Utilisez Mutant / Mutest pour améliorer vos tests, votre code

Sigilium

@getsigilium

Sigilium

Gestion centralisée des Signatures Emails

@thomasdarde

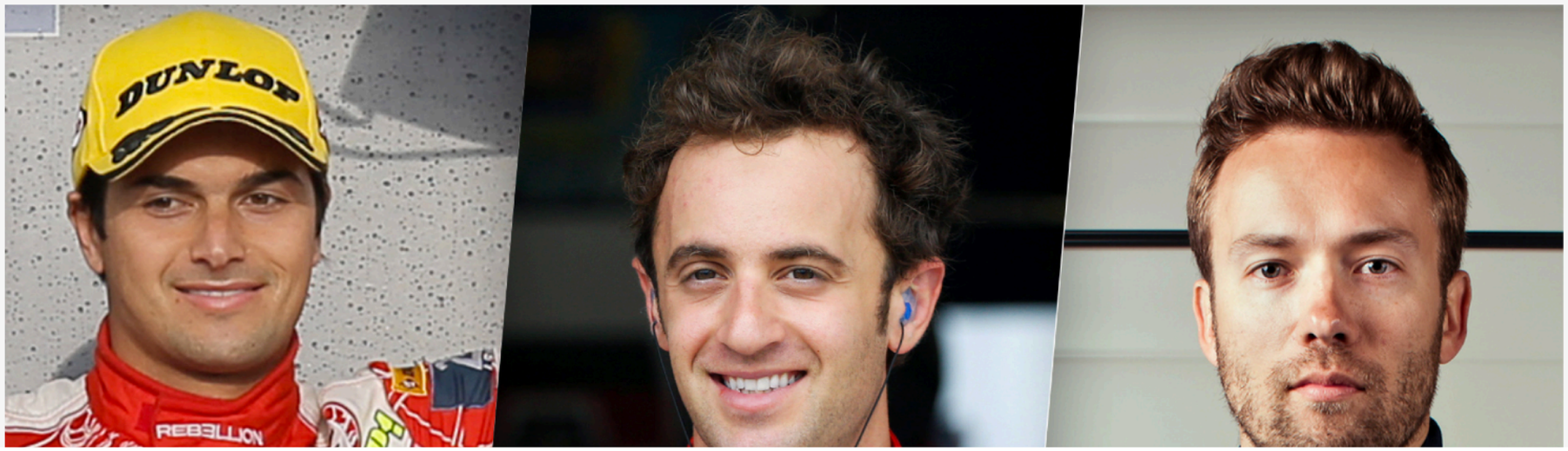
@getsigilium

Parmi nos clients: Motul


Vaillante
MOTUL
Basecamp.com
ROLAND ITEN
13
REBELLION
LIFESTYLE
LEMO
NORTHWIRE
DUNLOP

02/03

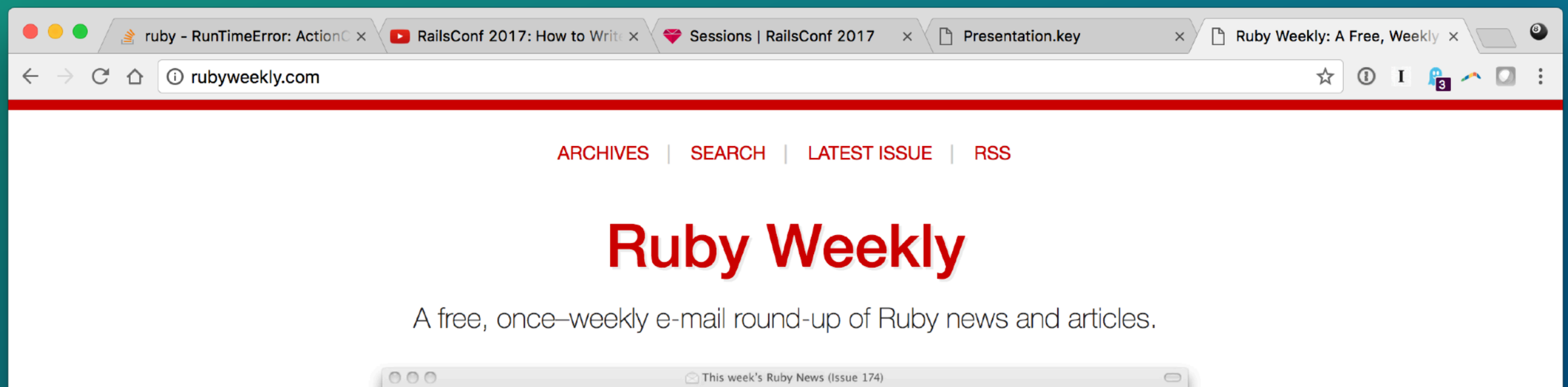
LMP2 #13 DRIVERS



Nicolas Prost, Bruno Senna et .....

Contexte

- Suite à une présentation de John Backus - @backus - john@cognitohq.com à RailsConf 2017
- https://railsconf.com/program#session-160
- Connu grâce à la newsletter http://rubyweekly.com/



Pourquoi Mutest

- Améliorer la couverture de vos tests
- Améliorer vos connaissances en Ruby
- Trouver du code non utilisé
- Simplifier votre code

@getsigilium

Line coverage

Lignes de code traversées par vos tests

/

Nombre total de lignes de code

@getsigilium

Mutation coverage

Combien de code peut être modifié

sans casser vos tests

@getsigilium

A la mano

@getsigilium

```ruby
class JobCandidateSearch
  def initialize(resumes)
    @resumes = resumes
  end

  def recent
    query = @resumes.search("Curious about #rails")
    query.first(2).map { |candidate| "@#{candidate.name}" }
  end
end
```

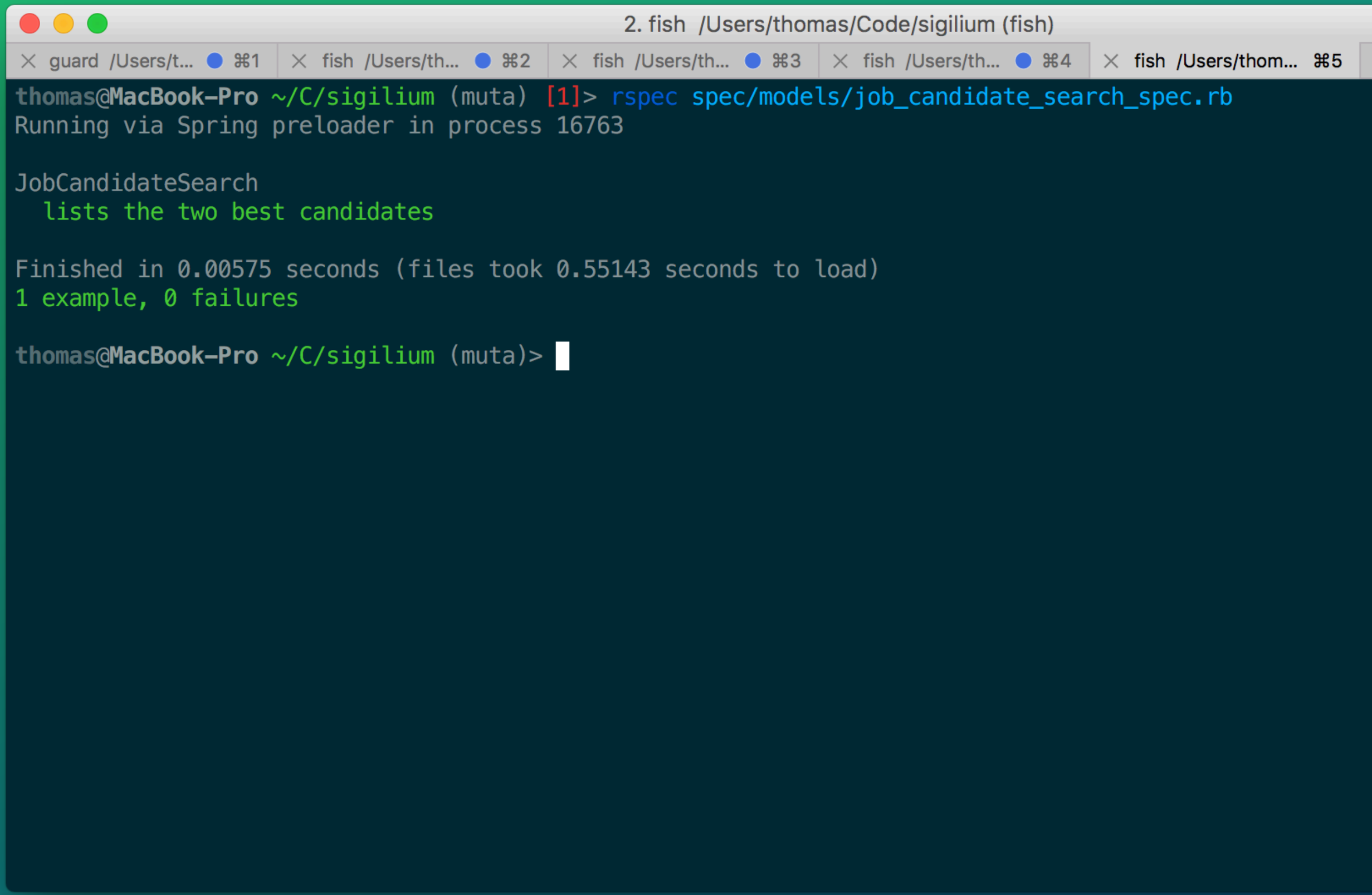
job_candidate_search.rb — sigilium

job_candidate_search.rb

```ruby
class JobCandidateSearch
  def initialize(resumes)
    @resumes = resumes
  end

  def recent
    query = @resumes.search('Curious about #rails')
    query.first(2).map { |candidate| "@#{candidate.name}" }
  end
end
```

job_candidate_search_spec.rb

```ruby
RSpec.describe JobCandidateSearch do
  it 'lists the two best candidates' do
    resumes = [double(name: 'Basile'), double(name: 'Nathan')]
    candidates = JobCandidateSearch.new(double(search: resumes))

    expect(candidates.recent).to eql(%w[@Basile @Nathan])
  end
end

```


2. fish /Users/thomas/Code/sigilium (fish)
guard /Users/t... ⌘1
fish /Users/th... ⌘2
fish /Users/th... ⌘3
fish /Users/th... ⌘4
fish /Users/thom... ⌘5
thomas@MacBook-Pro ~/C/sigilium (muta) [1]> rspec spec/models/job_candidate_search_spec.rb
Running via Spring preloader in process 16763
JobCandidateSearch
lists the two best candidates
Finished in 0.00575 seconds (files took 0.55143 seconds to load)
1 example, 0 failures
thomas@MacBook-Pro ~/C/sigilium (muta)>

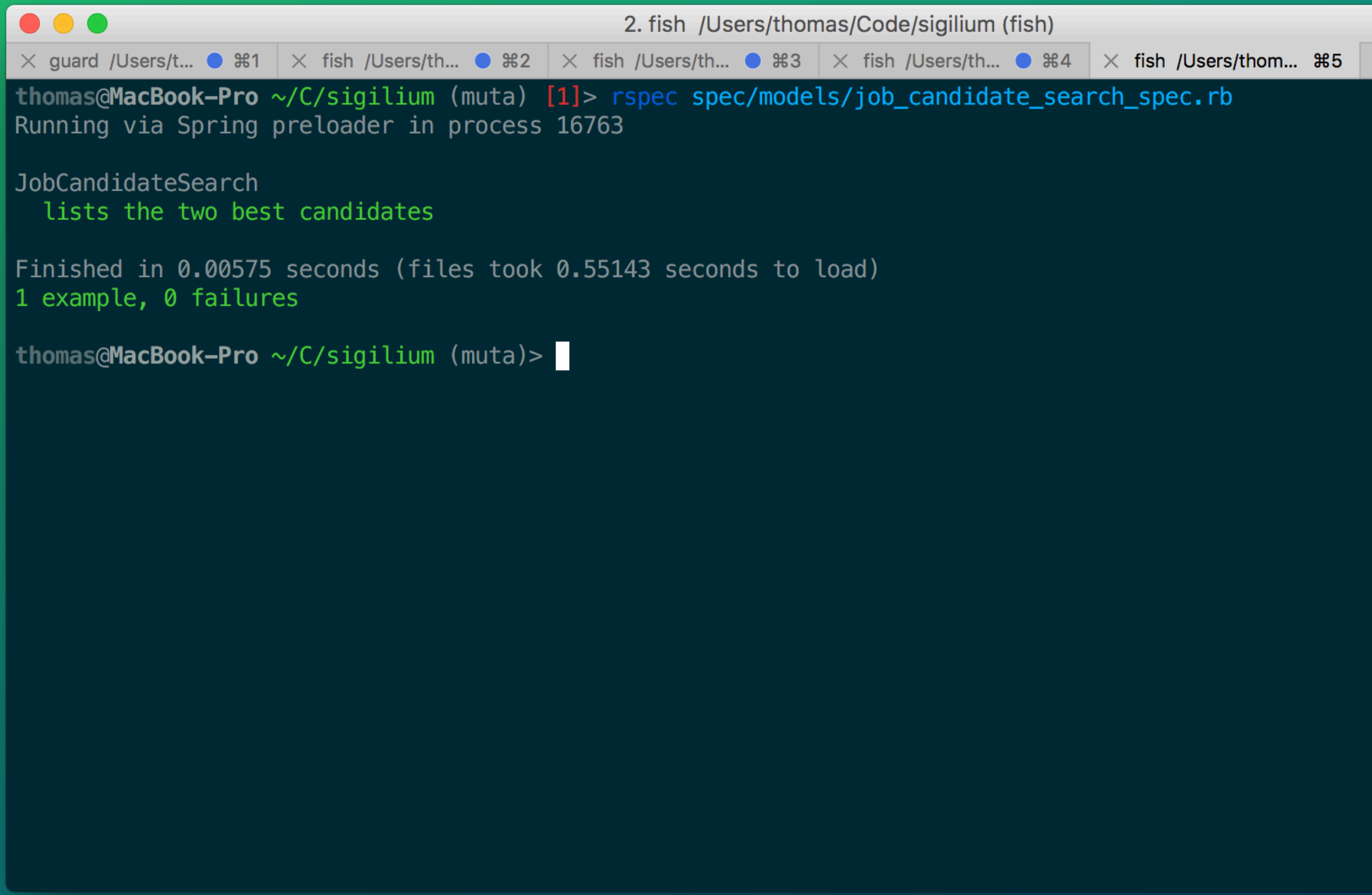
job_candidate_search.rb — sigilium

job_candidate_search.rb

```ruby
class JobCandidateSearch
  def initialize(resumes)
    @resumes = resumes
  end

  def recent
    query = @resumes.search('Curious about #php')
    query.first(2).map { |candidate| "@#{candidate.name}" }
  end
end

```

job_candidate_search_spec.rb

```ruby
RSpec.describe JobCandidateSearch do
  it 'lists the two best candidates' do
    resumes = [double(name: 'Basile'), double(name: 'Nathan')]
    candidates = JobCandidateSearch.new(double(search: resumes))

    expect(candidates.recent).to eql(%w[@Basile @Nathan])
  end
end


```


2. fish /Users/thomas/Code/sigilium (fish)
guard /Users/t... ⌘1
fish /Users/th... ⌘2
fish /Users/th... ⌘3
fish /Users/th... ⌘4
fish /Users/thom... ⌘5
thomas@MacBook-Pro ~/C/sigilium (muta) [1]> rspec spec/models/job_candidate_search_spec.rb
Running via Spring preloader in process 16763
JobCandidateSearch
lists the two best candidates
Finished in 0.00575 seconds (files took 0.55143 seconds to load)
1 example, 0 failures
thomas@MacBook-Pro ~/C/sigilium (muta)>

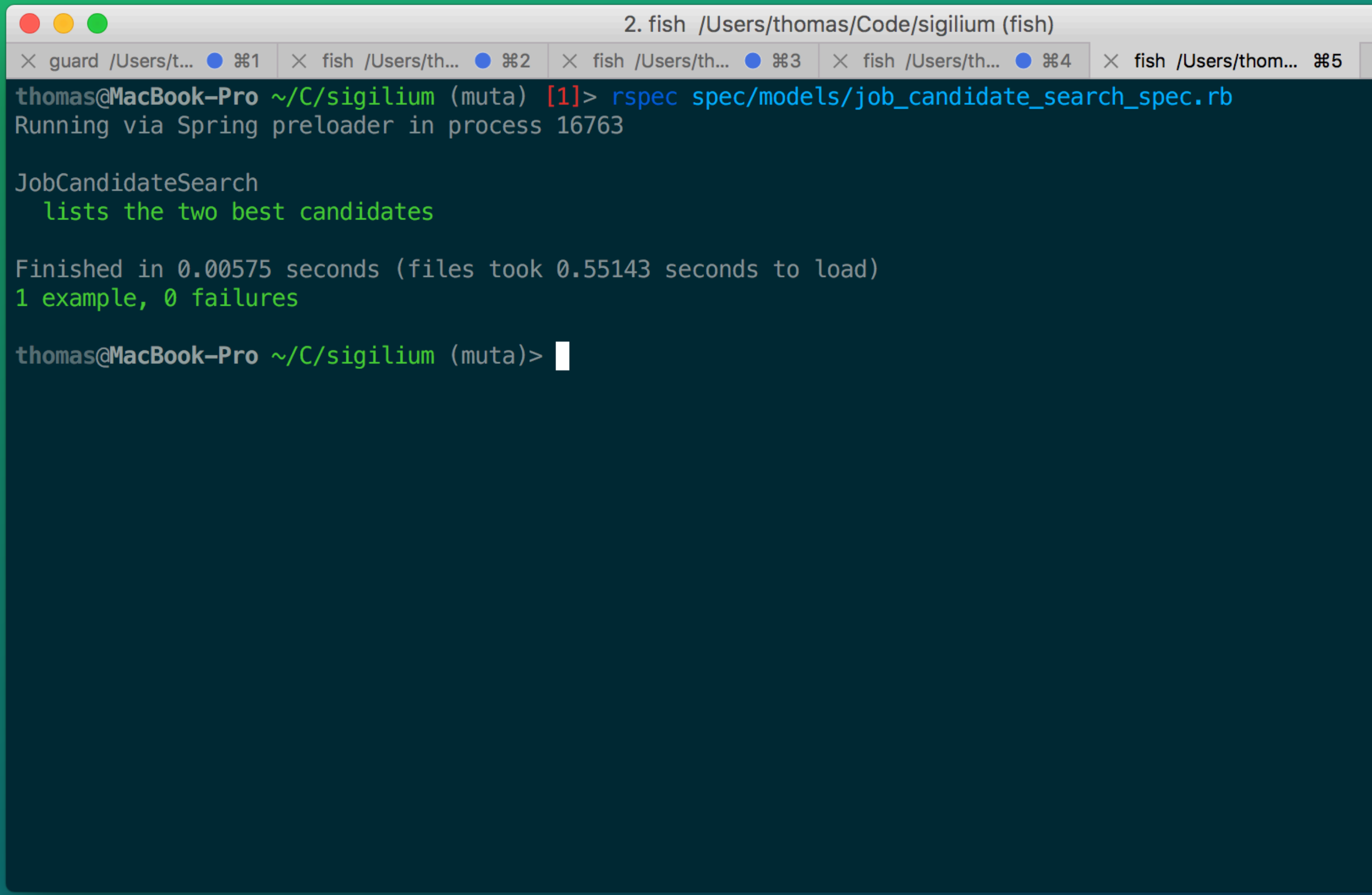
job_candidate_search.rb — sigilium

job_candidate_search.rb

```ruby
class JobCandidateSearch
  def initialize(resumes)
    @resumes = resumes
  end

  def recent
    query = @resumes.search(nil)
    query.first(2).map { |candidate| "@#{candidate.name}" }
  end
end

```

job_candidate_search_spec.rb

```ruby
RSpec.describe JobCandidateSearch do
  it 'lists the two best candidates' do
    resumes = [double(name: 'Basile'), double(name: 'Nathan')]
    candidates = JobCandidateSearch.new(double(search: resumes))

    expect(candidates.recent).to eql(%w[@Basile @Nathan])
  end
end

```


2. fish /Users/thomas/Code/sigilium (fish)
guard /Users/t... ⌘1
fish /Users/th... ⌘2
fish /Users/th... ⌘3
fish /Users/th... ⌘4
fish /Users/thom... ⌘5
thomas@MacBook-Pro ~/C/sigilium (muta) [1]> rspec spec/models/job_candidate_search_spec.rb
Running via Spring preloader in process 16763
JobCandidateSearch
lists the two best candidates
Finished in 0.00575 seconds (files took 0.55143 seconds to load)
1 example, 0 failures
thomas@MacBook-Pro ~/C/sigilium (muta)>

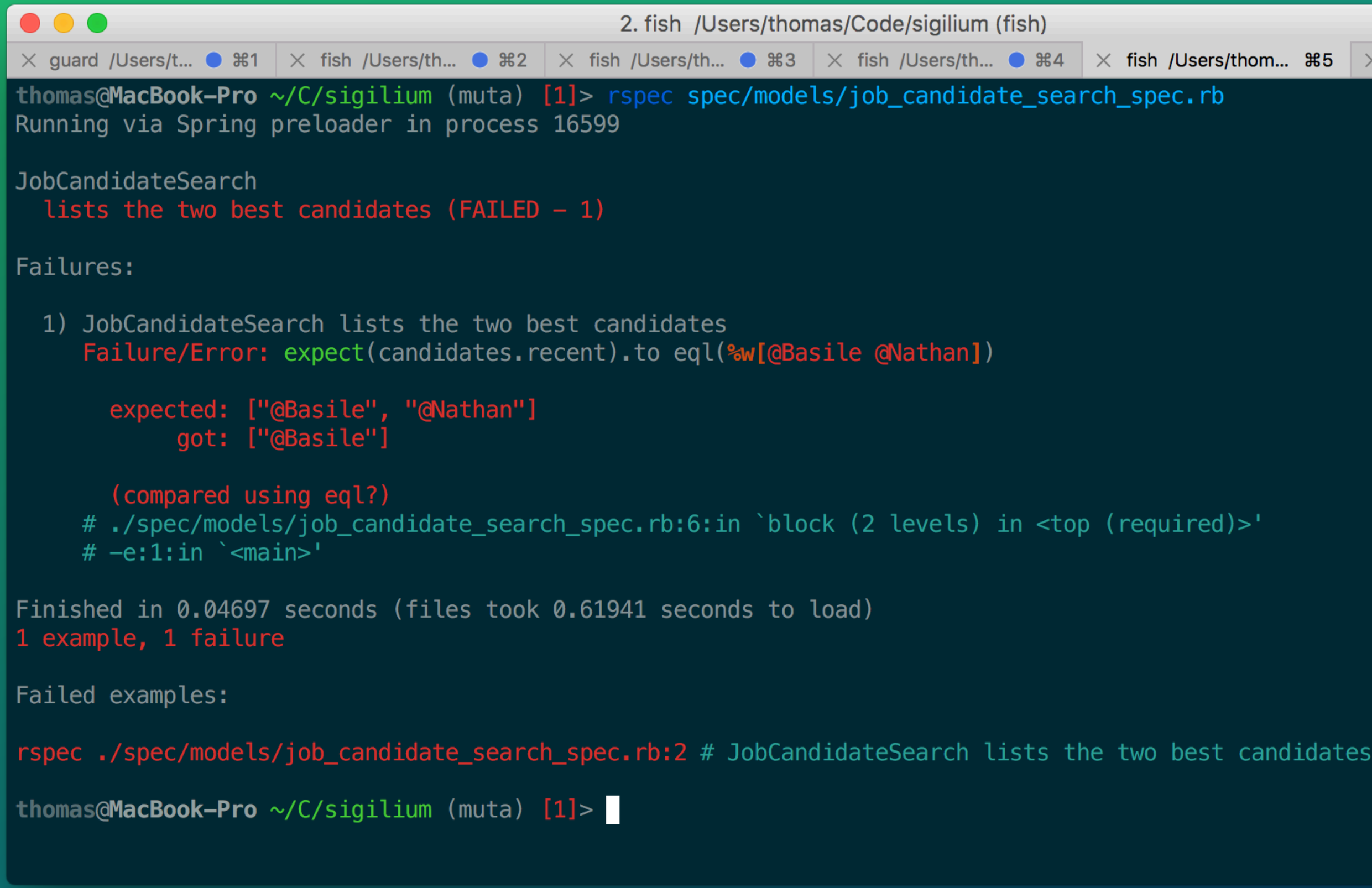
job_candidate_search.rb — sigilium

job_candidate_search.rb

```ruby
class JobCandidateSearch
  def initialize(resumes)
    @resumes = resumes
  end

  def recent
    query = @resumes.search(nil)
    query.first(1).map { |candidate| "@#{candidate.name}" }
  end
end

```

job_candidate_search_spec.rb

```ruby
RSpec.describe JobCandidateSearch do
  it 'lists the two best candidates' do
    resumes = [double(name: 'Basile'), double(name: 'Nathan')]
    candidates = JobCandidateSearch.new(double(search: resumes))

    expect(candidates.recent).to eql(%w[@Basile @Nathan])
  end
end

```


2. fish /Users/thomas/Code/sigilium (fish)
guard /Users/t... ⌘1
fish /Users/th... ⌘2
fish /Users/th... ⌘3
fish /Users/th... ⌘4
fish /Users/thom... ⌘5
thomas@MacBook-Pro ~/C/sigilium (muta) [1]> rspec spec/models/job_candidate_search_spec.rb
Running via Spring preloader in process 16599
JobCandidateSearch
lists the two best candidates (FAILED - 1)
Failures:
1) JobCandidateSearch lists the two best candidates
Failure/Error: expect(candidates.recent).to eql(%w[@Basile @Nathan])
expected: ["@Basile", "@Nathan"]
got: ["@Basile"]
(compared using eql?)
./spec/models/job_candidate_search_spec.rb:6:in `block (2 levels) in <top (required)>'
-e:1:in `<main>'
Finished in 0.04697 seconds (files took 0.61941 seconds to load)
1 example, 1 failure
Failed examples:
rspec ./spec/models/job_candidate_search_spec.rb:2 # JobCandidateSearch lists the two best candidates
thomas@MacBook-Pro ~/C/sigilium (muta) [1]>

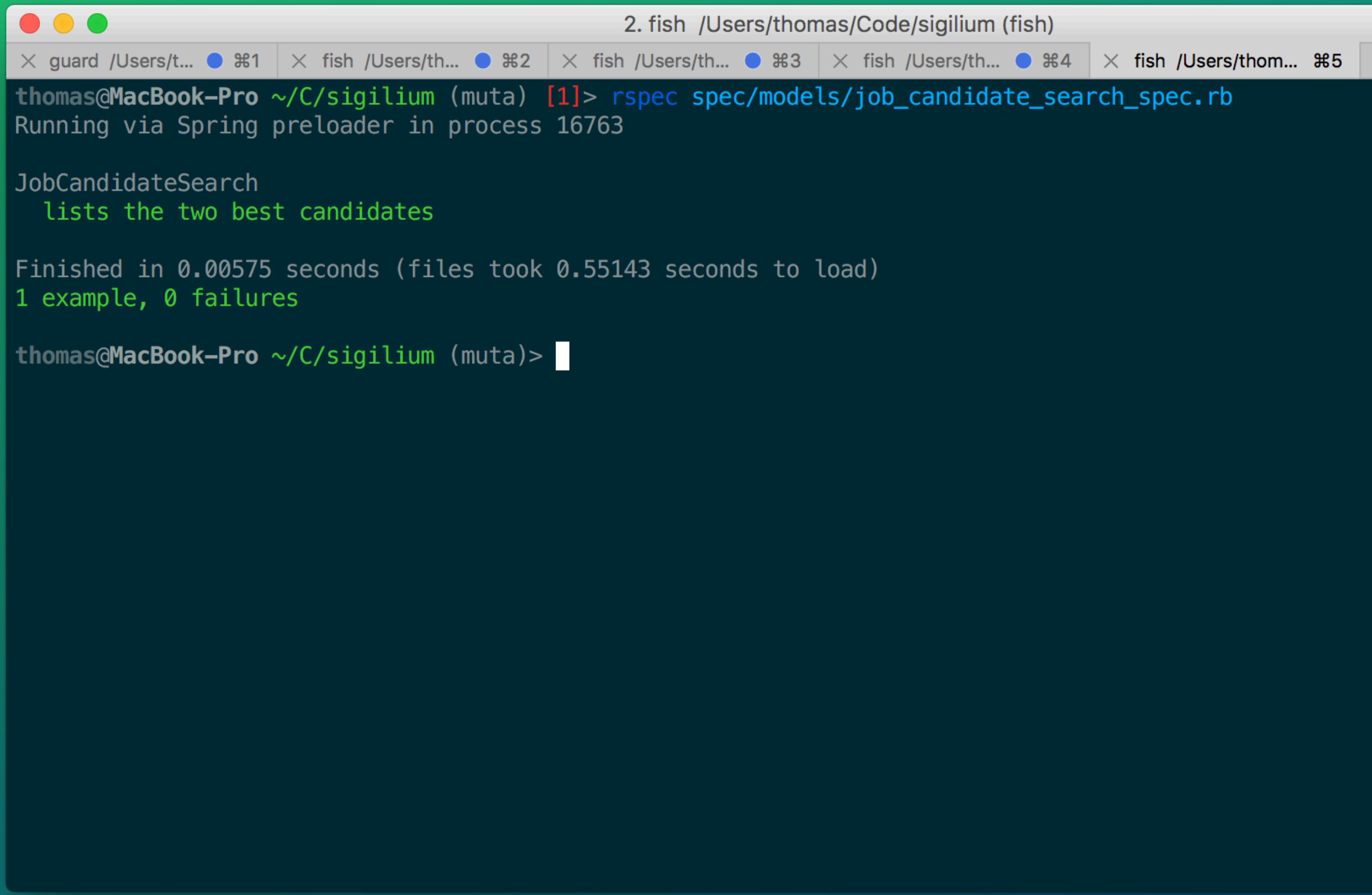
job_candidate_search.rb — sigilium

job_candidate_search.rb

```ruby
class JobCandidateSearch
  def initialize(resumes)
    @resumes = resumes
  end

  def recent
    query = @resumes.search("Curious about #rails")
    query.last(2).map { |candidate| "@#{candidate.name}" }
  end
end

```

job_candidate_search_spec.rb

```ruby
RSpec.describe JobCandidateSearch do
  it 'lists the two best candidates' do
    resumes = [double(name: 'Basile'), double(name: 'Nathan')]
    candidates = JobCandidateSearch.new(double(search: resumes))

    expect(candidates.recent).to eql(%w[@Basile @Nathan])
  end
end


```


2. fish /Users/thomas/Code/sigilium (fish)
guard /Users/t... ⌘1
fish /Users/th... ⌘2
fish /Users/th... ⌘3
fish /Users/th... ⌘4
fish /Users/thom... ⌘5
thomas@MacBook-Pro ~/C/sigilium (muta) [1]> rspec spec/models/job_candidate_search_spec.rb
Running via Spring preloader in process 16763
JobCandidateSearch
lists the two best candidates
Finished in 0.00575 seconds (files took 0.55143 seconds to load)
1 example, 0 failures
thomas@MacBook-Pro ~/C/sigilium (muta)>

job_candidate_search_spec.rb — sigilium

job_candidate_search.rb | job_candidate_search_spec.rb

```ruby
RSpec.describe JobCandidateSearch do

  it 'lists the two most recent gluttonous tweeters' do
    resumes = [
      double(author: 'Basile'),
      double(author: 'Nathan'),
      double(author: 'Gaëlle')
    ]

    search_engine = double('SearchEngine')
    candidates = JobCandidateSearch.new(search_engine)

    allow(search_engine)
      .to receive(:search)
      .with("Curious about #rails")
      .and_return(resumes)

    expect(candidates.recent).to eql(%w[@Basile @Nathan])
end


```

Limitations

- C’est long… et fastidieux
- Saurez-vous être plus rusé que vous même ?

Utilisation de mutest

```
group :development, :test do
  gem "parallel_tests"
  gem "mutest-rspec"
end
```

```
bundle exec mutest --include lib --use rspec ‘JobCandidateSearch'
```

```
mutest --include lib --require my_project --use rspec --since HEAD~1 'MyProject*'
```

Exemple Live

Autres exemples

Example #1: Une API interne

```ruby
class UsersController < ApplicationController
  def show
    render json: User.find(params[:id].to_i)
  rescue User::RecordNotFound => error
    render json: { error: error.to_s }
  end
end
```

```ruby
it 'returns a user when given a valid id' do
  expect(get(:show, id: 1)).to eq(id: 1, name: 'John')
end

it 'renders JSON error when given an invalid id' do
  expect(get(:show, id: 0))
    .to eq(error: "Could not find User with 'id'=0")
end
```

@backus

```
1   def show
2 -   render json: User.find(params[:id].to_i)
3 +   render json: User.find(Integer(params[:id]))
4   rescue User::RecordNotFound => error
5     render json: { error: error.to_s }
6   end
```

@backus

```
1   def show
2 -   render json: User.find(params[:id].to_i)
3 +   render json: User.find(params.fetch(:id).to_i)
4   rescue User::RecordNotFound => error
5     render json: { error: error.to_s }
6   end
```

@backus

```
1 def show
2 -   render json: User.find(params[:id].to_i)
3 +   render json: User.find(Integer(params.fetch(:id)))
4 rescue User::RecordNotFound => error
5   render json: { error: error.to_s }
6 end
```

@backus

```ruby
class UsersController < ApplicationController
  def created_after
    after = Date.parse(params[:after])
    render json: User.recent(after)
  end
end
```

@backus

```
1 def created_after
2 - after = Date.parse(params[:after])
3 + after = Date.iso8601(params[:after])
4     render json: User.recent(after)
5 end
```

@backus

```
1 def created_after
2 - after = Date.parse(params[:after])
3 + after = Date.iso8601(params[:after])
4   render json: User.recent(after)
5 end
```

Date.iso8601	Date.parse
“2017-05-01"	“2017-05-01"
	"H29.05.01"
	"Tue May 01 00:00:00 2017"
	"Tue, 01 May 2017 00:00:00 +0000"
	"Tue, 01 May 2017 00:00:00 GMT"
	"May"
	"I may be complete garbage"

@backus

Example #2: Des expressions régulières

@backus

Example #3: Apprenez en sur votre client HTTP

@backus

```ruby
usernames.select do |username|
  username =~ /^(John|Alain).+$/
end
```

@backus

```
1 usernames.select do |username|
2 -   username =~ /^(John|Alain).+$/
3 +   username =~ /\A(John|Alain).+$/
4 end
```

```
1 usernames.select do |username|
2 -   username =~ /^(John|Alain).+$/
3 +   username =~ /^(John|Alain).+\z/
4 end
```

@backus

```
1 usernames.select do |username|
2 -   username =~ /^(John|Alain).+$/
3 +   username =~ /^(Alain).+$/
4 end
```

```
1 usernames.select do |username|
2 -   username =~ /^(John|Alain).+$/
3 +   username =~ /^(John).+$/
4 end
```

@backus

```
1 usernames.select do |username|
2 -   username =~ /^(John|Alain).+$/
3 +   username =~ /^(?:John|Alain).+$/
4 end
```

@backus

```
1 usernames.select do |username|
2 -   username =~ /^(John|Alain).+$/
3 +   username.match?(/^(John|Alain).+$/)
4 end
```

@backus

```
1 usernames.select do |username|
2 -   username =~ /^(John|Alain).+$/
3 +   username.match?(/\A(?:John|Alain).+\z/)
4 end
```

@backus

```ruby
def stars_for(repo)
  url  = "https://api.github.com/repos/#{repo}"
  data = HTTParty.get(url).to_h

  data['stargazers_count']
end
```

@backus

```
1 def stars_for(repo)
2   url  = "https://api.github.com/repos/
#{repo}"
3 - data = HTTParty.get(url).to_h
4 + data = HTTParty.get(url)
5
6   data['stargazers_count']
7 end
```

@backus

Une meilleure couverture de test

@backus

```ruby
1 def can_buy_alcohol?(age)
2   age >= 21
3 end
```

@backus

```
1 def can_buy_alcohol?(age)
2 -   age >= 21
3 +   age >= 22
4 end
```

```
1 def can_buy_alcohol?(age)
2 -   age >= 21
3 +   age >= 20
4 end
```

@backus

Trouvez du code mort !

@backus

```ruby
class User < ActiveRecord::Base
  def self.find_by_name(name)
    find_by(name: name)
  end
end
```

@backus

```
1 class User < ActiveRecord::Base
2   def find_by_name(name)
3 -     find_by(name: name)
4 +     super
5   end
6 end
```

@backus

```
1 class PostsController <
ApplicationController
2   private
3
4   def authorized?(user = current_user)
5     # ...
6   end
7 end
```

@backus

```
 1 class PostsController <
ApplicationController
 2   def edit
 3     return unauthorized unless authorized?
(current_user)
 4   end
 5
 6   def admin_edit
 7     return unauthorized unless authorized?
(current_user)
 8   end
 9
10   private
11
12   def authorized?(user = current_user)
13     # ...
14   end
15 end
```

@backus

```
    1 class PostsController <
ApplicationController
    2 -   def authorized?(user = current_user)
    3 +   def authorized?(user)
    4       # ...
    5     end
    6 end
```

@backus

```
 1 class PostsController <
ApplicationController
 2   def edit
 3     return unauthorized unless authorized?
 4   end
 5
 6   def admin_edit
 7     return unauthorized unless authorized?
 8   end
 9
10   private
11
12   def authorized?(user = current_user)
13     # ...
14   end
15 end
```

@backus

```
    1 class PostsController <
ApplicationController
    2   def authorized?(user = current_user)
    3 +     user = current_user
    4       # ...
    5   end
    6 end
```

@backus

```ruby
module MyApp
  class User
    def posted?
      ::MyApp::Post.exists?(user_id: id)
    end
  end
end
```

@backus

```
1 module MyApp
2   class User
3     def posted?
4 -       ::MyApp::Post.exists?(user_id: id)
5 +       Post.exists?(user_id: id)
6     end
7   end
8 end
```

@backus

```
1 class PostsController <
ApplicationController
2   def show
3     render json: Post.find(params[:id]),
status: :ok
4   end
5 end
```

@backus

```
    1 class PostsController <
ApplicationController
    2   def show
    3 -     render json: Post.find(params[:id]),
status: :ok
    4 +     render json: Post.find(params[:id])
    5   end
    6 end
```

@backus

Simplifier du code

@backus

```ruby
class UserDecorator
  attr_reader :user

  def initialize(user)
    @user = user
  end

  def greeting
    "Welcome, #{@user.name}!"
  end
end
```

@backus

```
1  class UserDecorator
2    attr_reader :user
3
4    def initialize(user)
5      @user = user
6    end
7
8    def greeting
9  -       "Welcome, #{@user.name}!"
10 +       "Welcome, #{user.name}!"
11   end
12 end
```

@backus

```ruby
def greeting
  "Welcome, #{@usre.name}!"
end
```

```
NoMethodError: undefined method `name' for
               nil:NilClass
```

@backus

```ruby
1 def greeting
2   "Welcome, #{usre.name}!"
3 end
```

```
NameError: undefined local variable or method `usre`
```

```ruby
def expand_home(path)
  path.gsub(%r{\A~/}, ENV['HOME'])
end
```

```
1 def expand_home(path)
2 -  path.gsub(%r{\A~/}, ENV['HOME'])
3 +  path.sub(%r{\A~/}, ENV['HOME'])
4 end
```

```ruby
require 'logger'

logger = Logger.new($stdout)

logger.formatter =
  Proc.new do |severity, datetime, name, msg|
    "[#{severity}] #{datetime} #{name} -- #{msg}\n"
  end
```

```
1 require 'logger'
2
3 logger = Logger.new($stdout)
4
5 logger.formatter =
6 -   Proc.new do |severity, datetime, name,
msg|
7 +   lambda do |severity, datetime, name,
msg|
8       "[#{severity}] #{datetime} #{name} --
#{msg}\n"
9     end
```

```ruby
formatter = proc { |a, b| [a, b].inspect }
```

```ruby
formatter = proc { |a, b| [a, b].inspect }

formatter.call()          # => "[nil, nil]"
formatter.call(1)         # => "[1, nil]"
formatter.call(1, 2)      # => "[1, 2]"
formatter.call(1, 2, 3)   # => "[1, 2]"

formatter.call([])        # => "[nil, nil]"
formatter.call([1])       # => "[1, nil]"
formatter.call([1, 2])    # => "[1, 2]"
formatter.call([1, 2, 3]) # => "[1, 2]"
```

```ruby
formatter = lambda { |a, b| [a, b].inspect }

formatter.call()          # => ArgumentError
formatter.call(1)         # => ArgumentError
formatter.call(1, 2)      # => "[1, 2]"
formatter.call(1, 2, 3)   # => ArgumentError
formatter.call([])        # => ArgumentError
formatter.call([1])       # => ArgumentError
formatter.call([1, 2])    # => ArgumentError
formatter.call([1, 2, 3]) # => ArgumentError
```

AVEC MUTEST

Devenez un meilleur développeur !

Apprenez en sur RUBY

Apprenez en sur votre base de Code

Réduisez les bugs et régressions

Simplifiez du code !

@thomasdarde

@getsigilium

Sigilium

MERCI

@thomasdarde

@getsigilium

Dans la pratique

/^(John|Alain).+$/

/\A(John|Alain).+$/

/^(John|Alain).+\z/

/^(Alain).+$/

/^(?:John|Alain).+$/

/^(John).+$/

```
$ mutest --use rspec --since master
```

```
$ mutest                \
   --use rspec          \
  --since master \
  'YourApp::User#validate_email'
```