Debugging Ruby

I. Simple Ruby Code

ruby -r debug filename

- set breakpoints
- run by steps
- catch exceptions
- display stacktrace
- display variables
- evaluate expressions

Binding

Objects of class Binding encapsulate the execution context at some particular place in the code and retain this context for future use. The variables, methods, value of self, and possibly an iterator block that can be accessed in this context are all retained. Binding objects can be created using Kernel#binding, and are made available to the callback of Kernel#set_trace_func.

source && source_location

```
irb(main):003:0> puts p.method(:activate).source
  def activate
    update(active: true)
  end

irb(main):004:0> puts p.method(:activate).source_location
/home/berlin/projects/test/app/models/photo.rb
27
```

```ruby
class Demo
  def initialize(n)
    @secret = n
  end
  def get_binding
    return binding()
  end
end

k1 = Demo.new(99)
b1 = k1.get_binding
k2 = Demo.new(-3)
b2 = k2.get_binding

eval("@secret", b1)   #=> 99
eval("@secret", b2)   #=> -3
eval("@secret")       #=> nil
```

caller

Returns the current execution stack—an array containing strings in the form “file:line” or “file:line: in `method”‘. The optional start parameter determines the number of initial stack entries to omit from the result.

exception.backtrace

Returns any backtrace associated with the exception. The backtrace is an array of strings, each containing either “filename:lineNo: in `method'" or “filename:lineNo."

II. Pry

show-method

show-doc

show-source

binding.pry && object.pry

ls, cd, edit, play

find-method

```
class Object
  def pry(object=nil, hash={})
    if object.nil? || Hash === object
      Pry.start(self, object || {})
    else
      Pry.start(object, hash)
    end
  end
end
```

```
  # Start a Pry REPL on self.
  #
  # If `self` is a Binding then that will
be used to evaluate expressions;
  # otherwise a new binding will be
created.
```

By default typing `ls` shows you the local variables defined in the current context, and any public methods or instance variables defined on the current object.

Usage:

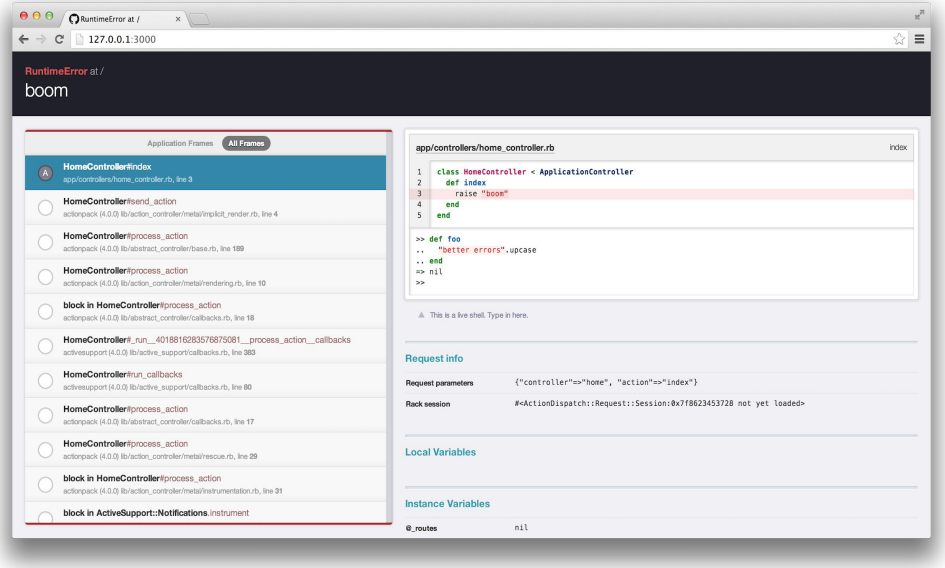
```
    ls [-m|-M|-p|-pM] [-q|-v] [-c|-i] [Object]
    ls [-g] [-l]

options:

    -m, --methods        Show public methods defined on the Object (default)
    -M, --module         Show methods defined in a Module or Class
    -p, --ppp            Show public, protected (in yellow) and private (in green) methods
    -q, --quiet          Show only methods defined on object.singleton_class and object.class
    -v, --verbose        Show methods and constants on all super-classes (ignores Pry.config.ls.ceiling)
    -g, --globals        Show global variables, including those builtin to Ruby (in cyan)
    -l, --locals         Show locals, including those provided by Pry (in red)
    -c, --constants      Show constants, highlighting classes (in blue), and exceptions (in purple)
    -i, --ivars          Show instance variables (in blue) and class variables (in bright blue)
    -G, --grep           Filter output by regular expression
    -h, --help           Show help
```

III. Rails

better_errors + binding_of_caller / byebug



- Full stack trace
- Source code inspection for all stack frames (with highlighting)
- Local and instance variable inspection
- Live shell (REPL) on every stack frame
- Links directly to the source line in your editor
- Useful information in non-HTML requests

rails_panel

Dejan Simic

rors.dev/

Hallo, wie geht's?

My name is Dejan and this is my blog. I'm a programmer based in Germany. Follow me on GitHub and Twitter for updates on my work and me.

Checkout some of my open-source projects:

RailsPanel - Chrome extension for Rails development (15.000+ users!)
auto_html - Rails engine for embedding rich content (5 years old, still maintained!)

Elements Network Sources Timeline Profiles Resources Audits Console Rails

Status	Controller#Action	Method	Format	Resp. Time
200	ArticlesController#show	GET	html	133 ms
200	ArticlesController#show	GET	html	32 ms
200	ArticlesController#edit	GET	html	45 ms
302	ArticlesController#update	POST	html	27 ms
200	ArticlesController#show	GET	html	49 ms
200	ArticlesController#index	GET	html	38 ms
200	ArticlesController#show	GET	html	50 ms
302	ArticlesController#destroy	POST	html	19 ms
200	AboutController#show	GET	html	29 ms

Breakdown Params ActiveRecord Rendering Log Error

Location	Type	SQL	Duration
controllers/application_controller.rb:8	User Load	SELECT `users`.* FROM `users` WHERE `users`.`id` = 1 LIMIT 1	1 ms
controllers/articles_controller.rb:49	Article Load	SELECT `articles`.* FROM `articles` WHERE `articles`.`id` = 38 LIMIT 1	1 ms
controllers/articles_controller.rb:50		BEGIN	1 ms
controllers/articles_controller.rb:50	SQL	DELETE FROM `articles` WHERE `articles`.`id` = 38	3 ms
controllers/articles_controller.rb:50		COMMIT	1 ms

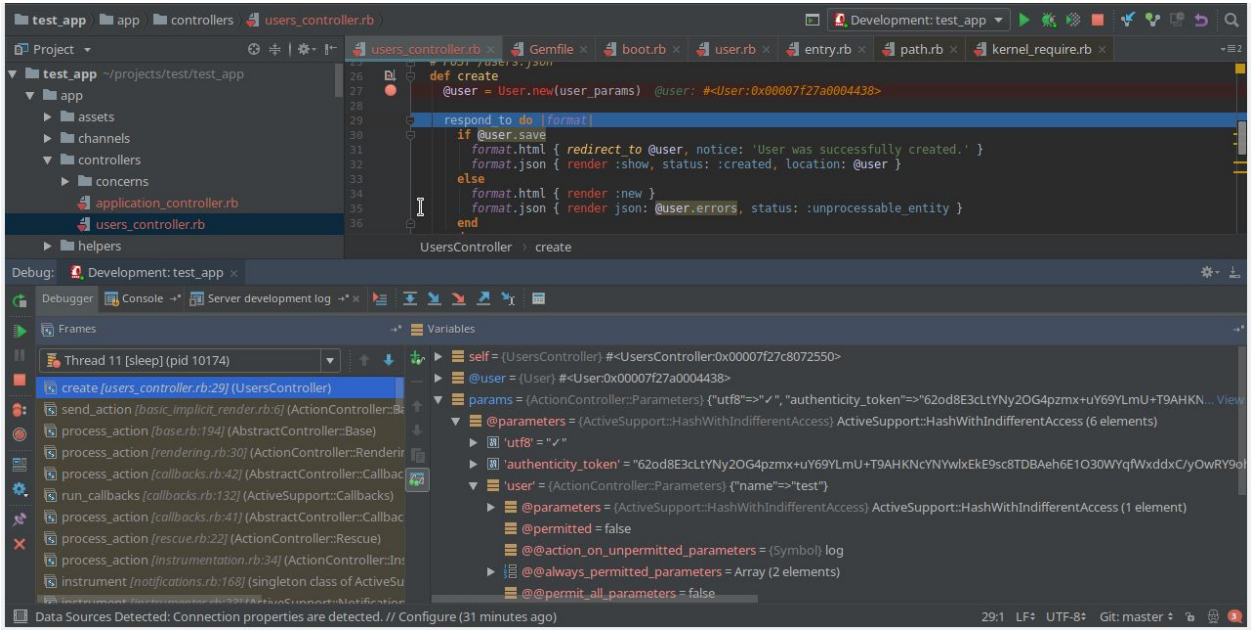
Number of executed queries: 5

IV. Editors

Sublime Text + Ruby Debugger

```
1   class ApplicationController < ActionController::Base
2     protect_from_forgery
3     before_filter :authenticated?, :prepare_for_mobile
4
5     private
6     def current_user
7       @current_user ||= User.find(session[:user_id]) if session[:user_id] && User.exists?(session[:user_id])
8     end
9
10    def authenticated?
11      if !current_user
12        flash[:notice] = "Please sign in"
13        session[:return_point] = request.url
14        redirect_to "/auth/facebook", id: "sign in"
15      end
16    end
17
18    def return_point
19      session[:return_point] ? session[:return_point] : root_path
20    end
21
22    def mobile_device?
23      if session[:mobile_param]
24        session[:mobile_param] == "1"
25      else
26        request.user_agent =~ /Mobile|webOS/
27      end
28    end
```

RubyMine


test_app
app
controllers
users_controller.rb
Development: test_app
Project
users_controller.rb
Gemfile
boot.rb
user.rb
entry.rb
path.rb
kernel_require.rb
test_app ~/projects/test/test_app
app
assets
channels
controllers
concerns
application_controller.rb
users_controller.rb
helpers
def create
@user = User.new(user_params) @user: #<User:0x00007f27a0004438>
respond_to do |format|
if @user.save
format.html { redirect_to @user, notice: 'User was successfully created.' }
format.json { render :show, status: :created, location: @user }
else
format.html { render :new }
format.json { render json: @user.errors, status: :unprocessable_entity }
end
UsersController
create
Debug:
Development: test_app
Debugger
Console
Server development log
Frames
Variables
Thread 11 [sleep] (pid 10174)
create [users_controller.rb:29] (UsersController)
send_action [basic_implicit_render.rb:6] (ActionController::Ba
process_action [base.rb:194] (AbstractController::Base)
process_action [rendering.rb:30] (ActionController::Renderin
process_action [callbacks.rb:42] (AbstractController::Callbac
run_callbacks [callbacks.rb:132] (ActiveSupport::Callbacks)
process_action [callbacks.rb:41] (AbstractController::Callbac
process_action [rescue.rb:22] (ActionController::Rescue)
process_action [instrumentation.rb:34] (ActionController::Ins
instrument [notifications.rb:168] (singleton class of ActiveSu
self = {UsersController} #<UsersController:0x00007f27c8072550>
@user = {User} #<User:0x00007f27a0004438>
params = {ActionController::Parameters} {"utf8"=>"✓", "authenticity_token"=>"62od8E3cLtYNy2OG4pzmx+uY69YLmU+T9AHKN... View
@parameters = {ActiveSupport::HashWithIndifferentAccess} ActiveSupport::HashWithIndifferentAccess (6 elements)
'utf8' = "✓"
'authenticity_token' = "62od8E3cLtYNy2OG4pzmx+uY69YLmU+T9AHKNcYNYwlxEkE9sc8TDBAeh6E1O30WYqfWxddxC/yOwRY9o
'user' = {ActionController::Parameters} {"name"=>"test"}
@parameters = {ActiveSupport::HashWithIndifferentAccess} ActiveSupport::HashWithIndifferentAccess (1 element)
@permitted = false
@@action_on_unpermitted_parameters = {Symbol} log
@@always_permitted_parameters = Array (2 elements)
@@permit_all_parameters = false
Data Sources Detected: Connection properties are detected. // Configure (31 minutes ago)
29:1
LF
UTF-8
Git: master

V. Further readings

- http://valgrind.org/ - application for detecting C-based memory leaks and race conditions.
- Query Reviewer - This Rails plugin not only runs "EXPLAIN" before each of your select queries in development, but provides a small DIV in the rendered output of each page with the summary of warnings for each query that it analyzed.
- MethodProfiler - MethodProfiler collects performance information about the methods in your objects and creates reports to help you identify slow methods. The collected data can be sorted in various ways, converted into an array, or pretty printed as a table.

Links

- https://stackoverflow.com/questions/3955688/how-do-i-debug-ruby-scripts
- https://www.tutorialspoint.com/ruby/ruby_debugger.htm
- http://guides.rubyonrails.org/debugging_rails_applications.html
- http://pryrepl.org/
- https://github.com/pry/pry/wiki
- https://medium.com/@jrmair/dig-deeper-with-pry-introducing-cruby-source-browsing-702cb8358690
- https://shbrt.co/2018/02/22/see-ruby-source.html
- https://medium.com/@jrmair/dig-deeper-with-pry-introducing-cruby-source-browsing-702cb8358690
- https://medium.com/@tiagoparreira/powering-your-ruby-rails-development-with-pry-3d5dbd2a8b80