The Assets Gem

The Shopify experience in every app

The problem

We've invested so much time and effort into giving the admin a great UI. How do we share this with other apps without repeating ourselves?

Two solutions

1. **Move common front end assets to another place, consume it from the admin and other apps**
2. **Copy common assets from the admin to another place to share with apps**

Two solutions

1. *Move* common front end assets to another place, consume it from the admin and other apps
2. *Copy* common assets from the admin to another place to share with apps

We have a lot of apps now!

- channels: Facebook, Twitter, Pinterest, Buy Button, Amazon, Facebook Messenger
- embedded apps: Digital downloads, Wholesale, Barcode printer, Shopify Scripts, Bulk Discounts
- integrations: Accounting, UberRUSH, Avalara
- experiments: React Reports

Github search: `shopify_assets user:shopify filename:Gemfile`

Move

Relocating the most global and shared assets to a common dependency of all apps that need the admin styles: fonts, layout, the grid, forms, etc.

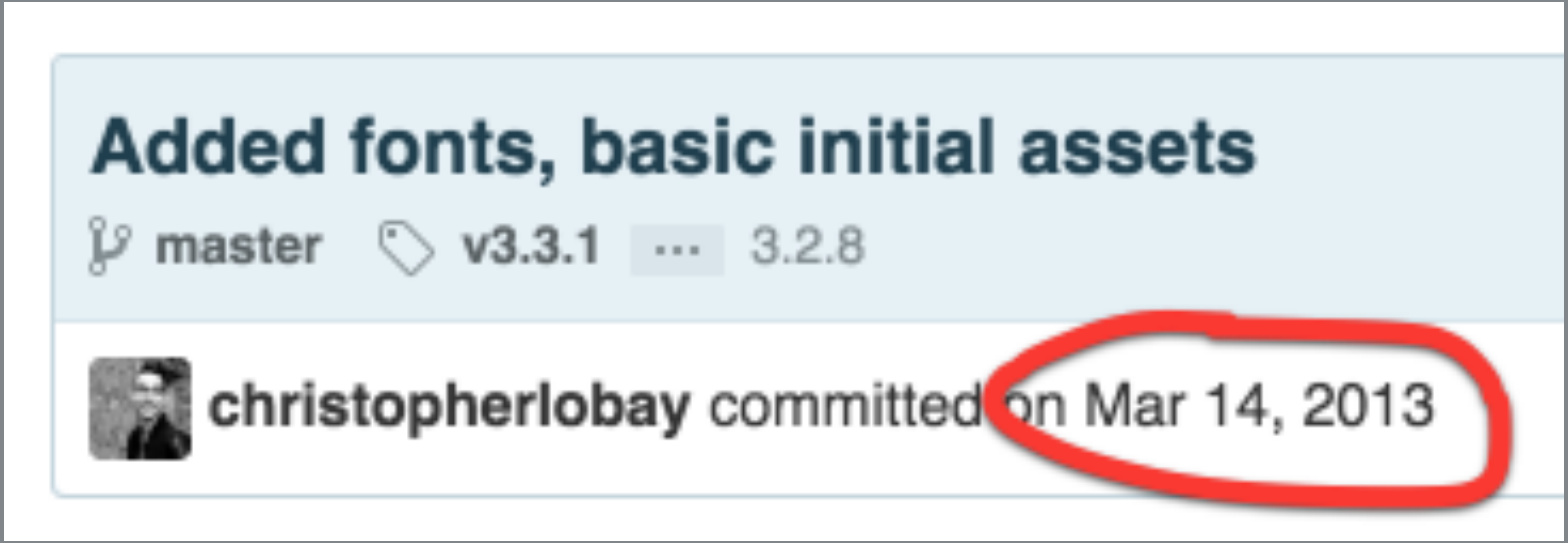
- Central location for all assets, no duplication
- Slows down development of the admin: developers need to make changes in a different repository and then update the dependency in the admin

Copy

Copy assets to a different repository as external apps need them

- Doesn't affect admin development
- Slows down development of apps: something broken in the admin needs to be fixed there then propogated to apps via dependency
- Forget backwards compatibility

Some history



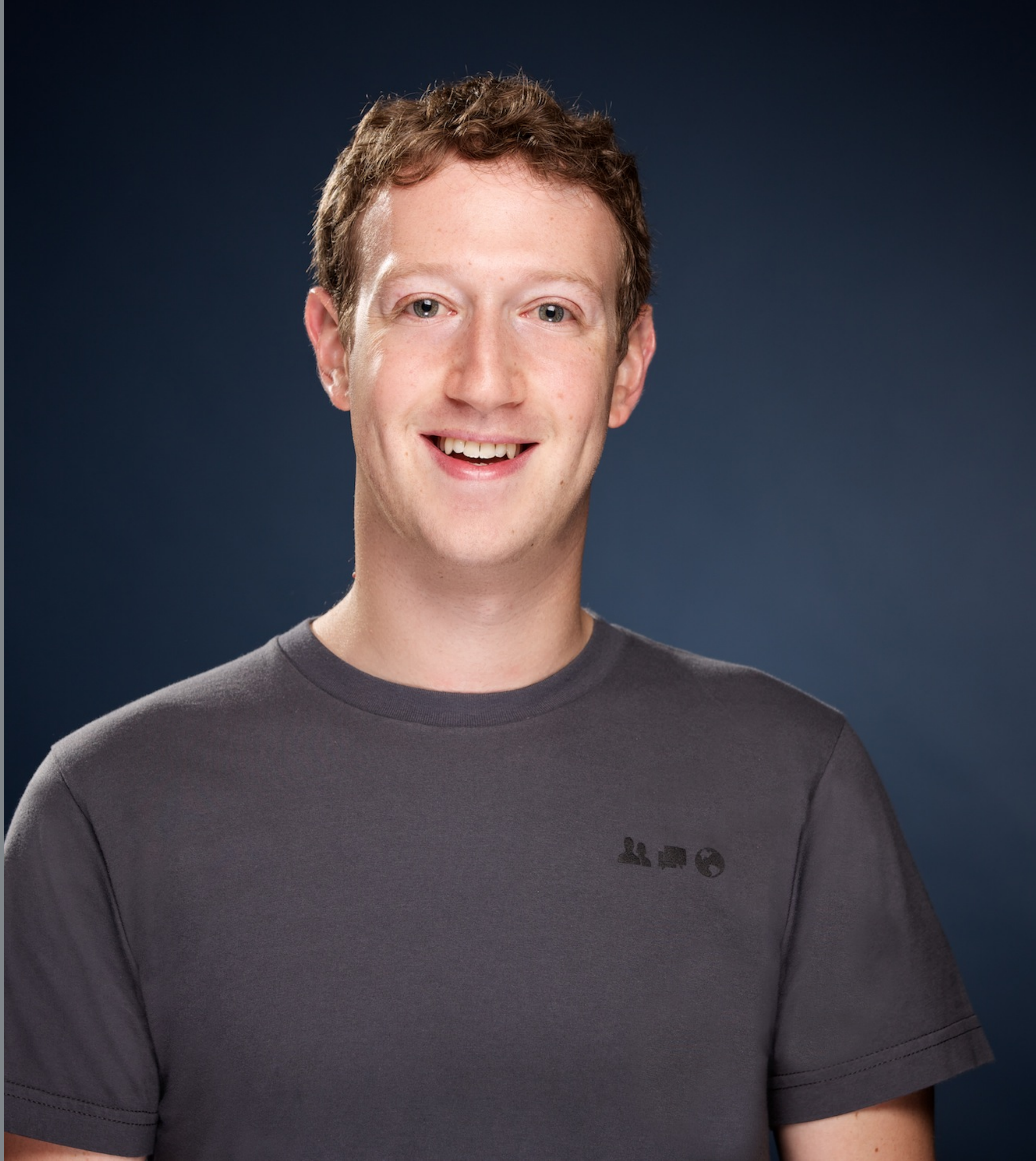
Added fonts, basic initial assets

christopherlobay committed on Mar 14, 2013

V1

- beloved `admin2` assets
- also contained assets used on brand sites

Then came Facebook and Buy Button

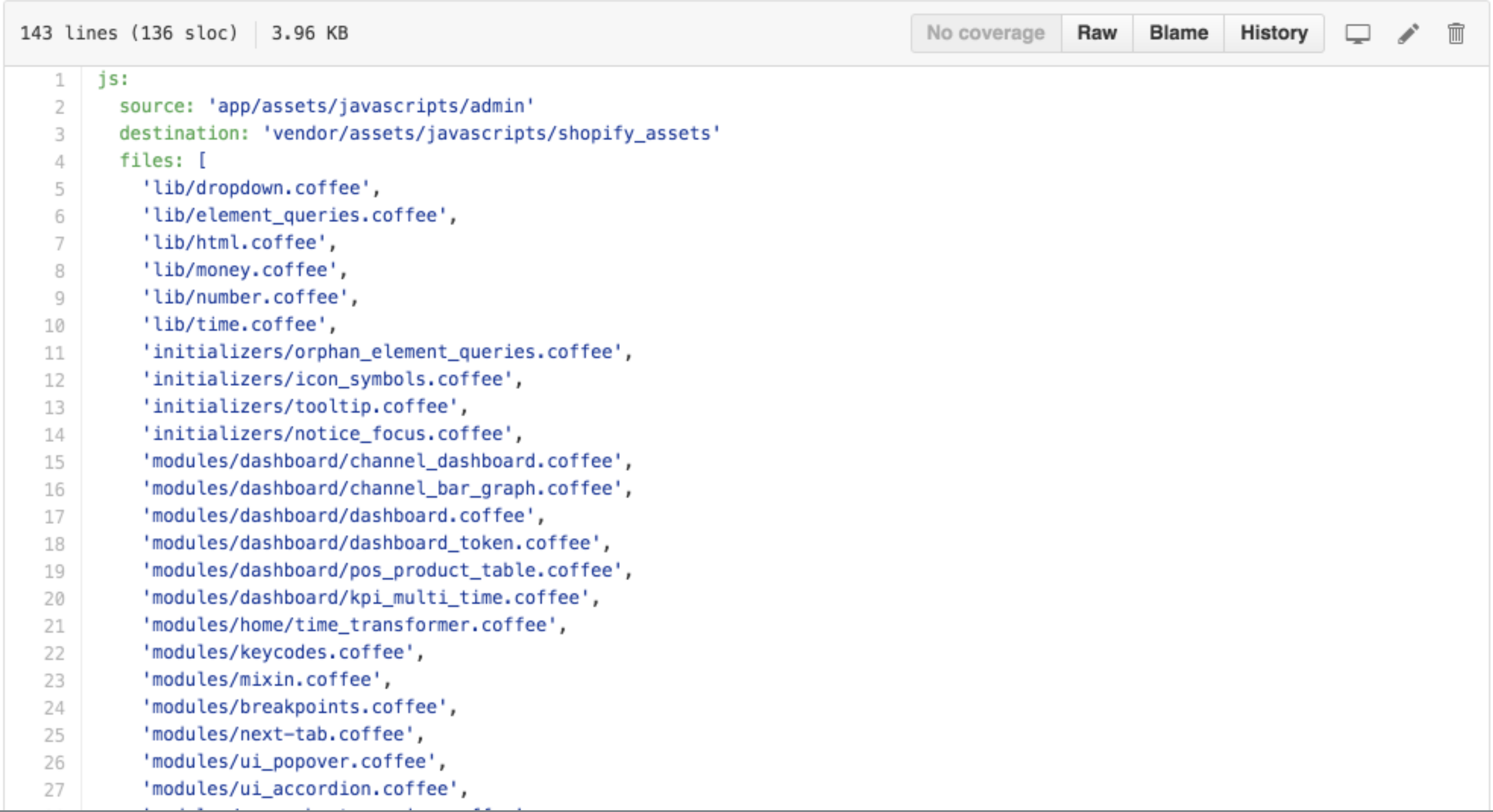


V2

- Automate the process as much as possible
- Make UI Components work in apps
- Reduce the amount of assets included in each app's application bundle

The process

1. We maintain a whitelist of files we want out of the admin.
2. We copy these files into the gem
3. Release a new version
4. Update in consuming apps
5. Rinse
6. Repeat



```
js:
  source: 'app/assets/javascripts/admin'
  destination: 'vendor/assets/javascripts/shopify_assets'
  files: [
    'lib/dropdown.coffee',
    'lib/element_queries.coffee',
    'lib/html.coffee',
    'lib/money.coffee',
    'lib/number.coffee',
    'lib/time.coffee',
    'initializers/orphan_element_queries.coffee',
    'initializers/icon_symbols.coffee',
    'initializers/tooltip.coffee',
    'initializers/notice_focus.coffee',
    'modules/dashboard/channel_dashboard.coffee',
    'modules/dashboard/channel_bar_graph.coffee',
    'modules/dashboard/dashboard.coffee',
    'modules/dashboard/dashboard_token.coffee',
    'modules/dashboard/pos_product_table.coffee',
    'modules/dashboard/kpi_multi_time.coffee',
    'modules/home/time_transformer.coffee',
    'modules/keycodes.coffee',
    'modules/mixin.coffee',
    'modules/breakpoints.coffee',
    'modules/next-tab.coffee',
    'modules/ui_popover.coffee',
    'modules/ui_accordion.coffee',
```

What's included

- UI Components
- Core CSS and Javascript
- SVGs
- Dependencies like Twine, Turbograft, Lodash, jQuery. Anything we need to replicate the Shopify UI

What's not included

- Area-specific CSS and JS
- old Data URI icons

A note about versions

We're using this weird format for versions in the gem:

```
9.2.1.shopify-41a4fd1735d4fc7378fe463cb527598dc0d60869
```

Getting set up

1. **Include the assets gem in your** `Gemfile`

```
gem 'shopify_assets',
git: 'https://shopify-
dep:<key>@github.com/Shopify/assets'
tag:
'9.2.1.shopify-41a4fd1735d4fc7378fe463cb
527598dc0d60869'
```

1. **Import the base styles into your** `application.scss`**:**

   ```
   @import 'shopify_assets/assets-base';
   ```

2. **Include some JS to help you use Shopify JS outside the admin:**

   ```
   #= require shopify_assets/channel
   ```

```
window.channel = new Channel.app()
channel.init()
channel.initTurboGraft()
```

Using admin styles

Import stylesheets as necessary:

```
@import 'shopify_assets/modules/ui-banner';
@import 'shopify_assets/modules/next-headings.scss';
```

Use UI components in your views!

```
<%= ui_banner(status: 'error') do |banner| %>
  <% banner.title 'My cool banner' %>
  <% banner.content do %>
    <p>Cool banner content.</p>
  <% end %>
<% end %>
```

Problems

- No Backwards compatibility. Updating the gem in your project could break *anything*.
- Not many tests. We ensure that the assets we bring in compile, but we're not copying over any of the unit tests from admin.
- Some components that depend on pieces of shopify don't work yet.

Contributions welcome!

Keeping up with the assets gem

Visit us in #assets-gem
Keep up with new versions on the github releases page: http://github.com/shopify/assets/releases

Questions?