

# **Rails Meetup 2020**

Hoelzle AG

Gregor Wassmann



Drehleuchte  
SATURNELLO  
Art. 2188001



Hella - Kennleuchte K-LED  
Rebelution 12/24V blitzend  
Art. 2XD455255001



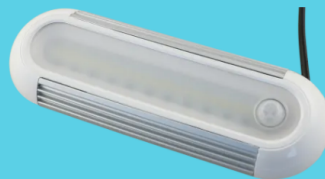
WAS - vordere LED-Blink-/  
Postionsl. 12/24 L+R  
Art. W204/1425



Schaltkasten m.  
Pressverschraubung 4  
Knöpfe  
Art. SK4PG



230V LED-Strahler 60W mit  
Tragegriff/Ständer  
Art. HL6060



Innenleuchte LED  
INPOINT III 12/24V weiss  
Art. 398900057



COBO - Schlussleuchte links  
Art. 1008701



LED-Arbeitsscheinwerfer 9-  
32V  
Art. FFL7080

[Shop](#)

## Aktuell

[Aktuell](#)**OSRAM Lichttestwochen**[Freie Stelle](#)**Kundendienst B2B, 80–100 %**







[github.com/hoelzle/rails-meetup-2020](https://github.com/hoelzle/rails-meetup-2020)

# Agenda

- Decorators vs Concerns
- Surprise

You should use...

```
class Price < SimpleDelegator  
end
```



# Thoughtbot — Evaluating Alternative Decorator Implementations In Ruby

# Decorators

- Module + Extend + Super
- Plain Old Ruby Object (PORO)
- Class + Method Missing
- **SimpleDelegator + Super + Getobj**

# Pros

- Encapsulation
- Easily Testable
- Transparent interface
- Simple
- No dependencies

It redefines `class` , but that may be the only drawback...

```
class Decorators::Price
class Decorators::SpecialPrice
class Decorators::CustomerDiscountPrice
class Decorators::PromotionPrice
class Decorators::ProductDiscountPrice
class Decorators::GroupDiscountPrice
class Decorators::RegularPrice
```

```
1  module Decorators      Alessandro Rodi, a year ago • Display customizations (#181) • e0e06c9
2      class CustomizationsPreload < Collection
3          def initialize(products, customer, *decorators)
4              super(products, customer, *decorators)
5              # TODO: will this break in Rails 6.0? https://github.com/rails/rails/issues/36638
6              ActiveRecord::Associations::Preloader.new.preload(
7                  products, :customizations, Customization.where(customer: customer).order(:created_at)
8              )
9          end
10     end
11 end
12
```

```
class ProductsController < ApplicationController
  def index
    @products = Decorators::CustomizationsPreload.new(
      @products, current_customer, Decorators::Price
    )
  end
end
```

# Refinements



```
module DiscountFactor
  refine Numeric do
    def factor
      1 - self / 100.0
    end
  end
end
```

```
module Decorators
  class RegularPrice < Decorators::Context
    using DiscountFactor

    def net_price
      price * discount.factor
    end

    # Omitted code here
  end
end
```

# See also

- Draper
- Trailblazer

[github.com/hoelzle/rails-meetup-2020](https://github.com/hoelzle/rails-meetup-2020)

# Thanks Rails Community ❤️

What I learned at the meetup was:

- <https://crystal-lang.org/>
- <https://www.luckyframework.org/>