Guille Carlos

- **@guillec**
- Pair Programming Meetup
- Bitpop (http://bitpop.in)
  - Ruby on Rails
  - Mobile
  - Want work?
  - Love burritos?

3 Parts

1. Intro
2. Making it more Awesome-ish?
3. Tips and Ideas

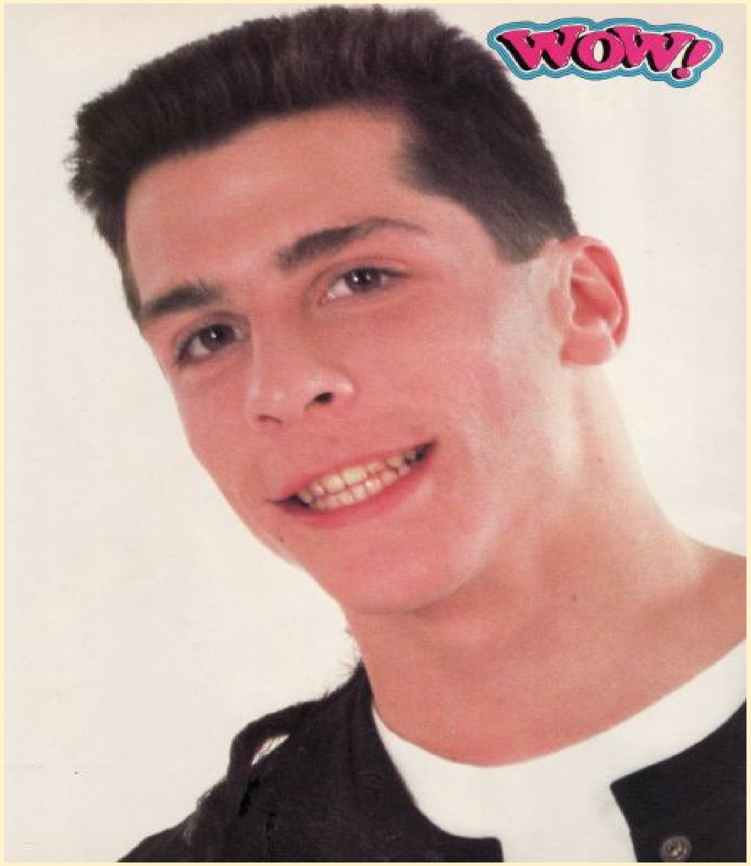
PART 1: Step by Step, Ooooh baby

for Building APIs

Step 1: We can have lots of fun (danny)

RAILS IS FUN!

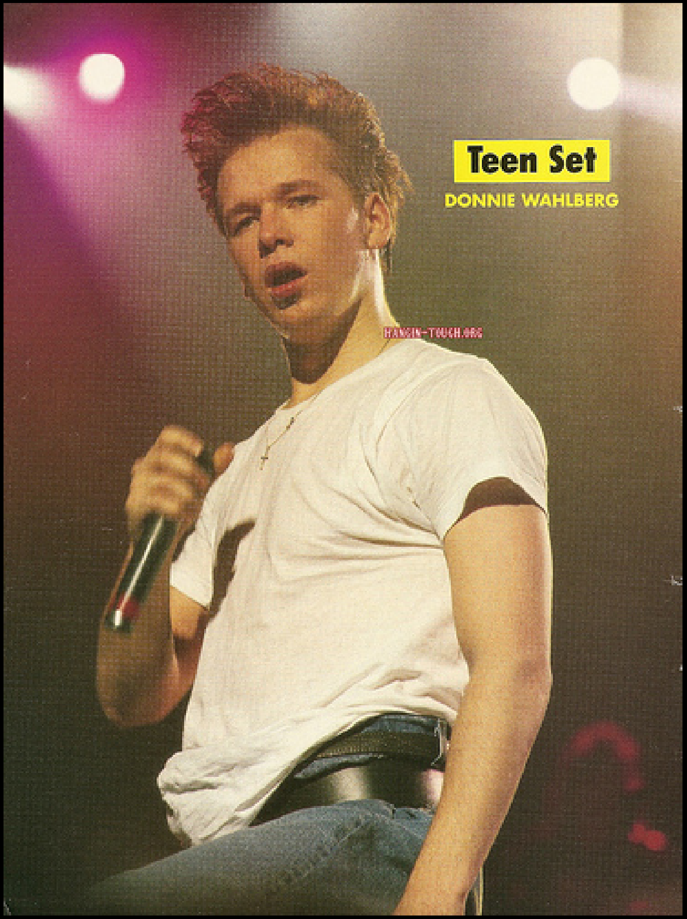
```
rails g model car
rails g controller cars
```



Step 2: There's so much we can do
(donnie)

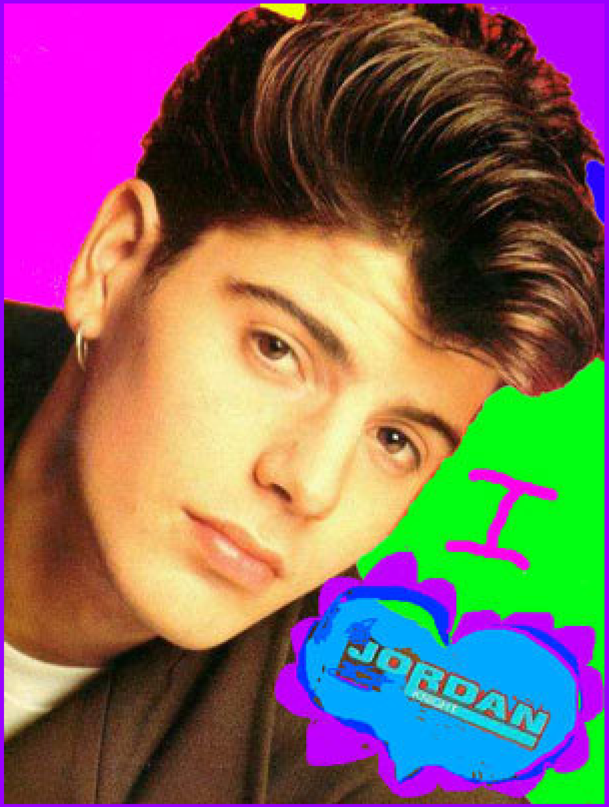
app/controllers/cars_controller.rb

```
def index
  @cars = Car.all
end
```



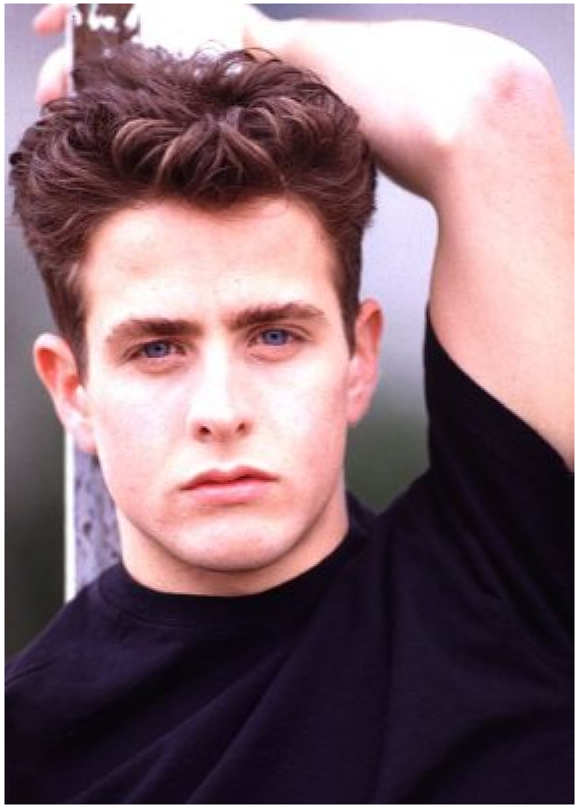
Step 3: it's just you, for me (jordan)

```
vim config/routes.rb

resources :cars
```



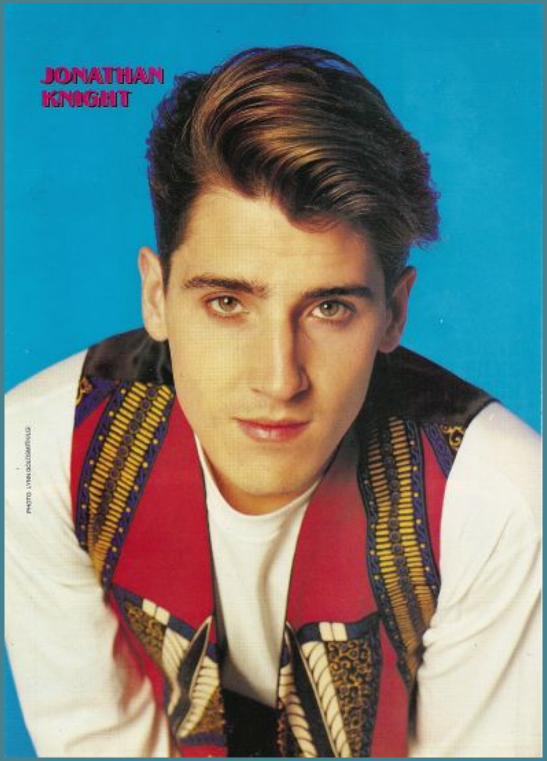
Step 4: I can give you more (joe)

```
respond_to :json, :xml, :html

def index
  @c = Car.all

  respond_to do |format|
    format.html
    format.json { render @c.to_json }
    ....
  end
end
```



Step 5: dont you know that the time has arrived. (jon)

http://localhost:3000/cars.json
http://localhost:3000/cars.html
http://localhost:3000/cars.xml



SECURITY WARNING:

Security issue with non-HTML formats

Please note that using default to_xml or to_json methods can lead to security holes, as these method expose all attributes of your model by default, including salt, crypted_password,permissions, status or whatever you might have. You might want to override these methods in your models, e.g.:

```
def to_xml
  super( :only => [ :login, :first_name, :last_name ] )
end
```

Or consider not using responds_to at all, if you only want to provide HTML.

PART 2: Fight!!

I CODED A
WEBSERVICE
I DONT KNOW HOW TO
VERSION IT
memegenerator.net

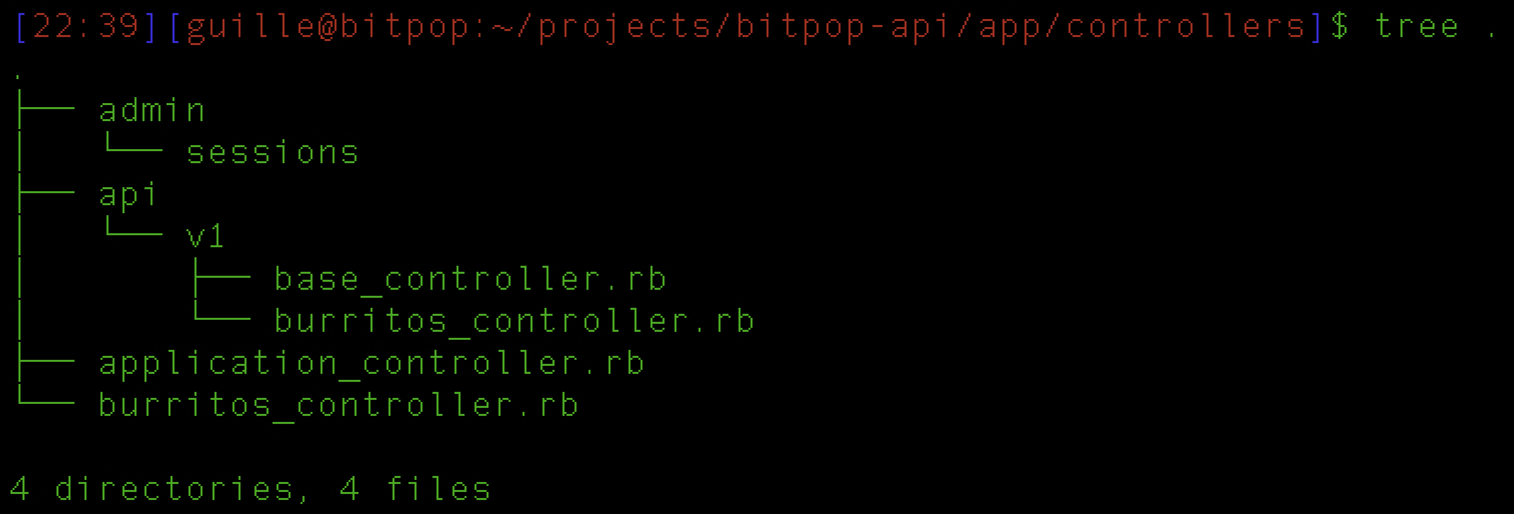
Example 1:
GET /api/burritos?version=1

Example 2:
GET /api/v1/burritos.json

Example 3:
GET /burritos
Accept: application/vnd.bitpop.api-v2+json

GET /api/v1/burritos.json

rails g controller api::v1::burritos



```
[22:39][guille@bitpop:~/projects/bitpop-api/app/controllers]$ tree .
.
├── admin
│   └── sessions
├── api
│   └── v1
│       ├── base_controller.rb
│       └── burritos_controller.rb
├── application_controller.rb
└── burritos_controller.rb

4 directories, 4 files
```

config/routes.rb

```ruby
namespace :api do
  namespace :v1 do
    resources :burritos
  end
end
```

rake routes

```
    api_v1_burritos GET    /api/v1/burritos(.:format)          api/v1/burritos#index
                    POST   /api/v1/burritos(.:format)          api/v1/burritos#create
 new_api_v1_burrito GET    /api/v1/burritos/new(.:format)      api/v1/burritos#new
edit_api_v1_burrito GET    /api/v1/burritos/:id/edit(.:format) api/v1/burritos#edit
     api_v1_burrito GET    /api/v1/burritos/:id(.:format)      api/v1/burritos#show
                    PUT    /api/v1/burritos/:id(.:format)      api/v1/burritos#update
                    DELETE /api/v1/burritos/:id(.:format)      api/v1/burritos#destroy
```

Edit app/controllers/api/v1/burritos_controllers.rb

```
class Api::V1::BurritosController < Api::V1::BaseController
  respond_to :json, :xml, :html

  def index
    respond_with Burrito.for(current_user)
  end

end
```

```
request: curl "http://bitpop-api.dev/api/v1/burritos.json"
response: { some json }
```

```
GET /burritos
Accept: application/vnd.bitpop.api-v2+json
```

vim config/initializers/mime_types.rb

```
Mime::Type.register "application/vnd.bitpop.api-v2+json", :api_v2
```

EDIT app/controllers/burritos_controllers.rb

```
class BurritosController < ApplicationController
  respond_to :html, :api_v2

  def show
    @burrito = Burrito.find(params[:id])
    respond_with(@burrito) do |format|
      format.api_v2 { render ... }
    end
  end

end
```

BUT!

If the resource responds to a `#to_#{format}` method, and a renderer exists for the format, then use the renderer; otherwise, attempt to render using an appropriate template.

rails / actionpack / lib / action_controller / metal / **renderers.rb**

```
add :json do |json, options|
  json = json.to_json(options) unless json.kind_of?(String)
  json = "#{options[:callback]}(#{json})" unless options[:callback].blank?
  self.content_type ||= Mime::JSON
  self.response_body  = json
end
```

CREATE app/renderers/api_v2_renderer.rb

```
ActionController::Renderers.add :api_v2 do |resource, options|
  self.content_type = Mime::API_V2
  self.response_body = resource.to_api_v2(options)
end
```

ADD app/models/burrito.rb

```
def to_api_v2(options)
  { id: id, name: name }.to_json(options)
end
```

EDIT app/controllers/burritos_controllers.rb

```
class BurritosController < ApplicationController
  respond_to :html, :api_v2

  def show
    @burrito = Burrito.find(params[:id])
    respond_with(@burrito)
  end
end
```

request:

```
curl -H 'ACCEPT: application/vnd.bitpop.api-v2+json' "http://bitpop-api.dev/burritos/1"
```

YO DAWG
THAT AINT RESTFUL
memegenerator.net

EDIT app/models/burrito.rb

```
class Burrito < ActiveRecord::Base

  def to_api_v2(options)
    { id: id, name: name, links: links }.to_json(options)
  end

end
```

```ruby
class Link < Struct.new(:ref, :href, :text)
end


class Burrito < ActiveRecord::Base
  include ActionView::Helpers::UrlHelper
  include ActionController::UrlFor
  include Rails.application.routes.url_helpers
  validates :name, :presence => true
  cattr_accessor :controller, :env

  def controller
    self.class.controller
  end

  def request
    controller.request
  end

  def self.for(user)
    Burrito.where(:secret => false)
  end

  def links
    [Link.new( index , burrito_url(self),  This burrito )]
  end

  def to_api_v2(options)
    { id: id, name: name, links: links }.to_json(options)
  end
end
```

HATEOAS

Hypertext As The Engine Of Application State

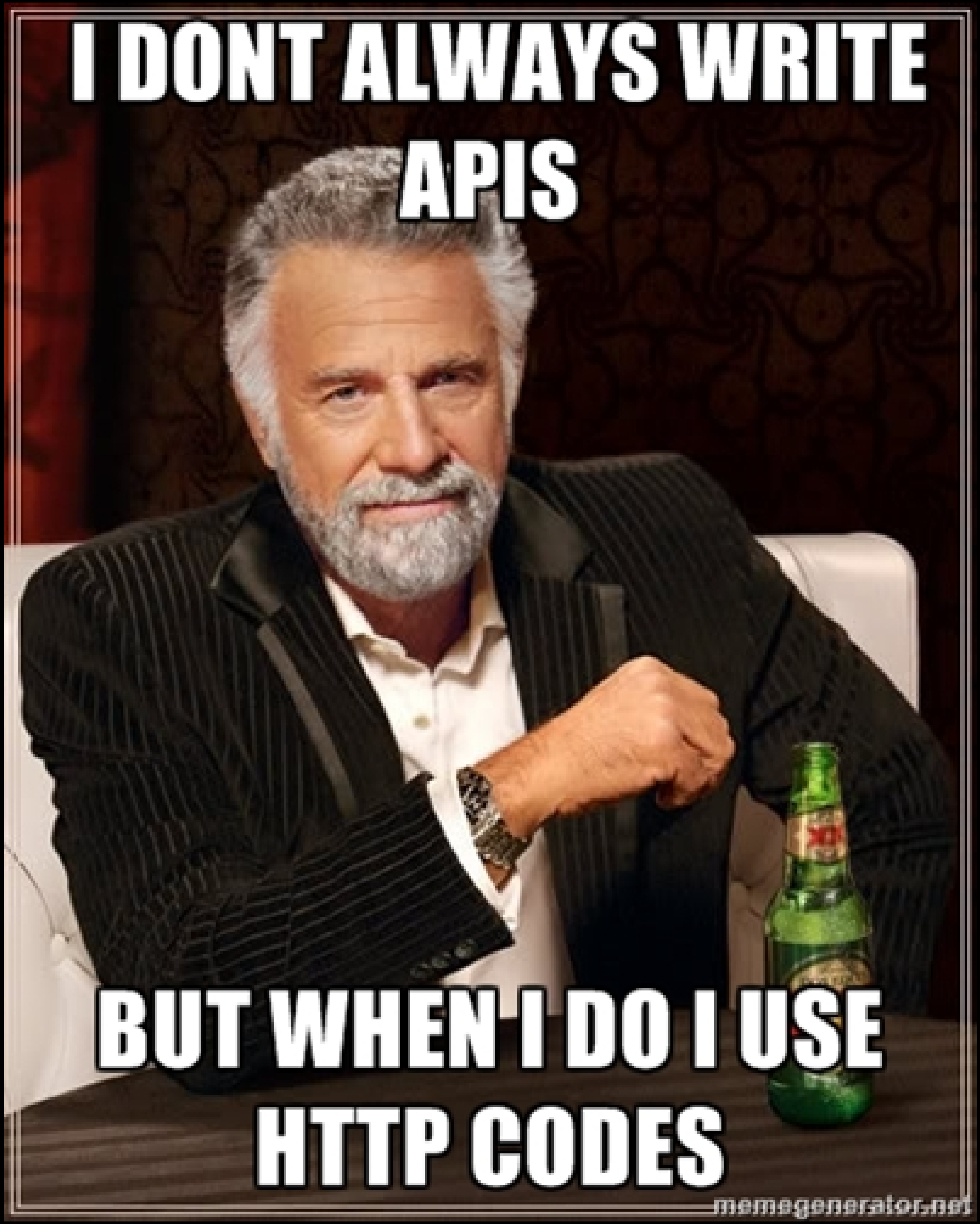
request:

```
curl -H 'ACCEPT: application/vnd.bitpop.api-v2+json' "http://bitpop-api.
dev/burritos/1"
```

response:

```
{
 "id":1,
 "name":"Burrito Test",
 "links": [{
     "ref":"self",
     "href":"http://bitpop-api.dev/burritos/1",
     "text":"This burrito"
  }]
}
```

PART 3


I DONT ALWAYS WRITE APIS
BUT WHEN I DO I USE HTTP CODES
memegenerator.net

gem install cheat

cheat status_codes

```
2xx Success

  200 => :ok
  201 => :created
  202 => :accepted
  203 => :non_authoritative_information
  204 => :no_content
  205 => :reset_content
  206 => :partial_content
  207 => :multi_status
  226 => :im_used
```

Rails is pretty good

```
curl -IL "http://bitpop-api.dev/aslkfsadlfkhsffshflksfhsfhslakfhsdfkljh"
```

```
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=utf-8
Content-Length: 761
X-Request-Id: 5416df2b4b017ae219ee12df1c5063ce
X-Runtime: 0.011654
Connection: keep-alive
```

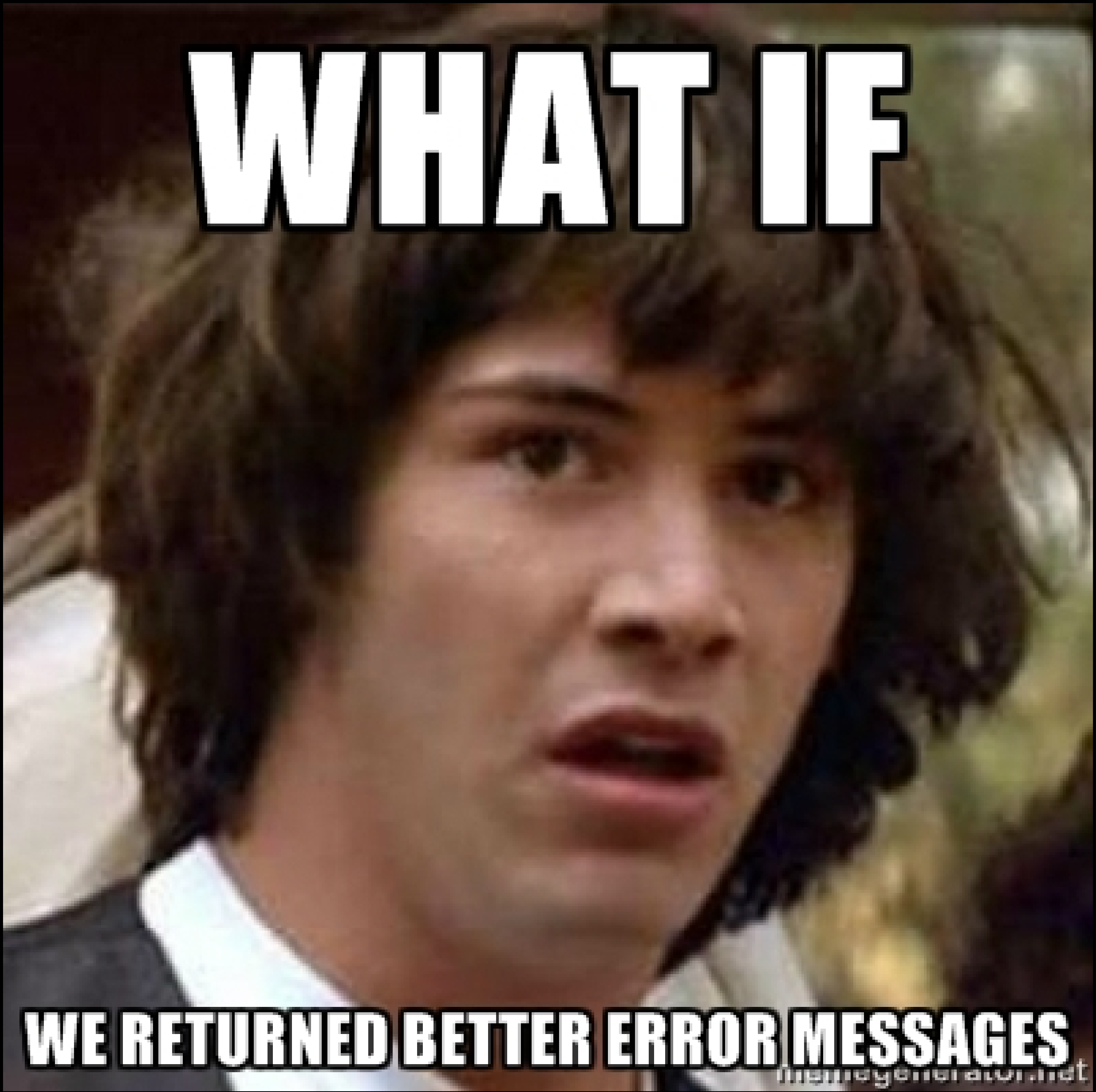
But could be better

curl -I "http://bitpop-spi.dev/api/v1/burritos.json?&token=INVALIDTOKEN"

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-UA-Compatible: IE=Edge
ETag: "a8ef9e470d4fdda863d8f7740597b534"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 9e9bbe34c2b5f402fd4349329dfb3b8f
X-Runtime: 0.006695
Connection: close

{"error":"Token is invalid"}
```

Better HTTP code

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json; charset=utf-8
X-UA-Compatible: IE=Edge
Cache-Control: no-cache
X-Request-Id: 5c6ca45f0c74b11dbf01ccd915616431
X-Runtime: 0.006564
Connection: close
```


WHAT IF
WE RETURNED BETTER ERROR MESSAGES
memegenerator.net

GET /burritos.json?token="BADTOKEN"

HTTP/1.1 500 Internal Server Error

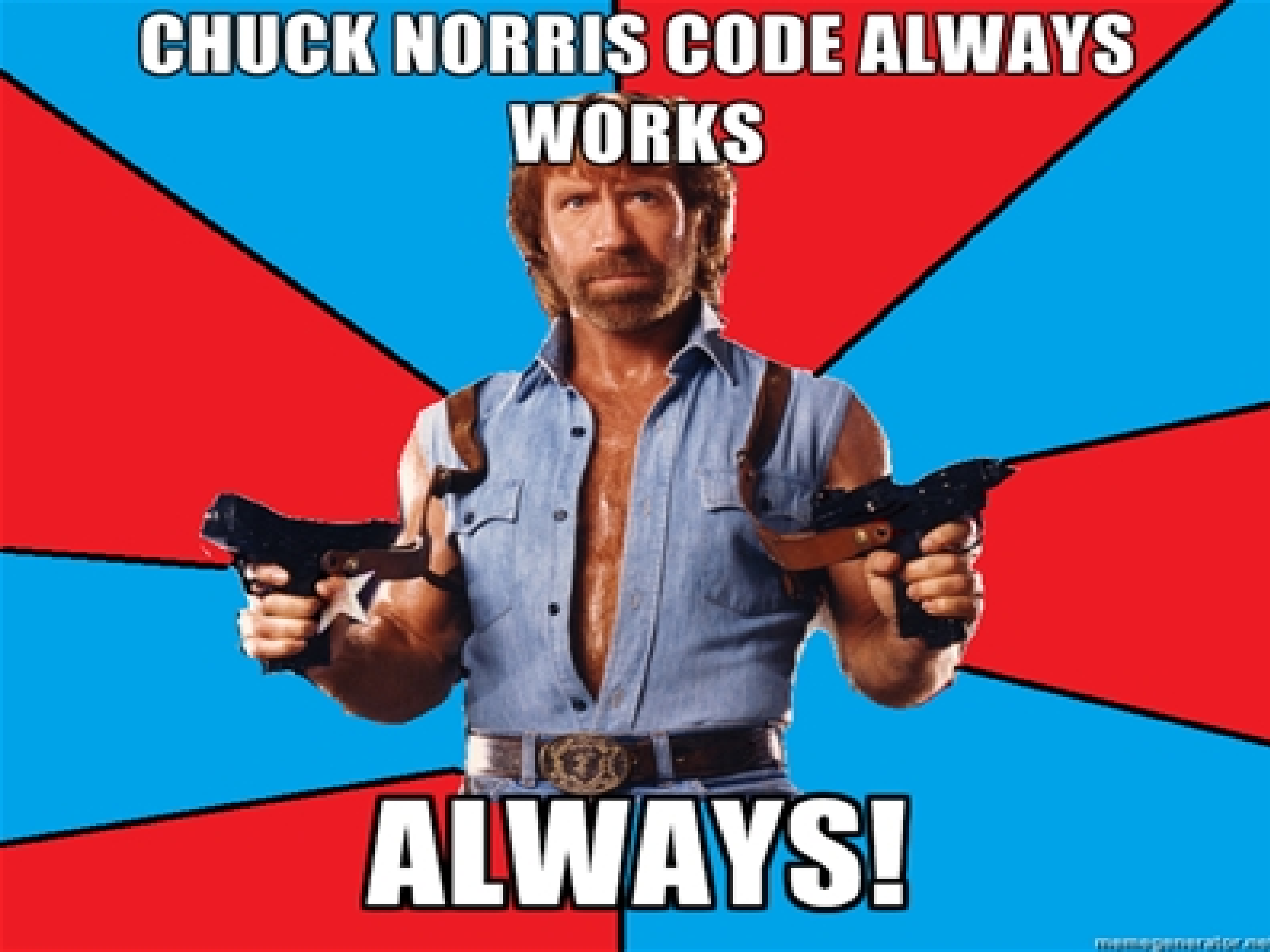
HTTP/1.1 401 Unauthorized

HTTP/1.1 401 Unauthorized

{ "errors": ["Bad=key if you forgot you key go here and get new one http://bitpop-api.dev/auth"] }

app/controllers/api/v1/base_controller.rb

```
def authenticate_user
  @current_user =User.find_by_authentication_token(params[:token])
  if @current_user.nil?
    respond_with( { :error => "Token is invalid" }, :status => 401 )
  end
end
```


CHUCK NORRIS CODE ALWAYS WORKS
ALWAYS!
memegenerator.net

gem rack-test

Test the actual URLs not just the actions

Sample spec

```
require "spec_helper"

describe "/ap1/v1/burritos", :type => :api do
...
end
```

spec/support/api/helper.rb

```
module ApiHelper
  include Rack::Test::Methods

  def app
    Rails.application
  end
end

Rspec.configure do |c|
  c.include ApiHelper, :type => :api
end
```

spec/api/v1/burritos_spec.rb

```
let(:url) { "/api/v1/burritos" }
```

vs

```
get :index
```

Nested Resources

What to return?

burrito has_many :condiments

return all?

What would be better for mobile?

Add query param to URL /burritos?condiments=true

Separate API Object

Create a new object for your API

Burrito and Api::Burrito

That's all Folks!