Prairie Hill Learning Center Website

AnderSon

April 16, 2015

Prairie Hill Learning Center

- ☐ fix change/forgot password issue
- ⊟ rebuild ccf volunteer app

  `./config/routes.rb`

```
Rails.application.routes.draw do

  namespace :api, defaults: {format: 'json'} do
    #namespace :v1 do
    resources :activities, :pages, :shifts, :volunteers, :users
    #end
  end

  resources :activities
  resources :shifts
  resources :volunteers

  match '/contacts', to: 'contacts#new', via: 'get'
  resources "contacts", only: [:new, :create]

  comfy_route :cms_admin, :path => '/admin'

  devise_for :users
  resources :pages

  root "pages#home"
```

```
  get "about" => "pages#about"
  get "news" => "pages#news"
  get "events" => "pages#events"
  get "programs" => "pages#programs"
  get "calendar" => "pages#calendar"
  get "contact" => "pages#contact"
  get "staffandboard" => "pages#staff"
  get "jobs" => "pages#jobs"
  get "donate" => "pages#donate"
  get "camp" => "pages#summer_camp"
  get "csv" => "pages#csvupload"
  get "uniq" => "pages#unique"
  get "ccf" => "shifts#volunteer"
  get "user_shifts" => "shifts#user_shifts"

  # Make sure this routeset is defined last
  comfy_route :cms, :path => '/', :sitemap => true
end
```

- ☒ backup volunteer data
  - ☒ check api access to user data
    - ☒ update api to authenticate requests
      `http://railscasts.com/episodes/352-securing-an-api?view=asciicast`
    - ☒ Basic
      ```
      http_basic_authenticate_with name: "admin", password: "secret"
      ```
    - ☒ ruby?
      `https://gist.github.com/kyletcarlson/7911188` `http://www.rubyinside.com/nethttp-cheat-sheet-2940.html`
      ```
      require "net/http"
      require "uri"

      uri = URI.pasre("http://www.prairiehill.com/api/users")
      ```
  - ☒ user info
  - ☒ last years activity/shift data
- ⊟ re-organize resource relationships
  - ☐ destroy volunteer resource?

* ⊟ Devise User/Volunteer

  ./db/migrate ./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

  validates :username, presence: true, length: {maximum: 255}, uniqueness

  has_many :shifts
  has_many :activities through: :shifts

  # Virtual attribute for authenticating by either username or email
  # This is in addition to a real persisted field like 'username'
  attr_accessor :login


  def self.find_first_by_auth_conditions(warden_conditions)
    conditions = warden_conditions.dup
    if login = conditions.delete(:login)
      # when allowing distinct User records with, e.g., "username" and "U
      # where(conditions).where(["lower(username) = :value OR lower(email
      where(conditions).where(["username = :value OR lower(email) = lower
    else
      where(conditions).first
    end
  end

  #### This is the correct method you override with the code above
  #### def self.find_for_database_authentication(warden_conditions)
  #### end
end
```

  · attributes
  · id
  · email
  · username
  · name

- admin
- $\text{first}_{\text{name}}$
- $\text{last}_{\text{name}}$
- phone
- ☐ has guest?
- ☒ has many shifts
- ☒ has many activities through shifts

* ⊟ Activity

./app/models/activity.rb

```
class Activity < ActiveRecord::Base

  has_many :shifts

  def self.to_csv(options = {})
    CSV.generate(options) do |csv|
      csv << column_names
      all.each do |activity|
csv << activity.attributes.values_at(*column_names)
      end
    end
  end
end
```

- ☒ has many shifts
- ☐ belongs to users

* ☐ Shifts

./app/models/shift.rb

```
class Shift < ActiveRecord::Base
  has_and_belongs_to_many :users, :dependent => :destroy
  accepts_nested_attributes_for :users


  def self.to_xlsx(options = {})

    workbook = WriteExcel.new('shifts.xlsx')
#    workbook = WriteExcel.new(STDOUT)
```

```
    @shiftTitles = all.pluck(:title).uniq
    @shiftTitles.each do |title|

      worksheet = workbook.add_worksheet

      # format = workbook.add_format
      # format.set_bold
      # format.set_color('red')
      # format.set_align('right')

      worksheet.write(0, 0, title)

      @shifts_by_title = all.where(title: title)
      @shifts_by_title.each do |shift|
worksheet.write(1, 1, 'hotdog' )#shift.title)
      end
    end

    workbook.close

  end


  def self.to_csv(options = {})
    CSV.generate(options) do |csv|
      csv << ["", "Time", "Volunteer", "Guest Volunteer"]
      @shiftTitles = all.pluck(:title).uniq
      @shiftTitles.each do |title|
csv << [title]
@shifts_by_title = all.where(title: title)
@shifts_by_title.each do |shift|
  csv << ["", shift.time, shift.volunteer, shift.guest]
end
      end
    end
  end

  # def self.to_csv(options = {})
  #   CSV.generate(options) do |csv|
```

```
#     csv << ["", "Time", "Volunteer", "Guest Volunteer"]
#     @shiftTitles = all.pluck(:title).uniq

#     @shiftTitles.each do |title|
#       csv << [title]

#       @shifts_by_title = all.where(title: title)
#       @shifts_by_title.each do |shift|

#         csv << ["", shift.time, shift.volunteer, shift.guest]
#       end
#     end

#   end
# end

# def self.to_csv(options = {})
#   CSV.generate(options) do |csv|
#     csv << column_names
#     all.each do |shift|
#       csv << shift.attributes.values_at(*column_names)
#     end
#   end
# end

def add_user_idee(id)

  user_ids_will_change!
  update_attribute(:user_ids, self.user_ids << id)

  self.save

end

def cancel_shift

  shift.volunteer = nil
  shift.save

end
```

```
end
```

        - ☐ has guest?
        - ☐ belongs to activity
        - ☐ belongs to users
        - ☐ has guest?

- ☐ build an API

  `https://codelation.com/blog/rails-restful-api-just-add-water`

  - ☐ add to `./Gemfile`

```
gem 'jbuilder'
gem 'kaminari'
gem 'responders'

source 'http://rubygems.org'
ruby '2.2.0'

gem 'rails', '4.2.1'
gem 'sass-rails'
gem 'compass-rails', '~> 2.0.alpha.0'
gem 'uglifier', '2.5.1'
gem 'coffee-rails', '4.0.1'
gem 'jquery-rails', '3.1.1'
gem 'turbolinks'
gem 'jquery-turbolinks'
gem 'jbuilder'
gem 'kaminari'
gem 'responders'
gem 'bootstrap-sass'
gem 'bcrypt'
gem 'devise'
gem 'pg'
gem 'comfortable_mexican_sofa', '1.12.7'
gem 'sdoc', '~> 0.4.0',          group: :doc
gem 'aws-sdk', '~> 1.46.0'
gem 'mail_form'
gem 'simple_form'
```

```
gem 'cells'
gem 'inherited_resources', github: 'josevalim/inherited_resources', branch: '
gem 'skrollr-rails'
gem 'rails_admin'
gem 'picturefill'
gem 'autoprefixer-rails'
gem 'chronic'
gem 'acts_as_xlsx'
gem 'axlsx'
gem 'axlsx_rails'
gem 'rubyzip'
gem 'writeexcel', '1.0.5'
gem 'figaro'
gem 'meta-tags'
gem 'metamagic'
gem 'safe_yaml', '1.0.4'
gem 'sitemap_generator'
gem 'dynamic_sitemaps'

# Spring speeds up development by keeping your application running in the bac
gem 'spring',        group: :development

group :development, :test do
  gem 'sqlite3'
  gem 'foreman'
  gem 'pry-rails'
  gem 'unicorn'
end

group :production do
#  gem 'pg', '0.15.1'
  gem 'rails_12factor'
#  gem 'unicorn'
  gem 'unicorn-rails'
end
```

- ☐ controllers
  - ☐ create file `./app/controllers/api/base_controller.rb`

```
mkdir app/controllers/api

module Api
  class BaseController < ApplicationController
    protect_from_forgery with: :null_session
    before_action :set_resource, only: [:destroy, :show, :update]
    respond_to :json

    private

    # Returns the resource from the created instance variable
    # @return [Object]
    def get_resource
      instance_variable_get("@#{resource_name}")
    end

    # Returns the allowed parameters for searching
    # Override this method in each API controller
    # to permit additional parameters to search on
    # @return [Hash]
    def query_params
      {}
    end

    # Returns the allowed parameters for pagination
    # @return [Hash]
    def page_params
      params.permit(:page, :page_size)
    end

    # The resource class based on the controller
    # @return [Class]
    def resource_class
      @resource_class ||= resource_name.classify.constantize
    end

    # The singular name for the resource class based on the controller
    # @return [String]
    def resource_name
      @resource_name ||= self.controller_name.singularize
```

```
    end

    # Only allow a trusted parameter "white list" through.
    # If a single resource is loaded for #create or #update,
    # then the controller for the resource must implement
    # the method "#{resource_name}_params" to limit permitted
    # parameters for the individual model.
    def resource_params
      @resource_params ||= self.send("#{resource_name}_params")
    end

    # Use callbacks to share common setup or constraints between actions
    def set_resource(resource = nil)
      resource ||= resource_class.find(params[:id])
      instance_variable_set("@#{resource_name}", resource)
    end
  end
end
```

* ☐ add the public resource methods to the same controller

```
# POST /api/{plural_resource_name}
def create
  set_resource(resource_class.new(resource_params))

  if get_resource.save
    render :show, status: :created
  else
    render json: get_resource.errors, status: :unprocessable_entity
  end
end

# DELETE /api/{plural_resource_name}/1
def destroy
  get_resource.destroy
  head :no_content
end

# GET /api/{plural_resource_name}
def index
```

```
  plural_resource_name = "@#{resource_name.pluralize}"
  resources = resource_class.where(query_params)
              .page(page_params[:page])
              .per(page_params[:page_size])

  instance_variable_set(plural_resource_name, resources)
  respond_with instance_variable_get(plural_resource_name)
end

# GET /api/{plural_resource_name}/1
def show
  respond_with get_resource
end

# PATCH/PUT /api/{plural_resource_name}/1
def update
  if get_resource.update(resource_params)
    render :show
  else
    render json: get_resource.errors, status: :unprocessable_entity
  end
end

module Api
  class BaseController < ApplicationController
    protect_from_forgery with: :null_session
    before_action :set_resource, only: [:destroy, :show, :update]
    respond_to :json

    # POST /api/{plural_resource_name}
    def create
      set_resource(resource_class.new(resource_params))

      if get_resource.save
render :show, status: :created
      else
render json: get_resource.errors, status: :unprocessable_entity
      end
    end
```

```
    # DELETE /api/{plural_resource_name}/1
    def destroy
      get_resource.destroy
      head :no_content
    end

    # GET /api/{plural_resource_name}
    def index
      plural_resource_name = "@#{resource_name.pluralize}"
      resources = resource_class.where(query_params)
  .page(page_params[:page])
  .per(page_params[:page_size])

      instance_variable_set(plural_resource_name, resources)
      respond_with instance_variable_get(plural_resource_name)
    end

    # GET /api/{plural_resource_name}/1
    def show
      respond_with get_resource
    end

    # PATCH/PUT /api/{plural_resource_name}/1
    def update
      if get_resource.update(resource_params)
render :show
      else
render json: get_resource.errors, status: :unprocessable_entity
      end
    end

    private

    # Returns the resource from the created instance variable
    # @return [Object]
    def get_resource
      instance_variable_get("@#{resource_name}")
    end

    # Returns the allowed parameters for searching
```

```
  # Override this method in each API controller
  # to permit additional parameters to search on
  # @return [Hash]
  def query_params
    {}
  end

  # Returns the allowed parameters for pagination
  # @return [Hash]
  def page_params
    params.permit(:page, :page_size)
  end

  # The resource class based on the controller
  # @return [Class]
  def resource_class
    @resource_class ||= resource_name.classify.constantize
  end

  # The singular name for the resource class based on the controller
  # @return [String]
  def resource_name
    @resource_name ||= self.controller_name.singularize
  end

  # Only allow a trusted parameter "white list" through.
  # If a single resource is loaded for #create or #update,
  # then the controller for the resource must implement
  # the method "#{resource_name}_params" to limit permitted
  # parameters for the individual model.
  def resource_params
    @resource_params ||= self.send("#{resource_name}_params")
  end

  # Use callbacks to share common setup or constraints between actions
  def set_resource(resource = nil)
    resource ||= resource_class.find(params[:id])
    instance_variable_set("@#{resource_name}", resource)
  end
end
```

```
end
```

* ☐ connect base controller to model controllers

  Pay attention that these inherit from *Api::BaseController*

  ./app/controllers/api/users_controller.rb

```
module Api
  class UsersController < Api::BaseController
    #http_basic_authenticate_with name: "admin", password: "secret"
    http_basic_authenticate_with name: "admin", password: ENV["API_PASS"

    private

    def activity_params
      params.require(:activity).permit(:email, :username, :name, :admin,
    end

    def query_params
      params.permit(:activity).permit(:email, :username, :name, :admin,
    end

  end
end
```

  ./app/controllers/api/activities_controller.rb

```
module Api
  class ActivitiesController < Api::BaseController

    private

    def activity_params
      params.require(:activity).permit(:work_area, :coordinator, :sign,
    end

    def query_params
      params.permit(:work_area, :coordinator, :sign, :num_tickets, :vol_
    end
```

```
  end
end
```

./app/controllers/api/pages_controller.rb

```
module Api
  class PagesController < Api::BaseController

    private

    def page_params
      params.require(:page).permit(:title, :description)
    end

    def query_params
      params.permit(:title, :description)
    end

  end
end
```

./app/controllers/api/shifts_controller.rb

```
module Api
  class ShiftsController < Api::BaseController

    private

    def shift_params
      params.require(:shift).permit(:title, :time, :vols_needed, :volunt
    end

    def query_params
      params.permit(:title,  :time, :vols_needed, :volunteers, :voluntee
    end

  end
end
```

./app/controllers/api/volunteers_controller.rb

```
module Api
  class VolunteersController < Api::BaseController

    private

    def volunteer_params
      params.require(:volunteer).permit(:name, :email, :phone)
    end

    def query_params
      params.permit(:name, :email, :phone)
    end

  end
end
```

- ☐ routing

  ./config/routes.rb

  ```
  namespace :api do
    resources :logs, :periods
  end
  ```

- ☐ serializing data

  ```
  mkdir app/views/api /shifts etc
  ```

  * ☐ ./app/views/api/users/index.json.jbuilder

    ```
    json.users @users do |user|
      json.id user.id
      json.email user.email
      json.username user.username
      json.name user.name
      json.admin user.admin
      json.first_name user.first_name
      json.last_name user.last_name
      json.phone user.phone
    ```

```
  #json.period_id log.period ? log.period_id : nil
end
```

* □ ./app/views/api/users/show.json.jbuilder

```
json.user do
  json.id  @user.id
  json.username @user.username
  json.name @user.name
  json.admin @user.admin
  json.first_name @user.first_name
  json.last_name @user.last_name
  json.phone @user.phone

  #json.period_id @log.period ? @log.period_id : nil
end
```

* □ ./app/views/api/activities/index.json.jbuilder

```
json.activities @activities do |act|
  json.id act.id
  json.work_area act.work_area
  json.coordinator act.coordinator
  json.sign act.sign
  json.comments act.comments

  #json.period_id log.period ? log.period_id : nil
end
```

* □ ./app/views/api/activities/show.json.jbuilder

```
json.activity do
  json.id  @activity.id
  json.work_area @activity.work_area
  json.coordinator @activity.coordinator
  json.sign @activity.sign
  json.comments @activity.comments

  #json.period_id @log.period ? @log.period_id : nil
end
```

* ☐ ./app/views/api/pages/index.json.jbuilder

```
json.pages @pages do |page|
  json.id page.id
  json.title page.title
  json.description page.description

  #json.period_id log.period ? log.period_id : nil
end
```

* ☐ ./app/views/api/pages/show.json.jbuilder

```
json.page do
  json.id  @page.id
  json.title @page.title
  json.description @page.description

  #json.period_id @log.period ? @log.period_id : nil
end
```

* ☐ ./app/views/api/shifts/index.json.jbuilder

```
json.shifts @shifts do |shift|
  json.id shift.id
  json.title shift.title
  json.time shift.time
  json.vols_needed shift.vols_needed
  json.volunteer shift.volunteer
  json.guest shift.guest

  #json.period_id log.period ? log.period_id : nil
end
```

* ☐ ./app/views/api/shifts/show.json.jbuilder

```
json.shift do
  json.id  @shift.id
  json.title @shift.title
  json.time @shift.time
```

```
  json.vols_needed @shift.vols_needed
  json.volunteer @shift.volunteer
  json.guest @shift.guest

  #json.period_id @log.period ? @log.period_id : nil
end
```

* ☐ `./app/views/api/volunteers/index.json.jbuilder`

```
json.volunteers @volunteers do |vol|
  json.id vol.id
  json.name vol.name
  json.email vol.email
  json.phone vol.phone

  #json.period_id log.period ? log.period_id : nil
end
```

* ☐ `./app/views/api/volunteers/show.json.jbuilder`

```
json.volunteer do
  json.id  @volunteer.id
  json.name @volunteer.name
  json.email @volunteer.email
  json.phone @volunteer.phone

  #json.period_id @log.period ? @log.period_id : nil
end
```

– ☐ security and performance concerns

* ☐ use fragment caching to make API efficient

  · ☐ `http://guides.rubyonrails.org/caching_with_rails.html#fragment-caching`

  · ☐ `https://github.com/rails/jbuilder` offers advantages in caching over libraries like `https://github.com/rails-api/active_model_serializers` because you can cache JSON templates the same way you would *erb* templates

        * ☐ secure your API, gems that we use everyday include CanCan(Can) and Devise to offer per user permissions on resources
        * ☐ include some more complex functionality like sideloading for convenience in end-user application development

- ☐ rebuild views in angular?
- ⊟ build mobile app for sign-up
  - ⊟ ruboto `http://public.dhe.ibm.com/software/dw/demos/jrubyandandroid/index.htm`
    * ☒ expose public api
    * ☐ connect application via http requests `https://developer.android.com/training/volley/index.html`
    * ☐ build mobile views
  - ☐ phonegap
- ☒ re-route `http://www.prairiehill.com` => heroku app

Essential Files

`./FILES.org`

excel export

`http://railscasts.com/episodes/362-exporting-csv-and-excel`

Description

After over 2 years of cumbersome working with the wordpress managed website content for the Prairie Hill website (whether due to the way that wordpress arranges itself or my own ignorance and lack of education in web design, php, etc), this is an attempt to try something new. Having a nice solid foundation in building an application with Rails, I feel like now is the time to build something from the ground up that will hopefully meet my needs for

control and understanding for building and modifying the backend functionality (most importantly without having to go through all of hassle of using the actual web content management editors and having so many extraneous steps, instead of just using my text editor...), as well as the front end need for admin staff to update content, which is what is important to them in the basic functionality of the site. I take care of the functionality and aesthetic; they give it the words.

NB

What we need to look at for functionality:

- mailer contact

  `http://rubyonrailshelp.wordpress.com/2014/01/08/rails-4-simple-form-and-mail-form`

  set up successfully in development

  - ☐ change heroku configs to prairiehill email authentication for production

- user accounts

  - ☐ We need USERs with authenticatable accounts

    These users will have various access to update content and that's really all that they need. However,

    * ☐ Admin/General user

      `https://github.com/plataformatec/devise/wiki/How-To:-Add-an-Admin-Role`

      We will have user accounts for general things like summer camp and country fair sign up

      We will also have admin users who also have access to CMS

      · ☐ install & configure RailsAdmin

      `https://github.com/sferik/rails_admin`

      · ☐ bundle the gem

      ```
      gem 'rails_admin'
      bundle install
      ```

- ☐ install RailsAdmin

  ```
  rails g rails_admin:install
  ```

- ☐ configure for Devise

  `https://github.com/sferik/rails_admin/wiki/Devise`

* ☐ Using ComfortableMexicanSofa for Content Management

  - ☐ already set up to use Paperclip for images
  - ☐ WYSIWYG

    `./app/assets/stylesheets/comfortable_mexican_sofa/admin/application.css`

  - ☒ editor window is very short

* ☐ Private content

  - ☐ admin vs common user accounts

* ☐ User profiles?
* ☐ Summer Camp Registration model?
* ☐ Volunteers/CCF

  - ☐ connect devise users with shifts?
  - ☐ Sign up views
  - ☐ if user signed in...
  - ☐ time to learn some jQuery!
  - ☐ FIRST: Shows Activity titles and a number of volunteers total needed

- ☐ SECOND: Clicking on one of the FIRST shows a view of specific times and number of volunteers still needed for each, just after a description of the activity itself
- ☐ checkboxes for selected desired shifts?
- ☐ ability to remove volunteer from shifts
- ☐ BLOG/NEWSfeed for news updates?
- ☐ PAGEs for general website content

ModelViewControl

Model

Pages

`./app/controllers/pages_controller.rb ./app/models/page.rb`

Page

- ☒ Create Static Pages

  `http://www.railstutorial.org/book/static_pages`

  - ☒ Generate a Pages controller

    `./app/controllers/static_pages_controller.rb ./config/routes.rb`

    ```
    rails g controller StaticPages home
    ```

Rails Generation

- Scaffolding

  - ☒ Disable scaffold stylesheet creation

    `./config/application.rb`

    ```
    config.generators do |g|
      g.stylesheets false
    end
    ```

- ☐ Generate a scaffold

  EXAMPLE

  ```
  rails g scaffold Page index
  ```

- ☐ migrate the database

  ```
  rake db:migrate
  ```

View

Skrollr

`https://github.com/reed/skrollr-rails`
???"@import 'skrollr';" in `./app/assets/stylesheets/bootstrap_and_customization.css.scss`?

- ☒ add skrollr script
  - ☒ make sure skrollr-rails is in the Gemfile

    `./Gemfile`

    ```
    gem 'skrollr-rails'
    ```

  - ☒ add the following script just before </body> tag

    `./app/views/layouts/application.html.erb`

    ```
    <script>
     (function($){
       skrollr.init({
         forceHeight: false,
         smoothScrolling: false
       }).refresh();
     } (jQuery));
    </script>
    ```

  - ☒ Place #skrollr-body div tag around <%= yield %> tag

    ```
    <div id="skrollr-body">
    ```

- ☒ require skrollr in application.js

  `./app/assets/javascripts/application.js`

  ```
  //= require skrollr
  ```

  - ☒ For IE compatibility

    ```
    //= require skrollr
    //= require skrollr.ie
    ```

  - ☒ This plugin makes hashlinks scroll nicely to their target position.

    ```
    //= require skrollr
    //= require skrollr.menu
    ```

Bootstrap-sass

- ☒ Create custom bootstrap stylesheet

  `./app/assets/stylesheets/bootstrap_and_customization.css.scss`

  - ☒ create file

    ```
    echo "@import 'bootsrap';" > app/assets/stylesheets/bootstrap_and_customizati
    ```

  NOTE Place new variables before "@import 'bootstrap'"

  - ☒ Fonts

    EXAMPLE:

    ```
    @import url(http://fonts.googleapis.com/css?family=Roboto:400,100,100italic,7
    ```

  - ☒ Variables

    ```
    $phill-grn: #3f8000;
    ```

- ☒ Require Bootstrap's Javascript, after $\text{jquery}_{\text{ujs}}$

  `./app/assets/javascripts/application.js`

  ```
  //= require jquery
  //= require jquery_ujs
  //= require bootstrap
  //= require turbolinks
  //= require_tree .
  ```

Assets

- Stylesheets

  `./app/assets/stylesheets/bootstrap_and_customization.css.scss`

- Javascripts

  - ☒ Replace turbolinks with jquery-turbolinks

    `./app/assets/javascripts/application.js`

    * ☒ Check for jquery-turbolinks in Gemfile

      `./Gemfile`

      ```
      gem 'jquery-turbolinks'
      bundle
      ```

    * ☒ remove turbolinks line

      ```
      //= require turbolinks
      ```

    * ☒ add jquery.turbolinks under bootstrap

      ```
      //= require bootstrap
      //= require jquery.turbolinks
      ```

      · ☒ Restart the server

- Images

  - ☒ css background images

    `./app/assets/stylesheets/bootstrap_and_customization.css.scss`

    ```
    background: image-url('image.jpg')
    ```

  - ☐ run the following command to precompile assets

    ```
    RAILS_ENV=production bundle exec rake assets:precompile
    ```

  - ☐ set video as background?

Views

- Application

  `./app/views/`

  - ☒ add viewport

    `./app/views/layouts/application.html.erb`

    ```
    <meta name="viewport" content="width=device-width, intial-scale=1.0">
    ```

  - ☐ Optional page refresh interval

    ```
    <meta http-equiv="REFRESH" content="60" />
    ```

- Pages

  `./app/views/pages/ ./app/views/pages/pages.org`

Control

AngularJS (Honeybadger tutorial)

This example from honeybadger may be my key to fixing the issue I am having with the the Prairie Hill volunteer sign-up. Let's try it out, first in this sample app. Once I understand what is going on and how to impliment Angular, maybe it will be a better solution than all of that erb crap I was trying to use...

`https://www.honeybadger.io/blog/2013/12/11/beginners-guide-to-angular-js-rails`

- Initial setup

  - ☒ create the project

    ```
    rails new rest --database=postgresql --skip-test-unit
    ```

  - ☐ create the PostgreSQL database user:

    ```
    createuser -P -s -e rest
    ```

- ☐ Add RSpec to your Gemfile & Install RSpec

  ./Gemfile

  ```
  gem "rspec-rails", "~> 2.14.0"

  bundle install

  rails g rspec:install
  ```

- ☐ Create the database:

  ```
  rake db:create
  ```

- Creating the Restaurant model
  - ☐ Create the Restaurant resource

    ```
    rails g scaffold restaurant name:string
    ```

  - ☐ Make sure restaurant names are unique

    ./db/migrate/

    ```
    class CreateRestaurants < ActiveRecord::Migration
      def change
        create_table :restaurants do |t|
          t.string :name

          t.timestamps
        end

        add_index :restaurants, :name, unique: true
      end
    end
    ```

    * ☐ Run the migration

      ```
      rake db:migrate
      ```

    * ☐ Add some specs...

      Need to start learning TDD, but I'm lazy right now

- Bringing AngularJS into the mix
  - ☒ Create the controller

    ```
    rails g controller static_pages index
    ```

  - ☒ Update routes

    `./config/routes.rb`

    ```
    root 'static_pages#index'
    ```

  - ☒ Download Angular

    ```
    wget http://code.angularjs.org/1.1.5/angular.js \
    http://code.angularjs.org/1.1.5/angular-mocks.js

    mv angular* app/assets/javascripts
    ```

  - ⊟ Add it to the asset pipeline

    `./app/assets/javascripts/application.js`

    * ☐ Remove turbolinks line

      Keeping it in for now as a test

    * ☐ Add the following two lines

      ```
      //= require angular
      //= require main
      ```

    * ☒ Set up the layout

      `./app/views/layouts/application.html.erb`

      naming the app via angular "phill" for simplicity keeping turbolinks code in for now until I see a real reason to take it out

- ☒ tested taking out turbolinks markup

```
<!DOCTYPE html>
<html ng-app="phill">
<head>
  <title>Rest</title>
  <%= stylesheet_link_tag    'application', media: 'all' %>
  <%= javascript_include_tag 'application' %>
  <%= csrf_meta_tags %>
</head>
<body>

<div ng-view>
  <%= yield %>
</div>

</body>
</html>
```

* ☒ Creating an Angular controller

```
mkdir -p app/assets/javascripts/angular/controllers
```

  - ☒ Create the controller

    ```
    ./app/assets/javascripts/angular/controllers/
    HomeCtrl.js.coffee

    @phill.controller 'HomeCtrl', ['$scope', ($scope) ->

    ]
    ```

  - ☒ Add an Angular route

    ```
    ./app/assets/javascripts/main.js.coffee

    # This line is related to our Angular app, not to our
    # HomeCtrl specifically. This is basically how we tell
    # Angular about the existence of our application.
    @phill = angular.module('phill', [])
    ```

```
# This routing directive tells Angular about the default
# route for our application. The term "otherwise" here
# might seem somewhat awkward, but it will make more
# sense as we add more routes to our application.
@phill.config(['$routeProvider', ($routeProvider) ->
  $routeProvider.
    otherwise({
      templateUrl: '../templates/home.html',
      controller: 'HomeCtrl'
    })
])
```

· ☒ Add an Angular template

```
mkdir public/templates
```

```
./public/templates/home.html
```

```
This is the home page
```

· ☒ An example of data binding

```
./app/assets/javascripts/angular/controllers/
HomeCtrl.js.coffee
```

```
@phill.controller 'HomeCtrl', ['$scope', ($scope) ->
  $scope.foo = 'bar'
]
```

```
./public/templates/home.html
```

```
Value of "foo": {{foo}}
```

- Doing it for real this time
  - ☐ Seed the database

    `./db/seeds.rb`

```
Restaurant.create([
  { name: "The French Laundry" },
  { name: "Chez Panisse" },
  { name: "Bouchon" },
  { name: "Noma" },
  { name: "Taco Bell" },
])
```

```
rake db:seed
```

– ☒ Creating a shift index page

```
mkdir public/templates/shifts
```

```
./public/templates/shifts/index.html
```

```
<a href="/#">index</a>
<ul ng-repeat="restaurant in restaurants">
  <li>
    <a ng-click="viewRestaurant(restaurant.id)">
      {{ restaurant.name }}
    </a>
  </li>
</ul>
```

OR rather

```
<a href="/#">Shifts</a>
<ul ng-repeat="shift in shifts">
  <li>
    <a ng-click="viewShift(shift.id)">
      {{ shift.title }}
    </a>
  </li>
</ul>
```

– ☒ Create the controller

```
./app/assets/javascripts/angular/controllers/ShiftIndexCtrl.
js.coffee
```

```
@rest.controller 'RestaurantIndexCtrl', ['$scope', '$location', '$http', ($sc
  $scope.restaurants = []
  $http.get('./restaurants.json').success((data) ->
    $scope.restaurants = data
  )
]
```

OR rather

```
@phill.controller 'ShiftIndexCtrl', ['$scope', '$location', '$http', ($scope,
  $scope.shifts = []
  $http.get('./shifts.json').success((data) ->
    $scope.shifts = data
  )
]
```

- ☒ Adjust routing configuration

  `./app/assets/javascripts/main.js.coffee`

```
@phill = angular.module('phill', [])

@phill.config(['$routeProvider', ($routeProvider) ->
  $routeProvider.
    when('/shifts', {
      templateUrl: '../templates/shifts/index.html',
      controller: 'ShiftIndexCtrl'
    }).
    otherwise({
      templateUrl: '../templates/home.html',
      controller: 'HomeCtrl'
    })
])
```

- Adding our first test

  fill in later

- Building out the shifts page

  When you generate scaffolding in Rails 4, it gives you some .jbuilder files:

  `./app/views/shifts/index.json.jbuilder`

– ☒ Add :id parameter for json.extract!

```
json.array!(@restaurants) do |restaurant|
  json.extract! restaurant, :id, :name
  json.url restaurant_url(restaurant, format: :json)
end
```

OR rather

```
json.array!(@shifts) do |shift|
  json.extract! shift, :id, :title, :vols_needed, :user_ids
  json.url shift_url(shift, format: :json)
end
```

– ☐ define pushShift()

./app/assets/javascripts/angular/controllers/ShiftIndexCtrl.
js.coffee

– ☒ define viewShift()

./app/assets/javascripts/angular/controllers/ShiftIndexCtrl.
js.coffee

```
@rest.controller 'RestaurantIndexCtrl', ['$scope', '$location', '$http', ($sc
  $scope.restaurants = []
  $http.get('./restaurants.json').success((data) ->
    $scope.restaurants = data
  )

  $scope.viewRestaurant = (id) ->
    $location.url "/restaurants/#{id}"
]
```

OR rather

```
@phill.controller 'ShiftIndexCtrl', ['$scope', '$location', '$http', ($scope,
  $scope.shifts = []
  $http.get('./shifts.json').success((data) ->
    $scope.shifts = data
```

```
  )

  $scope.viewShift = (id) ->
    $location.url "/shifts/#{id}"
]
```

– ☒ Create show template, route and controller

```
./public/templates/shifts/show.html

<h1>{{shift.title}}</h1>

./app/assets/javascripts/main.js.coffee

@rest = angular.module('rest', [])

@rest.config(['$routeProvider', ($routeProvider) ->
  $routeProvider.
    when('/restaurants', {
      templateUrl: '../templates/restaurants/index.html',
      controller: 'RestaurantIndexCtrl'
    }).
    when('/restaurants/:id', {
      templateUrl: '../templates/restaurants/show.html',
      controller: 'RestaurantShowCtrl'
    }).
    otherwise({
      templateUrl: '../templates/home.html',
      controller: 'HomeCtrl'
    })
])

./app/assets/javascripts/angular/controllers/ShiftShowCtrl.
js.coffee

@rest.controller 'RestaurantShowCtrl', ['$scope', '$http', '$routeParams', ($
  $http.get("./restaurants/#{$routeParams.id}.json").success((data) ->
    $scope.restaurant = data
  )
]
```

Routes

Views Directory
`./config/routes.rb`

- ☒ create root path

```
root 'static_pages#home'
```

- ☐ create paths for desired routes

```
get "about" => "pages#about"
get "news" => "pages#news"
get "programs" => "pages#programs"
get "calendar" => "pages#calendar"
get "contact" => "contacts#new"
get "staffandboard" => "pages#staff"
get "jobs" => "pages#jobs"
get "donate" => "pages#donate"
get "camp" => "pages#summer_camp"
get "csv" => "pages#csvupload"
get "ccf" => "shifts#volunteer"
```

Controllers

`./app/controllers/application_controller.rb`
`./app/controllers/pages_controller.rb`

Application skeleton BASICS

Useful commands

Rake

```
rake routes
```

Rails

```
rails console
```

```
rails s
rails s -e production
```

Heroku

```
heroku rename $NEW_NAME
heroku open
heroku logs --tail
heroku run rails console

heroku config:set <ENV_NAME>=<variable>
heroku config:unset
heroku config:get
```

Git

Essential Files

Gemfile

Create the default skeletal application

- ☒ create a new application

  ```
  rails new PrairieHillWebsite
  ```

- ☒ update README

  ```
  rm README.rdoc
  touch README.org
  ```

- ☒ rename application.css to application.css.scss

  ```
  ./app/assets/stylesheets/application.css.scss

  cd app/assets/stylesheets
  mv application.css application.css.scss
  ```

- ☒ Test the skeletal application

  - ☒ Start the Rails server

    ```
    rails s
    ```

  - ☒ open your browser to localhost, port 3000

    ```
    localhost:3000
    ```

- ☒ update the Gemfile

  ```
  ./Gemfile
  ```

  ```
  cat ~/RAILS-dev/DEFAULT-Gemfile > Gemfile
  ```

- ☒ update the bundle

  ```
  bundle update
  bundle install --without production
  ```

Set up Git and Heroku

- Git

  - ☒ initialize git repo

    ```
    git init
    ```

  - ☒ update .gitignore

    ```
    ./.gitignore
    ```

    ```
    echo ".env" >> .gitignore
    echo "Procfile" >> .gitignore
    ```

  - ☒ initial stage and commit of all files

    ```
    git add .
    git commit -am "initial commit"
    ```

  - ☒ add the origin

    ```
    git remote add origin https://github.com/son1112/PrairieHillWebsite.git
    ```

  - ☒ initial push

    ```
    git push -u origin master
    ```

- Heroku

  - ☒ Create and push a new heroku app

    ```
    heroku create
    git push heroku master
    ```

  - ☒ Rename the heroku app

    ```
    heroku rename phill-new
    ```