

Type

Fond

Forme

Intitule

J'ai vu que dans certains modeles la logique est differente, dans ceux là à la limite pour le moment tu peux redéfinir le set_sequence.

- Mais pas dans les array : [:foo, :bar] mais met un espace entre les éléments de l'array par contre !

Statut	commentaire/question
--------	----------------------

reponse

- Utilises des scope à la place "find_all" etc.

Pour te donner un exemple, vo un de nos projet (les scopes sont syntaxe qui va avec Rails 3 ici):

```
default_scope where(:trashed => false)
scope :starred,      where(:starred => true)
scope :confirmed,    where(:confirmed => true)
scope :not_confirmed, where(:confirmed => false)
scope :invited,      where(:invited => true)
scope :not_invited,  where(:invited => false)
scope :interested_in_beta,
=> true)
scope :not_interested_in_beta,
```

=> false)

```

scope :not_already_resent_c
where(:confirmation_resent_at => nil)
scope :having_comment, lambda {
  ({ :free_text.ne => nil } & { :free_text => nil } | { :free_text :
    scope :having_site, lambda {
      ("DISTINCT(enthusiasts.id)").join(
        ("enthusiasts.id NOT IN (?)", having_tag(true)) }
      scope :having_tag, lambda {
        ("enthusiasts.id").joins(:tags) : where(
          ("?", having_tag(true)) }
      scope :by_date, lambda {
        (:created_at.send(way || 'desc')) }
      scope :by_email, lambda {
        (:email.send(way || 'desc'), :created_at.send(way || 'desc')) }
      scope :by_starred, lambda {
        (:starred.send(way || 'asc'), :created_at.send(way || 'desc')) }
      scope :by_confirmed, lambda {
        (:confirmed_at.send(way || 'asc'), :created_at.send(way || 'desc')) }
      scope :by_interested_in_beta, lambda {
        (:interested_in_beta.send(way || 'asc'), :created_at.send(way || 'desc')) }
      scope :by_invited, lambda {
        (:invited_at.send(way || 'desc'), :created_at.send(way || 'desc')) }
      scope :with_tag, lambda { |tag|
        ({ :sites => :tags }, :tags).where({ :sites => { :tags => { :name => tag } } })
        scope :search, lambda { |q| where(
          (enthusiasts.email) LIKE LOWER(:q),
          (enthusiasts.free_text) LIKE LOWER(:q)) }
      }
    }
  }
}

```

(enthusiast_sites.hostname) LIKE
#{q}%", "%#{q}%"])

Ensuite t'as plus qu'a appeler
Enthusiast.starred.not_invited.has
exemple (tu peut enchaîner les s
requete SQL au final!). Vacheme

des méthode "find_by",

ici des scopes que l'on a sur
t définis avec la nouvelle

```

d_at => nil)
re(:starred => true)
ere(:confirmed_at.ne => nil)
where(:confirmed_at => nil)
re(:invited_at.ne => nil)
ere(:invited_at => nil)
where(:interested_in_beta

```

a, where(:interested_in_beta


```

confirmation_instructions,
> nil)
lambda { |bool| bool ? where
  ext.ne => "" }) : where
=> "" }) }
lambda { |bool| bool ? select
  is(:sites) : where
  ing_site(true)) }
lambda { |bool| bool ? select
  ere("enthusiasts.id NOT IN

```

```

lambda { |way| order
}
lambda { |way| order
  ed_at.desc) }
lambda { |way| order
  ted_at.desc) }
lambda { |way| order
  :created_at.desc) }
lambda { |way| order
  'desc'), :created_at.desc) }
lambda { |way| order
  :created_at.desc) }
  g| tag.present? ? includes
  :tags => { :name => tag } } |
  ag } } }) : where("") }
  here(["LOWER
  R(?) OR LOWER
  VER(?) OR LOWER

```

≡ LOWER(?), "%#{q}%", "%

ving_tag.by_date par
copes et ça te fait une
nt plus clean ! :)