

# Rails Meetup 2022

Hoelzle AG  
Gregor Wassmann









**HOELZLE**  
**fahrzeugelektrik.ch**

We're a team of four Ruby on Rails developers  
rethinking and migrating a legacy ERP system.

Please find slides and source code on  
[github.com/gregorw/discriminable/tree/demo](https://github.com/gregorw/discriminable/tree/demo)

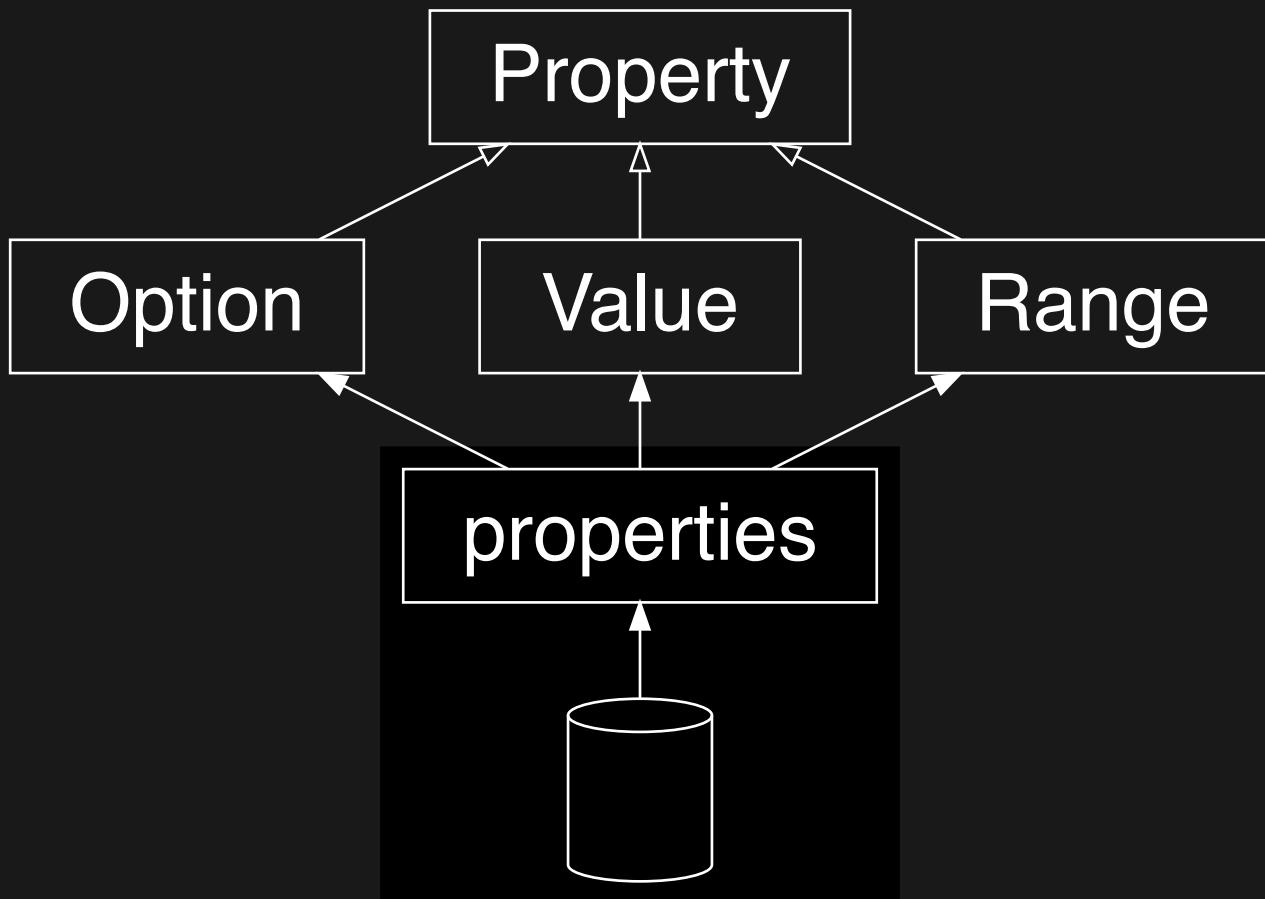
# Discriminable

[github.com/gregorw/discriminable](https://github.com/gregorw/discriminable)

More powerful STI

**Single-table inheritance** means using different classes for the same database table.

A **discriminator** figures out the class from a record based on some attribute value.



# Motivation

**HOELZE** fahrzeugelektrik.ch KONTAKT MIT ZUKUNFT

**Shop** **Marken** **Services** **Wissen**

Übersicht Neuheiten Aktionen **M**

Scheinwerfer **Arbeitsscheinwerfer**

 Halogen-Arbeitsscheinw.  
 LED-Arbeitsscheinw.

## Arbeitsscheinwerfer

[Filter einklappen](#)

<b>Spannung</b>	<b>LED</b>	<b>Lichtscheibe beheizbar</b>
<input type="checkbox"/> 12 V	<input type="radio"/> Ja	<input type="radio"/> Ja
<input type="checkbox"/> 24 V	<input type="radio"/> Nein	<input type="radio"/> Nein
<input type="checkbox"/> < 12V		
<input type="checkbox"/> 12/24 V		
<input type="checkbox"/> > 24 V		

---

<b>Lichtverteilung</b>	<b>Lichtstrom</b>	<b>IP-Grad</b>
<input type="radio"/> Kombi-Strahler	<input type="checkbox"/> ... 1000lm	<input type="radio"/> K
<input type="radio"/> Weitstrahler	<input type="checkbox"/> 3000 - 7000lm	
<input type="radio"/> Breitstrahler	<input type="checkbox"/> 7000lm ...	
	<input type="checkbox"/> 1000 - 3000lm	

---

<b>Form</b>	<b>Länge</b>	<b>Höhe</b>
<input type="radio"/> oval	0  1311 mm	24  260 mm
<input type="radio"/> eckig		
<input type="radio"/> rund		

---

<b>Breite / Tiefe</b>	<b>Durchmesser</b>	<b>Anschluss</b>
15  200	0  162	<input type="radio"/> Kabel

**HOELZE** fahrzeugelektrik.ch

HOELZE AG  
Rosengartenstrasse 11  
8608 Bubikon

# A

```
class Order
  enum state: { open: 1, completed: 2 }
end
```

# B

```
class Cart < Order
end
```

 Search or jump to... / Pull requests Issues Marketplace Explore

hoelzle / momo Private Edit Pins Unwatch 4 Fork 0 Star 0

Code Pull requests 16 Discussions Actions Wiki Security 41 Insights Settings

## Refactor Properties into separate classes #1647

Merged gregorw merged 12 commits into main from feature/discriminable on 2 May

Conversation 2 Commits 12 Checks 0 Files changed 22 +105 -70

gregorw commented on 25 Apr • edited

The `Property` model is pretty long and it includes logic about a number of different property types such as option, value and range properties. Using separate classes has a number of benefits:

- Separation of concerns / adhere to single responsibility principle
- Use separate partials using Rails naming convention
- Option for separate controllers
- Open-closed principle: Add more types without modifying the existing class
- Overall better code quality, extensibility and maintainability.
- Less database queries required.

—

Update

- Use nested classes
- Let `Property::Brand` inherit from `Property` directly — not `Property::Option`.

See #1670

Reviewers: franco, koffeinfrei

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

```
class Property < ApplicationRecord
  include Discriminable

  discriminable_attribute :kind
end

class Property::Number < Property
  discriminable_value 1
end

class Property::Option < Property
  discriminable_values 2, 3, 4 # The first mention becomes the d
end
```

# Benefits of separate classes

- Separation of concerns / adhere to **single responsibility principle**
- Use separate partials using Rails naming convention
- Option for separate controllers
- **Open-closed principle:** Add more types without modifying the existing class
- Overall better code quality, extensibility and maintainability
- **Less database queries** required

# Rails STI

<i>value</i>	string	integer	boolean	enum
single	class.name only	●	●	●
multiple	●	●	●	●

# Discriminable Gem

<i>value</i>	string	integer	boolean	enum	...
single	●	●	●	●	●
multiple	●	●	●	●	●

# Why discriminable?

- Useable with existing data (model)
- More flexibility
- Support for multiple values
- Refactor class names without the need for a data migration (loose coupling)
- 70 lines of code



# Discriminable

gem version 3.0.0

Ruby passing

maintainability A

test coverage 100%

This is a Ruby gem that implements single-table inheritance (STI) for ActiveRecord models using string, integer and boolean column types.

In other words, it allows to use any (existing) model attribute to discriminate between different subclasses in your class hierarchy. This makes storing class names in a `type` column redundant.

Also, it supports aliased attributes and *multiple* values per subclass.

## Installation

```
bundle add discriminable
```

or

```
gem install discriminable
```

## Usage

---

```
class Order < ActiveRecord::Base
  include Discriminable
```

# Experience of writing a Gem

- Great ecosystem (Rubygems, Bundler, Github Actions, etc.)
- Learning happens outside your comfort zone (Minitest, Github Matrix, Rails source code)
- Rails can (and should) learn from Java and Python, too

# Hash memoization

```
Hash.new do |map, value|
  map[value] = value.constantize._discriminable_values&.first
end
```

I hope discriminable will be useful to you.

Questions?

Thanks