

Rails の Active Record attributes API が便利

@vividmuimui

社内勉強会資料 LT資料 2017/11/03

注意

この資料は2017/11/03時点のもので、Rails5.2が出る前のものです。
資料内でGemを導入していますが、Rails5.2では必要なくなっています。

Active Record attributes API とは

rails5で入った機能

- release note:
http://guides.rubyonrails.org/5_0_release_notes.html#active-record-attributes-api
- api document:
<http://api.rubyonrails.org/classes/ActiveRecord/Attributes/ClassMethod>

分かりやすかった記事

- <http://y-yagi.tumblr.com/post/140725723370/rails-5%E3%81%AEactive-record-attributes-api%E3%81%AB%E3%81%A4%E3%81%84%E3%81%A6>
- https://www.wantedly.com/companies/wantedly/post_articles/31132

実装する時に確実に見るやつ

- <http://api.rubyonrails.org/classes/ActiveModel/Type/Value.html>

使い方・どんなやつか

実際実装したのを例にします。

1 typeクラスを定義する

```
class EmailType < ActiveRecord::Type::String
  # DBに入れる・検索する時にどう変換するか
  def serialize(value)
    normalize(super)
  end

  # DBから取り出す時にどう変換するか
  # def deserialize(value); end

  private

  # setterでどう変換するか
  def cast_value(value)
    normalize(super)
  end

  def normalize(value)
    return unless value
    value.downcase.remove(/\A[[:^graph:]]*/).remove(/[[[:^graph:]]*\z/)
  end
end
```

2 登録する

```
ActiveRecord::Type.register(:email, EmailType)
```

3 modelで使うよう宣言する

```
class User < ApplicationRecord
  # attribute :field, :type
  attribute :email, :email
end
```

設定終わり

4 使う

検索する時にserializeされて検索される

```
(main)> User.find_by(email: " SAMPLE@EXAMPLE.COM ")
User Load (0.4ms)  SELECT `users`.* FROM `users` WHERE `users`.`email` = 'sample@example.c
```

保存するときもserializeされる

```
(main)> FactoryBot.create(:user, email: " SAMPLE@EXAMPLE.COM ").email
(0.4ms)  BEGIN
SQL (0.3ms)  INSERT INTO `users` (`uid`, `identifier`, `email`, `password_digest`, `activat
(8.5ms)  COMMIT
=> "sample@example.com"
```

setter等では, cast_value(#cast)が呼ばれる

```
User.new(email: " SAMPLE@EXAMPLE.COM ").email  
=> "sample@example.com"
```

```
(main)> a = User.new  
=> #<User:0x00007fec9402c118 ....  
(main)> a.email = " SAMPLE@EXAMPLE.COM "  
=> " SAMPLE@EXAMPLE.COM "  
(main)> a.email  
=> "sample@example.com"
```

便利! 🎉

問題

現状のrailsでは ActiveRecord でしか使えず、ActiveModelなobjectでは使えない

つい最近、ActiveModelでも動くように実装され始めているようだが、現時点では使えない

<https://github.com/rails/rails/pull/30920>

ActiveModelでも使う

ここらへん読んで理解しつつ `active_model_attributes` gemを入れる

<https://github.com/rails/rails/pull/26728>

<https://karolgalanciak.com/blog/2016/12/04/introduction-to-activerecord-and-activemodel-attributes-api/>

- 近い将来、rails本体で ActiveModel で attributes apiが使えるようになるのでそのじゃまにならない方が良い
- PRにあるように、このgemで全部正しく動くわけではないので小さなモデルでしか使わないほうが良さそう

使い方

gemを入れる

```
gem "active_model_attributes"
```

2 typeクラスを定義する

```
class EmailType < ActiveRecord::Type::String
  # DBに入れる・検索する時にどう変換するか
  def serialize(value)
    normalize(super)
  end

  # DBから取り出す時にどう変換するか
  # def deserialize(value); end

  private

  # setterでどう変換するか
  def cast_value(value)
    normalize(super)
  end

  def normalize(value)
    return unless value
    value.downcase.remove(/\A[[:^graph:]]*/).remove(/[[:^graph:]]*\z/)
  end
end
```


3 登録する

```
ActiveModel::Type.register(:email, EmailType) # ActiveRecordではなくActiveModel
```

4 modelで使うよう宣言する

```
class SomeModel
  include ActiveRecord::Model
  include ActiveRecord::Attributes

  attribute :email, :email
end
```

設定終わり



使う

```
(main)> SomeModel.new(email: " SAMPLE@EXAMPLE.COM ").email  
=> "sample@example.com"  
(main)> a = SomeModel.new  
=> #<SomeModel:0x00007fec933324f8>  
(main)> a.email = " SAMPLE@EXAMPLE.COM "  
=> " SAMPLE@EXAMPLE.COM "  
(main)> a.email  
=> "sample@example.com"
```

便利! 🎉

おわり

before_validationでゴニョゴニョしてる時とか、
今回のような、ユーザの入力をnormalizeしたいときとかにはだいぶ有用そうだった
だいぶ便利なので、ほかにも使いみちは色々ありそう