Put this code to create a Micropost scaffold. Had to force it since I made a mistake the first time

$ rails generate scaffold Micropost content:string user\_id:integer initiator:in

teger responder:integer topic:string --force

rake db:migrate

I may not need initiator and responder since user\_id already exists. Will need to figure this out down the line

Rails generate model Relationship initiator\_id:integer responder\_id:integer

Copying

add\_index :relationships, :initiator\_id

add\_index :relationships, :responder\_id

add\_index :relationships, [:initiator\_id, :responder\_id], unique: true

to migrate file in order to create index. Right now I am not sure if the last statement will be ‘unique’. I may need to make it unique by initiator-respoder-topic\_id or something. Right now I don’t have a ‘topic\_id’ anywhere or a table about topics..maybe I should make one

rake db:migrate

there is some good information on populating a database for testing purposes at <http://railscasts.com/episodes/126-populating-a-database>

Added

resources :users do

member do

get :responders, :initiators

end

end

to routes.db

created a new partial file \_stats and added this to it

<% @user ||= current\_user %>

<div class="stats">

<a href="<%= responders\_user\_path(@user) %>">

<strong id="responders" class="stat">

<%= @user.responder\_users.count %>

</strong>

responders

</a>

<a href="<%= initiators\_user\_path(@user) %>">

<strong id="initiators" class="stat">

<%= @user.initiators.count %>

</strong>

initiators

</a>

</div>

Then need to render the \_stats file in the home page

When creating micropost, I did not create an index as shown in the tutorial. Not sure how to do a migrate. Will need to check

Changed routes to microposts#index

Added the following to Microposts controller

before\_action :signed\_in\_user, only: [:create, :destroy]

this command can be used to generate the views for devise gem since it does not do so automatically

rails generate devise:views

Added code to Application helper

def current\_user?(user)

user == current\_user

end

may need to delete the following in routes under users

resources :users do

member do

get :whoresponders, :whoinitiators

end

end

may need to delete whoresponders and whoinitiators from user.rb. Same methods have been replicated in the microposts controller

try putting user controller stuff in application controller

check if @users = @user.responder\_users.paginate(page: params[:page]) is wrking in user.rb

I added this code for checking in show\_communicate

<%# I added this div%>

<div class="user\_avatars">

<% @users.each do |user| %>

<%#= render user %> i

<%= render 'microposts/users' %>

<% end %>

</div>

Deleted the following code from micropost\_controller

def whoresponders

@title = "Responders"

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

@users = @user.responder\_users.paginate(page: params[:page])

#@users = @user.responders.paginate(page: params[:page])

render 'show\_communicate'

end

def whoinitiators

@title = "Initiators"

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

@users = @user.initiators.paginate(page: params[:page])

render 'show\_communicate'

end

Deleted the following code from users\_controller and added it to applications controller

def whoresponders

@title = "Responders"

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

@users = @user.responder\_users.paginate(page: params[:page])

#@users = @user.responders.paginate(page: params[:page])

render 'show\_communicate'

end

def whoinitiators

@title = "Initiators"

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

@users = @user.initiators.paginate(page: params[:page])

render 'show\_communicate'

end

good website on application controller: <http://api.rubyonrails.org/classes/ActionController/Base.html>

deleted this from user.rb

def whoresponders

@title = "Responders"

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

@users = @user.responder\_users.paginate(page: params[:page])

render '@users'

@users.name

#@users = @user.responders.paginate(page: params[:page])

render 'show\_communicate'

end

def whoinitiators

@title = "Initiators"

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

@users = @user.initiators.paginate(page: params[:page])

render 'show\_communicate'

end

I added an index.html.erb in the devise folder to show the profile of a user

Deleted this after resources: microposts

do

member do

get :whoresponders, :whoinitiators

end

end

In relationships\_controller not sure why we do this:

@user = User.find(params[:relationship][:responder\_id])

Why cant we just do the above on responder\_id? Why do we need the :relationship

To learn

Naming conventions

Self. As in microposts.rb

Passing variables to partials

<http://stackoverflow.com/questions/4700617/pass-a-variable-into-a-partial-rails-3>

<%= render :partial => 'post', :collection => @posts, :spacer\_template => 'post\_divider' %>

<%= form\_for(@user, :url => url\_for(:controller => 'microposts', :action => 'new')) do |f| %>

Deleted

def home

@micropost = current\_user.microposts.build if signed\_in?

End

Fro microposts controller

Added @conversation\_items = current\_user.conversation.paginate(page: params[:page]) to applications controller home but still does not work

Also added the above to user.rb under the show method in users controller. It is duplicated but will need to remove one

Deleted \_conversation partials from microposts and moved them to user models

12/28

Deleted image validation from pin model

Using collection item to render partials and using ‘h’ to prevent user from inputting html tags in the inputs

<http://ruby.about.com/od/rubyonrails/ss/blogpart4.htm> (i think it is in page 6)

more code <http://ruby.about.com/od/rubyonrails/>

<http://ruby.about.com/od/rubyonrails/ss/What-Is-Activerecord.htm> provides some active record functions

add conversation\_list function to users.rb. this is to list out all conversations and render it in conversations\_list file

remove # from

class UsersController < ApplicationController

before\_action :set\_user#, only: [:show, :edit, :update, :destroy]

in users controller since I wanted the set\_user function to not trigger for index

added

@users = Pin.all.order("created\_at DESC").paginate(:page => params[:page], :per\_page => 5)

To users controller

Creating new objects manually

MyModel.create(:name => params[:name]) or MyModel.create(:name => params[:name])

def create\_forum

Forum.create(:upload\_id => self.id, :title => self.name...)

end

Cloning and object

<http://stackoverflow.com/questions/4300790/create-without-using-a-form-in-rails>

redirect vs. rendering

<http://guides.rubyonrails.org/layouts_and_rendering.html>

rendering a smaller collection

<http://stackoverflow.com/questions/1833659/render-partial-with-a-manipulated-collection>

render :partial => 'dogs/summary', :as => :dog,

:collection => @dogs.select{|dog| dog.gender == "M"}

Selecting a few items from a collection or an object

<http://stackoverflow.com/questions/3191448/limit-each-do-loop>

<% @feed.recent(10).each do |feed| %>

Feed.all.order('created\_at DESC').take(10)

@array.each\_with\_index do |feed, i|

break if i == 10;

<http://stackoverflow.com/questions/5624548/limit-each-list-in-rails>

<http://stackoverflow.com/questions/5624548/limit-each-list-in-rails>

Just limit it on your ActiveRecord level (or SQL)

@shops = Shop.limit(20) # Rails 3+

@shops = Shop.all :limit => 10 # Rails 2+

Or use Ruby

<% @shops[0,20].each do |shop| %>

<li><%= shop.name %></li>

<% end %>

@shops.take(20).each.

here is the code with for loop having an index: this was in users/index.html.erb

<ul>

<% @users.each\_with\_index do |first\_user, i| %>

break if i==1

</br>

<% @users.each\_with\_index do |second\_user, j| %>

break if j==1

<% if first\_user.communicating?(second\_user) %>

<% @conversation\_items = first\_user.conversation\_list(first\_user, second\_user) %>

<%= first\_user.name %>

<%= second\_user.name %>

<% if @conversation\_items.any? %>

<ol class="microposts">

<%= render partial: 'conversation\_item', collection: @conversation\_items %>

</ol>

<%#= will\_paginate @conversation\_items, renderer: BootstrapPagination::Rails %>

<% end %>

<%#= conversation\_item.content %>

<li><%#= second\_user.name %></li>

<% else %>

<% end %>

<% end %>

<% end %>

</ul>