

# Let the Right One In

A gentle intro to devise

Sergio Arbeo



Serabe



Serabe

serabe.com/cursos/  
devise-crash-  
course.tar.gz

What is  
authentication?

What is authorisation?

Devise deals with  
authentication only

# How can a user log in in a system?

- Username + password
- OAuth 2
- Multi-factor authentication
- Biometry

# How would you implement password-based authentication?

- Hash password (slow hashing)
- Add salting
- Never log the password in any way.

# Install Devise

*You already did! (almost)*



rg devise:install

rg devise User

serabe@Weisz:~/programming/cc/cursos/devise\_crash\_course/devise\_blog (\*)

> rake routes

	Prefix	Verb	URI Pattern	Controller#Action
	new_user_session	GET	/users/sign_in(:format)	devise/sessions#new
	user_session	POST	/users/sign_in(:format)	devise/sessions#create
	destroy_user_session	DELETE	/users/sign_out(:format)	devise/sessions#destroy
	user_password	POST	/users/password(:format)	devise/passwords#create
	new_user_password	GET	/users/password/new(:format)	devise/passwords#new
	edit_user_password	GET	/users/password/edit(:format)	devise/passwords#edit
		PATCH	/users/password(:format)	devise/passwords#update
		PUT	/users/password(:format)	devise/passwords#update
cancel_user_registration		GET	/users/cancel(:format)	devise/registrations#cancel
	user_registration	POST	/users(:format)	devise/registrations#create
	new_user_registration	GET	/users/sign_up(:format)	devise/registrations#new
	edit_user_registration	GET	/users/edit(:format)	devise/registrations#edit
		PATCH	/users(:format)	devise/registrations#update
		PUT	/users(:format)	devise/registrations#update
		DELETE	/users(:format)	devise/registrations#destroy
	posts	GET	/posts(:format)	posts#index
		POST	/posts(:format)	posts#create
	new_post	GET	/posts/new(:format)	posts#new
	edit_post	GET	/posts/:id/edit(:format)	posts#edit
	post	GET	/posts/:id(:format)	posts#show
		PATCH	/posts/:id(:format)	posts#update
		PUT	/posts/:id(:format)	posts#update
		DELETE	/posts/:id(:format)	posts#destroy
	root	GET	/	posts#index

serabe@Weisz:~/programming/cc/cursos/devise\_crash\_course/devise\_blog (\*)

> rake routes

	Prefix	Verb	URI Pattern	Controller#Action
	new_user_session	GET	/users/sign_in(:format)	devise/sessions#new
	user_session	POST	/users/sign_in(:format)	devise/sessions#create
	destroy_user_session	DELETE	/users/sign_out(:format)	devise/sessions#destroy
	user_password	POST	/users/password(:format)	devise/passwords#create
	new_user_password	GET	/users/password/new(:format)	devise/passwords#new
	edit_user_password	GET	/users/password/edit(:format)	devise/passwords#edit
		PATCH	/users/password(:format)	devise/passwords#update
		PUT	/users/password(:format)	devise/passwords#update
cancel_user_registration	GET		/users/cancel(:format)	devise/registrations#cancel
user_registration	POST		/users(:format)	devise/registrations#create
new_user_registration	GET		/users/sign_up(:format)	devise/registrations#new
edit_user_registration	GET		/users/edit(:format)	devise/registrations#edit
		PATCH	/users(:format)	devise/registrations#update
		PUT	/users(:format)	devise/registrations#update
		DELETE	/users(:format)	devise/registrations#destroy
	posts	GET	/posts(:format)	posts#index
		POST	/posts(:format)	posts#create
	new_post	GET	/posts/new(:format)	posts#new
edit_post	GET		/posts/:id/edit(:format)	posts#edit
post	GET		/posts/:id(:format)	posts#show
		PATCH	/posts/:id(:format)	posts#update
		PUT	/posts/:id(:format)	posts#update
		DELETE	/posts/:id(:format)	posts#destroy
root	GET		/	posts#index



serabe@Weisz:~/programming/cc/cursos/devise\_crash\_course/devise\_blog (\*)

> rake routes

	Prefix	Verb	URI Pattern	Controller#Action
	new_user_session	GET	/users/sign_in(:format)	devise/sessions#new
	user_session	POST	/users/sign_in(:format)	devise/sessions#create
	destroy_user_session	DELETE	/users/sign_out(:format)	devise/sessions#destroy
	user_password	POST	/users/password(:format)	devise/passwords#create
	new_user_password	GET	/users/password/new(:format)	devise/passwords#new
	edit_user_password	GET	/users/password/edit(:format)	devise/passwords#edit
		PATCH	/users/password(:format)	devise/passwords#update
		PUT	/users/password(:format)	devise/passwords#update
	cancel_user_registration	GET	/users/cancel(:format)	devise/registrations#cancel
	user_registration	POST	/users(:format)	devise/registrations#create
	new_user_registration	GET	/users/sign_up(:format)	devise/registrations#new
	edit_user_registration	GET	/users/edit(:format)	devise/registrations#edit
		PATCH	/users(:format)	devise/registrations#update
		PUT	/users(:format)	devise/registrations#update
		DELETE	/users(:format)	devise/registrations#destroy
	posts	GET	/posts(:format)	posts#index
		POST	/posts(:format)	posts#create
	new_post	GET	/posts/new(:format)	posts#new
	edit_post	GET	/posts/:id/edit(:format)	posts#edit
	post	GET	/posts/:id(:format)	posts#show
		PATCH	/posts/:id(:format)	posts#update
		PUT	/posts/:id(:format)	posts#update
		DELETE	/posts/:id(:format)	posts#destroy
	root	GET	/	posts#index

serabe@Weisz:~/programming/cc/cursos/devise\_crash\_course/devise\_blog (\*)

> rake routes

	Prefix	Verb	URI Pattern	Controller#Action
	new_user_session	GET	/users/sign_in(:format)	devise/sessions#new
	user_session	POST	/users/sign_in(:format)	devise/sessions#create
	destroy_user_session	DELETE	/users/sign_out(:format)	devise/sessions#destroy
	user_password	POST	/users/password(:format)	devise/passwords#create
	new_user_password	GET	/users/password/new(:format)	devise/passwords#new
	edit_user_password	GET	/users/password/edit(:format)	devise/passwords#edit
		PATCH	/users/password(:format)	devise/passwords#update
		PUT	/users/password(:format)	devise/passwords#update
cancel_user_registration	GET		/users/cancel(:format)	devise/registrations#cancel
user_registration	POST		/users(:format)	devise/registrations#create
new_user_registration	GET		/users/sign_up(:format)	devise/registrations#new
edit_user_registration	GET		/users/edit(:format)	devise/registrations#edit
		PATCH	/users(:format)	devise/registrations#update
		PUT	/users(:format)	devise/registrations#update
		DELETE	/users(:format)	devise/registrations#destroy
	posts	GET	/posts(:format)	posts#index
		POST	/posts(:format)	posts#create
	new_post	GET	/posts/new(:format)	posts#new
	edit_post	GET	/posts/:id/edit(:format)	posts#edit
	post	GET	/posts/:id(:format)	posts#show
		PATCH	/posts/:id(:format)	posts#update
		PUT	/posts/:id(:format)	posts#update
		DELETE	/posts/:id(:format)	posts#destroy
	root	GET	/	posts#index



```
serabe@Weisz:~/programming/cc/cursos/devise_crash_course/devise_blog (*)
```

```
> rake routes
```

	Prefix	Verb	URI Pattern	Controller#Action
	new_user_session	GET	/users/sign_in(:format)	devise/sessions#new
	user_session	POST	/users/sign_in(:format)	devise/sessions#create
	destroy_user_session	DELETE	/users/sign_out(:format)	devise/sessions#destroy
	user_password	POST	/users/password(:format)	devise/passwords#create
	new_user_password	GET	/users/password/new(:format)	devise/passwords#new
	edit_user_password	GET	/users/password/edit(:format)	devise/passwords#edit
		PATCH	/users/password(:format)	devise/passwords#update
		PUT	/users/password(:format)	devise/passwords#update
	cancel_user_registration	GET	/users/cancel(:format)	devise/registrations#cancel
	user_registration	POST	/users(:format)	devise/registrations#create
	new_user_registration	GET	/users/sign_up(:format)	devise/registrations#new
	edit_user_registration	GET	/users/edit(:format)	devise/registrations#edit
		PATCH	/users(:format)	devise/registrations#update
		PUT	/users(:format)	devise/registrations#update
		DELETE	/users(:format)	devise/registrations#destroy
	posts	GET	/posts(:format)	posts#index
		POST	/posts(:format)	posts#create
	new_post	GET	/posts/new(:format)	posts#new
	edit_post	GET	/posts/:id/edit(:format)	posts#edit
	post	GET	/posts/:id(:format)	posts#show
		PATCH	/posts/:id(:format)	posts#update
		PUT	/posts/:id(:format)	posts#update
		DELETE	/posts/:id(:format)	posts#destroy
	root	GET	/	posts#index

serabe@Weisz:~/programming/cc/cursos/devise\_crash\_course/devise\_blog (\*)

> rake routes

	Prefix	Verb	URI Pattern	Controller#Action
	new_user_session	GET	/users/sign_in(:format)	devise/sessions#new
	user_session	POST	/users/sign_in(:format)	devise/sessions#create
	destroy_user_session	DELETE	/users/sign_out(:format)	devise/sessions#destroy
	user_password	POST	/users/password(:format)	devise/passwords#create
	new_user_password	GET	/users/password/new(:format)	devise/passwords#new
	edit_user_password	GET	/users/password/edit(:format)	devise/passwords#edit
		PATCH	/users/password(:format)	devise/passwords#update
		PUT	/users/password(:format)	devise/passwords#update
	cancel_user_registration	GET	/users/cancel(:format)	devise/registrations#cancel
	user_registration	POST	/users(:format)	devise/registrations#create
	new_user_registration	GET	/users/sign_up(:format)	devise/registrations#new
	edit_user_registration	GET	/users/edit(:format)	devise/registrations#edit
		PATCH	/users(:format)	devise/registrations#update
		PUT	/users(:format)	devise/registrations#update
		DELETE	/users(:format)	devise/registrations#destroy
	posts	GET	/posts(:format)	posts#index
		POST	/posts(:format)	posts#create
	new_post	GET	/posts/new(:format)	posts#new
	edit_post	GET	/posts/:id/edit(:format)	posts#edit
	post	GET	/posts/:id(:format)	posts#show
		PATCH	/posts/:id(:format)	posts#update
		PUT	/posts/:id(:format)	posts#update
		DELETE	/posts/:id(:format)	posts#destroy
	root	GET	/	posts#index



posts\_controller.rb x

```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```

posts\_controller.rb x


```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```

posts\_controller.rb x

```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```

posts\_controller.rb x

```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```




posts\_controller.rb x

```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```

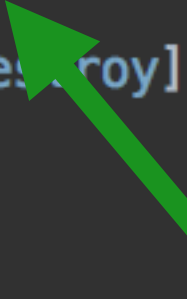
posts\_controller.rb x

```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```



posts\_controller.rb x

```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```





posts\_controller.rb x

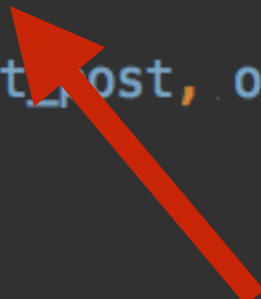
```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```





posts\_controller.rb x

```
1 1 class PostsController < ApplicationController
2 1   before_action :authenticate_user!,
3 2     only: [:new, :create, :edit, :update, :destroy]
4 3
5 4   before_action :set_post, only: [:show, :edit, :update, :destroy]
6 5
7 6   # GET /posts
8 7   # GET /posts.json
9 8   def index
10 9     @posts = Post.all
11 10   end
12 11
13 12   # GET /posts/1
14 13   # GET /posts/1.json
15 14   def show
16 15   end
17 16
18 17   # GET /posts/new
19 18   def new
20 19     @post = Post.new
```



posts\_controller.rb x

```
1 25 class PostsController < ApplicationController
2 24   before_action :authenticate_user!,
3 23   except: [:index, :show]
4 22
```

Run test: devise\_blog



0 of 1 test: 1 failed

Terminated

test\_publish\_a\_post

`/Users/serabe/.rvm/rubies/ruby-2.1.5/bin/ruby -e $stdout.sync=true;$stderr.sync=tru`

Testing started at 18:31 ...

Running via Spring preloader in process 19228

Started

Minitest::Assertion: Expected: "/posts/new"

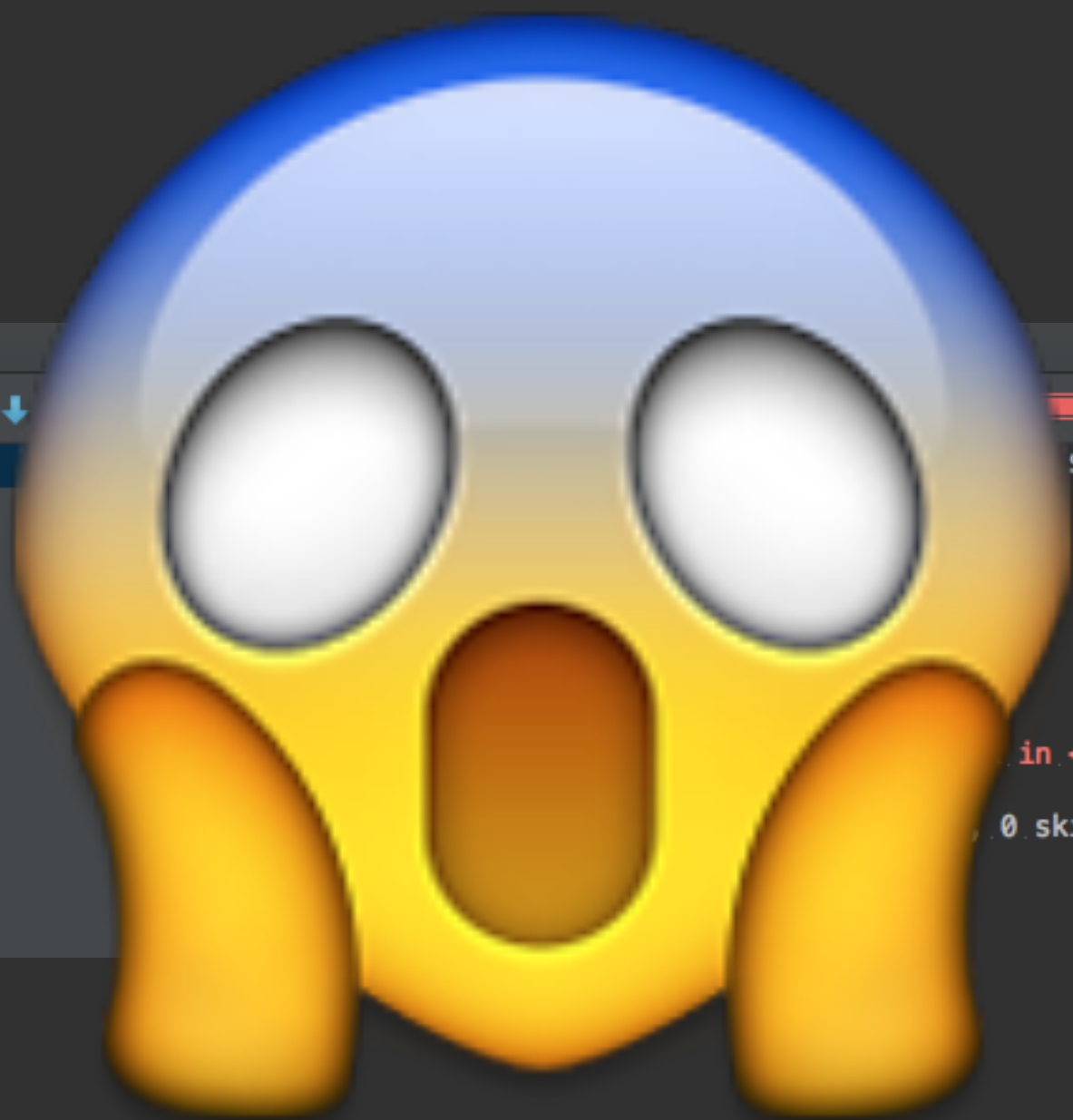
Actual: "/users/sign\_in"

`test/features/publishing_test.rb:11:in `block in <class:PublishingTest>'`

Finished in 0.41464s

1 tests, 2 assertions, 1 failures, 0 errors, 0 skips

Process finished with exit code 0



Run  test: devise\_blog

  Terminated

 test\_publish\_a\_post


 0 of 1 test: 1 failed

`$stdout.sync=true;$stderr.sync=tru`

`in <class:PublishingTest>`

`0 skips`

0 of 1 test: 1 failed

▼  Terminated `/Users/serabe/.rvm/rubies/ruby-2.1.5/bin/ruby -e $stdout.sync=true;$stderr.sync=true`

```
! test_publish_a_post
```

```
Running via Spring preloader in process 19228
Started
```

Started

```

minitest:Assertion: Expected: "/posts/new"
Actual: "/users/sign in"

```

```
Finished in 0.41464s
```

```
1 tests, 2 assertions, 1 failures, 0 errors, 0 skips
```

```
/Users/serabe/.rvm/rubies/ruby-2.1.5/bin/ruby -e $stdout.sync=true;$stderr.sync=true
```

```
Testing started at 18:31 ...
```

Running via Spring preloader in process 19228

Started

```
Minitest::Assertion: Expected: "/posts/new"
```

```
Actual: "/users/sign_in"
```

```
test/features/publishing test.rb:11:in `block in <class:PublishingTest>'
```

Finished in 0.41464s

```
1 tests, 2 assertions, 1 failures, 0 errors, 0 skips
```

```
Process finished with exit code 0
```

posts\_controller.rb x

publishing\_test.rb x

```
1 24 require "test_helper"
2 23
3 22 class PublishingTest < Capybara::Rails::TestCase
4 21   test "cannot publish a post if not logged in" do
5 20     visit posts_path
6 19
7 18     # assert_css page, '.posts_table tbody tr', count: 0
8 17
9 16     click_link 'New Post'
10 15
11 14     assert_equal new_user_session_path, current_path
12 13
13 12
14 11     # within '#new_post' do
```

Run: Development: devise\_blog test: devise\_blog

All Tests 406ms

```
/Users/serabe/.rvm/rubies/ruby-2.1.5/bin/ruby -e $stdout.sync=true;$st
Testing started at 18:53 ...
Running via Spring preloader in process 19452
Started

Finished in 0.41515s
2 tests, 7 assertions, 0 failures, 0 errors, 0 skips

Process finished with exit code 0
```

All 2 tests pas





Run: Development: devise\_blog test: devise\_blog

▶

OK

☰

↓a

↓Z

»

All Tests 406ms

↑

↓

↻

📄

🖨

🗑

Testing started at 18:53 ...

Running via Spring preloader in process 19452

Started

Finished in 0.41515s

2 tests, 7 assertions, 0 failures, 0 errors, 0 skips

Process finished with exit code 0

All 2 tests pas

What should we do  
next?

```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmple@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```

```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmple@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```

```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmp@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```



```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmple@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```

```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmp@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```



```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmple@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21   assert_equal new_post_path, current_path
29 20
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```

```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmple@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```

```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmp@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11   visit posts_path
39 10
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```

```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmp@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```



```
12 37 test "can publish a post if logged in" do
13 36   visit posts_path
14 35   user = User.create email: 'exmple@example.com', password: '123123123'
15 34
16 33   assert_css page, '.posts_table tbody tr', count: 0
17 32
18 31   click_link 'New Post'
19 30
20 29   assert_equal new_user_session_path, current_path
21 28
22 27   within '#new_user' do
23 26     fill_in 'Email', with: user.email
24 25     fill_in 'Password', with: user.password
25 24
26 23     click_button 'Sign in'
27 22   end
28 21
29 20   assert_equal new_post_path, current_path
30 19
31 18
32 17   within '#new_post' do
33 16     fill_in 'Title', with: 'My Article'
34 15     fill_in 'Body', with: 'My awesome new article on Devise'
35 14
36 13     click_button 'Create Post'
37 12   end
38 11
39 10   visit posts_path
40 9
41 8   assert_css page, '.posts_table tbody tr', count: 1
42 7
43 6   within '.posts_table tbody tr:first' do
44 5     assert_content page, 'My Article'
45 4     assert_content page, 'My awesome new article on Devise'
46 3   end
47 2 end
48 1 end
```

Better Navigation

```
posts_controller.rb x publishing_test.rb x authentication_test.rb x
1 9      require "test_helper"
2 8
3 7      class AuthenticationTest < Capybara::Rails::TestCase
4 6          test "a registered user can log in" do
5 5              visit root_path
6 4
7 3              assert_content page, 'Log In'
8 2          end
9 1      end
10 10
11 1
```



posts\_controller.rb x

publishing\_test.rb x

authentication\_test.rb x

```
1 9      require "test_helper"
2 8
3 7      class AuthenticationTest < Capybara::Rails::TestCase
4 6          test "a registered user can log in" do
5 5              visit root_path
6 4
7 3              assert_content page, 'Log In'
8 2          end
9 1      end
10 10
11 1
```

# views/layouts/application

```
posts_controller.rb x publishing_test.rb x authentication_test.rb x application/_header.html.erb x a
1 20 <!DOCTYPE html>
2 19 <html>
3 18 <head>
4 17   <title>DeviseBlog</title>
5 16   <%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track' => true %>
6 15   <%= javascript_include_tag 'application', 'data-turbolinks-track' => true %>
7 14   <%= csrf_meta_tags %>
8 13 </head>
9 12 <body>
10 11   <%= render partial: 'header' %>
11 10
12 9   <p class="notice"><%= notice %></p>
13 8   <p class="alert"><%= alert %></p>
14 7
15 6   <%= yield %>
16 5
17 4 </body>
18 3 </html>
19 2
20 1
```

# views/application/\_header

```
posts_controller.rb × publishing_test.rb × authentication_test.rb × application/_header.html.erb × application.html.erb ×  
13 1 <nav>  
22   <%= link_to 'Log In', new_user_session_path %>  
31 </nav>  
44  
51
```

```
posts_controller.rb x publishing_test.rb x authentication_test.rb x
1 9      require "test_helper"
2 8
3 7      class AuthenticationTest < Capybara::Rails::TestCase
4 6          test "a registered user can log in" do
5 5              visit root_path
6 4
7 3              assert_content page, 'Log In'
8 2          end
9 1      end
10 10
11 1
```

posts\_controller.rb x publishing\_test.rb x authentication\_test.rb x application/\_header.html.erb x

```
1 22 require "test_helper"
2 21
3 20 class AuthenticationTest < Capybara::Rails::TestCase
4 19   test "a registered user can log in" do
5 18     user = User.create email: 'example@example.com', password: '1234567890'
6 17
7 16     visit root_path
8 15
9 14     assert_content page, 'Log In'
10 13
11 12     click_link 'Log In'
12 11
13 10     within '#new_user' do
14 9       fill_in 'Email', with: user.email
15 8       fill_in 'Password', with: user.password
16 7
17 6       click_button 'Sign in'
18 5     end
19 4
20 3     assert_content page, 'Log Out'
21 2   end
22 1 end
```

posts\_controller.rb x publishing\_test.rb x authentication\_test.rb x application/\_header.html.erb x application.html.erb

```
1 11 <nav>
2 10   <% if user_signed_in? %>
3 9     <%= link_to 'Log Out',
4 8       destroy_user_session_path,
5 7       method: :delete %>
6 6   <% else %>
7 5     <%= link_to 'Log In',
8 4       new_user_session_path %>
9 3   <% end %>
10 2 </nav>
```

posts\_controller.rb x publishing\_test.rb x authentication\_test.rb x application/\_header.html.erb x application.html.erb

```
1 11 <nav>
2 10   <% if user_signed_in? %>
3 9     <%= link_to 'Log Out',
4 8         destroy_user_session_path,
5 7         method: :delete %>
6 6   <% else %>
7 5     <%= link_to 'Log In',
8 4         new_user_session_path %>
9 3   <% end %>
10 2 </nav>
```



posts\_controller.rb x publishing\_test.rb x authentication\_test.rb x application/\_header.html.erb x application.html.erb

```
1 11 <nav>
2 10   <% if user_signed_in? %>
3 9     <%= link_to 'Log Out',
4 8         destroy_user_session_path,
5 7         method: :delete %>
6 6   <% else %>
7 5     <%= link_to 'Log In',
8 4         new_user_session_path %>
9 3   <% end %>
10 2 </nav>
```

posts\_controller.rb × publishing\_test.rb × authentication\_test.rb × application/\_header.html.erb ×

```
1 22 require "test_helper"
2 21
3 20 class AuthenticationTest < Capybara::Rails::TestCase
4 19   test "a registered user can log in" do
5 18     user = User.create email: 'example@example.com', password: '1234567890'
6 17
7 16     visit root_path
8 15
9 14     assert_content page, 'Log In'
10 13
11 12     click_link 'Log In'
12 11
13 10     within '#new_user' do
14 9       fill_in 'Email', with: user.email
15 8       fill_in 'Password', with: user.password
16 7
17 6       click_button 'Sign in'
18 5     end
19 4
20 3     assert_content page, 'Log Out'
21 2   end
22 1 end
```

Can you add an  
author to our posts?

```
class OnlyAuthorCanEditTest < Capybara::Rails::TestCase
  test "shows the author of an article" do
    user = User.create email: 'example@example.com', password: '1234567890'
    visit root_path

    click_link 'New Post'

    within '#new_user' do
      fill_in 'Email', with: user.email
      fill_in 'Password', with: user.password

      click_button 'Sign in'
    end

    within '#new_post' do
      fill_in 'Title', with: 'My Article'
      fill_in 'Body', with: 'My awesome new article on Devise'

      click_button 'Create Post'
    end

    visit post_path(Post.last)

    assert_content page, 'Written by example@example.com'
  end
end
```

```
class OnlyAuthorCanEditTest < Capybara::Rails::TestCase
  test "shows the author of an article" do
    user = User.create email: 'example@example.com', password: '1234567890'
    visit root_path

    click_link 'New Post'

    within '#new_user' do
      fill_in 'Email', with: user.email
      fill_in 'Password', with: user.password

      click_button 'Sign in'
    end

    within '#new_post' do
      fill_in 'Title', with: 'My Article'
      fill_in 'Body', with: 'My awesome new article on Devise'

      click_button 'Create Post'
    end

    visit post_path(Post.last)

    assert_content page, 'Written by example@example.com'
  end
end
```

```
class OnlyAuthorCanEditTest < Capybara::Rails::TestCase
  test "shows the author of an article" do
    user = User.create email: 'example@example.com', password: '1234567890'
    visit root_path

    click_link 'New Post'

    within '#new_user' do
      fill_in 'Email', with: user.email
      fill_in 'Password', with: user.password

      click_button 'Sign in'
    end

    within '#new_post' do
      fill_in 'Title', with: 'My Article'
      fill_in 'Body', with: 'My awesome new article on Devise'

      click_button 'Create Post'
    end

    visit post_path(Post.last)

    assert_content page, 'Written by example@example.com'
  end
end
```

```
class OnlyAuthorCanEditTest < Capybara::Rails::TestCase
  test "shows the author of an article" do
    user = User.create email: 'example@example.com', password: '1234567890'
    visit root_path

    click_link 'New Post'

    within '#new_user' do
      fill_in 'Email', with: user.email
      fill_in 'Password', with: user.password

      click_button 'Sign in'
    end

    within '#new_post' do
      fill_in 'Title', with: 'My Article'
      fill_in 'Body', with: 'My awesome new article on Devise'

      click_button 'Create Post'
    end

    visit post_path(Post.last)

    assert_content page, 'Written by example@example.com'
  end
end
```



```
class OnlyAuthorCanEditTest < Capybara::Rails::TestCase
  test "shows the author of an article" do
    user = User.create email: 'example@example.com', password: '1234567890'
    visit root_path

    click_link 'New Post'

    within '#new_user' do
      fill_in 'Email', with: user.email
      fill_in 'Password', with: user.password

      click_button 'Sign in'
    end

    within '#new_post' do
      fill_in 'Title', with: 'My Article'
      fill_in 'Body', with: 'My awesome new article on Devise'


      click_button 'Create Post'
    end

    visit post_path(Post.last)

    assert_content page, 'Written by example@example.com'
  end
end
```


# Migration

```
16 class AddUserToPosts < ActiveRecord::Migration
25   def change
34     add_reference :posts, :author, index: true, foreign_key: true
43   end
52 end
```




# Migration

```
16 class AddUserToPosts < ActiveRecord::Migration
25   def change
34     add_reference :posts, :author, index: true, foreign_key: true
43   end
52 end
```



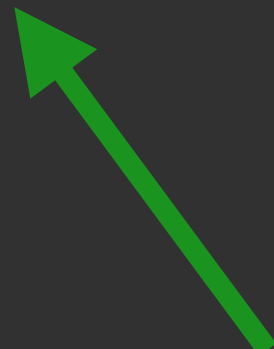
# Migration

```
16 class AddUserToPosts < ActiveRecord::Migration
25   def change
34     add_reference :posts, :author, index: true, foreign_key: true
43   end
52 end
```



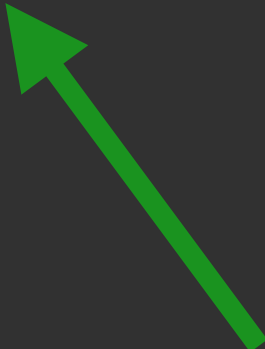
# Migration

```
16 class AddUserToPosts < ActiveRecord::Migration
25   def change
34     add_reference :posts, :author, index: true, foreign_key: true
43   end
52 end
```




# Migration

```
16 class AddUserToPosts < ActiveRecord::Migration
25   def change
34     add_reference :posts, :author, index: true, foreign_key: true
43   end
52 end
```



# Post Model


```
15 • class Post < ActiveRecord::Base
24   belongs_to :author,
33           class_name: 'User',
42           required: true
51 end
```





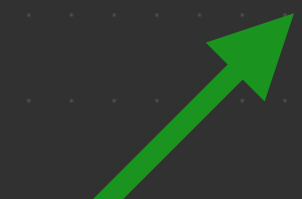
# Post Model

```
15 • class Post < ActiveRecord::Base
24   belongs_to :author,
33   class_name: 'User',
42   required: true
51 end
```




# Post Model

```
15 • class Post < ActiveRecord::Base
24   belongs_to :author,
33   class_name: 'User',
42   required: true
51 end
```




# Post Model

```
15 • class Post < ActiveRecord::Base
24   belongs_to :author,
33   class_name: 'User',
42   required: true
51 end
```




# User Model

```
1 10 class User < ActiveRecord::Base
2 9   # Include default devise modules. Others available are:
3 8   # :confirmable, :lockable, :timeoutable and :omniauthable
4 7   devise :database_authenticatable, :registerable,
5 6         :recoverable, :rememberable, :trackable, :validatable
6 5
7 4   has_many :posts,
8 3         foreign_key: 'author_id'
9 2 end
```



# User Model

```
1 10 class User < ActiveRecord::Base
2 9   # Include default devise modules. Others available are:
3 8   # :confirmable, :lockable, :timeoutable and :omniauthable
4 7   devise :database_authenticatable, :registerable,
5 6         :recoverable, :rememberable, :trackable, :validatable
6 5
7 4   has_many :posts,
8 3         foreign_key: 'author_id'
9 2 end
```



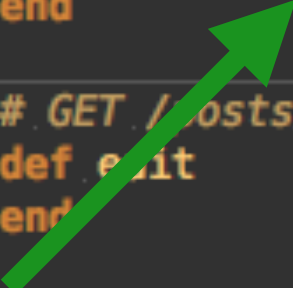
# User Model

```
1 10 class User < ActiveRecord::Base
2 9   # Include default devise modules. Others available are:
3 8   # :confirmable, :lockable, :timeoutable and :omniauthable
4 7   devise :database_authenticatable, :registerable,
5 6         :recoverable, :rememberable, :trackable, :validatable
6 5
7 4   has_many :posts,
8 3         foreign_key: 'author_id'
9 2 end
```



# Post Controller


```
21 3 def new
22 4   @post = current_user.posts.build
23 5 end
24 6
25 7 # GET /posts/1/edit
26 8 def edit
27 9 end
28 10
29 11 # POST /posts
30 12 # POST /posts.json
31 13 def create
32 14   @post = current_user.posts.build(post_params)
33 15
34 16   respond_to do |format|
35 17     if @post.save
36 18       format.html { redirect_to @post, notice: 'Post was successfully created.' }
37 19       format.json { render :show, status: :created, location: @post }
38 20     else
39 21       format.html { render :new }
40 22       format.json { render json: @post.errors, status: :unprocessable_entity }
41 23     end
42 24   end
43 25 end
```






# Post Controller

```
21 3 def new
22 4   @post = current_user.posts.build
23 5 end
24 6
25 7 # GET /posts/1/edit
26 8 def edit
27 9 end
28 10
29 11 # POST /posts
30 12 # POST /posts.json
31 13 def create
32 14   @post = current_user.posts.build(post_params)
33 15
34 16   respond_to do |format|
35 17     if @post.save
36 18       format.html { redirect_to @post, notice: 'Post was successfully created.' }
37 19       format.json { render :show, status: :created, location: @post }
38 20     else
39 21       format.html { render :new }
40 22       format.json { render json: @post.errors, status: :unprocessable_entity }
41 23     end
42 24   end
43 25 end
```



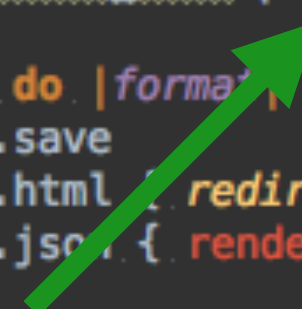
# Post Controller

```
21 3 def new
22 4   @post = current_user.posts.build
23 5 end
24 6
25 7 # GET /posts/1/edit
26 8 def edit
27 9 end
28 10
29 11 # POST /posts
30 12 # POST /posts.json
31 13 def create
32 14   @post = current_user.posts.build(post_params)
33 15
34 16   respond_to do |format|
35 17     if @post.save
36 18       format.html { redirect_to @post, notice: 'Post was successfully created.' }
37 19       format.json { render :show, status: :created, location: @post }
38 20     else
39 21       format.html { render :new }
40 22       format.json { render json: @post.errors, status: :unprocessable_entity }
41 23     end
42 24   end
43 25 end
```



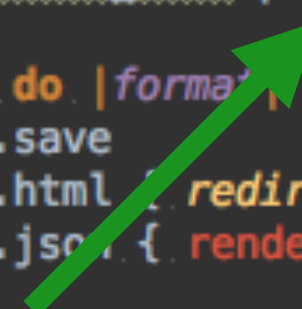
# Post Controller

```
21 3 def new
22 4   @post = current_user.posts.build
23 5 end
24 6
25 7 # GET /posts/1/edit
26 8 def edit
27 9 end
28 10
29 11 # POST /posts
30 12 # POST /posts.json
31 13 def create
32 14   @post = current_user.posts.build(post_params)
33 15
34 16   respond_to do |format|
35 17     if @post.save
36 18       format.html { redirect_to @post, notice: 'Post was successfully created.' }
37 19       format.json { render :show, status: :created, location: @post }
38 20     else
39 21       format.html { render :new }
40 22       format.json { render json: @post.errors, status: :unprocessable_entity }
41 23     end
42 24   end
43 25 end
```



# Post Controller

```
21 3 def new
22 4   @post = current_user.posts.build
23 5 end
24 6
25 7 # GET /posts/1/edit
26 8 def edit
27 9 end
28 10
29 11 # POST /posts
30 12 # POST /posts.json
31 13 def create
32 14   @post = current_user.posts.build(post_params)
33 15
34 16   respond_to do |format|
35 17     if @post.save
36 18       format.html { redirect_to @post, notice: 'Post was successfully created.' }
37 19       format.json { render :show, status: :created, location: @post }
38 20     else
39 21       format.html { render :new }
40 22       format.json { render json: @post.errors, status: :unprocessable_entity }
41 23     end
42 24   end
43 25 end
```



# views/posts/show

```
<p>  
  <strong>Body:</strong>  
  <%= @post.body %>  
</p>
```

```
<p>  
  <strong>Written by</strong>  
  <%= @post.author.email %>  
</p>
```

```
class OnlyAuthorCanEditTest < Capybara::Rails::TestCase
  test "shows the author of an article" do
    user = User.create email: 'example@example.com', password: '1234567890'
    visit root_path

    click_link 'New Post'

    within '#new_user' do
      fill_in 'Email', with: user.email
      fill_in 'Password', with: user.password

      click_button 'Sign in'
    end

    within '#new_post' do
      fill_in 'Title', with: 'My Article'
      fill_in 'Body', with: 'My awesome new article on Devise'

      click_button 'Create Post'
    end

    visit post_path(Post.last)

    assert_content page, 'Written by example@example.com'
  end
end
```

Can you add the  
confirmable module?



# Confirmable

- Create the migration (look the previous one)
- Configure it
  - Model
  - Initializer
- Add `confirmed_at: Time.now` to your tests.
- Bonus points if you tried testing it!

# Confirmable

- Create the model (re-use the previous one)
- Configure it
  - Model
  - Initializer
- Add `confirmed_at`: Time of our tests.
- Bonus points if you tried testing it!

# Confirmable

- Create the migration (look the previous one)
- Configure it
  - Model
  - Initializer
- Add `confirmed_at: Time.now` to your tests.
- Bonus points if you tried testing it!

# Thank you!

Sergio Arbeo



Serabe



Serabe