But why does it work?

Rian McGuire

Melbourne Ruby 2023-08-30

Benefits

- Better understand your system, so it won't surprise you later
- Build confidence in dealing with unfamiliar problems
- Broaden your skills and learn about other parts of the stack
- Write better software

What's in this talk?

- Two live demos
  - Questions and interruptions encouraged
- Tips for exploring unfamilar code

Demo 1 - Magic Methods

```ruby
class DemoController < ApplicationController
  schema :create do
    required(:name).value(:string)
  end

  def create
    if safe_params.failure?
      render status: 400, json: safe_params.errors
    else
      demo = Demo.create!(safe_params.to_h)
      render status: 201, json: demo
    end
  end
end
```

Demo 1 - Magic Methods

```
Rails.application.routes.draw do
  # Define your application routes per the DSL in
  # https://guides.rubyonrails.org/routing.html

  # Defines the root path route ("/")
  # root "articles#index"
  resources :demo
end
```

Demo 1 - Magic Methods

- Use `method(:foo).source_location` to find the code that defined a method
- A debugging tool like `binding.break` can help
- Use `bundle open <gem>` to locate gem code that's installed in your project

Demo 2 - The Ruby VM

- Sometimes the documentation is not enough

Demo 2 - The Ruby VM

```ruby
def do_work(size)
    array = Array.new(size, 0)
    1_000_000.times do
        array.unshift(0)
        array.pop
    end
end
```

Demo 2 - The Ruby VM

- Search is your friend - use known strings like function names or error messages to find code
- If you're reading an unfamilar language or codebase, you don't have to understand everything at once

Conclusion

- The layers below are software, too
- Look for opportunities to improve your understanding
- Be curious

Thank you!