SimpleDelegator活用のご提案

平成Ruby会議 01

Daisuke Fujimura

2019-12-13

自己紹介

藤村大介

- https://twitter.com/ffu_
- https://github.com/fujimura
- スタートアップ数社 ⇨ マチマチCTO ⇨ フリーランス
- 現在は数社で技術顧問のお仕事
- RubyとHaskellが好き
- WEB+DB PRESS Vol.110 名前付け大全を書いた

今日のお話

SimpleDelegatorは便利なのにあまり使われていない印象があるので宣伝したい！

SimpleDelegatorとは

- オブジェクト指向でいうところの「委譲」ができるライブラリ
- Ruby標準ライブラリ ‘delegate’ に含まれる

委譲ってなに

- TODO: 時間あれば書く

基本的な使用例

```ruby
class User
  def born_on
    Date.new(1989, 9, 10)
  end
end

class UserDecorator < SimpleDelegator
  def birth_year
    born_on.year
  end
end

decorated_user = UserDecorator.new(User.new)
decorated_user.birth_year  #=> 1989
decorated_user.__getobj__  #=> #<User: ...>
```

出典: https://ruby-doc.org/stdlib-2.6.5/libdoc/delegate/rdoc/SimpleDelegator.html

主な使いみち

TODO: 詳しく

- デコレータ
- 複合したオブジェクト
- 局所的な拡張

例：APIのレスポンスとカラム名を揃えたい

- APIのレスポンスがオブジェクト
- モデルとインターフェイスが違う

```
class Entry < SimpleDelegator
  def title
    subject
  end
end

entries = api.get('/entries/').map { |r| Entry.new(r) }
puts entries.first.title # => `subject`の値
```

- 継承だと`api.get`で返ってくるオブジェクトを拡張できない

例：ソート順を変えたい

```
class UserSortedByBirthday < SimpleDelegator
  def <=>(other)
    birthday > other.birthday
  end
end
users = User.limit(10).map {|u| UserSortedByBirthday.new(u) }
users.sort #=> 誕生日で並べ替えられる
```

例：現在のユーザーによってオブジェクトの値を変えたい

```
class PersonalizedPost < SimpleDelegator
  def initialize(post, user)
    @user = user
    __setobj__(post)
  end

  def comments
    __getobj__.comments.filter {|c| !c.author.blocked_by?(@user) }
  end
end

posts = Post.first(10).map {|p| PersonalizedPost.new(p, current_user) }
posts.first.comments # ブロックされたユーザーのコメントは現れない
```

- パーソナライズに便利

例：クエリオブジェクト作るの面倒

```ruby
class DailyNewComment < SimpleDelegator
  def initialize(user, start_at)
    comments = user.visible_comments.where(
      'created_at >= :from AND created_at < :to',
      from: start_at,
      to: 24.hours.since(start_at)
    )

    __setobj__(comments)
  end
end

comments = DailyNewComment.new(user, start_at)
```

- `DailyNewCommentQuery.new(...).result` などとするより簡潔

例：アクセサを足したい

```ruby
class UserWithToken < SimpleDelegator
  attr_accessor :token
end

token = generate_token
users = User.limit(10).map {|user|
  u = UserWithTimestamp.new(user)
  u.token = token
  u
}

users.first.token #=> 生成したトークン
```

まとめ

TODO: 主な使いみち軸で総括

補足

- 遅いってマジ？
- DelegateClassとの違いは？わからん！誰か教えてくれ！

TODO

- 全体にbefore/afterにする