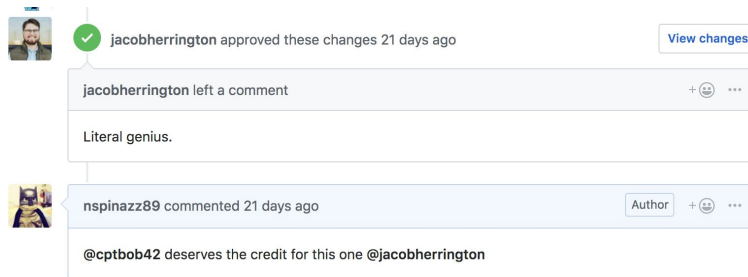# My Time on the Watch

Bugs and Back Again, an Engine Tale

It started easy enough...

- A few bugs around engine mail

- A one line fix which earned a compliment
  from Herrington...

- ...that I couldn't take credit for

Then comes the sidebar...

- Dashboard
- Reports
- Orders
  - Orders
  - Premade Carts
  - Subscriptions
- Products
  - Products
  - Option Types
  - PropertyTypes
  - Taxonomies
  - Display Order
  - Stock
- Promotions
  - Promotions
  - Promotion Categories
- CMS

- Rearrange the sidebar

- Sounds simple enough...

- Just need to add a solidus update with a

  menu position argument right?

Nope

- Many menu items are inserted individually by solidus plugins
- So just write a class that rearranges the menu items after they have all been inserted...right?
- Nope snippet to the right shows that several of the menu items being inserted have their own position argument usage.

```
def menu_items
  @menu_items ||= [
    MenuItem.new(
      ORDER_TABS,
      'shopping-cart',
      condition: -> { can?(:admin, Spree::Order) },
      position: 0
    ),
    MenuItem.new(
      PRODUCT_TABS,
      'th-large',
      condition: -> { can?(:admin, Spree::Product) },
      partial: 'spree/admin/shared/product_sub_menu',
      position: 1
    ),
    MenuItem.new(
      PROMOTION_TABS,
      'bullhorn',
      partial: 'spree/admin/shared/promotion_sub_menu',
      condition: -> { can?(:admin, Spree::Promotion) },
      url: :admin_promotions_path,
      position: 2
    ),
    MenuItem.new(
      STOCK_TABS
```

The (Ugly?) Solution

- config/initializers/sidebar_menu.rb

- Functions as an after_initalize

- Empties out the existing menu items and then rebuilds them in the way we want them

```ruby
if Spree::Backend::Config.respond_to?(:menu_items)
  Spree::Backend::Config.configure do |config|
    config.menu_items = []
    config.menu_items << config.class::MenuItem.new(
      DASHBOARD_TABS,
      'dashboard',
      label: 'dashboard',
      condition: -> { can?(:display, Spree::Order) },
      position: 0
    )
    config.menu_items << config.class::MenuItem.new(
      REPORT_TABS,
      'file',
      label: :reports,
      condition: -> { can?(:admin, :reports) },
      position: 1
    )
    config.menu_items << config.class::MenuItem.new(
      ORDER_TABS,
      'shopping-cart',
      label: :orders,
```