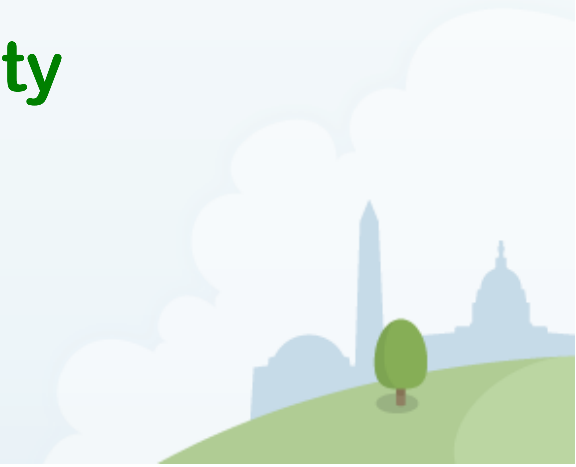
RESTful API On Grape

andy@intridea.com
2013/07/27 @XianOpenParty



Introduction

王要弟 == Andy Wang

WangYaodi.com

Senior Ruby/Rails Enginner @intridea

Weibo/Twitter/Github/RubyChina: @yorzi

What’s Grape?

Generalized
Rapid
APIs
Erector

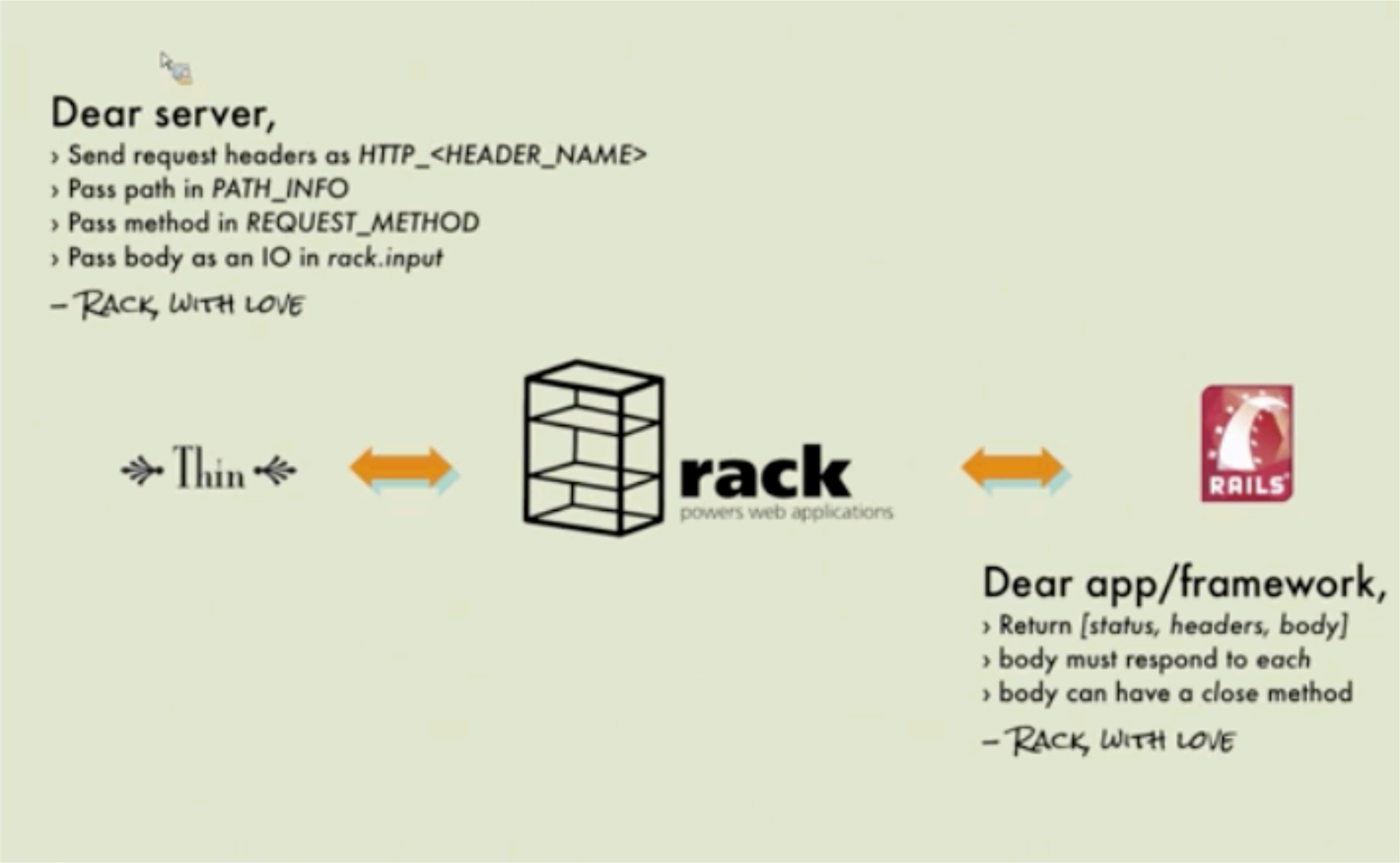
https://github.com/intridea/grape/blob/master/README.md

“An opinionated micro-framework for creating RESTful APIs in Ruby”

REpresentational
State
Transfer

http://www.ruanyifeng.com/blog/2011/09/restful.html

Grape APIs are Rack applications



Dear server,

- › Send request headers as *HTTP_<HEADER_NAME>*
- › Pass path in *PATH_INFO*
- › Pass method in *REQUEST_METHOD*
- › Pass body as an IO in *rack.input*

– RACK, WITH LOVE

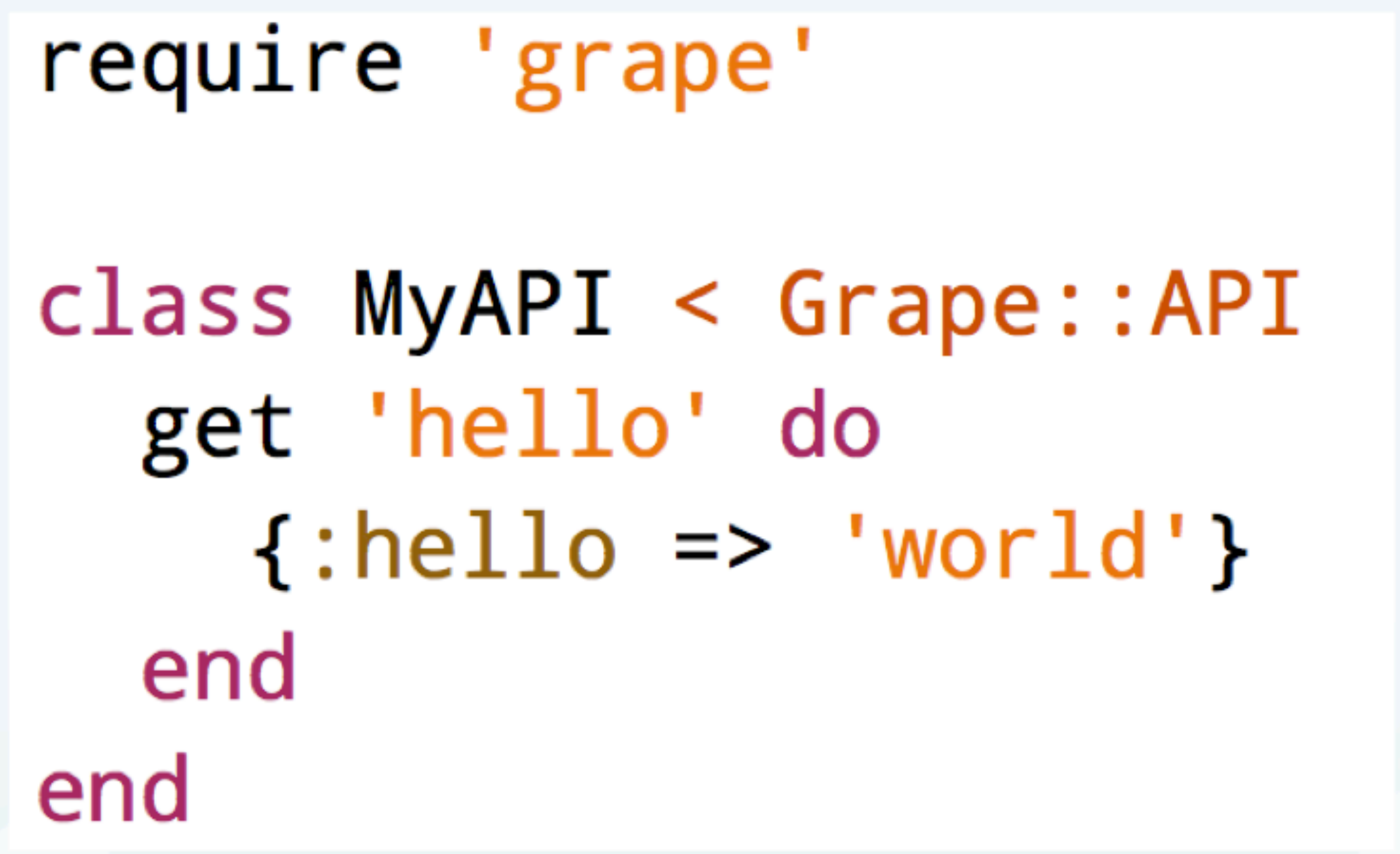
Dear app/framework,

- › Return *[status, headers, body]*
- › body must respond to each
- › body can have a close method

– RACK, WITH LOVE

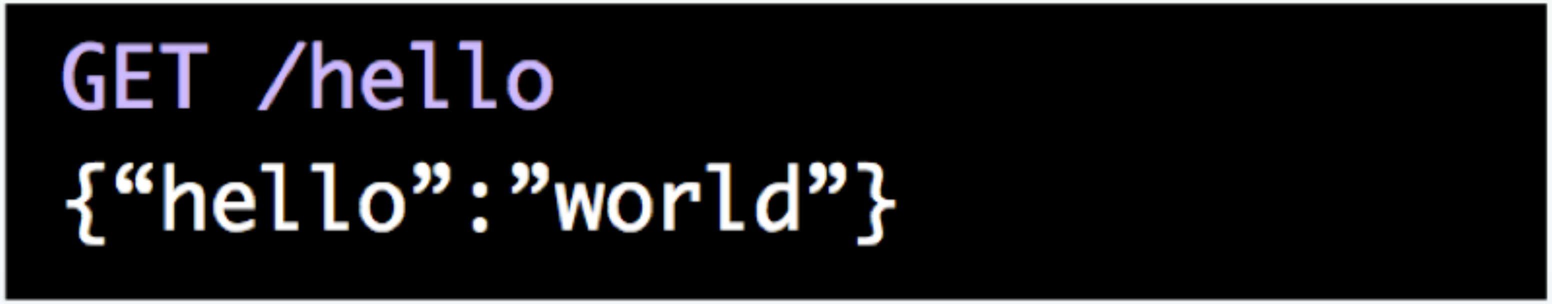
http://guides.rubyonrails.org/rails_on_rack.html

Basic API



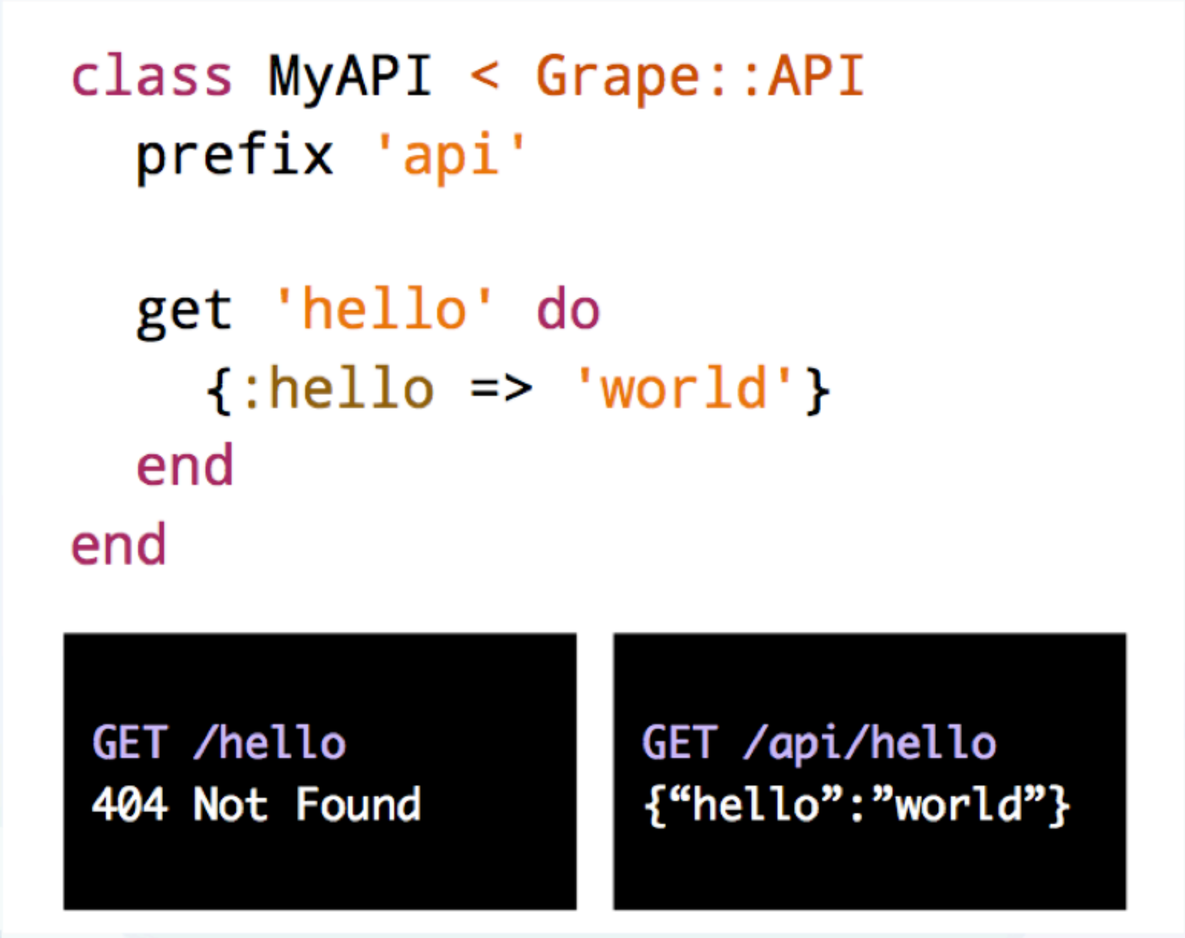
```ruby
require 'grape'

class MyAPI < Grape::API
  get 'hello' do
    {:hello => 'world'}
  end
end
```



```
GET /hello
{“hello”:”world”}
```

Add Prefix



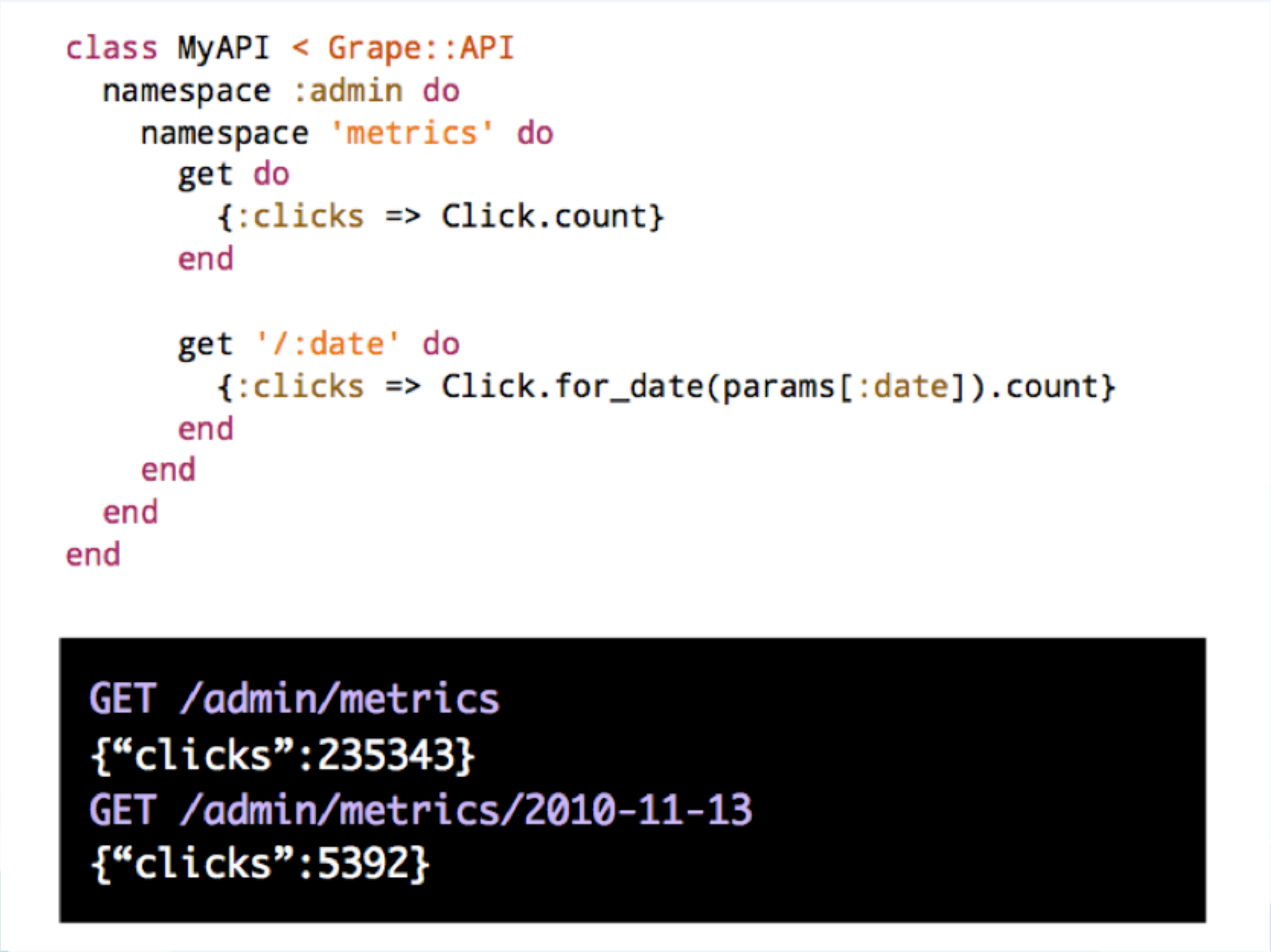
```
class MyAPI < Grape::API
  prefix 'api'

  get 'hello' do
    {:hello => 'world'}
  end
end
```

```
GET /hello
404 Not Found
```

```
GET /api/hello
{“hello”:”world”}
```

Add Namespace

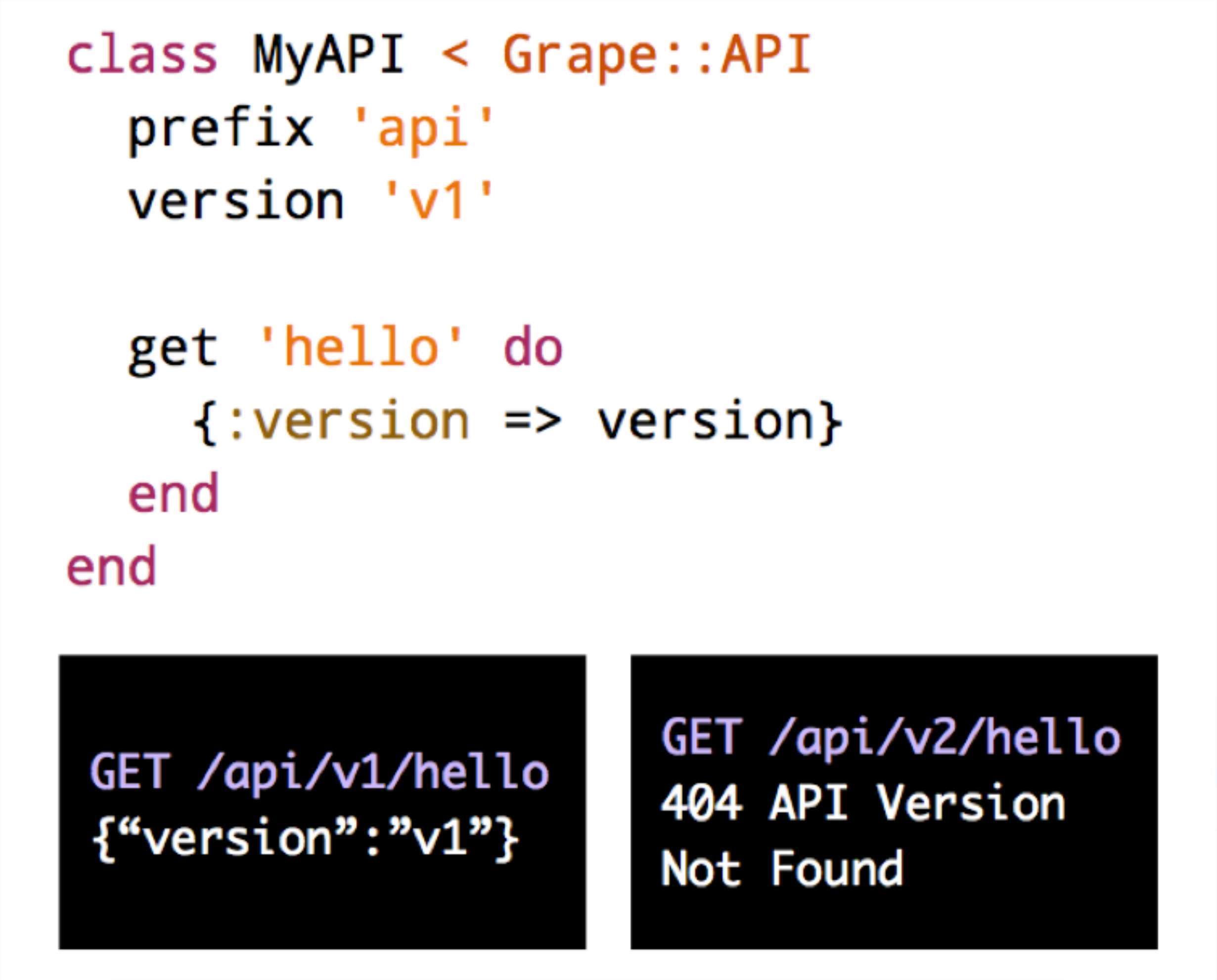


```ruby
class MyAPI < Grape::API
  namespace :admin do
    namespace 'metrics' do
      get do
        {:clicks => Click.count}
      end

      get '/:date' do
        {:clicks => Click.for_date(params[:date]).count}
      end
    end
  end
end
```

```
GET /admin/metrics
{“clicks”:235343}
GET /admin/metrics/2010-11-13
{“clicks”:5392}
```

Add Version



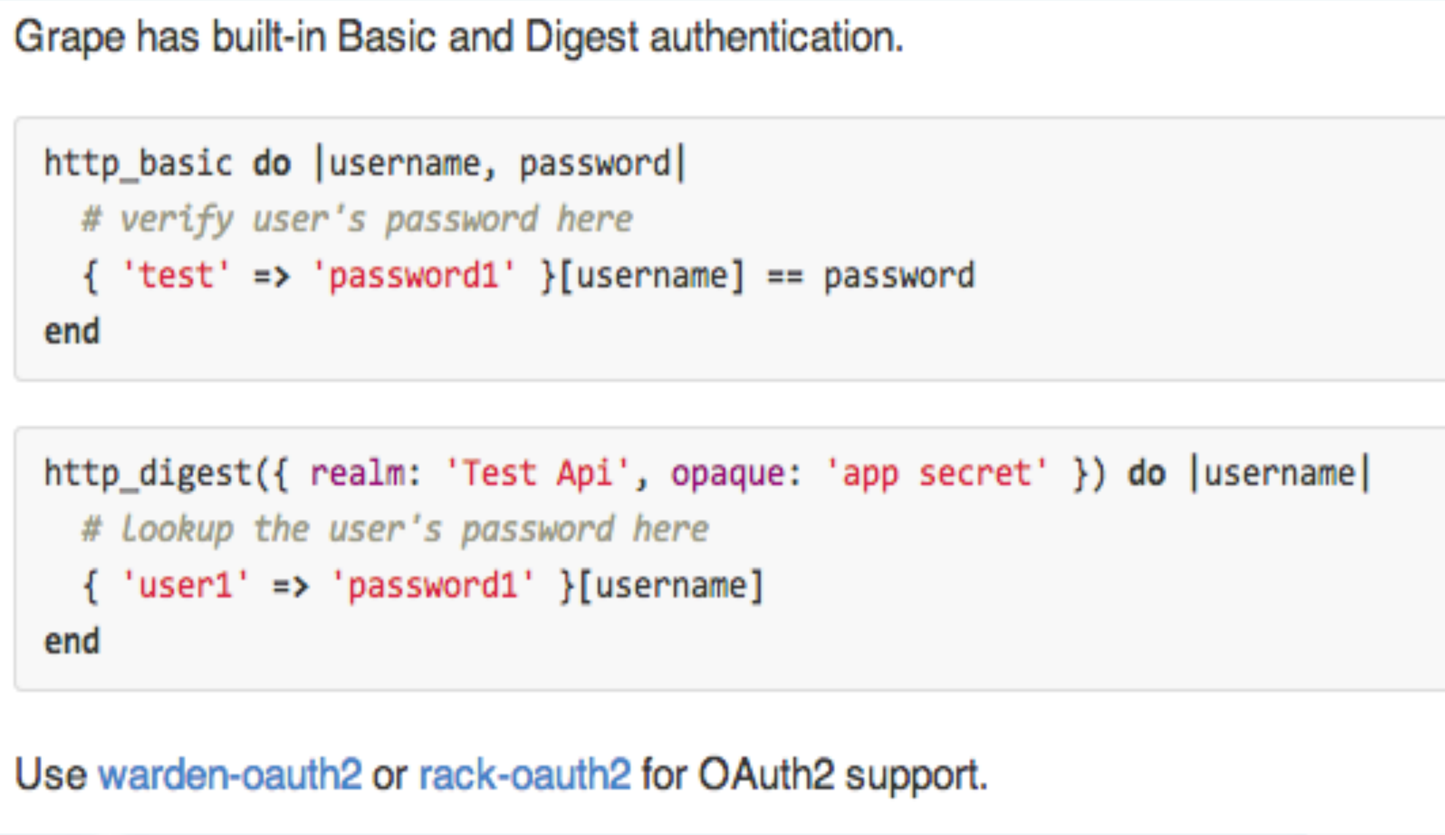
```ruby
class MyAPI < Grape::API
  prefix 'api'
  version 'v1'

  get 'hello' do
    {:version => version}
  end
end
```

```
GET /api/v1/hello
{“version”:”v1”}
```

```
GET /api/v2/hello
404 API Version
Not Found
```

Authentication



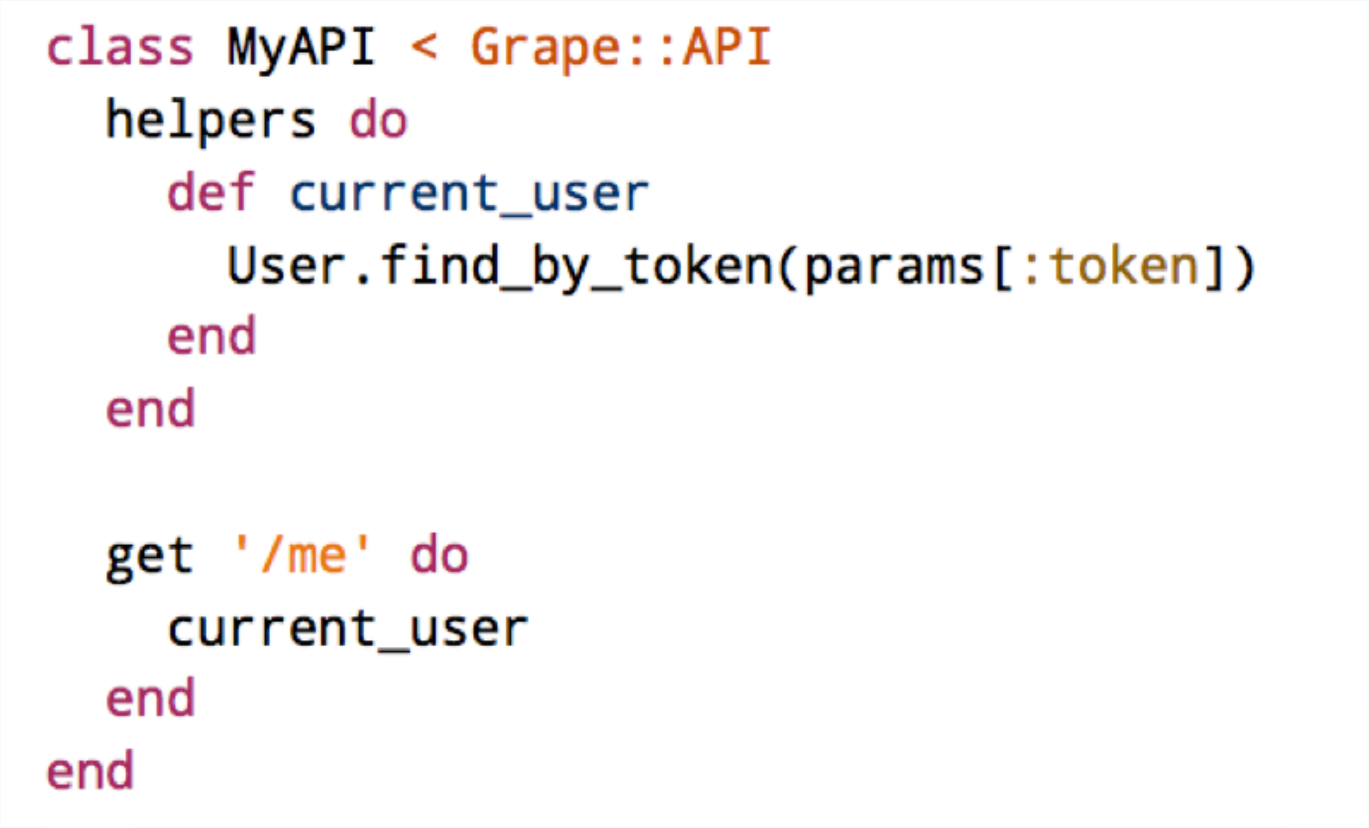
Grape has built-in Basic and Digest authentication.

```
http_basic do |username, password|
  # verify user's password here
  { 'test' => 'password1' }[username] == password
end
```

```
http_digest({ realm: 'Test Api', opaque: 'app secret' }) do |username|
  # lookup the user's password here
  { 'user1' => 'password1' }[username]
end
```

Use warden-oauth2 or rack-oauth2 for OAuth2 support.

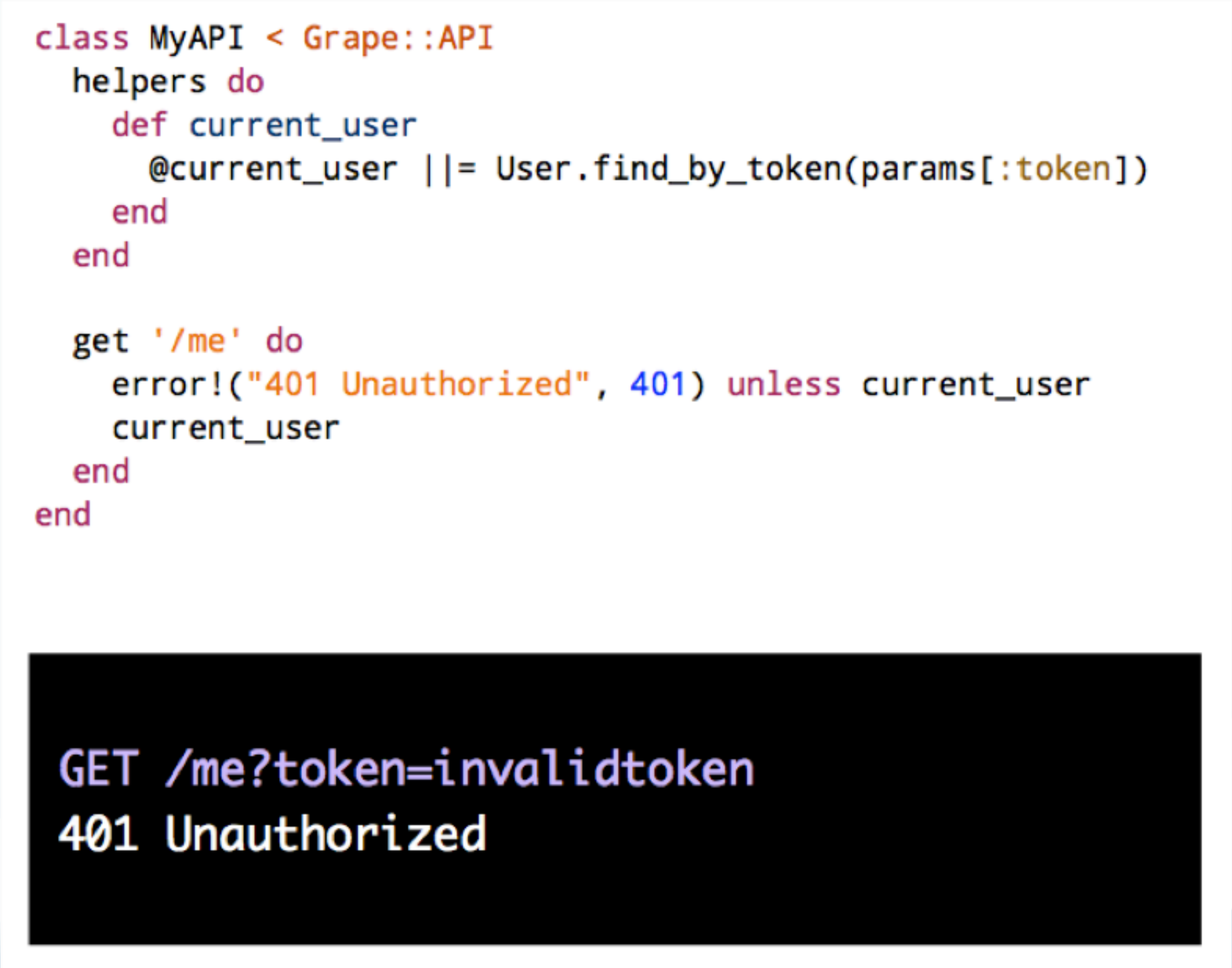
Use Helpers



```
class MyAPI < Grape::API
  helpers do
    def current_user
      User.find_by_token(params[:token])
    end
  end

  get '/me' do
    current_user
  end
end
```

Raise Error



```ruby
class MyAPI < Grape::API
  helpers do
    def current_user
      @current_user ||= User.find_by_token(params[:token])
    end
  end

  get '/me' do
    error!("401 Unauthorized", 401) unless current_user
    current_user
  end
end
```

```
GET /me?token=invalidtoken
401 Unauthorized
```

Exception Handling

JSONP & CORS

Rabl Template

Cookies

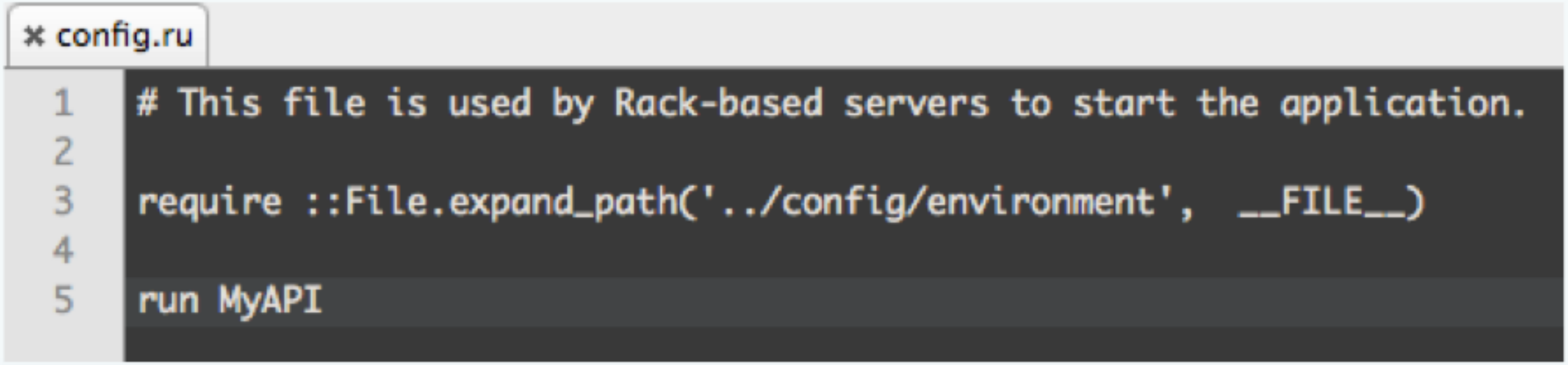
Redirecting

Logging

More Features

https://github.com/intridea/grape/blob/master/README.md

Run Rack App

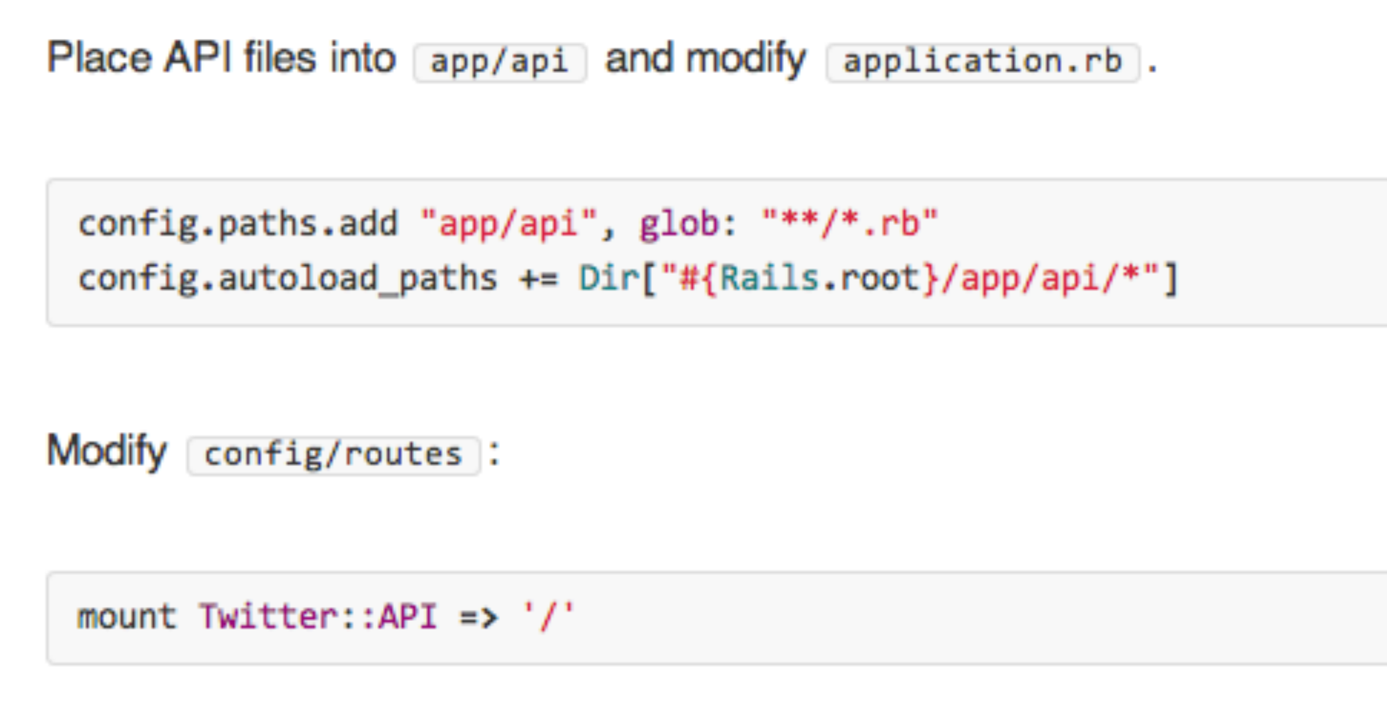


```ruby
# This file is used by Rack-based servers to start the application.

require ::File.expand_path('../config/environment',  __FILE__)

run MyAPI
```

Work with Rails



Place API files into `app/api` and modify `application.rb`.

```
config.paths.add "app/api", glob: "**/*.rb"
config.autoload_paths += Dir["#{Rails.root}/app/api/*"]
```

Modify `config/routes`:

```
mount Twitter::API => '/'
```

Resources

https://github.com/intridea/grape/wiki

Question?

Thank You!