Who? What? When?

Human readable audit reporting for PaperTrail

A ⚡ Talk ;)

ME: Jason Dinsmore

@dinjas on all the things

H/NT

(Ruby/React) consulting company in Vancouver, WA

Rails 4.2

Primarily back-office app
(powers a business)

Business Model

Acts as middleman between buyers and sellers
(& takes a cut)

✂

Users

Representatives for buyers, sellers, and business all access the system

Data

Much data is shared
between ALL the user types

Who? What? When?

We’re frequently asked
about the history of data

PaperTrail

github.com/paper-trail-gem/paper_trail

```
gem 'paper_trail'
```

```
rails g paper_trail:install --with-changes
```

```
class PaperTrail::Version < ActiveRecord::Base {
                :id => :integer,
         :item_type => :string,
           :item_id => :integer,
             :event => :string,
         :whodunnit => :string,
            :object => :text,
        :created_at => :datetime,
    :object_changes => :text
}
```

```ruby
class User < ApplicationRecord
  devise :database_authenticatable,
         :registerable,
         :validatable

  has_paper_trail

  validates :email, presence: true
  validates :name, presence: true

  alias_attribute :to_s, :name
end
```

Zeus’ ⚡ App

A (not so) real-world example

```
class Lightning < ApplicationRecord {
            :id => :integer,
          :name => :string,
     :intensity => :integer,
        :charge => :string,
     :bolt_type => :string,
     :strike_at => :datetime,
    :created_at => :datetime,
    :updated_at => :datetime
}
```

Hi-d-ho, go to demo!

PaperTrailScrapbook

github.com/tjchambers/paper_trail_scrapbook

```
gem 'paper_trail_scrapbook'
```

Note: relies on presence of object_changes
column in PaperTrail::Versions table

LifeHistory

Shows history of a given database object

```
puts PaperTrailScrapbook::LifeHistory.new(obj).story
```

UserJournal

Shows changes made by a user
(optionally scope to class or time range)

```
puts PaperTrailScrapbook::UserJournal.new(user).story
```

Ho-d-hi, back to demo!

```ruby
# frozen_string_literal: true

class ApplicationRecord < ActiveRecord::Base
  self.abstract_class = true

  def story
    return unless PaperTrail.request.enabled_for_model?(self.class)
    PaperTrailScrapbook::LifeHistory.new(self).story
  end

end
```

```ruby
# frozen_string_literal: true

module ApplicationHelper
  def history_link_for_object(object)
    link_to '#',
            class: 'scrapbook',
            title: 'History',
            data:  { id:    object.id,
                     klass: object.class.name } do
      fa_icon('book')
    end
  end
end
```

```haml
    - @lightnings.each do |lightning|
      %tr
        %td= lightning.name
        %td= lightning.intensity
        %td= lightning.bolt_type
        %td= lightning.charge
        %td= lightning.strike_at.strftime("%m/%d/%-Y")
        %td= lightning.created_at.strftime("%m/%d/%-Y")
        %td= lightning.updated_at.strftime("%m/%d/%-Y")
        %td= history_link_for_object(lightning)

#history_modal
  %h3#history_modal_header
  #history_modal_body
```

```ruby
# frozen_string_literal: true

class LifeStoryController < ApplicationController
  before_action :authenticate_user!

  def history
    obj = klass.find(obj_id)

    history = CGI.escapeHTML(obj.story).presence || 'No history available'

    render json: { name: "History for '#{obj}'", history: history }
  end

  private

  def obj_id
    Integer(params.fetch(:id))
  end

  def klass
    params.fetch(:klass).constantize
  end
end
```

Resources:

Demo app:
//github.com/dinjas/lightning_app

These slides:
lightning_app/docs/paper_trail_scrapbook.pdf

Blog:
//hint.io/blog/paper-trail-scrapbook

PaperTrail gem:
//github.com/paper-trail-gem/paper_trail

PaperTrailScrapbook gem:
//github.com/tjchambers/paper_trail_scrapbook