SEARCHLOGIC

ActiveRecord extension that allows you to search using a hash.


©Disney

Provides an object based interface to constructing hash values.


SECRET BONUS
NUMBER 01
2000

Pagination,
sorting,
helpers.

Ben Johnson aka binarylogic

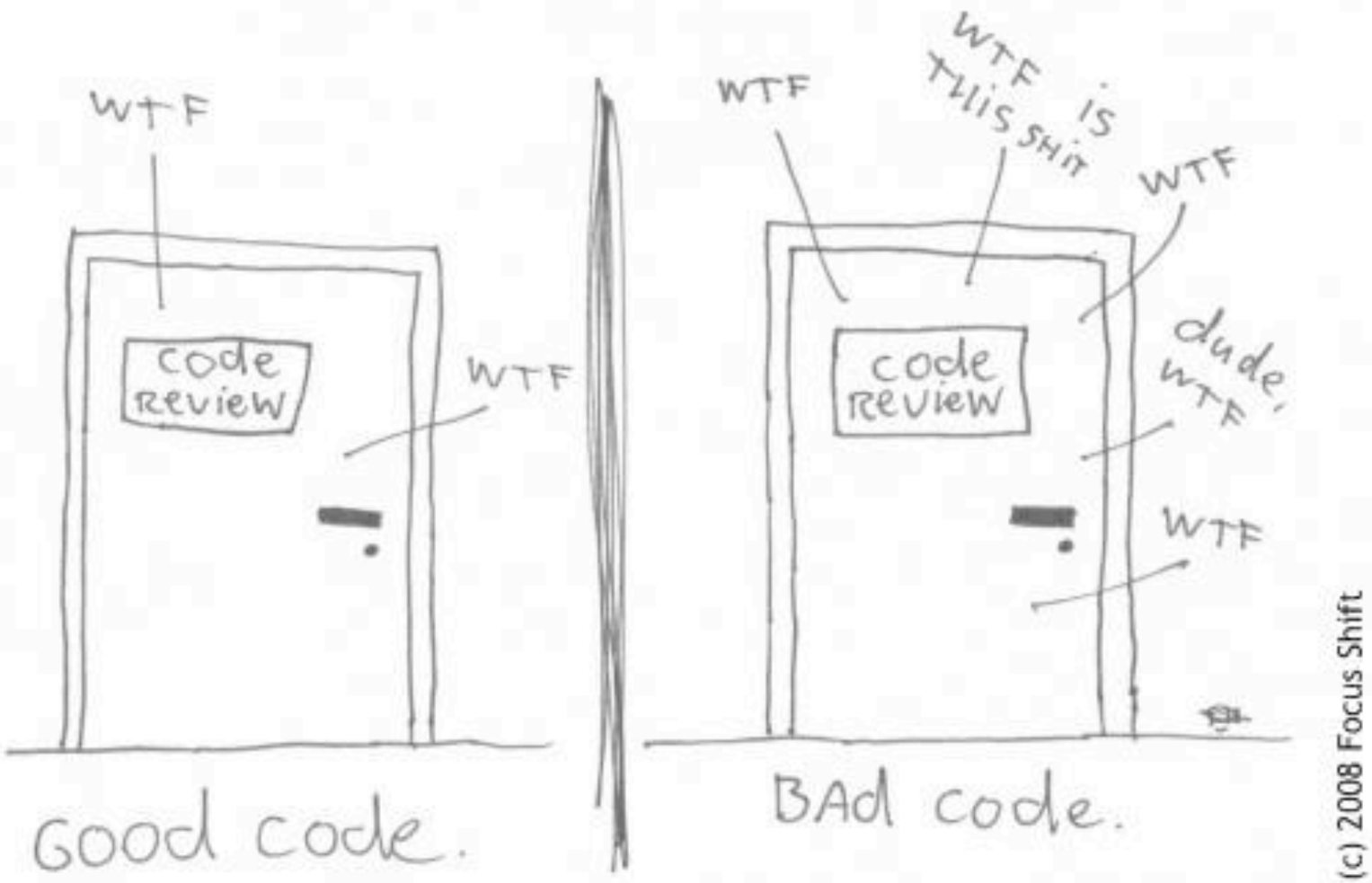
http://binarylogic.com

http://github.com/binarylogic

when you look at things on the internet, the one question you will ask is the same one everyone asks, and to which there is no answer: WHY WOULD A PERSON DO THAT

Isolation
Encapsulation
Integration

The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE



```ruby
User.all(
  :conditions => {
    # first_name like '%Ben%'
    :first_name_contains => "Ben",
    # email like '%binarylogic.com'
    :email_ends_with => "binarylogic.com",
    # created_at > Time.now
    :created_after => Time.now,
    # HOUR(created_at) > 5 (depends on DB type)
    :hour_of_created_at => 5,
    # relationship attribute searching
    :orders => {
      :line_items => {
        :created_after => Time.now
      }
    }
  },
  :order_as => "ASC",
  # order user_groups.name ASC
  :order_by => {:user_group => :name}
)
```

```
:begins_with
:ends_with
:equals
:greater_than
:greater_than_or_equal_to
:ilike
:less_than
:less_than_or_equal_to
:keywords
:like
:not_begin_with
:not_end_with
:not_equal
:not_have_keywords
:not_ilike
:not_like
```

```
def index
  @search = User.new_search(params[:search])
  @search.per_page = 20
  @search.page = 3
  @users, @users_count = @search.all, @search.count
end

- form_for @search do |f|
  - f.fields_for @search.conditions do |users|
    = users.text_field :first_name_contains
    # nice rails plugin for replacing date_select
    = users.calendar_date_select :created_after
    - users.fields_for users.object.orders do |orders|
      = orders.select :total_gt, (1..100)
  = f.submit "Search"

- unless @users_count.zero?
  %table
    %tr
      %th= order_by_link :account => :name
      %th= order_by_link :first_name
      %th= order_by_link :last_name
      %th= order_by_link :email
    - @users.each do |user|
      %tr
        %td= user.account? ? user.account.name : "-"
        %td= user.first_name
        %td= user.last_name
        %td= user.email

  == Per page: #{per_page_select}
  == Page: #{page_select}
- else
  No users were found.
```

View helpers let you modify 4 aspects of your results:

1. Order by single column or array of columns: order_by
2. Descend or ascend yr data: order_as
3. Change how many items are on a page: per_page
4. Paginate through the reults: page

Actions come in three different types: link, links, & select

order_by_link(:first_name)
order_by_link([:first_name, :last_name])
order_by_link({:orders => :total})
order_by_link([{:orders => :total}, :first_name])
Note: Automatically alternates between asc/desc.
page_link(10)
page_select

```
def index
  @search = User.new_search(params[:search])
  @search.per_page = 20
  @search.page = 3
  @users, @users_count = @search.all, @search.count
end

- form_for @search do |f|
  - f.fields_for @search.conditions do |users|
    = users.text_field :first_name_contains
    # nice rails plugin for replacing date_select
    = users.calendar_date_select :created_after
    - users.fields_for users.object.orders do |orders|
      = orders.select :total_gt, (1..100)
  = f.submit "Search"

- unless @users_count.zero?
  %table
    %tr
      %th= order_by_link :account => :name
      %th= order_by_link :first_name
      %th= order_by_link :last_name
      %th= order_by_link :email
    - @users.each do |user|
      %tr
        %td= user.account? ? user.account.name : "-"
        %td= user.first_name
        %td= user.last_name
        %td= user.email

  == Per page: #{per_page_select}
  == Page: #{page_select}
- else
  No users were found.
```

```ruby
@search = User.new_search(:conditions => {:age_gt => 18})
@search.conditions.first_name_contains = "Slick"
# can set this to "true" or "1" or "yes"
@search.conditions.any = true
# will join all conditions with "or" instead of "and"
@search.all
```

```ruby
@search = User.new_search(:conditions => {:age_gt => 18})
@search.conditions.or_first_name_contains = "Ben"
@search.conditions.or_last_name_contains = "Johnson"
# the and_ is optional, calling just id_gt is the same thing
@search.conditions.and_id_gt = 5
# will join conditions in the orders they
# were set with their specified join condition
@search.all
# => age > 17 OR first_name like '%Ben%'
# OR lsat_name like '%Johnson%' AND id > 5
```

```ruby
@current_user.orders.find(:all, :conditions => {:total_lte => 500})
@current_user.orders.count(:conditions => {:total_lte => 500})
@current_user.orders.sum('total', :conditions => {:total_lte => 500})
@current_user.orders.build_search(:conditions => {:total_lte => 500})
```

```ruby
class User < ActiveRecord::Base
  named_scope :sexy, {
    :conditions => {
      :first_name => "Ben",
      :email_ends_with => "binarylogic.com"
    },
    :per_page => 20
  }
end
```

```ruby
class User < ActiveRecord::Base
  named_scope :sexy, {
    :conditions => {
      :first_name => "Ben",
      :email_ends_with => "binarylogic.com"
    },
    :per_page => 20
  }
end

@search = User.sexy.build_search(:conditions => {:age_lte => 50})
@search = User.sexy.build_search(params[:search])
```


PHENOMINAL COSMIC POWER
itteh bitteh kitteh
ICANHASCHEEZBURGER.COM

Documentation
http://searchlogic.rubyforge.org

Tutorial
http://www.binarylogic.com/2008/9/7/tutorial-pagination-ordering-and-searching-with-searchlogic

Live example of the tutorial (with source)
http://searchlogic_example.binarylogic.com