

(/)

1. Adding Authentication with Devise

Created by Piotr Steininger, @polishprince (<https://twitter.com/polishprince>)

Updated by Ernesto Jimenez, @ernesto_jimenez (https://twitter.com/ernesto_jimenez)

This guide assumes that you have already built a Rails Girls app by following the app development guide (/app).

1. Add devise gem

Open up your `Gemfile` and add this line

```
gem 'devise'
```

and run

```
bundle install
```

to install the gem. **Also remember to restart the Rails server.**

2. Set up devise in your app

Run the following command in the terminal.

```
rails g devise:install
```

3. Configure Devise

Ensure you have defined default url options in your environments files. Open up

`config/environments/development.rb` and add this line:

```
config.action_mailer.default_url_options = { :host => 'localhost:3000' }
```

before the `end` keyword.

Open up `app/views/layouts/application.html.erb` and add:

```
<% if notice %>
  <p class="alert alert-success"><%= notice %></p>
<% end %>
<% if alert %>
  <p class="alert alert-danger"><%= alert %></p>
<% end %>
```

right above

```
<%= yield %>
```

Open up `app/views/ideas/show.html.erb` and remove the line that says:

```
<p id="notice"><%= notice %></p>
```

Do the same for `app/views/comments/show.html.erb`. These lines are not necessary as we've put the notice in the `app/views/layouts/application.html.erb` file.

Setup the User model

We'll use a bundled generator script to create the User model.

```
rails g devise user
rake db:migrate
```

Coach: Explain what user model has been generated. What are the fields?

Create your first user

Now that you have set everything up you can create your first user. Devise creates all the code and routes required to create accounts, log in, log out, etc.

Make sure your rails server is running, open `http://localhost:3000/users/sign_up` (`http://localhost:3000/users/sign_up`) and create your user account.

Add sign-up and login links

All we need to do now is to add appropriate links or notice about the user being logged in in the top right corner of the navigation bar.

In order to do that, edit `app/views/layouts/application.html.erb` add:

```
<p class="navbar-text pull-right">
  <% if user_signed_in? %>
    Logged in as <strong><%= current_user.email %></strong>.
    <%= link_to 'Edit profile', edit_user_registration_path, :class => 'navbar-link' %> |
    <%= link_to "Logout", destroy_user_session_path, method: :delete, :class => 'navbar-link' %>
  <% else %>
    <%= link_to "Sign up", new_user_registration_path, :class => 'navbar-link' %>
    |
    <%= link_to "Login", new_user_session_path, :class => 'navbar-link' %>
  <% end %>
</p>
```

right after

```
<ul class="nav">
  <li class="active"><a href="/ideas">Ideas</a></li>
</ul>
```

Finally, force the user to redirect to the login page if the user was not logged in.

Open up `app/controllers/application_controller.rb` and add:

```
before_action :authenticate_user!
```

after `protect_from_forgery with: :exception`.

Open your browser and try logging in and out from.

Coach: Talk about the `user_signed_in?` and `current_user` helpers. Why are they useful?

What next?

- Add extra fields to the User model
- Add relationships between users and ideas
- Restrict users to only be able to edit their own ideas and delete their own comments
- Expand to use roles or permissions (use one of the popular authorization gem like CanCan)

This work is licensed under a Creative Commons Attribution-Share Alike 3.0 License
(<http://creativecommons.org/licenses/by-sa/3.0/>)

Rails Girls (<http://railsgirls.com>) · Crafted with excitement from Helsinki, Finland · Follow us on Twitter
(<http://twitter.com/railsgirls>) and Facebook (<https://www.facebook.com/railsgirls>)