Sleight of Hand for the Ruby Man

Aaron Bedra
Relevance, Inc.
http://thinkrelevance.com

Metaprogramming is made possible by one simple principal

CODE IS DATA

DATA IS CODE

YAML configuration file

```yaml
server: thin
host: localhost
port: 31337

database:
  adapter: postgresql
  database: wicked_awesome_app_development
```

Turing the config into accessor methods

```
    class Configuration
  def self.load(config_file)
    config = Configuration.new
    config.config_hash = parse_yaml(config_file)
    config.parse_config(config_hash)
    config
  end

  def self.parse_yaml(config_file)
    YAML::load_file(config_file)
  end

  def self.parse_config(config_hash)
    mod = Module.new { config.each_pair { |k,v| define_method(k) { v } } }
    self.extend mod
  end
end
```

define_method()

- add new methods on the fly
- keep configuration flexible

```
CPUID:GenuineIntel 5.2.c irql:1f  SYSVER 0xf0000565

Dll Base DateStmp - Name               Dll Base DateStmp - Name
80100000 3202c07e - ntoskrnl.exe       80010000 31ee6c52 - hal.dll
80001000 31ed06b4 - atapi.sys          80006000 31ec6c74 - SCSIPORT.SYS
802c6000 31ed06bf - aic78xx.sys        802cd000 31ed237c - Disk.sys
802d1000 31ec6c7a - CLASS2.SYS         8037c000 31eed0a7 - Ntfs.sys
fc698000 31ec6c7d - Floppy.SYS         fc6a8000 31ec6ca1 - Cdrom.SYS
fc90a000 31ec6df7 - Fs_Rec.SYS         fc9c9000 31ec6c99 - Null.SYS
fc864000 31ed868b - KSecDD.SYS         fc9ca000 31ec6c78 - Beep.SYS
fc6d8000 31ec6c90 - i8042prt.sys       fc86c000 31ec6c97 - mouclass.sys
fc874000 31ec6c94 - kbdclass.sys       fc6f0000 31f50722 - VIDEOPORT.SYS
feffa000 31ec6c62 - mga_mil.sys        fc890000 31ec6c6d - vga.sys
fc708000 31ec6ccb - Msfs.SYS           fc4b0000 31ec6cc7 - Npfs.SYS
fefbc000 31eed262 - NDIS.SYS           a0000000 31f954f7 - win32k.sys
fefa4000 31f                                                  .SYS
feb8c000 31e
feacf000 31f                                                  ys
fc550000 316
fc718000 31e                                                  .sys
fc870000 31e                                                  SYS
fc5b0000 31e
fea3b000 31f

Address  dwo
fec32d84 801                                                  SecDD.SYS
801471c8 801                                                  toskrnl.exe
801471dc 801                                                  toskrnl.exe
80147304 803                                                  toskrnl.exe

Restart and set the recovery options in the system control panel
or the /CRASHDEBUG system start option.
```

Application Error
Bluescreen has performed an illegal operation. Bluescreen must be closed.
OK
YOU FAIL AT FAILING
No, that's not a double negative.
DIY.DESPAIR.COM

This trick can fail!

\___\\|/___/ Failboat example #1 \\___\\|/___/

```
server: thin
host: localhost
port: 31337

database:
  adapter: postgresql
  database: wicked_awesome_app_development

to_s: you suck at programming the internets
```

no, its not that bad yet
so far its just raining cats
ICANHASCHEEZBURGER.COM

Let’s add a check to the parse_config method

```
 def self.parse_config(config_hash)
  mod = Module.new do
    config.each_pair do |k,v|
      unless self.methods.include?(k)
        define_method(k) { v }
      end
    end
  end
  self.extend mod
end
```

λ

Let’s say...

- You are using a lighter weight Ruby framework such as Sinatra or Camping where the “controller” actions are blocks.
- You can’t legitimately unit test these actions

We can refactor it, we have the technology

Letting a presenter do the dirty work

```
    module Presenters
   class Index
     ... # delegation, ivar accessors, etc.
     def present_it
       @data = lambda {
         controller.erb :login, :locals => {
                                     :presenter => self
                                   }
       }
     end
   end
 end
```

@data is code

```
   get ‘/index/’ do
  @presenter = Presenters::Index.new(self)
  @presenter.present_it
  @presenter.data.call
end
```

What we've accomplished

- We have successfully pulled all our code into testable places and delivered nice clean code for our "controller".
- Through the use of lambda we can return anything back to the "controller" we want.

1:1
ACTUAL
SIZE

act
NOW!

AS SEEN ON
TV

Let the inspector find out whodunnit

- The inspector gives you hooks that enable you to find out exactly where a method is defined.
- This is extremely useful when you are overriding default methods in rails plugins.

Imagine if...

- You are using streamlined to develop a quick admin interface for your application.
- You update your code one morning to find that a co-worker has checked in some code that seems to be causing strange errors.
- That co-worker left last Friday for a 3 week vacation in the mountains and can't be reached.
- Your test suite doesn't expose any problems.

The stacktrace says that the offending method is Streamlined.ui_for

```
yourapp> ./script/console
>> require ‘inspector’
=> [“Inspector”]
>> Inspector.where_is_this_defined {Streamlined.ui_for
(:foo)}
=> "Streamlined received message 'ui_for', Line #6 of
yourapp/lib/extensions/streamlined/streamlined.rb"
```

Go from this

To this!

Links

- github.com/spicycode/the-inspector/tree/master
- mwrc2008.confreaks.com/03bowkett.html
- www.pragprog.com/screencasts/v-dtrubyom/the-ruby-object-model-and-metaprogramming
- aaronbedra.com
- workingwithrails.com/person/5499-aaron-bedra

relevance

Questions?