RubyMotion

A brave new world for iOS development

Michael Erasmus

About Me

About Me

Ruby, JavaScript, Web

About Me

Ruby, JavaScript, Web

“Backend Guy”

About Me

Ruby, JavaScript, Web

“Backend Guy”

Hey, let’s write iPhone apps!

So much to learn...

So much to learn...

iOS

So much to learn...

iOS

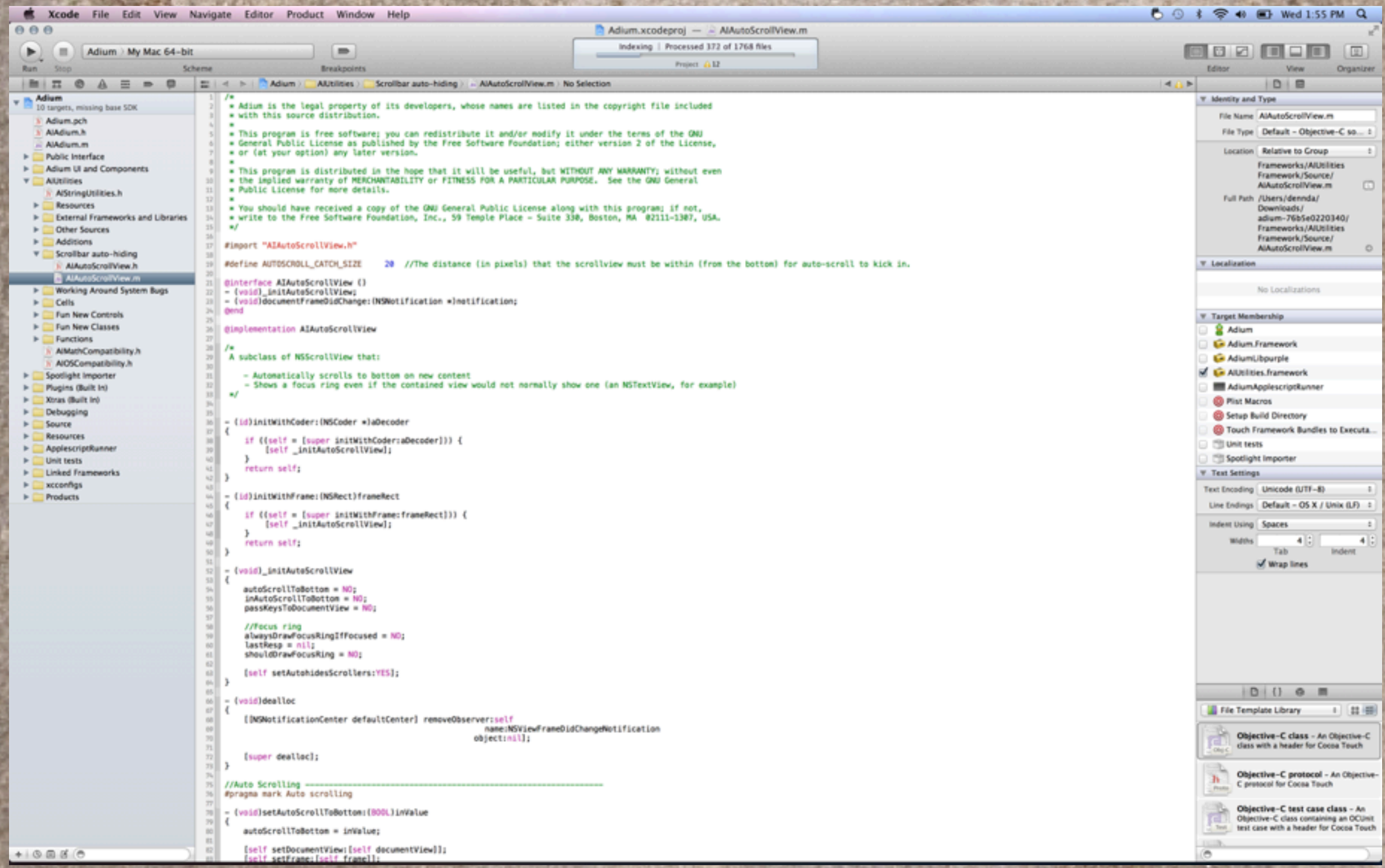
Objective-C

So much to learn...

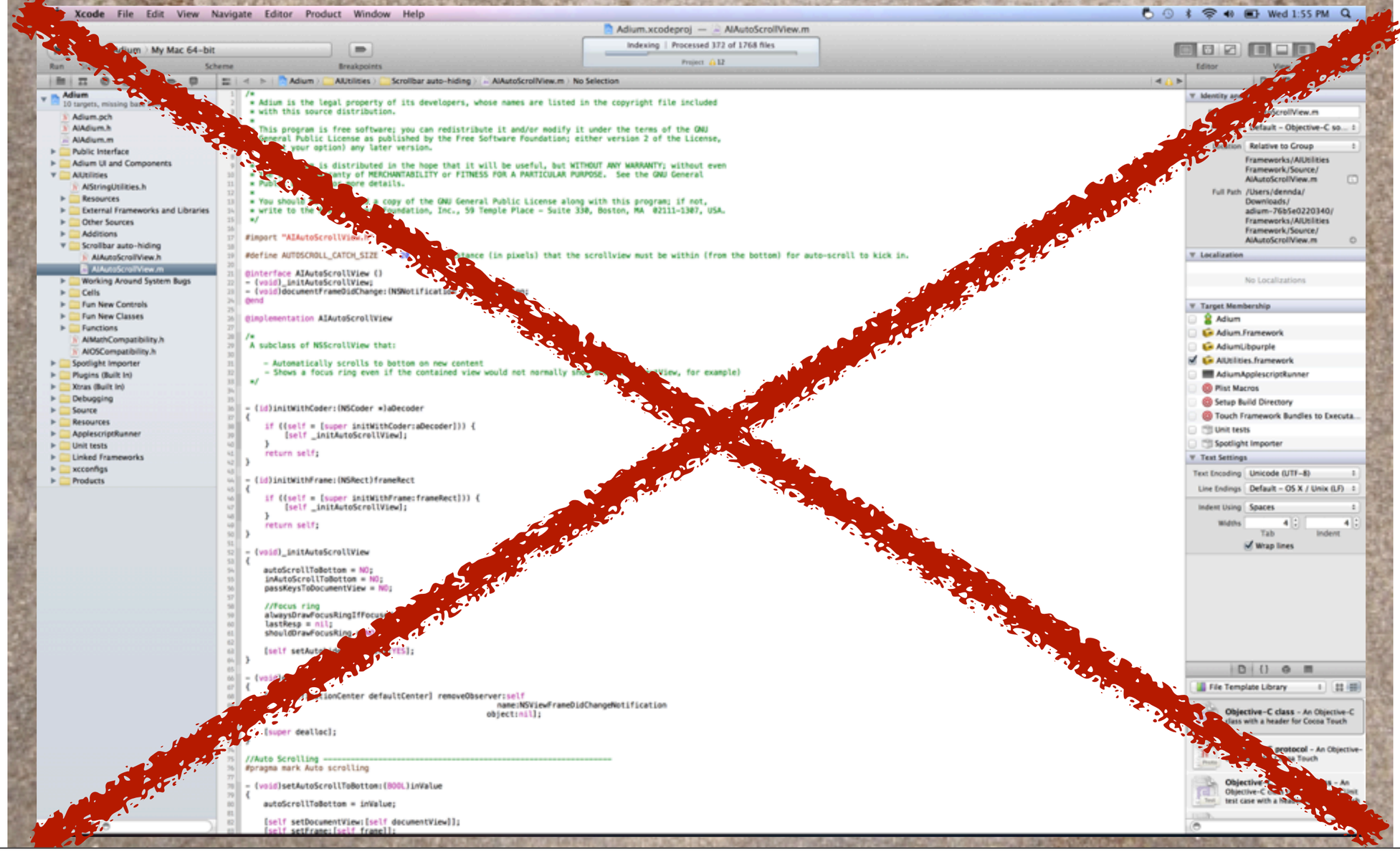
iOS

Objective-C

Uhm... this is awkward



Uhm... this is awkward



RubyMotion to the Rescue!

So...what exactly is it?

"a revolutionary **toolchain** that lets you quickly develop and **test native** iOS and OS X applications for **iPhone, iPad** and **Mac**, all using the awesome **Ruby** language you know and love."

rubymotion.com

So...what exactly is it?

"a revolutionary **toolchain** that lets you quickly develop and **test native** iOS and OS X applications for **iPhone, iPad** and **Mac**, all using the awesome **Ruby** language you know and love."

rubymotion.com

Toolchain

Toolchain

Ruby, Rake, Bundler, RubyGems, irb

Toolchain

Ruby, Rake, Bundler, RubyGems, irb

Vim, Emacs, Sublime Text 2

Toolchain

Ruby, Rake, Bundler, RubyGems, irb

Vim, Emacs, Sublime Text 2

UIKit, AppKit, CocoaPods

Native

Not a “bridge”

Compiled machine language

Performance on par with Obj-C

App store friendly

Ruby

Ruby 1.9 dialect

A couple of syntax tweaks

Some of the standard lib

Multiplatform

iPhone

iPad

Mac OSX

Testing

MacBacon

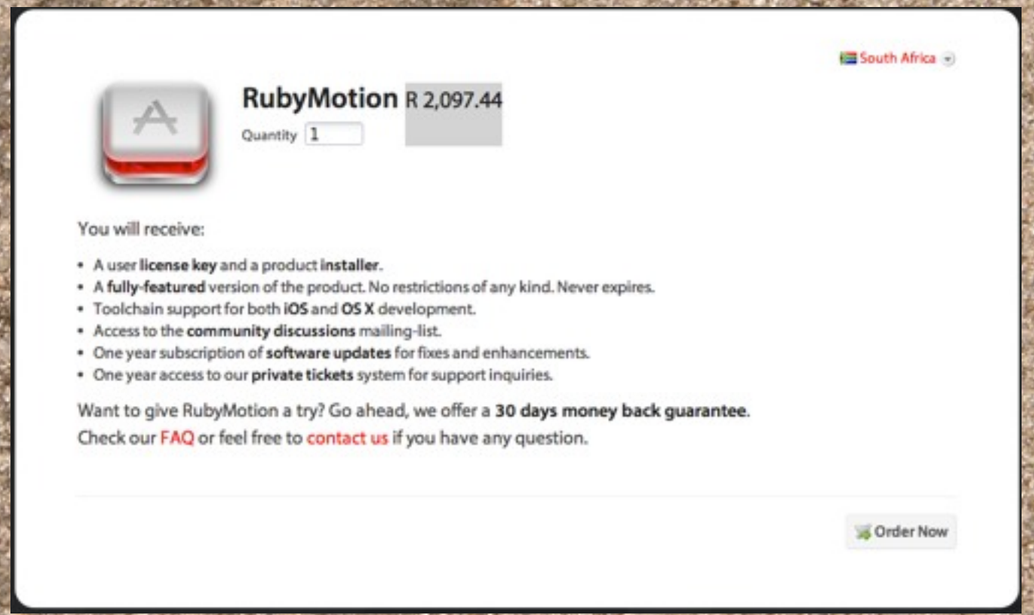
RSpec like

Focus on testing


What's the catch?
chan4chan.com
ROFLRAZZI

Well...

It’s not open source



Well...

Some of it's open source

https://github.com/HipByte/RubyMotion

Where does it come from?

Where does it come from?

http://www.hipbyte.com/

Where does it come from?

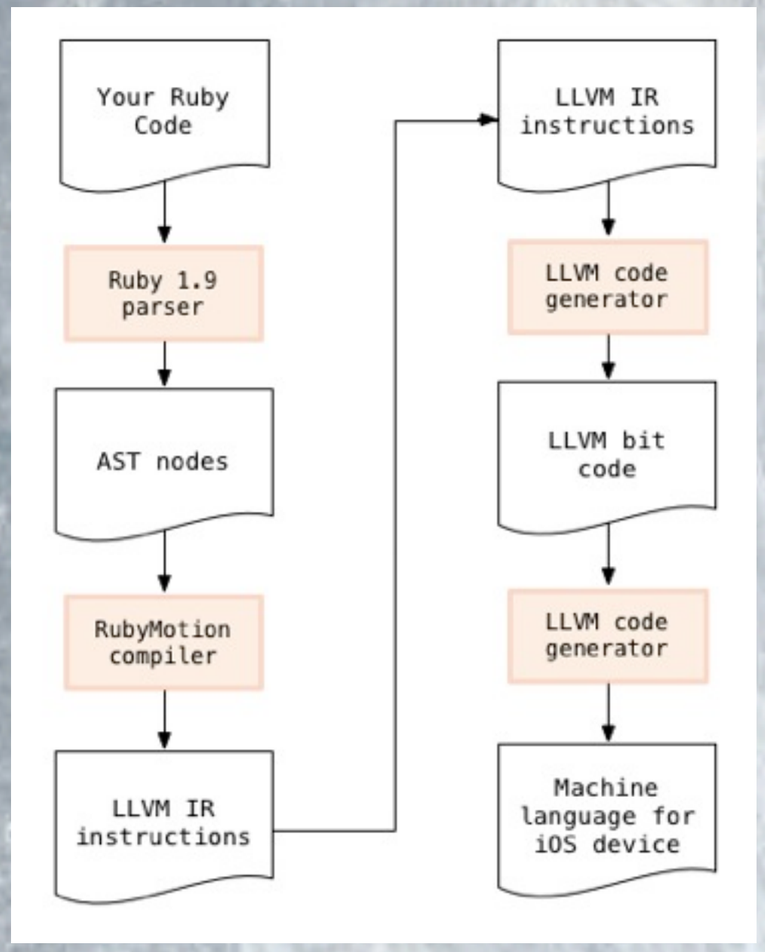
RubyCocoa -> MacRuby -> RubyMotion

How does it work?

How does it work?

LLVM

How does it work?



Take code like this:

```
- (void)viewDidLoad {
  [super viewDidLoad];
  responseData = [[NSMutableData data] retain];
  NSURLRequest *request = [NSURLRequest requestWithURL:[NSURL URLWithString:@"http://www.example.com/api.json"]];
  [[NSURLConnection alloc] initWithRequest:request delegate:self];
}

- (void)connection:(NSURLConnection *)connection didReceiveResponse:(NSURLResponse *)response {
  [responseData setLength:0];
}

- (void)connection:(NSURLConnection *)connection didReceiveData:(NSData *)data {
  [responseData appendData:data];
}

- (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error {
  label.text = [NSString stringWithFormat:@"Connection failed: %@", [error description]];
}

- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
  [connection release];

  NSString *responseString = [[NSString alloc] initWithData:responseData encoding:NSUTF8StringEncoding];
  [responseData release];

  NSError *error;
  SBJSON *json = [[SBJSON new] autorelease];
  NSArray *myArray = [json objectWithString:responseString error:&error];
  [responseString release];

  if (myArray == nil)
    label.text = [NSString stringWithFormat:@"JSON parsing failed: %@", [error localizedDescription]];
  else {
    NSMutableString *text = [NSMutableString stringWithString:@"Array Data:\n"];

    for (int i = 0; i < [myArray count]; i++)
      [text appendFormat:@"%@\n", [myArray objectAtIndex:i]];

    label.text =  text;
  }
}
```

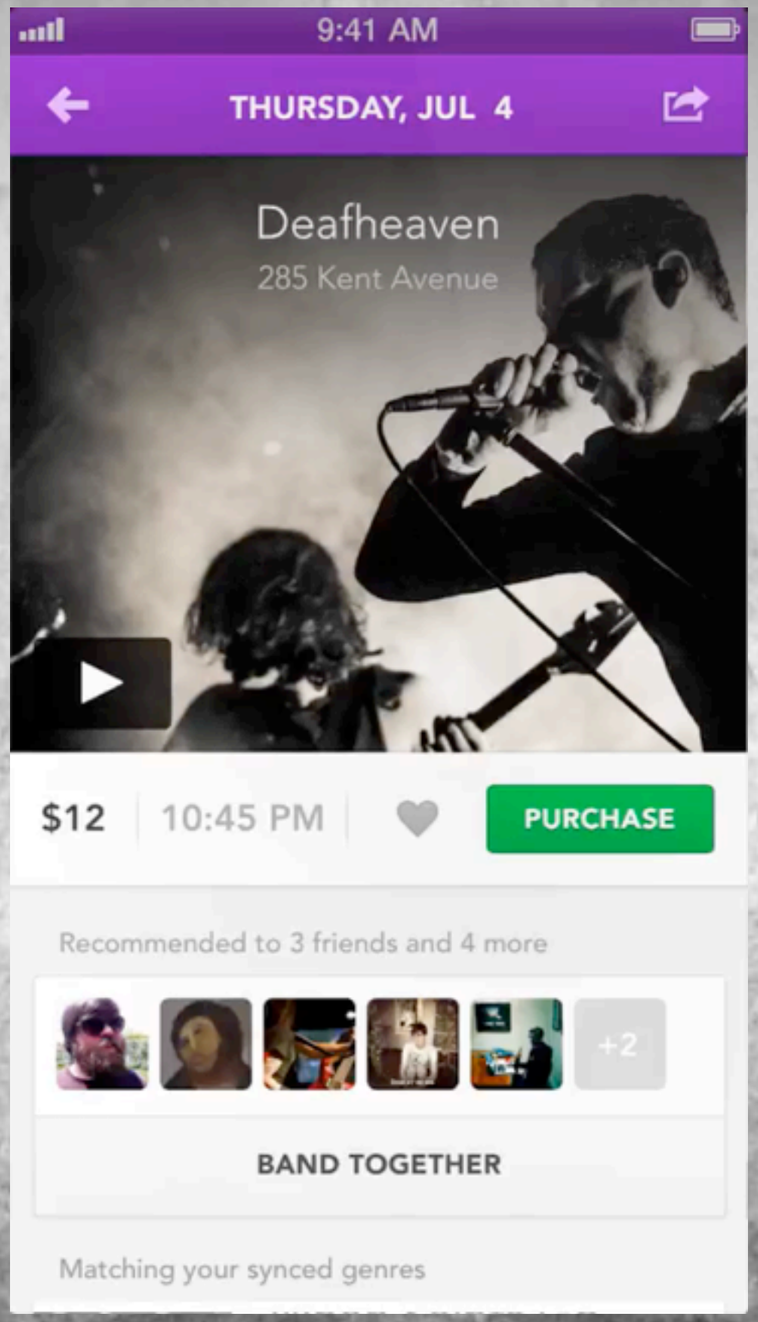
Write it like this:

```ruby
require 'net/http'
require 'json'

url = "http://www.example.com/api.json"
response = Net::HTTP.get_response(URI.parse(url))

if response.code.to_i == 200
  json = JSON.parse(response.body)
  puts json.inspect
else
  puts "An Error Occurred (#{response.code}): #{response.body}"
end
```

Demo

Is it worth it?

http://jukely.com


9:41 AM
THURSDAY, JUL 4
Deafheaven
285 Kent Avenue
$12
10:45 PM
PURCHASE
Recommended to 3 friends and 4 more
+2
BAND TOGETHER
Matching your synced genres

http://thetempleapp.com/


Temple
Hydration, Food, Fitness and More
Download on the App Store
The health and fitness tracker so easy and fun you'll actually use it.
One tap entry and a cool, fresh design. Record your daily activities quickly and get back to your life
Get fit, hydrate more, drop weight, etc.
Features
Screenshots
177
Like
Carrier
10:51 AM
temple
Progress
streak (great days in a row)
months
Jan Feb Mar Apr May
This week
Fitness
Fluids
Fuel
Carrier
1:04 PM
Today
Fitness
Quick
Light
Moderate
Intense
Fluids
0 sec
113%
Half Glass
Glass
Bottle
Large Bottle
Undo
Fuel
23 min
85%
Snack
Small Meal
Full Meal
Large Meal

37signals Basecamp

Some awesome gems

Bubblewrap

http://bubblewrap.io

```
> App.documents_path
# "/Users/mattetti/Library/Application Support/iPhone Simulator/5.0/Applications/EEC6454E-1816-451
> App.resources_path
# "/Users/mattetti/Library/Application Support/iPhone Simulator/5.0/Applications/EEC6454E-1816-451
> App.name
# "testSuite"
> App.identifier
# "io.bubblewrap.testSuite"
> App.alert("BubbleWrap is awesome!")
# creates and shows an alert message.
> App.run_after(0.5) {  p "It's #{Time.now}"   }
# Runs the block after 0.5 seconds.
> App.open_url("http://matt.aimonetti.net")
# Opens the url using the device's browser. (accepts a string url or an instance of `NSURL`.
> App::Persistence['channels'] # application specific persistence storage
# ['NBC', 'ABC', 'Fox', 'CBS', 'PBS']
> App::Persistence['channels'] = ['TF1', 'France 2', 'France 3']
# ['TF1', 'France 2', 'France 3']
```

```ruby
# Uses the front camera
BW::Device.camera.front.picture(media_types: [:movie, :image]) do |result|
  image_view = UIImageView.alloc.initWithImage(result[:original_image])
end

# Uses the rear camera
BW::Device.camera.rear.picture(media_types: [:movie, :image]) do |result|
  image_view = UIImageView.alloc.initWithImage(result[:original_image])
end

# Uses the photo library
BW::Device.camera.any.picture(media_types: [:movie, :image]) do |result|
  image_view = UIImageView.alloc.initWithImage(result[:original_image])
end
```

```
def viewWillAppear(animated)
  @foreground_observer = App.notification_center.observe UIApplicationWillEnterForegroundNotificat
    loadAndRefresh
  end

  @reload_observer = App.notification_center.observe ReloadNotification do |notification|
    loadAndRefresh
  end
end

def viewWillDisappear(animated)
  App.notification_center.unobserve @foreground_observer
  App.notification_center.unobserve @reload_observer
end

def reload
  App.notification_center.post ReloadNotification
end
```

Sugarcube

https://github.com/rubymotion/sugarcube

Promotion

https://github.com/clearsightstudio/ProMotion

```ruby
class AppDelegate < PM::Delegate
  def on_load(app, options)
    open RootScreen.new(nav_bar: true)
  end
end

class RootScreen < PM::Screen
  title "Root Screen"

  def push_new_screen
    open NewScreen
  end
end

class NewScreen < PM::TableScreen
  title "Table Screen"

  def table_data
    [{
      cells: [
        { title: "About this app", action: :tapped_about },
        { title: "Log out", action: :log_out }
      ]
    }]
  end
end
```

Walt

https://github.com/clayallsopp/Walt

```
@view = UIView.alloc.initWithFrame(....)

Walt.animate(
  assets: [{
    id: "logo",
    position: [100, 0],
    size: [110, 40],
    url: "http://bit.ly/S98Ta5"
  }],
  animations: [{
    duration: 2,
    operations: [{
      move: "logo",
      to: 150,
      axis: :y
    }],
    after: {
      duration: 2,
      operations: [{
        rotate: "logo",
        to: 360
      }]
    }
  }],
  in: @view
)
```

Joybox

https://joybox.io

Questions?

@michael_erasmus

motioncasts.tv