**USEFUL WIKI PAGES**

<https://github.com/plataformatec/devise/wiki/_pages>

**TESTING SETUP**

➜ Before you start, do the test setup

**SIMPLE FORM**

➜ Add simple\_form gem

gem ‘simple\_form’

➜ Run simple form with twitter bootstrap generator

**DEVISE**

➜ Put the devise gem in Gemfile

gem ‘devise’

$ bundle install

➜ Run devise generator

$ rails generate devise:install

➜ Add default\_url\_options to development.rb & production.rb appropriately

config.action\_mailer.default\_url\_options = { :host => 'localhost:3000' }

➜ Edit config/initializers/devise.rb

a) Change mailer\_sender setting option for the return email address for messages that Devise sends from the application.

**config.mailer\_sender** = "[info@railsforcharity.org](mailto:info@railsforcharity.org)"

b) Comment out sign\_out\_via so that the logout links work

**# config.sign\_out\_via** = :delete

➜ Generate a model and routes for a User

$ rails generate devise User

Devise will modify the **config/routes.rb** file to add:

devise\_for :users

which provides a complete set of routes for user signup and login. If you run rake routes you can see the routes that this line of code creates.

➜ Select the following devise modules in user.rb

# Devise modules. Others available are: :token\_authenticatable, :lockable, :timeoutable

devise :database\_authenticatable, :registerable, :omniauthable, :confirmable, :recoverable, :rememberable, :trackable, :validatable

#### ➜ **Prevent Logging of Passwords**

We don’t want passwords written to our log file.

Modify the file **config/application.rb** to include:

config.filter\_parameters += [:password, :password\_confirmation]

Note that filter\_parameters is an array.

#### ➜ **Enable Users to Have Names**

By default, Devise uses an email address to identify users. We’ll add a “name” attribute as well. Your application may not require a user to provide a name. But showing you how to add a name will help you see what you need to do if you decide to make changes to the default Devise user model.

Devise created a migration file to establish the schema for the SQLite database with a migration file named something like **db/migrate/xxxxxxx\_devise\_create\_users.rb**. We won’t modify the migration file. Instead we’ll add an additional migration that adds the “name” field to the User record.

rails generate migration AddNameToUsers name:string

Run the migration and prepare the test database to pick up the “name” field:

$ bundle exec rake db:migrate  
$ bundle exec rake db:test:prepare

You’ll also want to prevent malicious hackers from creating fake web forms that would allow changing of passwords through the mass-assignment operations of update\_attributes(attrs) and new(attrs). Devise already added this to the **models/user.rb** file:

attr\_accessible :email, :password, :password\_confirmation, :remember\_me

but you’ll need to add the “name” attribute:

attr\_accessible :name, :email, :password, :password\_confirmation, :remember\_me

If you’ve chosen to create the application with the Devise Confirmable module, also add :confirmed\_at:

attr\_accessible :name, :email, :password, :password\_confirmation, :remember\_me, :confirmed\_at

#### ➜ **Copy the following to application\_helper.rb**

def user\_not\_signed\_in?

!user\_signed\_in?

end

➜ To, customize devise views, run the command

$ rails generate devise:views

➜ **Convert Devise views from erb to haml**

$ gem install html2haml

$ **for** i in `find app/views/devise -name '\*.erb'` ; **do** html2haml -e $i **${**i%erb**}**haml ; rm $i ; **done**

**OR**

➜ **Convert Devise views from erb to slim** (<https://github.com/plataformatec/devise/wiki/How-To:-Create-Haml-and-Slim-Views>)

$ gem install haml hpricot ruby\_parser slim haml2slim

$ **for** i in `find app/views/devise -name '\*.erb'` ; **do** html2slim -e $i **${**i%erb**}**haml ; rm $i ; **done**

➜ Change all the devise views and add :class => ‘btn btn-primary’ to submit buttons for twitter bootstrap and ‘’ for Zurb Foundation.

example: %div= f.submit "Send me reset password instructions", :class => 'btn btn-primary'

and also align the %br’s (line breaks) properly

List of files to change:

→ tree app/views/devise/

app/views/devise/

├── confirmations

│ └── new.html.haml

├── passwords

│ ├── edit.html.haml

│ └── new.html.haml

├── registrations

│ ├── edit.html.haml

│ └── new.html.haml

├── sessions

│ └── new.html.haml

├── shared

│ └── \_links.haml

└── unlocks

└── new.html.haml

➜ Add a “Thank You” note to all the mailers in views/devise/mailers

%br

%br

Thanks!

%br

Your App

➜ Add users routes in routes.rb

resources :users

➜ Generate users controller ( $ rails g controller users )

➜ Add the following to users\_controller.rb

before\_filter :authenticate\_user!, :except => [:show, :index]

before\_filter :set\_user, :only => [:edit, :update, :show]

def edit

@user = current\_user

end

def show

end

def index

end

def update

@user = User.find(params[:id])

respond\_to do |format|

if @user.update\_attributes(params[:user])

format.html { redirect\_to :back, notice: 'User was successfully updated.' }

format.json { head :no\_content }

else

format.html { render action: "edit" }

format.json { render json: @event.errors, status: :unprocessable\_entity }

end

end

end

private

def set\_user

@user = User.find(params[:id])

end

➜ Create users edit haml file and paste the following

= simple\_form\_for @user, :validate => true, :html => {:multipart => true} do |f|

= f.input :name

= f.input :email

= f.submit 'Save Changes', :class => 'btn btn-primary'

➜ Edit the config/devise.en.yml file and change the following strings according to your application

mailer:

confirmation\_instructions:

subject: '[Your App] Please confirm your account.'

reset\_password\_instructions:

subject: '[Your App] Reset password instructions.'

unlock\_instructions:

subject: '[Your App] Account unlocking instructions.'

➜ Add destroy user functionality

app/models/user.rb

- has\_many :authentications

+ has\_many :authentications, :dependent => :destroy

app/views/users/edit.html.haml

+= link\_to t('views.users.edit.delete\_account'), user\_registration\_path, :method => :delete,

:confirm => 'Are you sure you want to destroy your account?', :class => "close"

config/locales/devise.en.yml

- destroyed: 'Bye! Your account was successfully cancelled. We hope to see you again soon.'

+ destroyed: 'We''re really Sorry to see you go! Your account was successfully cancelled. We hope to see you again soon.'