

[翻訳]
Immutable
ActiveRecord

2025/01/28
Hikaru Kazama @ Gotanda.rb#61

自己紹介

- 名前：Hikaru Kazama (@megane42)
- 職場：株式会社ギフティ
- 趣味：かっこいいワンタイムパスワード集め



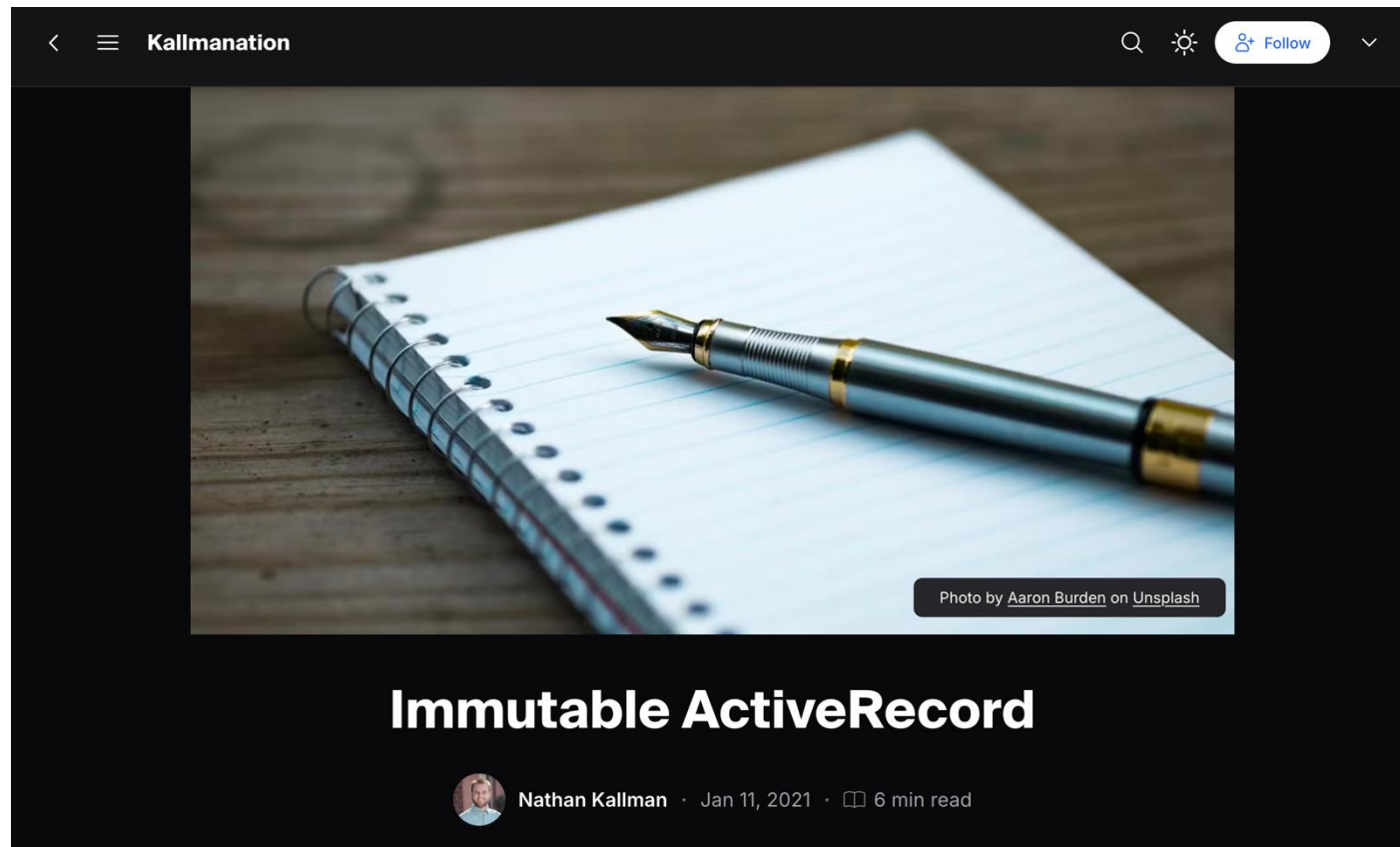
megane42 token is:

890 000

Your token expires in **11**

参考記事

- Nathan Kallman 氏のブログをほぼそのまま引用しただけ
 - <https://www.kallmanation.com/immutable-activerecord>



背景

- PointCard という ActiveRecord クラスに状態を持たせたい
 - 有効
 - 無効
 - 凍結

背景

- PointCard に status カラムを作るんじゃなくて、ステータス変更イベントレコードを積み上げて、現在ステータスはそこから「導出」するようにしたい！

こんな感じ

- **PointCardStatusChanging**
 - id
 - point_card_id
 - status
 - changed_at

課題

- PointCardStatusChanging が mutable である



```
# やめてね
```

```
PointCardStatusChanging.last.update(changed_at: "2000-01-01")
```

```
# やめて
```

```
PointCardStatusChanging.last.update(status: "disabled")
```

解法: ActiveRecord::Core#readonly?

- ActiveRecord::Core に実装されていて、すべての AR オブジェクトに対して実行できる
- #readonly? はデフォルトで false を返す
- #readonly! を呼び出すと、以後 true を返すようになる

解法: ActiveRecord::Core#readonly?

- ActiveRecord は #create, #update, #destroy のたびに #readonly? を実行しており、戻り値が true だったら **ActiveRecord::ReadOnlyRecord** エラーを起こす

ということ

- Immutable にしたクラスにこんなメソッドを定義（オーバーライド）してやれば、新規作成以外の変更ができなくなる！



```
def readonly?  
  !new_record?  
end
```

注意点

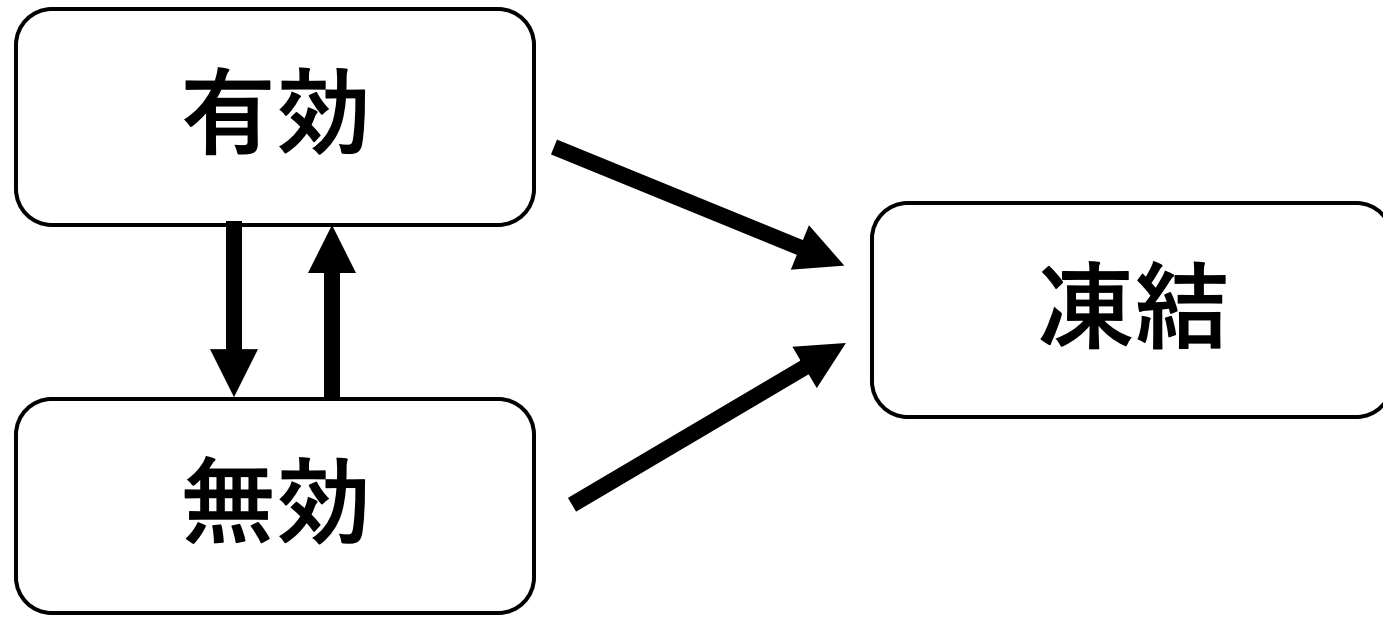
- `update_columns` みたいな「AR コールバックが実行されないメソッド」を実行したときは無力

まとめ

- AR にはオブジェクトを read only にする機構が備わっている
- #new_record? とのコンボでいい感じに immutable にできる

おまけ: 別の悩み

- 状態遷移ルールを無視してイベントレコードを作れてしまう
 - 例えば、一度「凍結」したら元には戻れないとする



おまけ: 別の悩み



```
point_card = PointCard.last  
point_card.point_card_status_changings  
    .create(status: "freezed")
```

```
# やめてね
```


```
point_card.point_card_status_changings  
    .create(status: "activated")
```

おまけ: 別の悩み

- 実際は `PointCard#activate!` とかを実装するんだらうけど、ガン無視して `PointCardStatusChanging.create` されることを誰も止められない

おまけ: 別の悩み

- PointCardStatusChanging のバリデーションとして状態遷移を実装することもできるけど、それはそれで大変
 - status の実装方法が露出しすぎという話もある



```
point_card.point_card_status_changings.build(status: "activated")
point_card.point_card_status_changings.build(status: "freezed")
point_card.point_card_status_changings.build(status: "activated")
point_card.point_card_status_changings.build(status: "freezed")
point_card.save # <= ちゃんと弾いてくれる？
```