

Incorporating Devise into your Application

1) Add Devise to Your Gemfile

Open the Gemfile in the root directory of the project and add *gem 'devise'*

After adding the gem, install it by running the *bundle install* command

2) Set Up Devise

Initialize Devise in the application by running *rails generate devise:install*

3) Configure Action Mailer

In order for Devise to work, you need to set up the default URL options for the ActionMailer in your development environment. Open *config/environments/development.rb* and add:

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

This'll ensure that Devise can generate the correct URLs in emails sent during development.

4) Define the Root Route

Devise recommends defining a root route in your application. Generate a controller and view for the home page: *rails generate controller home index*

Then, set the root path in *config/routes.rb*:

```
root 'home#index'
```

5) Generate the User Model

Create a User model with Devise's authentication features by running: *rails generate devise User*

This command will generate the User model along with the necessary database migrations. Apply the migrates with: *rails db:migrate*

6) Add Flash Messages to Application Layout

If you want to actually display the flash messages (like alerts and notices) provided by Devise, modify your application layout. Open *app/views/layouts/application.html.erb* and add the following lines above the *<%= yield %>* statement:

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

```
<p class="alert"><%= alert %></p>
```

These lines render flash messages in your application's views.

7) Configure Navigation Links

In your home page view (*app/views/home/index.html.erb*), add conditional links for signing in and signing out:

```
<% if user_signed_in? %>

  <p>Welcome, <%= current_user.email %>!</p>

  <%= link_to 'Sign Out', destroy_user_session_path, method: :delete %>

<% else %>

  <%= link_to 'Sign In', new_user_session_path %>

  <%= link_to 'Sign Up', new_user_registration_path %>

<% end %>
```

These links provide navigation options based on the user's authentication status.

8) And You're Done!

If all goes well, you should see the home page with options to sign up or sign in. Go ahead and test the authentication flow by signing up, signing in, and signing out. For more advanced configurations and customization.

Some extra resources:

<https://learnnetto.com/tutorials/how-to-set-up-a-rails-7-app-with-devise-gem-for-user-authentication>

<https://hackernoon.com/how-to-add-user-authentication-using-devise-bootstrap-in-rails-7>

<https://www.youtube.com/watch?v=k9g2sPXarwg>