Creating Blocitoff Steps

Here are the steps I took to create the Blocitoff To-Do App:

In my “code” directory

**$ rails new Blocitoff –T**

**$ cd Blocitoff**

$ mv README.rdoc README.md

$ rake db:create

Edited GEM File:

**- gem 'sqlite3'  
 group :production do**

**gem 'pg'**

**gem 'rails\_12factor'**

**end**

**group :development do**

**gem 'sqlite3'**

**end**

**gem 'bootstrap-sass'**

**gem 'faker'**

**gem 'devise'**

**gem 'figaro', '1.0'**

**gem 'carrierwave'**

**gem 'mini\_magick'**

**gem 'fog'**

**gem 'simple\_form'**

**gem 'whenever', :require => false**

**$ bundle install --without production**

$ rails generate simple\_form:install - -bootstrap

Signed in to my [**GitHub**](https://github.com/) account and created a new repo. (<https://github.com/ArtJSnyder/Bloccitoff.git>)

$ git init

$ git add .

$ git commit -m 'First commit and README update'

$ git remote add origin https://github.com/ArtJSnyder/Bloccitoff.git

$ git push -u origin master

$ rails s <Just to check to make sure the basic app was working>

$ git branch #=> should be on your master branch

$ git checkout -b welcome-controller

$ git branch #=> should be on the new welcome-controller branch

**$ rails g devise:install**

$ rails g controller welcome index about

added a root path to the config/routes.rb file:

**root to: 'welcome#index'**

Edit:

app/views/layouts/application.html.erb

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Blocitoff</title>**

**<%= stylesheet\_link\_tag 'application', media: 'all', 'data-turbolinks-track' => true %>**

**<%= javascript\_include\_tag 'application', 'data-turbolinks-track' => true %>**

**<%= csrf\_meta\_tags %>**

**</head>**

**<body>**

**<div class="container">**

**<ul class="nav nav-tabs">**

**<li><%= link\_to "Home", root\_path %></li>**

**<li><%= link\_to "About", welcome\_about\_path %></li>**

**<div class="pull-right user-info"> <!-- note the user-info class is referenced in spec/features/ to test signin link -->**

**<% if current\_user %>**

**<%= image\_tag(current\_user.avatar.tiny.url) if current\_user.avatar? %>**

**Hello <%= link\_to (current\_user.name || current\_user.email), current\_user %>!**

**<%= link\_to edit\_user\_registration\_path, class: 'btn btn-primary btn-sm' do %>**

**<span class="glyphicon glyphicon-user"></span>**

**<% end %>**

**<%= link\_to destroy\_user\_session\_path, method: :delete, class: 'btn btn-primary btn-sm' do %>**

**<span class="glyphicon glyphicon-log-out"></span>**

**<% end %>**

**<% else %>**

**<%= link\_to "Sign In", new\_user\_session\_path %> or**

**<%= link\_to "Sign Up", new\_user\_registration\_path %>**

**<% end %>**

**</div>**

**</ul>**

**<% if flash[:notice] %>**

**<div class="alert alert-success">**

**<button type="button" class="close" data-dismiss="alert">&times;</button>**

**<%= flash[:notice] %>**

**</div>**

**<% elsif flash[:error] %>**

**<div class="alert alert-danger">**

**<button type="button" class="close" data-dismiss="alert">&times;</button>**

**<%= flash[:error] %>**

**</div>**

**<% elsif flash[:alert] %>**

**<div class="alert alert-warning">**

**<button type="button" class="close" data-dismiss="alert">&times;</button>**

**<%= flash[:alert] %>**

**</div>**

**<% end %>**

**<%= yield %>**

**</div>**

**</body>**

**</html>**

app/views/welcome/about.html.erb

<br>

<div class="jumbotron">

<h1>Blocitoff</h1>

<p>Blocitoff is a web-based resource for creating To-Do Lists to help you stay organized!</p>

<p>

<% if !@user.present? %>

<%= link\_to "Sign In", new\_user\_session\_path, class: 'btn btn-primary' %> or

<%= link\_to "Sign Up", new\_user\_registration\_path, class: 'btn btn-primary' %> today!

<% else %>

Click here for <%= link\_to "My To-Do Items", user\_path(@user), class: 'btn btn-primary' %>

<% end %>

</p>

</div>

<div class="row">

<div class="col-md-4">

<h2>Keep lists clean and up to date.</h2>

<p>Blocitoff will automaically delete tasks seven days after creation to keep only the most important items on your list.</p>

</div>

<div class="col-md-4">

<h2>Keeps you motivated and focused.</h2>

<p>Blocitoff will show you many days remain before a to-do item is automatically deleted.</p>

</div>

<div class="col-md-4">

<h2>Keeps you organized with ease.</h2>

<p>Blocitoff will allow you to mark to-do items as complete and have them removed fro your list.</p>

</div>

</div>

app/assets/stylesheets/welcome.css.scss

**.jumbotron {**

**background: #ff0000;**

**border: 5px #accedd solid;**

**margin: 15px;**

**padding: 5px;**

**}**

$ mv app/assets/stylesheets/application.css app/assets/stylesheets/application.css.scss

Then open it up and remove any code there. Add the following line (it should be the only line):

app/assets/stylesheets/application.css.scss

**@import "bootstrap";**

Include the Bootstrap file in **app/assets/javascripts/application.js**:

app/assets/javascripts/application.js

***//= require jquery***

***//= require jquery\_ujs***

***//= require turbolinks***

**+ *//= require bootstrap***

***//= require\_tree .***

$ git add .

**$ git rm app/assets/stylesheets/application.css**

$ git commit -m 'welcome controller, boostrap, & css'

$ git checkout master

$git merge welcome-controller

$git branch –d welcome-controller

$git push

$ git checkout -b users

Open **config/environments/development.rb** and add the following lines at the bottom of the file (above **end**):

config/environments/development.rb

**...**

**config.action\_mailer.default\_url\_options = { host: 'http://learning-2-code-171173.usw1.nitrousbox.com:3000/' }**

**config.action\_mailer.delivery\_method = :smtp**

**config.action\_mailer.perform\_deliveries = true**

**end**

**$ rails g devise:views**

**$ rails g devise User**

modify the migration file though, so open up the migration file and add the **t.string :name** customization below:

db/migrate/[timestamp]\_devise\_create\_users.rb

**class DeviseCreateUsers < ActiveRecord::Migration**

**def change**

**create\_table(:users) do |t|**

***## Customization***

**t.string :name**

**...**

**end**

**end**

**end**

*We're adding this, because we want our users to be able to store names.*

You'll also want to uncomment all four lines under **## Confirmable**:

**confirmable***is an option used to validate / confirm user's email after they sign-up for Bloccit.*

db/migrate/[timestamp]\_devise\_create\_users.rb

**...**

***# t.string :confirmation\_token***

***# t.datetime :confirmed\_at***

***# t.datetime :confirmation\_sent\_at***

***# t.string :unconfirmed\_email # Only if using reconfirmable***

**t.string :confirmation\_token**

**t.datetime :confirmed\_at**

**t.datetime :confirmation\_sent\_at**

**t.string :unconfirmed\_email *# Only if using reconfirmable***

Now you can run the migration:

Terminal

**$ rake db:migrate**

Open our new **app/models/user.rb** file and add **, :confirmable** to the list of devise "modules" (to the right of **:validatable**):

app/models/user.rb

**class User < ActiveRecord::Base**

***# Include default devise modules. Others available are:***

***# :confirmable, :lockable, :timeoutable and :omniauthable***

**devise :database\_authenticatable, :registerable,**

**:recoverable, :rememberable, :trackable, :validatable, :confirmable**

**end**

<NEED TO EDIT DEVISE VIEWS TO ADD NAME FIELD TO NEW AND EDIT REGISTRATIONS>

$ heroku login

$heroku create

Terminal

$ git push heroku master

You can now see the web address for your application in Production by typing the following:

Terminal

$ heroku apps:info

You can open the app directly like so:

Terminal

$ heroku open

**$ heroku addons:create sendgrid:starter**

* Verify that Sendgrid was installed by typing the following on the command line:

Terminal

**$ heroku addons**

If you see **sendgrid:starter**, you've successfully installed Sendgrid.

Create a file in **config/initializers** named **setup\_mail.rb**, and add the following code:

config/initializers/setup\_mail.rb

**if Rails.env.development? || Rails.env.production?**

**ActionMailer::Base.delivery\_method = :smtp**

**ActionMailer::Base.smtp\_settings = {**

**address: 'smtp.sendgrid.net',**

**port: '587',**

**authentication: :plain,**

**user\_name: ENV['SENDGRID\_USERNAME'],**

**password: ENV['SENDGRID\_PASSWORD'],**

**domain: 'heroku.com',**

**enable\_starttls\_auto: true**

**}**

**end**

**$ figaro install**

config/application.yml

**SENDGRID\_PASSWORD: ajs48047**

**SENDGRID\_USERNAME: ArtJSnyder**

**AWS\_ACCESS\_KEY\_ID: AKIAI5ZKSKUEXABTJ67Q**

**AWS\_SECRET\_ACCESS\_KEY: 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**

**AWS\_REGION: us-west-1**

**development:**

**AWS\_BUCKET: ajs-learning-development**

**production:**

**AWS\_BUCKET: ajs-learning-production**

Create a new file in your **config** directory named **application.example.yml**. This file -- which will not be ignored and so **will show up on GitHub** -- exists simply to demonstrate to collaborators what variables are expected, and how they should be formatted. Add the following to it:

config/application.example.yml

**SENDGRID\_PASSWORD:**

**SENDGRID\_USERNAME:**

*This****says that****we're storing our Sendgrid password and username as ENV variables, but doesn't actually share that the values of these keys. This is important --****do not provide values in the example file, as it will be displayed on GitHub.***

**$ figaro heroku:set -e production**

To make sure your Heroku environment variables are correct, run:

Terminal

**$ heroku config**

**NEXT STEPS NOT REQUIRED ON NITROUS MACHINE**

**Terminal**

**$ brew update**

**$ brew install imagemagick**

**$ rails generate uploader Avatar**

**$ rails g migration AddAvatarToUsers avatar:string**

**$ rake db:migrate**

. Declare**mount\_uploader** in **user.rb**:

app/models/user.rb

**has\_many :posts**

**mount\_uploader :avatar, AvatarUploader**

*We give***mount\_uploader***a second argument --***AvatarUploader***-- to tell it which file to use for uploading avatars.*

Create an **app/controllers/users\_controller.rb** file:

app/controllers/users\_controller.rb

**class UsersController < ApplicationController**

**before\_action :authenticate\_user!, except: [:show]**

**def update**

**if current\_user.update\_attributes(user\_params)**

**flash[:notice] = "User information updated"**

**redirect\_to edit\_user\_registration\_path**

**else**

**flash[:error] = "Invalid user information"**

**redirect\_to edit\_user\_registration\_path**

**end**

**end**

**private**

**def user\_params**

**params.require(:user).permit(:name, :avatar, :email\_favorites)**

**end**

**end**

EDIT:

app/uploaders/avatar\_uploader.rb

***# encoding: utf-8***

**class AvatarUploader < CarrierWave::Uploader::Base**

***# Include RMagick or MiniMagick support:***

***# include CarrierWave::RMagick***

***# include CarrierWave::MiniMagick***

**include CarrierWave::MiniMagick**

***# Choose what kind of storage to use for this uploader:***

**storage :file**

***# storage :fog***

**storage :fog**

***# Override the directory where uploaded files will be stored.***

***# This is a sensible default for uploaders that are meant to be mounted:***

**def store\_dir**

**"uploads/**#{model.**class**.to\_s.underscore}**/**#{mounted\_as}**/**#{model.id}**"**

**end**

***# Provide a default URL as a default if there hasn't been a file uploaded:***

***# def default\_url***

***# # For Rails 3.1+ asset pipeline compatibility:***

***# # ActionController::Base.helpers.asset\_path("fallback/" + [version\_name, "default.png"].compact.join('\_'))***

***#***

***# "/images/fallback/" + [version\_name, "default.png"].compact.join('\_')***

***# end***

***# Process files as they are uploaded:***

**- *# process :scale => [200, 300]***

**process :resize\_to\_fill => [200, 300]**

**- *#***

**- *# def scale(width, height)***

**- *# # do something***

**- *# end***

***# Create different versions of your uploaded files:***

***# version :thumb do***

***# process :resize\_to\_fit => [50, 50]***

***# end***

**version :tiny do**

**process resize\_to\_fill: [20, 20]**

**end**

**version :small do**

**process resize\_to\_fill: [30, 30]**

**end**

**version :profile do**

**process resize\_to\_fill: [45, 45]**

**end**

***# Add a white list of extensions which are allowed to be uploaded.***

***# For images you might use something like this:***

***# def extension\_white\_list***

***# %w(jpg jpeg gif png)***

***# end***

**def extension\_white\_list**

**%w(jpg jpeg gif png)**

**end**

***# Override the filename of the uploaded files:***

***# Avoid using model.id or version\_name here, see uploader/store.rb for details.***

***# def filename***

***# "something.jpg" if original\_filename***

***# end***

**end**

Create a file named **fog.rb** in the **config/initializers/** directory. Add the following code:

*Initializer names are arbitrary; all initializers are run on server launch. We could have called our Fog initializer something else, but that would have been confusing.*

config/initializers/fog.rb

**CarrierWave.configure do |config|**

**config.fog\_credentials = {**

**provider: 'AWS',**

**aws\_access\_key\_id: ENV['AWS\_ACCESS\_KEY\_ID'],**

**aws\_secret\_access\_key: ENV['AWS\_SECRET\_ACCESS\_KEY'],**

**region: ENV['AWS\_REGION']**

**}**

**config.fog\_directory = ENV['AWS\_BUCKET']**

**config.fog\_public = true**

**end**

***# Ref:***

***# https://support.cloud.engineyard.com/entries/20996881-Use-CarrierWave-and-Optionally-Fog-to-Upload-and-Store-Files#update3***

***# http://stackoverflow.com/questions/7946819/carrierwave-and-amazon-s3***

Open Devise **registrations#edit** and replace the code w/ the following:

app/views/devise/registrations/edit.html.erb

**<h2>Edit <%= resource\_name.to\_s.humanize %></h2>**

**<div class="row">**

**<div class="col-md-8">**

**<h3>Change email or password</h3>**

**<%= form\_for(resource, as: resource\_name, url: registration\_path(resource\_name), html: { method: :patch }) do |f| %>**

**<%= devise\_error\_messages! %>**

**<div class="form-group">**

**<%= f.label :email %>**

**<%= f.email\_field :email, class: 'form-control', placeholder: "Enter email" %>**

**</div>**

**<% if devise\_mapping.confirmable? && resource.pending\_reconfirmation? %>**

**<div class="form-group">**

**<p>Waiting confirmation for:</p>**

**<%= resource.unconfirmed\_email %>**

**</div>**

**<% end %>**

**<div class="form-group">**

**<%= f.label :password %>**

**<%= f.password\_field :password, class: 'form-control', placeholder: "Enter password" %>**

**<i>(leave blank if you don't want to change it)</i>**

**</div>**

**<div class="form-group">**

**<%= f.label :password\_confirmation %>**

**<%= f.password\_field :password\_confirmation, class: 'form-control', placeholder: "Enter password confirmation" %>**

**</div>**

**<div class="form-group">**

**<%= f.password\_field :current\_password, class: 'form-control', placeholder: "Enter password" %>**

**<i>(we need your current password to confirm your changes)</i>**

**</div>**

**<div class="form-group">**

**<%= f.submit "Update", class: 'btn btn-success' %>**

**</div>**

**<% end %>**

**<h3>Edit personal information</h3>**

**<%= form\_for(current\_user) do |f| %>**

**<div class="form-group">**

**<%= f.label :name %>**

**<%= f.text\_field :name, class: 'form-control', placeholder: "Enter name", autofocus: true %>**

**</div>**

**<% if current\_user.avatar? %>**

**<div class="form-group">**

**<p>Current avatar</p>**

**<%= image\_tag( current\_user.avatar.profile.url ) %>**

**</div>**

**<% end %>**

**<div class="form-group">**

**<%= f.label :avatar %>**

**<%= f.file\_field :avatar %>**

**<%= f.hidden\_field :avatar\_cache %>**

**</div>**

**<div class="form-group">**

**<%= f.submit "Update", class: 'btn btn-success' %>**

**</div>**

**<% end %>**

**<h3>Cancel my account</h3>**

**<div class="form-group">**

**<p><%= button\_to "Cancel my account", registration\_path(resource\_name), data: { confirm: "Are you sure?" }, method: :delete, class: 'btn btn-danger' %></p>**

**</div>**

**</div>**

**</div>**

config/routes.rb

**devise\_for :users**

**resources :users, only: [:update]**

Add the following to allow Devise to accept the NAME: field:

app/controllers/application\_controller.rb

**before\_action :configure\_permitted\_parameters, if: :devise\_controller?**

**protected**

**def configure\_permitted\_parameters**

**devise\_parameter\_sanitizer.for(:sign\_up) << :name**

**end**

config/environments/production.rb

**# http://artbloccit.herokuapp.com**

**config.action\_mailer.default\_url\_options = { host: 'artbloccit.herokuapp.com' }**

**$ figaro heroku:set -e production**

**$ git push heroku master**

**$ heroku run rake db:migrate**

**$ heroku restart**

$ rails g model Item name:string user:references:index

A user has\_many :items and an item belongs\_to :user. Update user.rb < has\_many :items, dependent: :destroy > and item.rb < belongs\_to :user Note: this is automatically added by the generator because we used references> to reflect this association.

Nest the items resource under resources :users inroutes.rb. You know you will need two **item** actions (:create, :destroy) so feel free to add them both at this time, although you only need create to get started.

In routes.rb

resources :users, only: [:update, :show] do

resources :items, only: [:create, :destroy]

end

$ rake db:migrate

$ rails g controller items create destroy

 Modify ItemsController  <write methods for create & destroy>

class ItemsController < ApplicationController

def destroy

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

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

if @item.destroy

flash[:notice] = "To-Do Item was removed."

else

flash[:error] = "To-Do Item couldn't be deleted. Try again."

end

respond\_to do |format|

format.html

format.js

end

end

def create

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

@item = current\_user.items.build( item\_params )

@new\_item = Item.new

if @item.save

flash[:notice] = "To-Do item was created."

else

flash[:error] = "There was an error saving the To-Do item. Please try again."

end

respond\_to do |format|

format.html

format.js

end

end

private

def item\_params

params.require(:item).permit(:name, :user\_id)

end

end

end

 Modify UsersController <add method for user show>

def show

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

@items = Item.where(user\_id: @user.id)

@item = @items.new

end

Create an items/\_item.html.erb partial in a new items view directory so users can submit new items. Create a partial named \_item.html.erb in the items directory. Use it in users#show to show the body of each item already associated with the user.

Create an items/\_item.html.erb and add this code:

<%= content\_tag :div, class: 'media', id: "item-#{item.id}" do %>

<div class="media-body">

<div class="row">

<div class="col-md-4"><%= time\_ago\_in\_words(item.created\_at) %> ago (days remaining <%= item.days\_left %>)</div>

<div class="col-md-7"><%= item.name %></div>

<div class="col-md-1"><%= link\_to "", [item.user, item], method: :delete, remote: true, class: 'glyphicon glyphicon-ok', data: { confirm: 'Are you sure you want to mark this item as complete?' } %></div>

<hr>

<% end %>

</div>

</div>

Create an items/\_form.html.erb and add this code:

<div class="col-md-8">

<%= simple\_form\_for([@user, @user.items.new], remote: true, html: { class: 'form-horizontal' }) do |f| %>

<%= f.input :name, placeholder: "Enter To-Do Item" %>

<div class="form-group">

<%= f.button :submit, { class: 'btn btn-success' } %>

</div>

</div>

<% end %>

Create a users/show.htlm view to show the user profile and call the two partials to display the items and to produce the new form to create a new item:

<div class="row">

<div class="col-md-8">

<div class="media">

<% if @user.avatar? %>

<%= image\_tag @user.avatar.url, class: 'pull-left' %>

<% end %>

<div class="media-body">

<h2 class="media-heading"><%= @user.name %> Profile</h2>

<p>E-Mail: <%= @user.email %></p>

<p>Name: <%= @user.name %></p>

<%= link\_to "Edit", edit\_user\_registration\_path(@user), class: 'btn btn-success' %>

</div></div></div></div><hr>

<div class="row">

<h3>

<div class="col-md-4">To-Do Item Created:</div>

<div class="col-md-6">To-Do Item Name:</div>

<div class="col-md-2">Check When Complete:</div>

</h3></div><hr>

<div class='js-items'>

<%= render partial: 'items/item', collection: @items %>

</div>

<div class='new-item'>

<div class="col-md-8">

<%= render partial: 'items/form', locals: { item: @item } %>

Rename items/destroy.html.erb to items/destroy.js.erb  and add this code:

<% if @item.destroyed? %>

$('#item-' +<%= @item.id %>).fadeOut(850);

<% else %>

$('#item-' +<%= @item.id %>).prepend("<div class='alert alert-danger'><%= flash[:error] %></div>");

<% end %>

Rename items/create.html.erb to items/create.js.erb  and add this code:.

<% if @item.valid? %>

$('.js-items').prepend("<%= escape\_javascript(render(@item)) %>");

$('.new-item').html("<%= escape\_javascript(render partial: 'items/form', locals: {user: @user, item: @new\_item}) %>");

<% else %>

$('.new-item').html("<%= escape\_javascript(render partial: 'items/form', locals: { user: @user, item: @item }) %>");

<% end %>

Add the following code to models/item.rb

default\_scope { order('created\_at DESC') }

def days\_left

7 - (DateTime.now.to\_date - created\_at.to\_date).to\_i

end

Add this to application\_contoller.rb

before\_action :configure\_permitted\_parameters, if: :devise\_controller?

def after\_sign\_in\_path\_for(resource) #This overrides Devise default and redirects path to user show

user\_path(resource)

end

protected

def configure\_permitted\_parameters

devise\_parameter\_sanitizer.for(:sign\_up) << :name

end