



# 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
```

```
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]
```

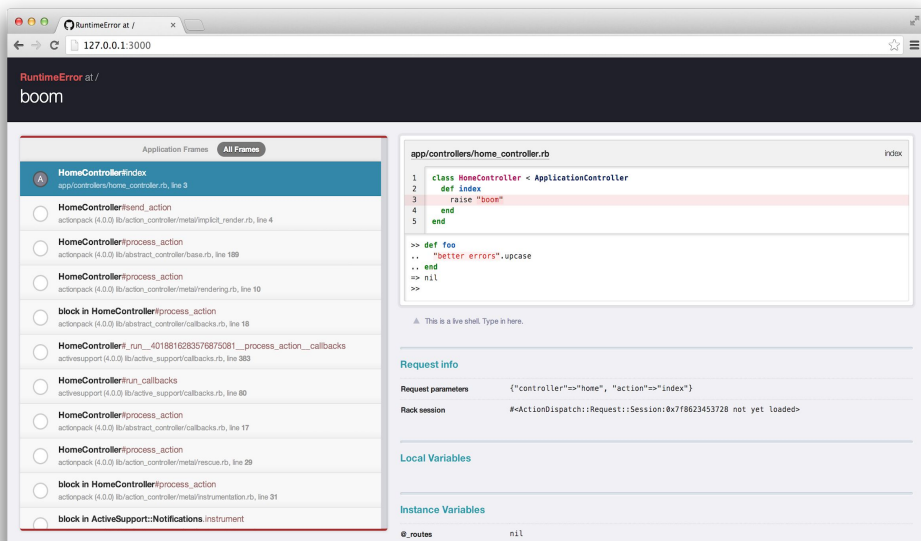
options:

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

The screenshot shows a web browser window with the address bar at `rors.dev/`. The page content is on a yellow background and includes a red heading, a paragraph about the user Dejan, and links to GitHub, Twitter, RailsPanel, and auto\_html. Below the page content is a Chrome DevTools console window with the 'Rails' tab selected. The console displays a table of Rails requests and a detailed view of the active record queries.

## 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!)

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

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

Number of executed queries: 5

## IV. Editors

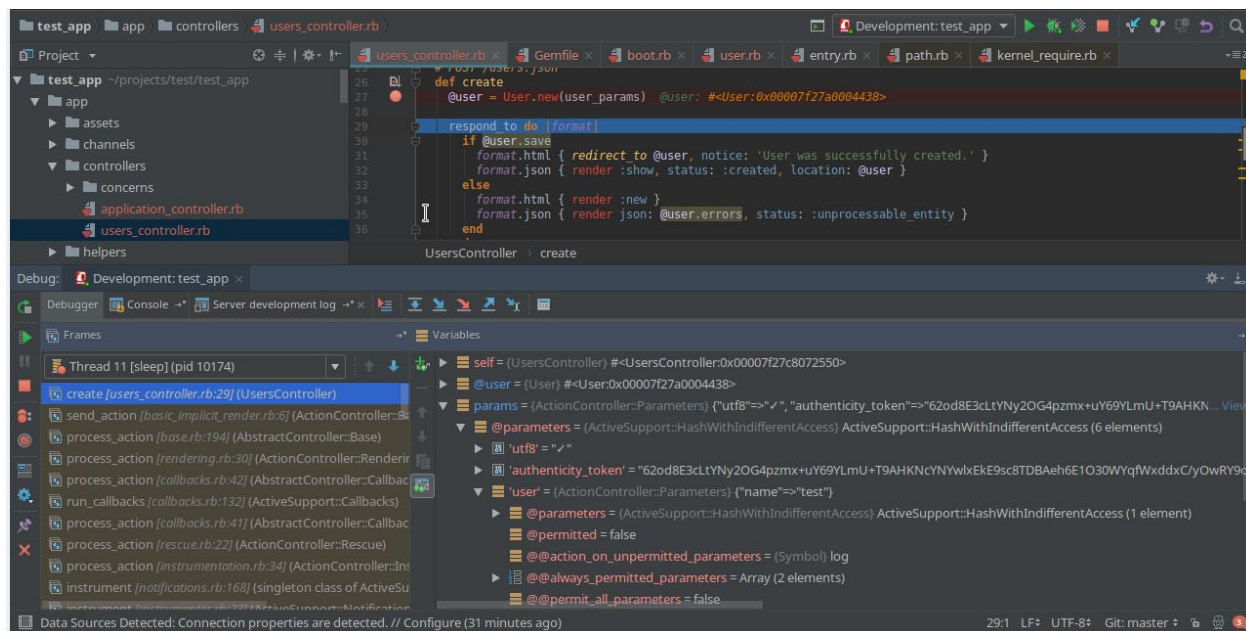




# Sublime Text + Ruby Debugger

```
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
29
```

# RubyMine





# Links

- <https://stackoverflow.com/questions/3955688/how-do-i-debug-ruby-scripts>
- [https://www.tutorialspoint.com/ruby/ruby\\_debugger.htm](https://www.tutorialspoint.com/ruby/ruby_debugger.htm)
- [http://guides.rubyonrails.org/debugging\\_rails\\_applications.html](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>