

Who? What? When?

Human readable audit reporting for PaperTrail

A  Talk ;)

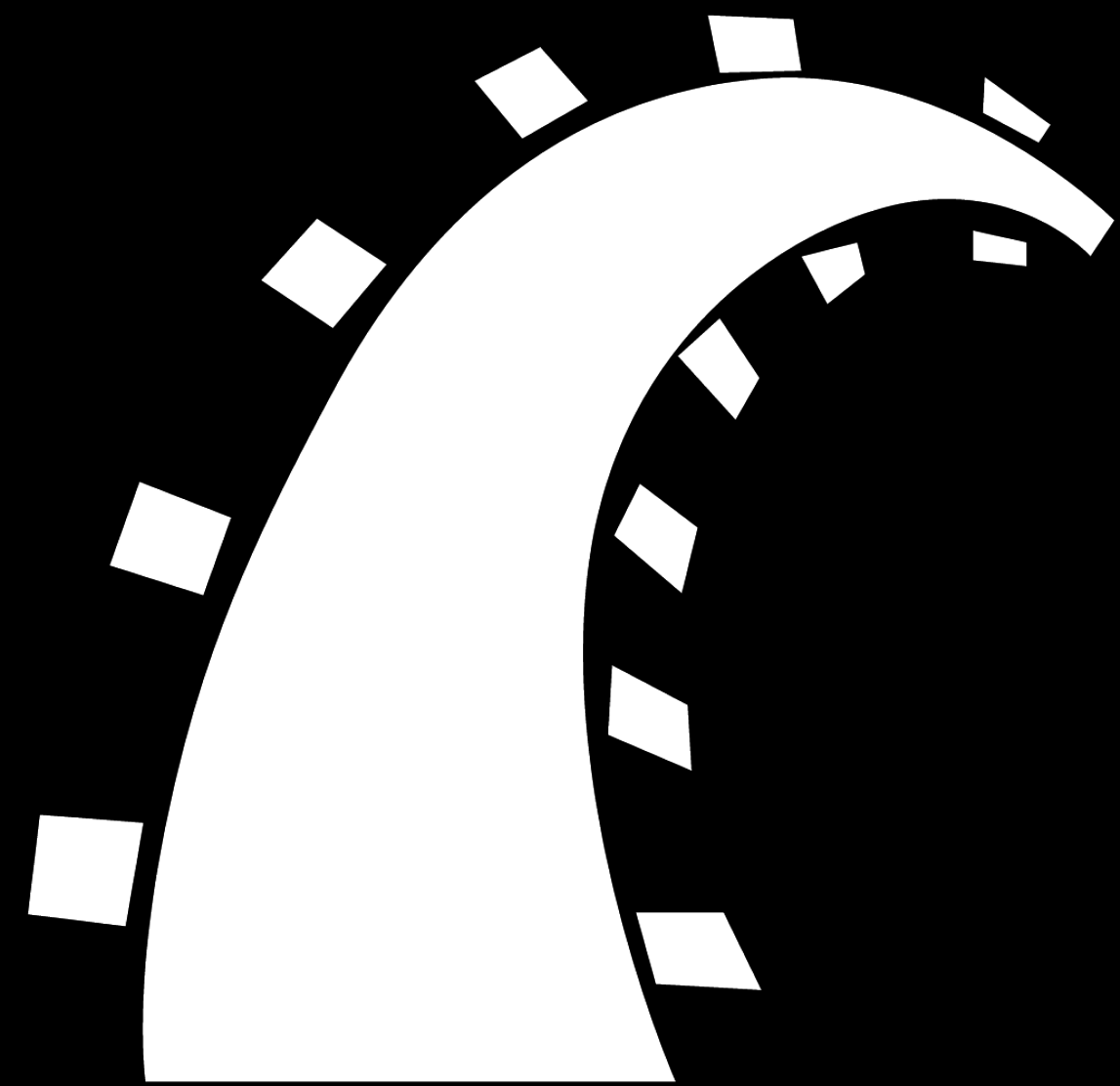
ME: Jason Dinsmore

@dinjas on all the things





(Ruby/React) consulting company in Vancouver, WA

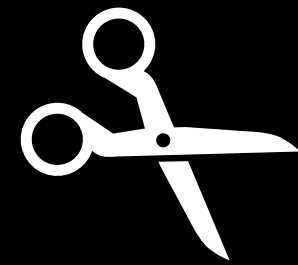


Rails 4.2

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  
}
```

```
16 class User < ApplicationRecord
17   devise :database_authenticatable,
18         :registerable,
19         :validatable
20
21   has_paper_trail
22
23   validates :email, presence: true
24   validates :name, presence: true
25
26   alias_attribute :to_s, :name
27 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!

```
1 # frozen_string_literal: true
2
3 class ApplicationRecord < ActiveRecord::Base
4   self.abstract_class = true
5
6   def story
7     return unless PaperTrail.request.enabled_for_model?(self.class)
8     PaperTrailScrapbook::LifeHistory.new(self).story
9   end
10
11 end
```

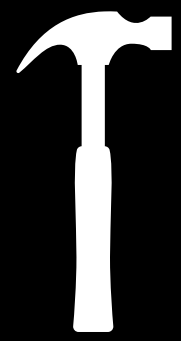
```
1 # frozen_string_literal: true
2
3 module ApplicationHelper
4   def history_link_for_object(object)
5     link_to '#',
6           class: 'scrapbook',
7           title: 'History',
8           data: { id: object.id,
9                 klass: object.class.name } do
10       fa_icon('book')
11     end
12   end
13 end
```

```
22     - @lightnings.each do |lightning|
23         %tr
24             %td= lightning.name
25             %td= lightning.intensity
26             %td= lightning.bolt_type
27             %td= lightning.charge
28             %td= lightning.strike_at.strftime("%m/%d/%-Y")
29             %td= lightning.created_at.strftime("%m/%d/%-Y")
30             %td= lightning.updated_at.strftime("%m/%d/%-Y")
31             %td= history_link_for_object(lightning)
32
33 #history_modal
34     %h3#history_modal_header
35     #history_modal_body
```



```
1 # frozen_string_literal: true
2
3 class LifeStoryController < ApplicationController
4   before_action :authenticate_user!
5
6   def history
7     obj = klass.find(obj_id)
8
9     history = CGI.escapeHTML(obj.story).presence || 'No history available'
10
11     render json: { name: "History for '#{obj}'", history: history }
12   end
13
14   private
15
16   def obj_id
17     Integer(params.fetch(:id))
18   end
19
20   def klass
21     params.fetch(:klass).constantize
22   end
23 end
```

Resources:



Demo app:

[//github.com/dinjas/lightning_app](https://github.com/dinjas/lightning_app)



These slides:

[lightning_app/docs/paper_trail_scrapbook.pdf](https://github.com/dinjas/lightning_app/docs/paper_trail_scrapbook.pdf)

Blog:

[//hint.io/blog/paper-trail-scrapbook](http://hint.io/blog/paper-trail-scrapbook)

PaperTrail gem:

[//github.com/paper-trail-gem/paper_trail](https://github.com/paper-trail-gem/paper_trail)

PaperTrailScrapbook gem:

[//github.com/tjchambers/paper_trail_scrapbook](https://github.com/tjchambers/paper_trail_scrapbook)