

# Clearance

**Jason Morrison**  
**[jmorrison@thoughtbot.com](mailto:jmorrison@thoughtbot.com)**



```
$ SUDO GEM INSTALL \
THOUGHTBOT-Clearance \
--SOURCE HTTP://GEMS.GITHUB.COM
```

```
$ CURL "HTTP://GITHUB.COM/THOUGHTBOT/ \
CLEARANCE/TREE/MASTER%2FREADME.TEXTILE? \
RAW=TRUE"
```



**Thank you.**





# Encore!



Looking for traditional values and various interests. ☎88866

**D:** **Rails developer** seeks simple, modular, well-tested authentication plugin. Accessorizing well with suspenders is a plus.

DWM, 32, Italian, black/hazel, sweet and passionate. ☎88209

breznoutis - a person







**“Who goes there?”**



**email + confirmation +  
password + forgot  
= authentication.**

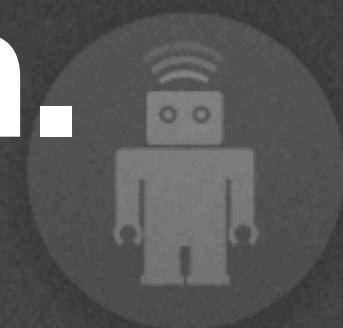
**No authorization,  
no roles, no ACLs,  
no HTTP basic auth,  
no OpenID (in core),  
no admin (in core),  
no, no, no!**







**Keep It Simple, Sucka.**





*mess*  
**Generator<sup>^</sup> free!**









# (but it's small)

# models and controllers

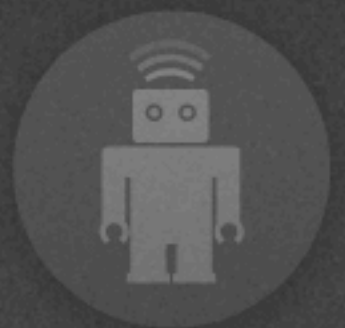
```
[~/dev/clearance/lib/clearance/app] find . | xargs wc -l  
434 total
```

# units and functionals

```
[~/dev/clearance/lib/clearance/test] find . | xargs wc -l  
822 total
```



**Let's see it!**









<http://github.com/thoughtbot/clearance>





# Have it your way

```
class User < ActiveRecord::Base

  include Clearance::Model

  def encrypt(password)
    Digest::SHA1.hexdigest "--#{salt}--#{password}--"
  end

  protected

  def initialize_salt
    self.salt = Digest::SHA1.hexdigest(
      "--#{Time.now.to_s}--#{email}--") if new_record?
  end
end
```



# Have it your way

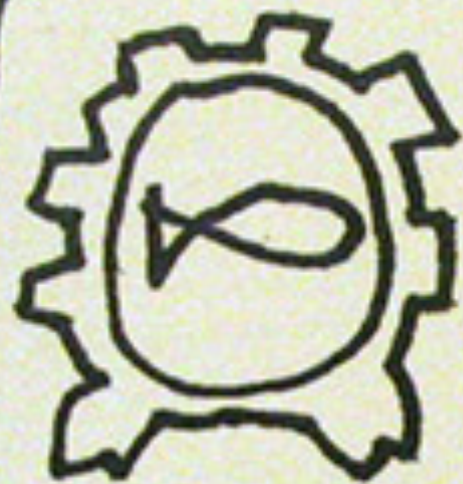
```
# similar deal for UsersController, ConfirmationsController
class SessionsController < ApplicationController
  include Clearance::SessionsController

  def url_after_create
    video_url(:awesome_and_exciting_welcome)
  end

  def url_after_destroy
    video_url(:wistful_farewell)
  end
end
```







# Security Clearance

Holder/Bearer hereby  
enabled access to all areas





# Get goin'

```
class User < ActiveRecord::Base

  include Clearance::Model

  acts_as_geocodable :normalize_address => true

  # don't forget this guy!
  attr_accessible :first_name, :last_name,
                 :street, :locality, :region, :postal_code,
                 :website, :about

end
```





# Get goin'

```
class UsersController < ApplicationController
  include Clearance::UsersController

  before_filter :authenticate, :except => [:new, :create]
  before_filter :can_only_edit_self, :only => [:edit, :update]

  protected

  def can_only_edit_self
    unless current_user == User.find(params[:id])
      flash[:error] = 'Oh, snap! Get outta here.'
      redirect_to root_url
    end
  end
end
```



# Future Work





# TODO.textile

- Some refactoring & documentation to do



# Loot from merb-auth

- Store `current_user` on the session, not controller
- HTTP fluency
  - 401 Unauthorized
  - 405 Resource not allowed
- Make a strategy:
  - Email confirmation
  - Forgot password
  - Salted passwords





# clearance-admin

`<% if current_user.admin? -%>`

`Admin::UsersController`

`logged_in_admin_context {}`

`should_only_allow_admins_on 'get :index'`



# clearance-openid

Extract from  
<http://hoptoadapp.com>







Guard Dog

[http://headrush.typepad.com/creating\\_passionate\\_users/2007/03/seven\\_blog\\_virt.html](http://headrush.typepad.com/creating_passionate_users/2007/03/seven_blog_virt.html)

Kiss

<http://flickr.com/photos/andraspfaff/623258079/>

Awesome Delorean

<http://flickr.com/photos/f1rstborn/757609629/>

“Shh!”

[http://flickr.com/photos/cupcake\\_eater/2721122278/](http://flickr.com/photos/cupcake_eater/2721122278/)

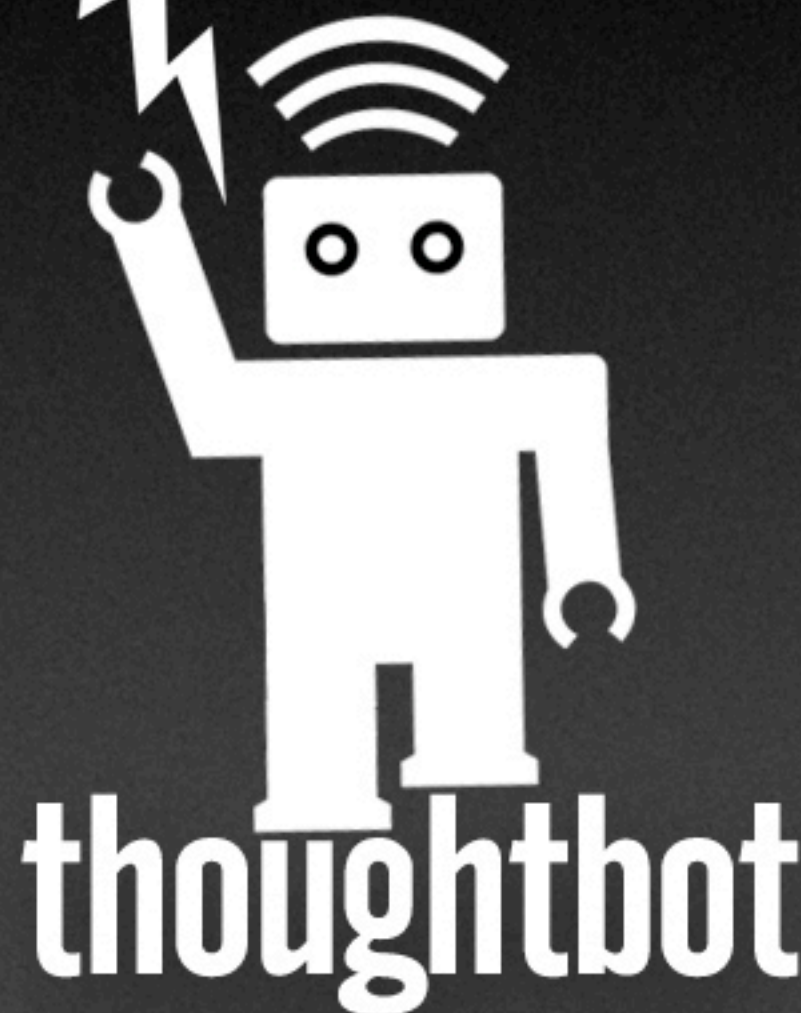
Personals Ad

<http://flickr.com/photos/eggplant/3211654/>

Baseball Photosiñata

<http://flickr.com/photos/mcbeth/156411746/>

# THANKS!

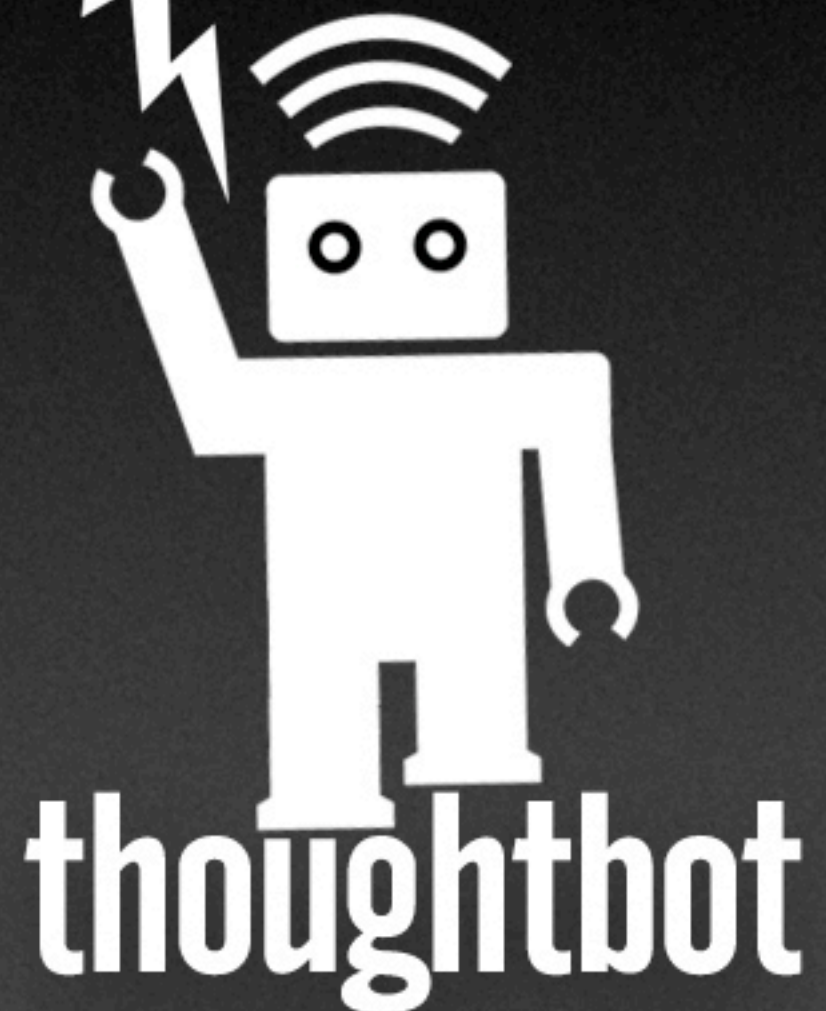


THANKS!

```
$ tail -n 8 README.textile
```

h2. Authors

- \* thoughtbot, inc.
- \* Dan Croak
- \* Jason Morrison
- \* Mike Burns
- \* Josh Nichols
- \* Mike Breen





THANKS!

**github.com**  
**/thoughtbot**  
**/clearance**

