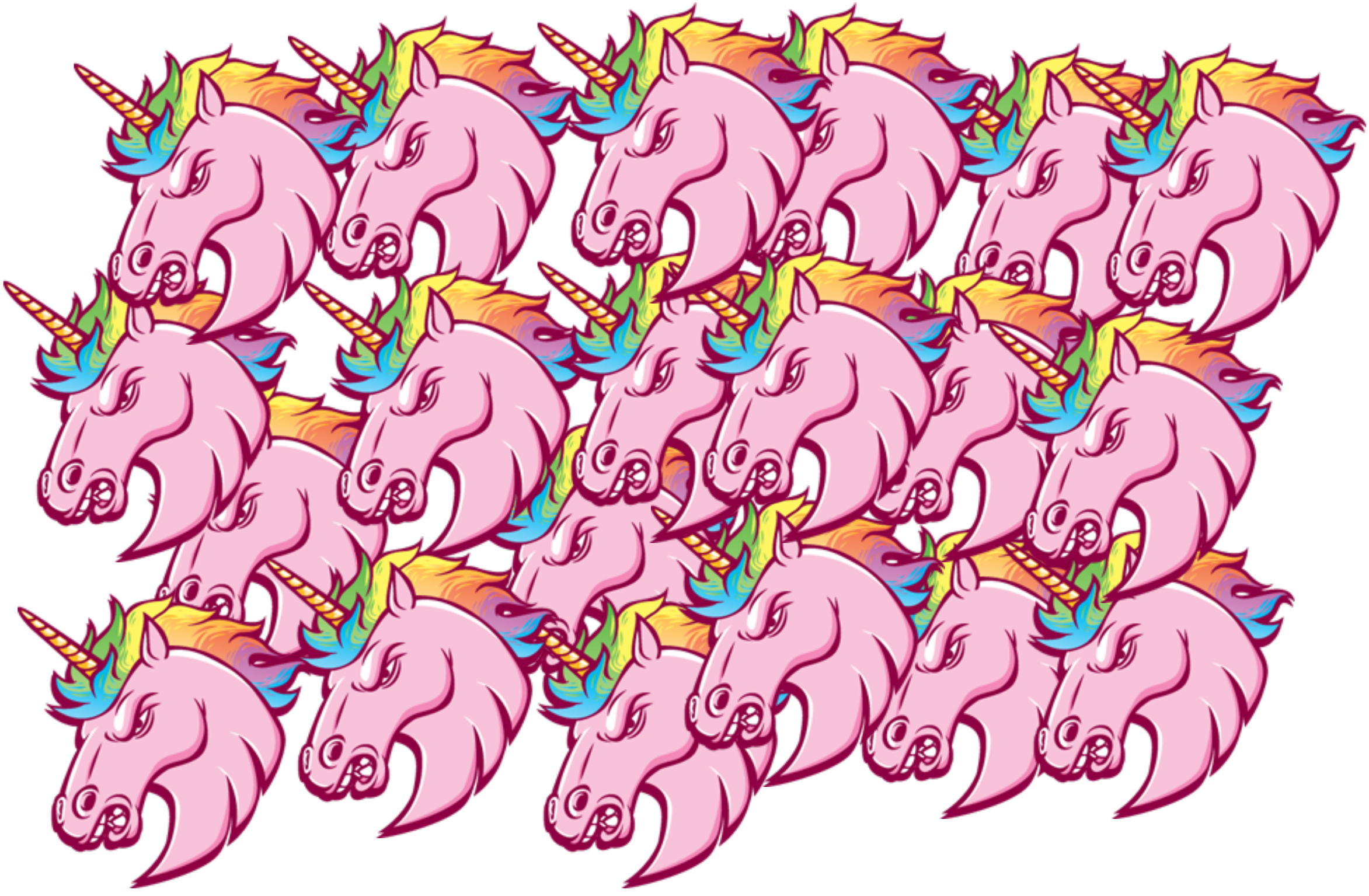
- Cross compilation is still not there yet.
- Better to have Crystal installed in the target machine.
  - capistrano-kemal: Capistrano 3 plugin for Kemal
    - cap production | staging deploy

Let’s talk about a

Real World Application

Rails is the elephant.

How many Unicorns does it take to carry an elephant?



ETOOMANYUNICORNS

```
5845 ubuntu      20   0  546M  210M  6104 S  0.0  2.8  0:04.46 unicorn worker[20]
5893 ubuntu      20   0  390M  118M  7828 S  0.0  1.6  0:00.00 unicorn master -D -
5279 ubuntu      20   0  390M  118M  7828 S  0.0  1.6  0:03.30 unicorn master -D -
5763 ubuntu      20   0  503M  168M  6068 S  0.0  2.3  0:00.00 unicorn worker[0] -
5767 ubuntu      20   0  488M  153M  6072 S  0.0  2.1  0:00.00 unicorn worker[1] -
5771 ubuntu      20   0  502M  166M  6084 S  0.0  2.2  0:00.00 unicorn worker[2] -
5775 ubuntu      20   0  510M  175M  6148 S  0.0  2.3  0:00.00 unicorn worker[3] -
5781 ubuntu      20   0  501M  166M  6144 S  0.0  2.2  0:00.00 unicorn worker[4] -
5785 ubuntu      20   0  525M  189M  6076 S  0.0  2.5  0:00.00 unicorn worker[5] -
5779 ubuntu      20   0  525M  189M  6076 S  0.0  2.5  0:02.41 unicorn worker[5] -
5787 ubuntu      20   0  493M  158M  6016 S  0.0  2.1  0:00.00 unicorn worker[6] -
5793 ubuntu      20   0  505M  171M  6104 S  0.0  2.3  0:00.00 unicorn worker[7] -
5795 ubuntu      20   0  504M  169M  6024 S  0.0  2.3  0:00.00 unicorn worker[8] -
5799 ubuntu      20   0  546M  211M  6140 S  0.0  2.8  0:00.00 unicorn worker[9] -
5803 ubuntu      20   0  502M  167M  6124 S  0.0  2.2  0:00.00 unicorn worker[10]
5801 ubuntu      20   0  502M  167M  6124 S  0.0  2.2  0:02.74 unicorn worker[10]
5813 ubuntu      20   0  505M  170M  6104 S  0.0  2.3  0:00.00 unicorn worker[11]
5817 ubuntu      20   0  496M  162M  6088 S  0.0  2.2  0:00.00 unicorn worker[12]
5821 ubuntu      20   0  506M  171M  6136 S  0.0  2.3  0:00.00 unicorn worker[13]
5827 ubuntu      20   0  502M  167M  6136 S  0.0  2.2  0:00.00 unicorn worker[14]
5823 ubuntu      20   0  502M  167M  6136 S  0.0  2.2  0:02.89 unicorn worker[14]
5831 ubuntu      20   0  501M  167M  6132 S  0.0  2.2  0:00.00 unicorn worker[15]
5825 ubuntu      20   0  501M  167M  6132 S  0.0  2.2  0:02.99 unicorn worker[15]
5835 ubuntu      20   0  502M  167M  6096 S  0.0  2.2  0:00.00 unicorn worker[16]
5839 ubuntu      20   0  501M  166M  6200 S  0.0  2.2  0:00.00 unicorn worker[17]
5843 ubuntu      20   0  505M  169M  6088 S  0.0  2.3  0:00.00 unicorn worker[18]
5837 ubuntu      20   0  505M  169M  6088 S  0.0  2.3  0:02.47 unicorn worker[18]
5847 ubuntu      20   0  503M  167M  6052 S  0.0  2.2  0:00.00 unicorn worker[19]
5851 ubuntu      20   0  546M  210M  6104 S  0.0  2.8  0:00.00 unicorn worker[20]
5857 ubuntu      20   0  502M  167M  6108 S  0.0  2.2  0:00.00 unicorn worker[21]
5855 ubuntu      20   0  499M  165M  6028 S  0.0  2.2  0:00.00 unicorn worker[22]
5853 ubuntu      20   0  499M  165M  6028 S  0.0  2.2  0:02.50 unicorn worker[22]
5861 ubuntu      20   0  493M  158M  6040 S  0.0  2.1  0:00.00 unicorn worker[23]
5865 ubuntu      20   0  502M  166M  6128 S  0.0  2.2  0:00.00 unicorn worker[24]
5869 ubuntu      20   0  500M  165M  6136 S  0.0  2.2  0:00.00 unicorn worker[25]
5873 ubuntu      20   0  511M  175M  6100 S  0.0  2.3  0:00.00 unicorn worker[26]
```

~150 MB per worker
32 workers
~5GB

unicorn-worker-killer

Kemal to the rescue!

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
13485	ubuntu	20	0	146M	22496	3256	S	0.0	0.3	0:27.82	./app

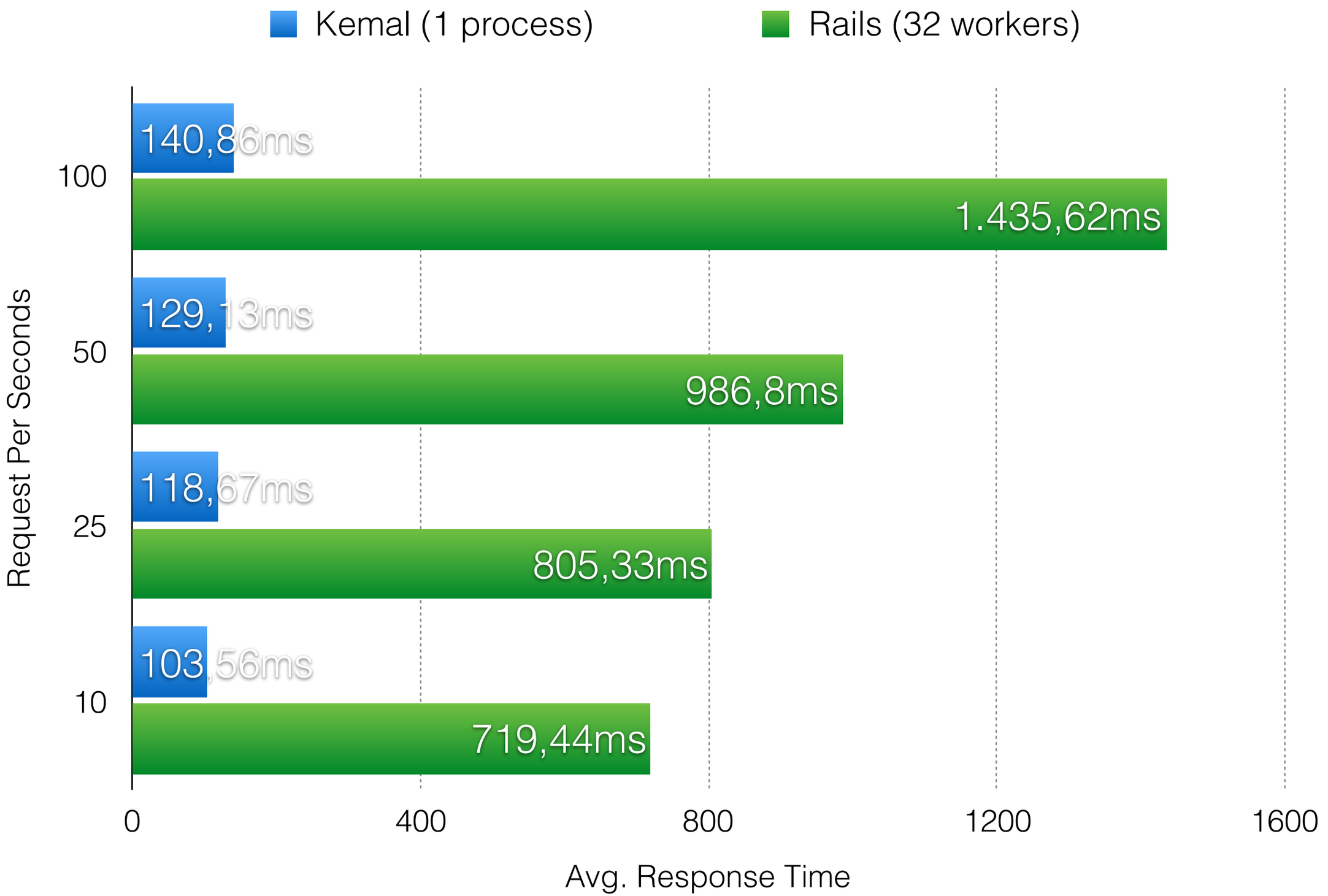
~25 MB per process
1 process
~25 MB

VS

~150 MB per worker
32 workers
~5GB

~25 MB per process
1 process
~25 MB

~200x less memory
Faster response times


Kemal (1 process)
Rails (32 workers)
Request Per Seconds
100
50
25
10
140,86ms
1.435,62ms
129,13ms
986,8ms
118,67ms
805,33ms
103,56ms
719,44ms
0
400
800
1200
1600
Avg. Response Time

Future

Unified Params

```ruby
post "/" do |env|
  env.params["name"]
  env.params["json_object"]
  env.params["query_param"]
  env.params["url_param"]
end
```

Caching

Parallelism

```
Kemal.run { |cfg| cfg.server.bind(reuse_port: true) }
```

Tutorials, Screencasts, Docs, Book?

Improve Benchmarks

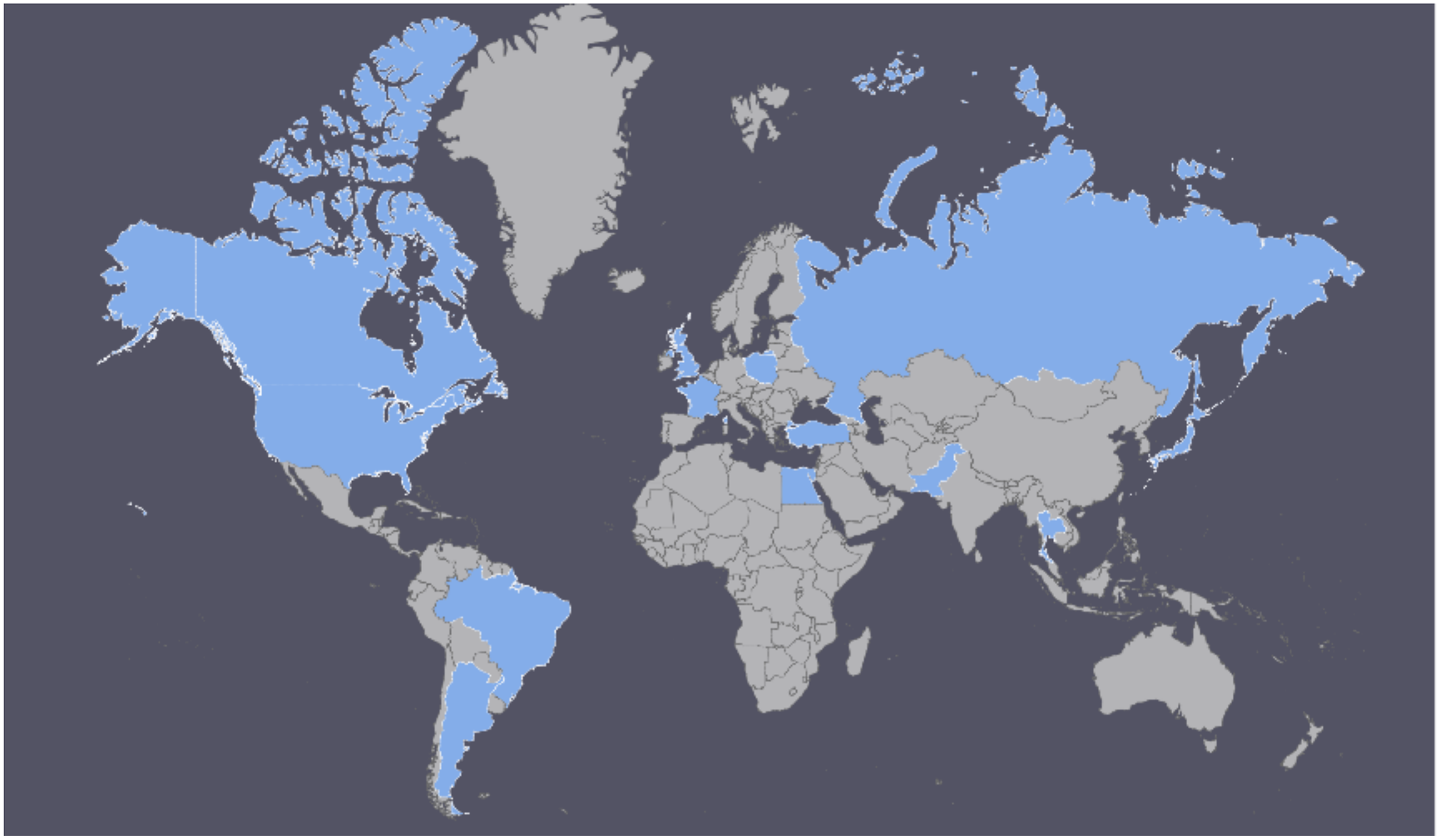
flask-pypy	19,582	3.3%					
revel	18,930	3.4%	Ful	Go	Non	Non	Lin	Rea
silex	18,096	3.2%	Mcr	PHP	Non	ngx	Lin	Rea
silex-orm	16,341	2.9%	Mcr	PHP	Non	ngx	Lin	Rea
kemal (postgresql)	12,845	2.3%	Ful	Cry	Non	Non	Lin	Rea
web-simple	12,645	2.3%	Mcr	Prl	Plk	Sta	Lin	Rea
kelp	12,265	2.2%	Ful	Prl	Plk	Sta	Lin	Rea
kelp	11,905	2.1%	Ful	Prl	Plk	Sta	Lin	Rea
codeigniter	11,824	2.1%	Ful	PHP	Non	ngx	Lin	Rea

1.0

Kemal in 2017

- 500+ commit
- 1500+ stars on Github
- 41 contributors

Contributors



Github WAF Showcase

kemalcr / kemal

Fast, Effective, Simple web framework for Crystal

Crystal ★ 1,483 87 Updated 9 days ago Starred by 30 people you know

web2py / web2py

Free and open source full-stack enterprise framework for agile development of secure database-driven web-based applications, written and programmable in Python.

Python ★ 1,355 675 Updated 2 days ago

pakyow / pakyow

An open-source platform for the modern web

Ruby ★ 745 63 Updated 6 days ago Starred by 4 people you know

frappe / frappe

Full Stack Web Framework in Python & JS. Used to build ERPNext

Python ★ 572 435 Updated 4 hours ago Starred by 1 person you know

Patreon

https://www.patreon.com/sdogruyol

- RainforestQA
- Protel
- Twentify
- Bulutfon
- Duo Design
- LI-COR Biosciences
- Westfield
- Nikola Motor Company

Be sure to add yourself Kemal Wiki

crystalweekly.com

crystalforrubyists.com

Thanks!

Serdar Doğruyol
@sdogruyol

Questions?