

**Favorite**のパター  
ン化 (したい  
)

**@tkawa (from Sendagaya.rb)**

# はじめに

---

Ruby on Rails Advent Calendar 2013 5日目

「Favoriteの設計実装はパターンとして使える」

<http://qiita.com/tkawa/items/ac01dbc1e441e78ffcfd>

一見設計しにくい「操作」はパターンに落とし込めるのでは、という話です。

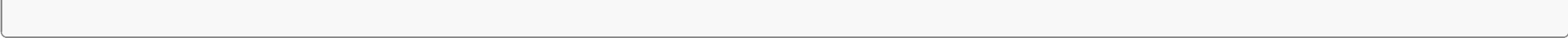
# リストの項目をソートしたい

---

```
class TodoList < ActiveRecord::Base
  has_many :todo_items, -> { order(position: :desc)
}

  def sort!
    # self.todo_items の position を書き換えてsaveす
    るコード
  end
end

class TodoItem < ActiveRecord::Base
  # name:      string
  # position:  integer
  belongs_to :todo_list
end
```



# コントローラは？

---

```
class TodoList < ActiveRecord::Base
  def sort!
    ...
  end
end
```

コントローラはどうなる？ `sort!` を実行するのはどれ？？

```
POST    /todo_lists/1          POST    /todo_lists/1/sort
PUT     /todo_lists/1          PUT     /todo_lists/1/sort
PATCH /todo_lists/1          PATCH  /todo_lists/1/sort
```

# やりがち

---

```
class TodoListsController < ApplicationController
  # POST /todo_lists/1/sort
  def sort
    @todo_list.sort!
    ...
  end
end
```

```
resources :todo_lists do
  post :sort, on: :member
end
```

# 惜しい

---

アクションを増やすのはよくない

これをやり続けるとアクションがどんどん増えていく。CRUD以外の責務は危険信号。

これは本当にTodoListsControllerの責務なのか？

**/todo\_lists/1/sort** とは何？

URL（リソース）はGETできることが基本。（だから動詞は変）

これをGETすると何を返すべきか？

「ソート状態」→いい考え！

でもGETを実装するにはまたアクション増やさないと...



# 改良

---

## 「ソート状態」リソースにする

動詞の操作名は、過去分詞にすると「状態（〇〇済み）」になる。

## **/todo\_lists/1/sorted**

イメージは `todo_list.sorted?` のような感じ。

→GETするとtrue/falseを返す。

よって `todo_list.sorted = true` をすればソートされるように

→PUTすればいい！

# 改良

---

```
class SortedController < ApplicationController
  # PUT /todo_lists/1/sorted
  def update
    @todo_list.sort!
    ...
  end
end
```

```
resources :todo_lists do
  resource :sorted, only: [:show, :update]
end
```

favoriteやmembershipでよくある構造と同じ！



# コントローラ増やすのめんどく さい

---

`todo_list.sorted = true`のイメージなので、それをそのまま実装する手もある。部分変更はPATCH。

```
class TodoListsController < ApplicationController
  # PATCH /todo_lists/1
  def update
    if todo_list_params[:sorted]
      @todo_list.sort!
      ...
    end
  end
end
```

例示のための単純な実装だけど、コントローラに実装すると `if` が増えてしまうので、やるならモデルに実装したほうがいい。

# コントローラを増やすメリット

---

「ソート状態」リソースには他にもいろいろ考えられて...

```
class SortingController < ApplicationController
  # PUT /todo_lists/1/sorting
  def update
    @todo_list.sort_by! (params[:sorting])
    ...
  end
end
```

```
PUT /todo_lists/1/sorting
sorting[by]=name&sorting[in]=desc
```

# 発展

---

## 適用例：

- ツイートに☆をつける
- リストの項目をソート
- ブログの記事を下書きから公開にする
- 注文の金額を請求額にあわせて修正
- 無料プランを有料プランにアップデート
- ...

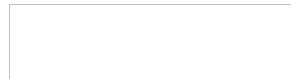
true/falseのパターンはよくあるので、コントローラを自前で書く代わりにgemにできそう？

# 宣伝

---

## RESTful Web APIs読書会

毎月第2・第4木曜日に開催中  
次回は 1/9(木)です。



<http://www.circleaf.com/groups/19>