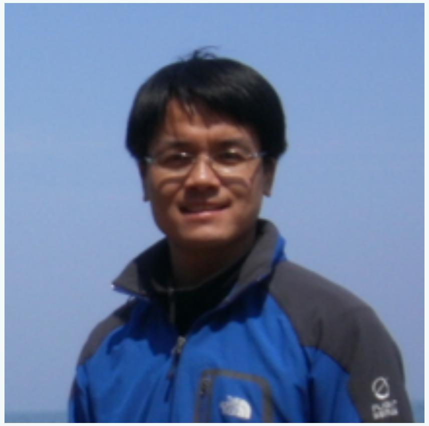
Rapid Ruby API On Grape

wangyaodi@gmail.com

2011/07/24

About Me



王要弟
Andy Wang

◆ Symbol - yorzi
Details - about.me/wangyaodi
Senior Engineer at Intridea Inc.

More Links:

- ❖ http://github.com/yorzi
- ❖ http://twitter.com/yorzi
- ❖ http://more-sites/yorzi

What’s Grape?

Michael Bleigh

@mbleigh

https://github.com/mbleigh

https://github.com/intridea/omniauth

https://github.com/mbleigh/acts-as-taggable-on

Ok, BUT What's Grape?

"An opinionated micro-framework for creating REST-like APIs in Ruby"

https://github.com/intridea/grape

http://www.infoq.com/articles/rest-introduction

Ruby makes hard things easy

But APIs still aren’t easy

APIs in Rails are too entangled

APIs in Sinatra are too manual

Grape

Generalized
Rapid
API
Erector

How To Use Grape?

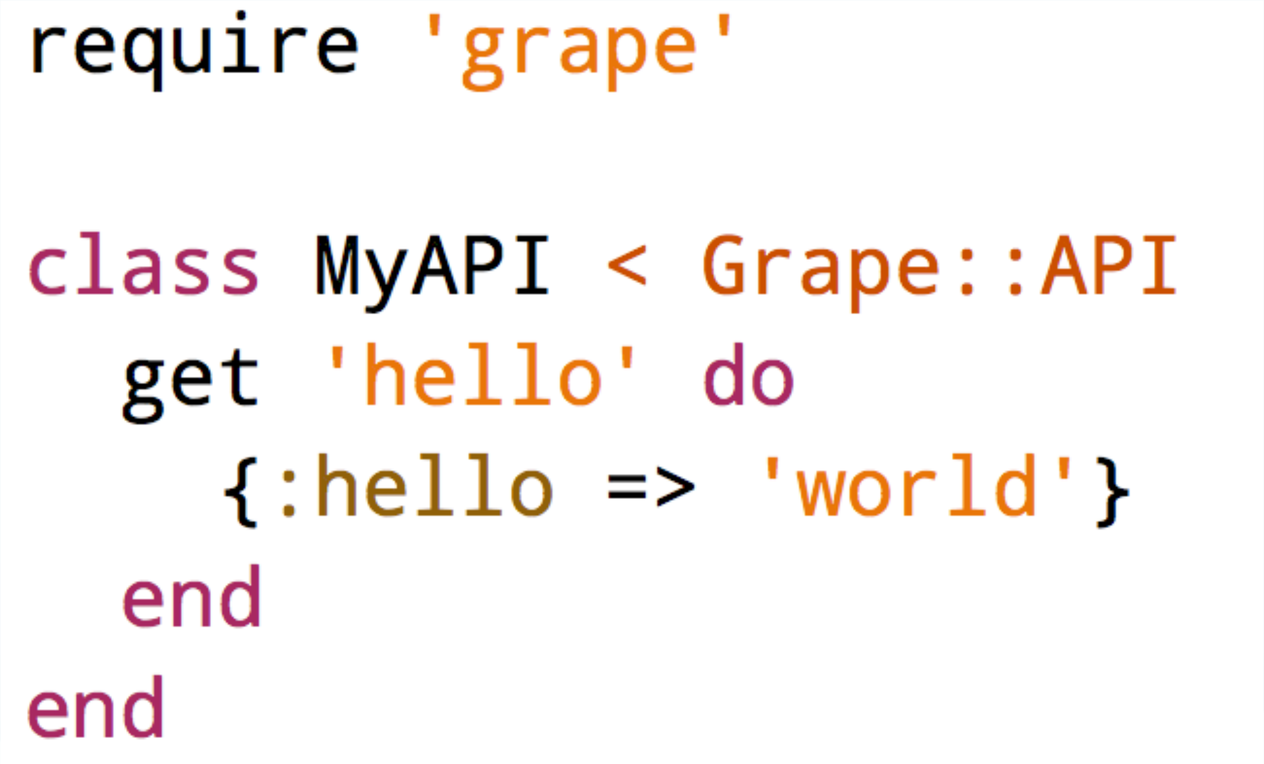
```
gem install grape
```

github.com/intridea/grape

Remember: Grape APIs are Rack applications

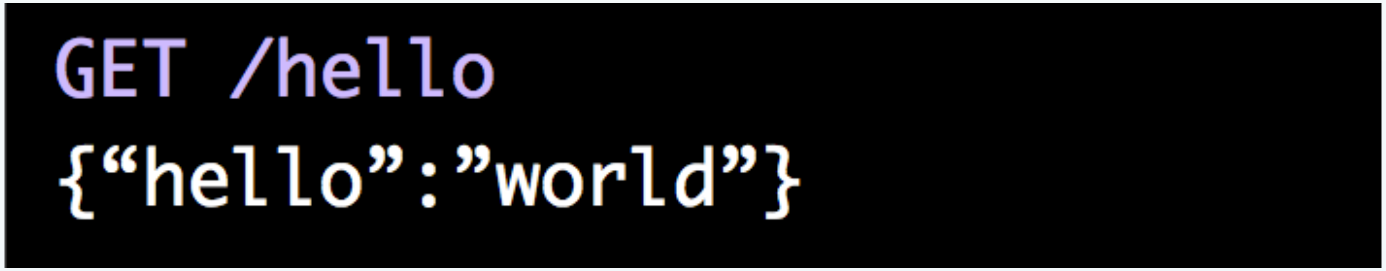
http://ruby.about.com/od/rack/a/What-Is-Rack.htm

Basic API



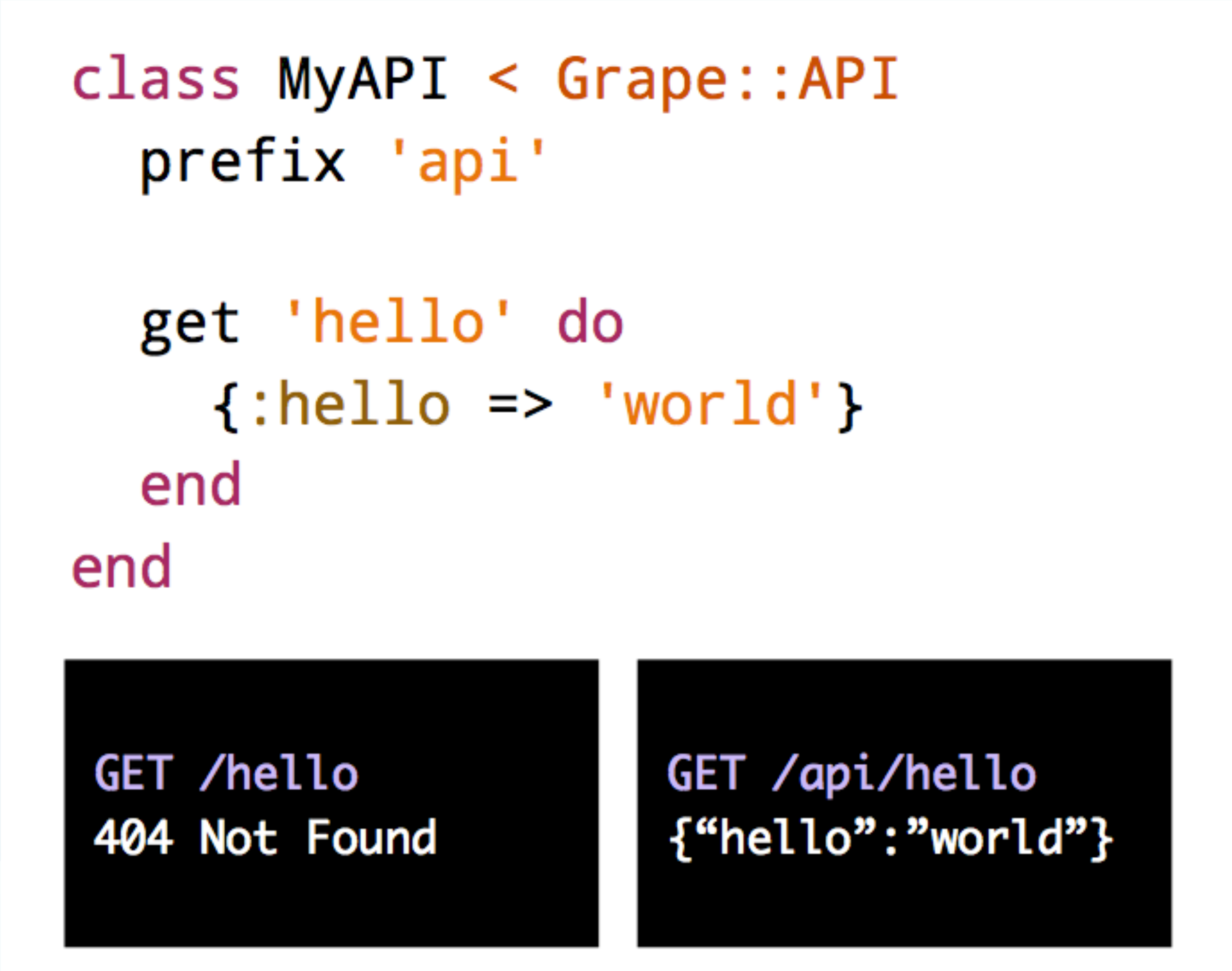
```ruby
require 'grape'

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

Result, JSON Serialization



```
GET /hello
{“hello”:”world”}
```

Add Prefix

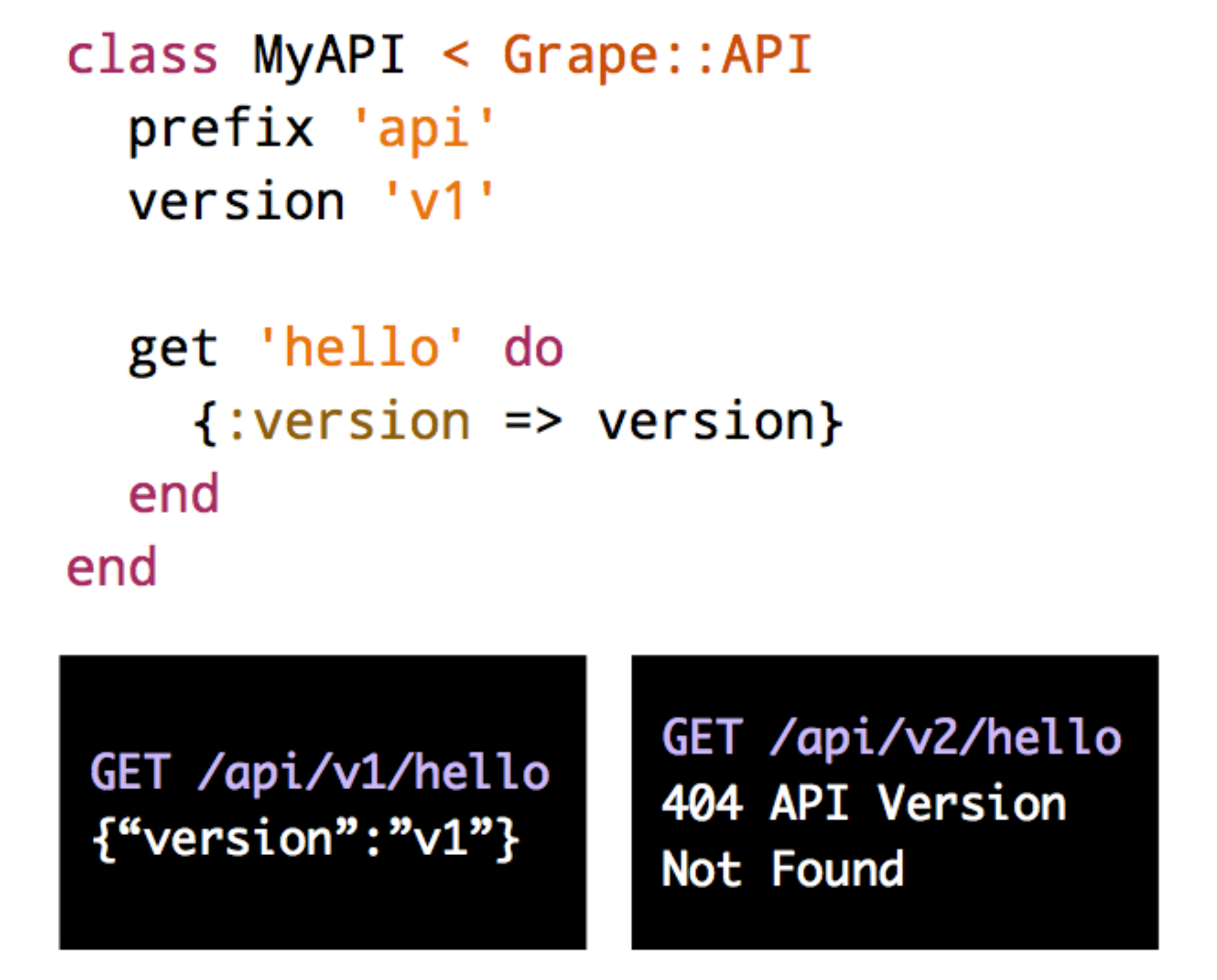


```ruby
class MyAPI < Grape::API
  prefix 'api'

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

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

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

Add Version

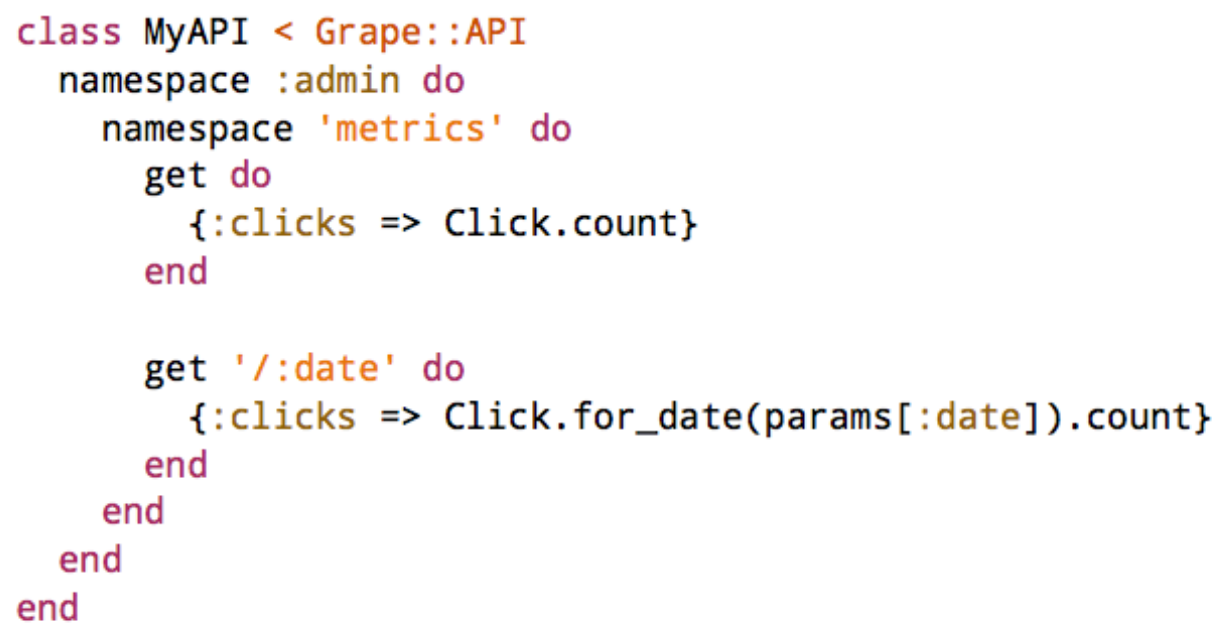


```
class MyAPI < Grape::API
  prefix 'api'
  version 'v1'

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

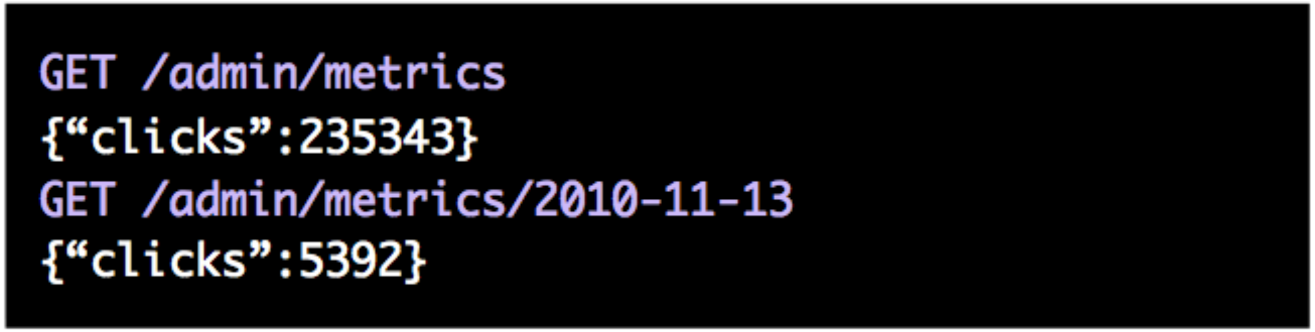
```
GET /api/v1/hello
{"version":"v1"}
```

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

Add Namespace

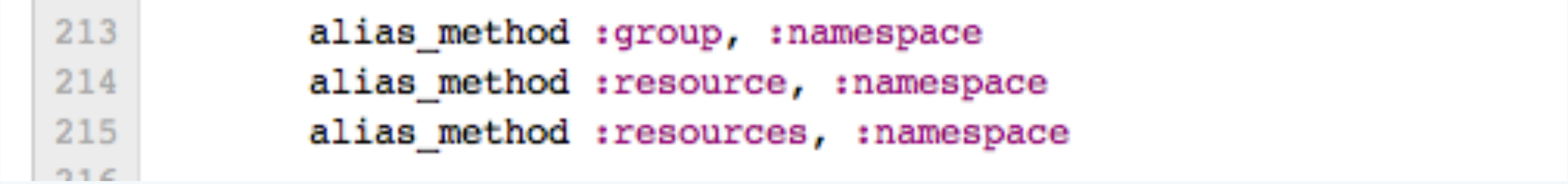


```ruby
class MyAPI < Grape::API
  namespace :admin do
    namespace 'metrics' do
      get do
        {:clicks => Click.count}
      end

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



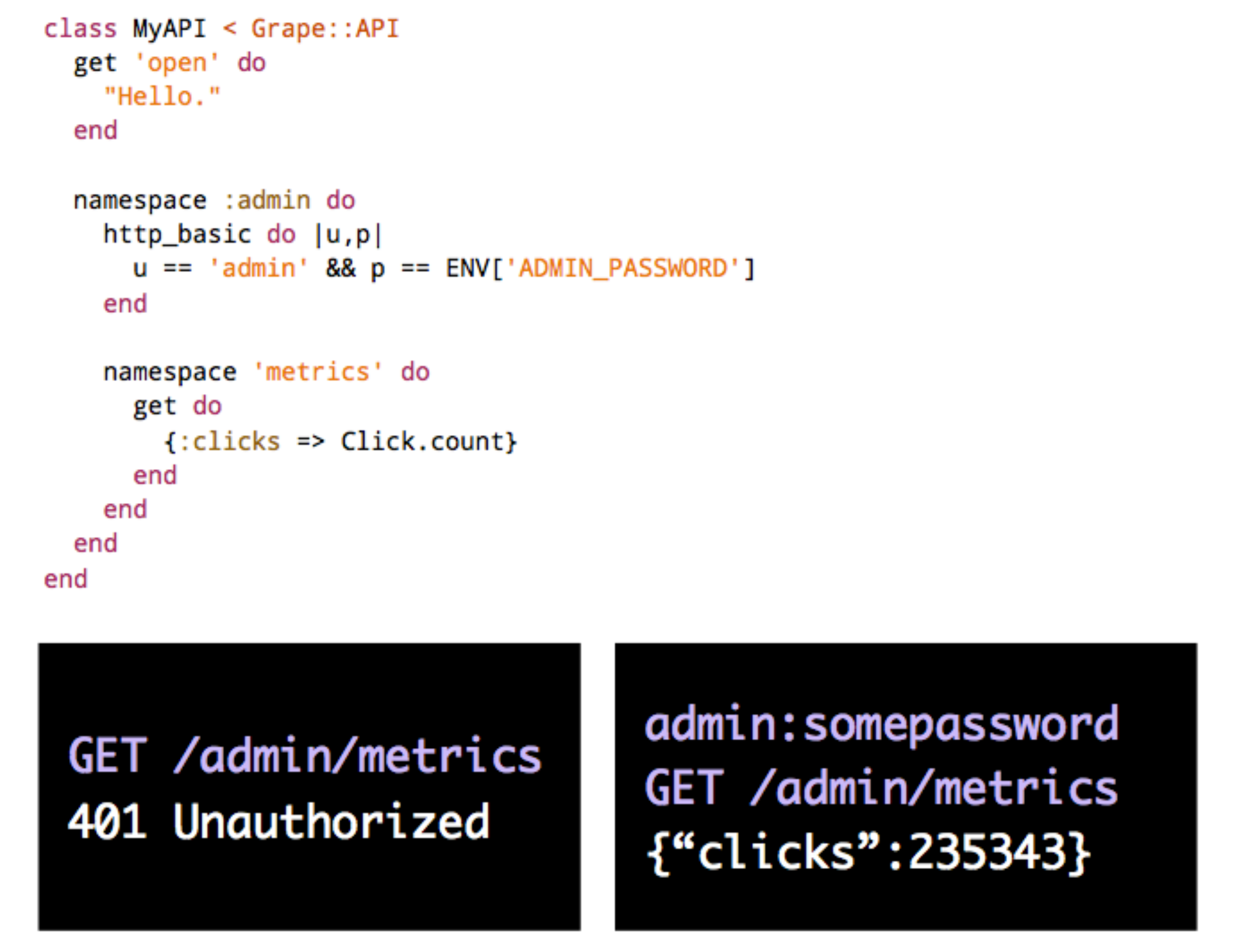
```
GET /admin/metrics
{“clicks”:235343}
GET /admin/metrics/2010-11-13
{“clicks”:5392}
```

Other Namespacing Key words



```
        alias_method :group, :namespace
        alias_method :resource, :namespace
        alias_method :resources, :namespace
```

Add Basic Auth

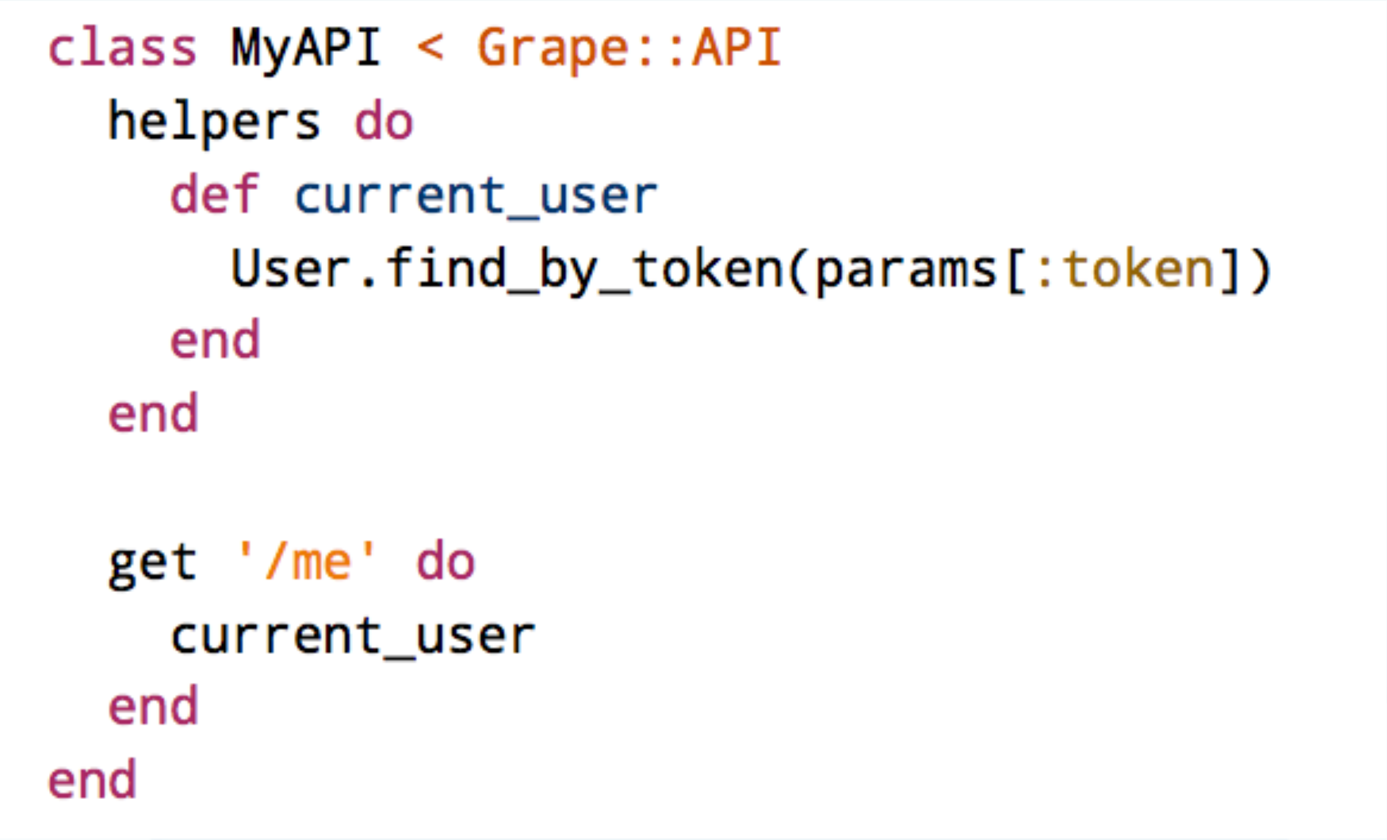


```ruby
class MyAPI < Grape::API
  get 'open' do
    "Hello."
  end

  namespace :admin do
    http_basic do |u,p|
      u == 'admin' && p == ENV['ADMIN_PASSWORD']
    end

    namespace 'metrics' do
      get do
        {:clicks => Click.count}
      end
    end
  end
end
```

```
GET /admin/metrics
401 Unauthorized
```

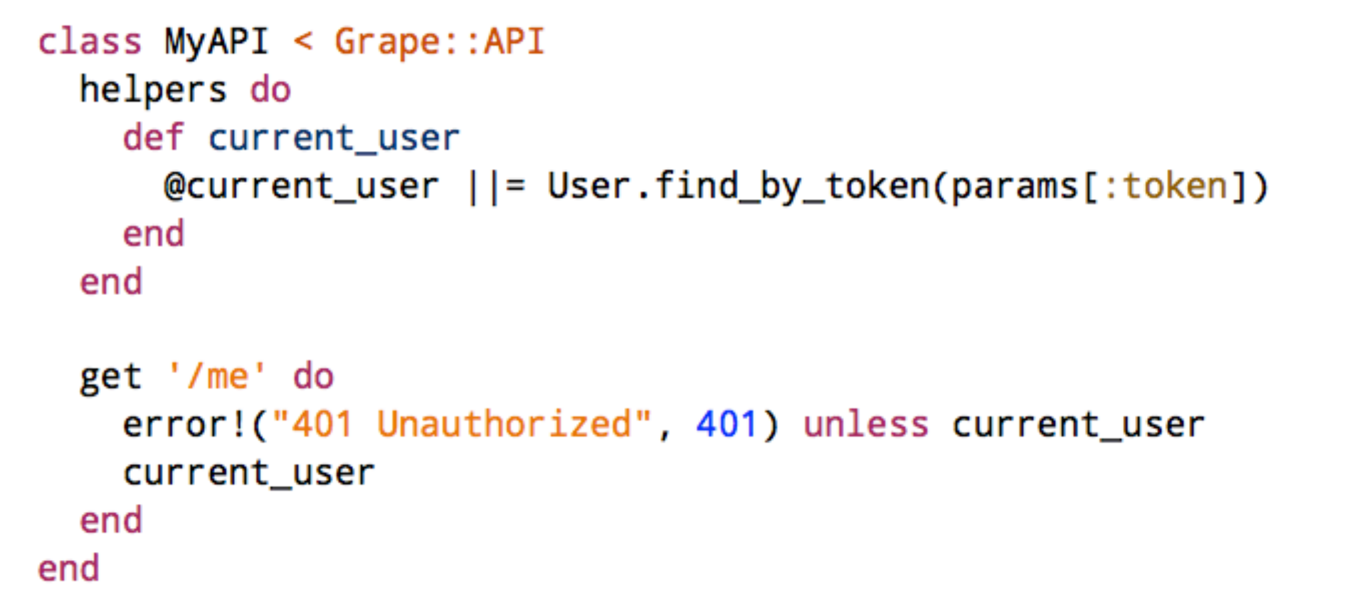
```
admin:somepassword
GET /admin/metrics
{“clicks”:235343}
```

Use Helpers

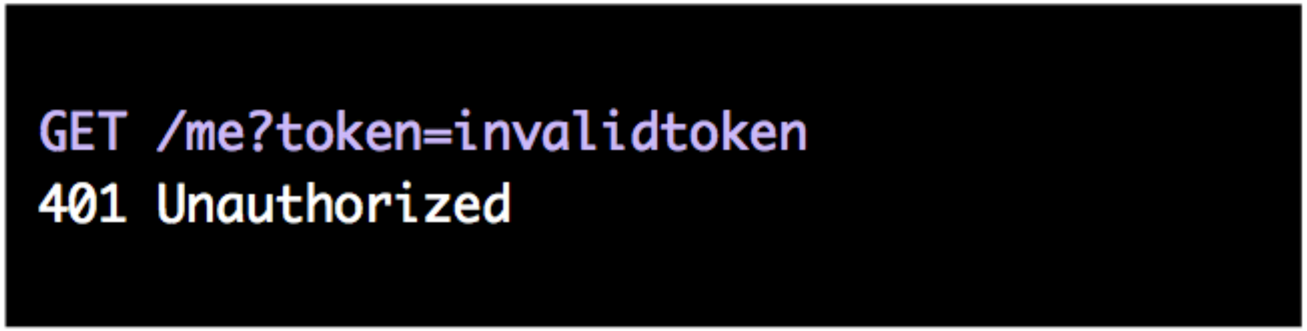


```
class MyAPI < Grape::API
  helpers do
    def current_user
      User.find_by_token(params[:token])
    end
  end

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

Raise Error

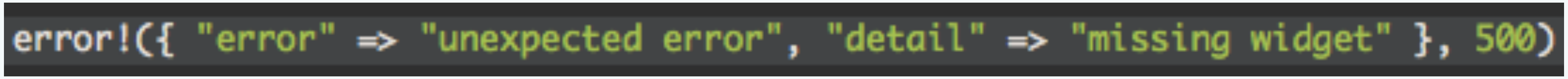


```
class MyAPI < Grape::API
  helpers do
    def current_user
      @current_user ||= User.find_by_token(params[:token])
    end
  end

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



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

Error Details

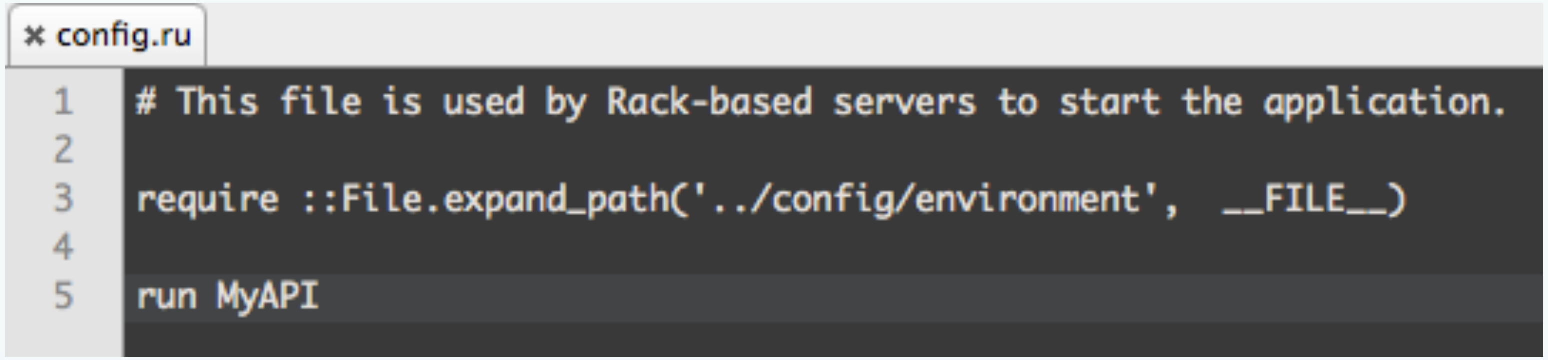


```
error!({ "error" => "unexpected error", "detail" => "missing widget" }, 500)
```

How To Run Grape APIs?

Remember: Grape APIs are Rack applications

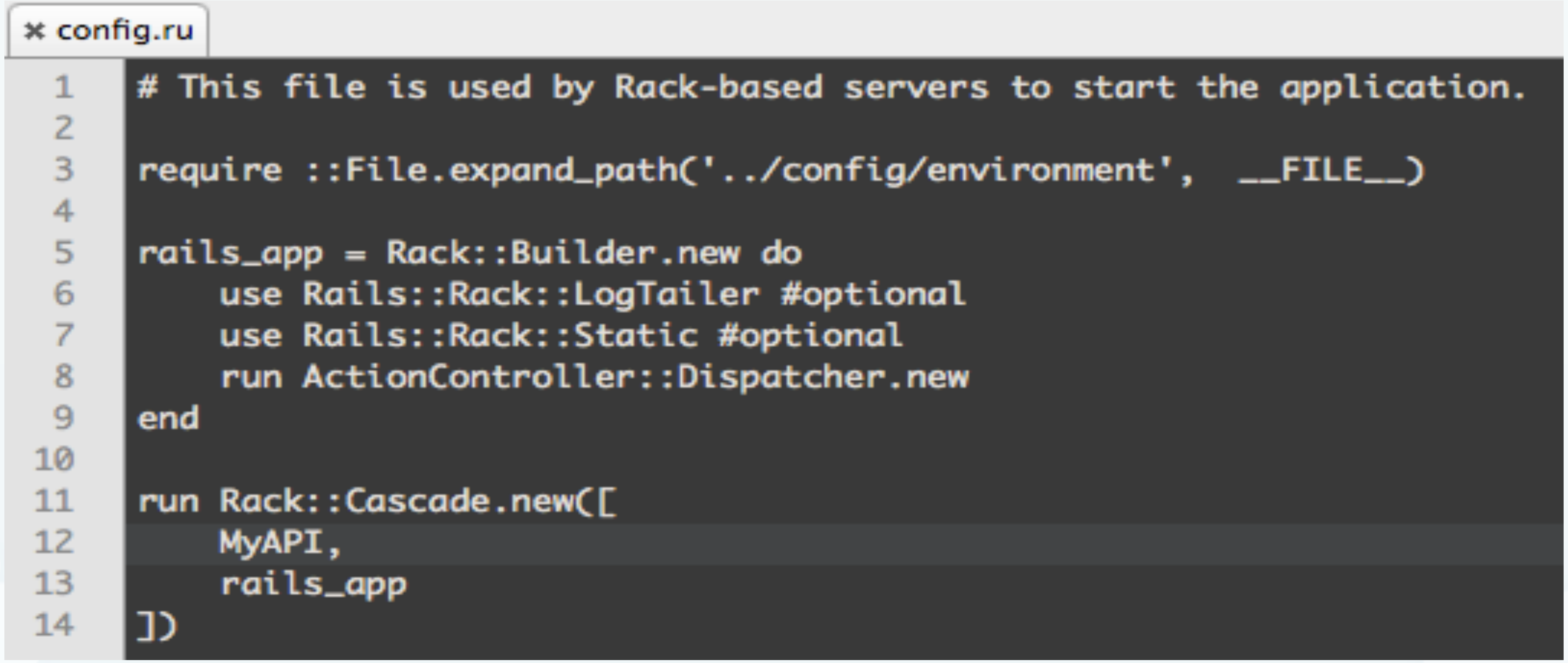
Run As a Rack App



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

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

run MyAPI
```

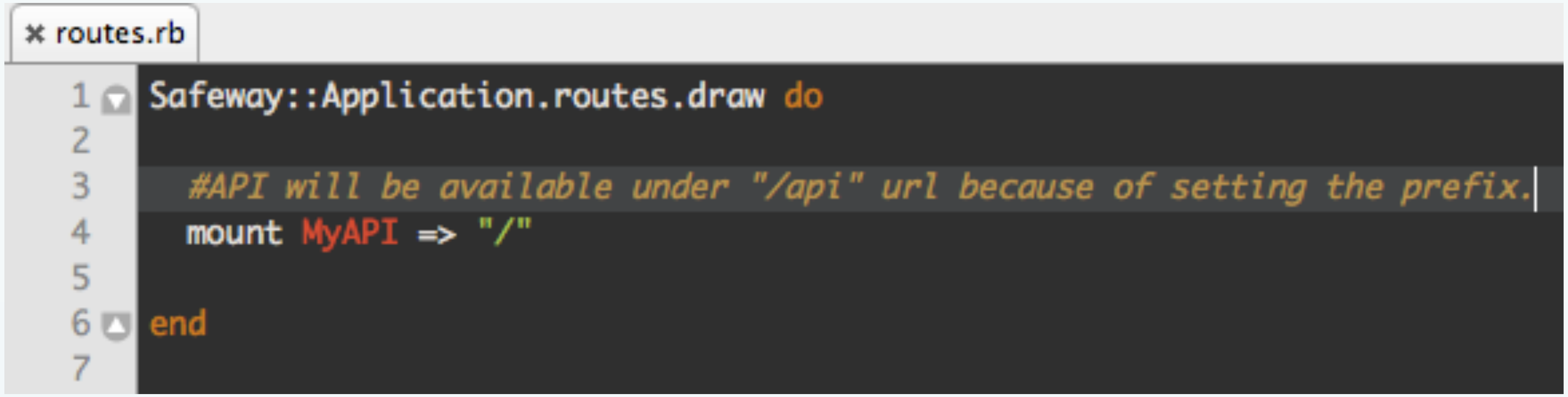
Run `rackup` in Rack App(API) folder

Grape with Rails 2.3.x



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

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

rails_app = Rack::Builder.new do
    use Rails::Rack::LogTailer #optional
    use Rails::Rack::Static #optional
    run ActionController::Dispatcher.new
end

run Rack::Cascade.new([
    MyAPI,
    rails_app
])
```

Grape with Rails 3.x



```ruby
Safeway::Application.routes.draw do

  #API will be available under "/api" url because of setting the prefix.
  mount MyAPI => "/"

end

```

Any Future Feature?

YES!

Near Future

*Support for XML formatting
*Multiple-file APIs (modularity)
*OAuth 1.0a and 2.0 authentication support
Filters/Callbacks

https://github.com/makandra/modularity
http://code.dblock.org/modularizing-a-ror-grape-api

Medium Future

YARD extensions for self-documenting APIs
Plugin System for easy extension
Internal API Access
Rate Limiting
Lifecycle Automation

Far Future

Streaming API support
PubSubHubBub

Resources & References

RubyConf 2010 presentation about Grape.

http://confreaks.net/videos/475-rubyconf2010-the-grapes-of-rapid

https://github.com/downloads/intridea/grape/The%20Grapes%20of%20Rapid.pdf

Example Rails implementation

http://martinciu.com/2011/01/mounting-grape-api-inside-rails-application.html

Demo & Sample

Demo API with Sinatra App.
https://github.com/mbleigh/conf_ask
http://www.arailsdemo.com/posts/30

Well Documented Sample API.
https://gist.github.com/892282

Questions?

Thank You!