PatternFly Community Meeting

April 2018

PATTERNFLY

Agenda

- Accessibility Update

- Migrating from Angular-PF (angular v1) to:

  - PF-NG (angular 4+)

  - PF-React

- RCUE Update

- PatternFly Repo Teams updates

- PatternFly Next Update

- PatternFly Metrics Update

redhat.

Announcements

- Check out the [PatternFly Library Overview page](#)

- [Forum survey](#)

- Recent blogs
  - [PatternFly Next Update](#)
  - [Accessibility is Usability](#)

- [We have a mailing list, sign up!](#)

Accessibility Update

PATTERNFLY

Proposal for Accessibility in PatternFly 3

Starting on Global Accessibility Awareness Day, May 17, any new contributions would:

- ○ Be keyboard accessible

- ○ Follow the 5 rules of ARIA

redhat.

1st rule of ARIA

If you can use a native HTML element or attribute, then do so.

Under what circumstances may this not be possible?

- the HTML element is not yet accessible
- the HTML element can not be styled as required
- the HTML element does not yet exist

redhat.

2nd rule of ARIA

Do not change native semantics, unless you really have to.

Do not do this:

```
<h2 role="tab">
  heading tab
</h2>
```

Do this:

```
<div role="tab">
  <h2>heading tab</h2>
</div>
```

3rd rule of ARIA

All interactive ARIA controls must be usable with the keyboard.

- Complex gestures must have a keyboard equivalent
- Interactive widgets must respond to standard keystrokes

redhat.

4th rule of ARIA

Do not use role="presentation" or aria-hidden="true" on a visible focusable element.

Do not do this:

```
<button role="presentation">
  press me
</button>
```

Do not do this:

```
<button aria-hidden="true">
  press me
</button>
```

5th rule of ARIA

All interactive elements must have an accessible name.

Do not do this:

```
<label>user name</label>
<input type="text">
```

Do this:

```
<label for="uname">
  user name
</label>
<input type="text" id="uname">
```

redhat.

Proposal for Accessibility in PatternFly 3

- Thoughts or concerns about this proposal?

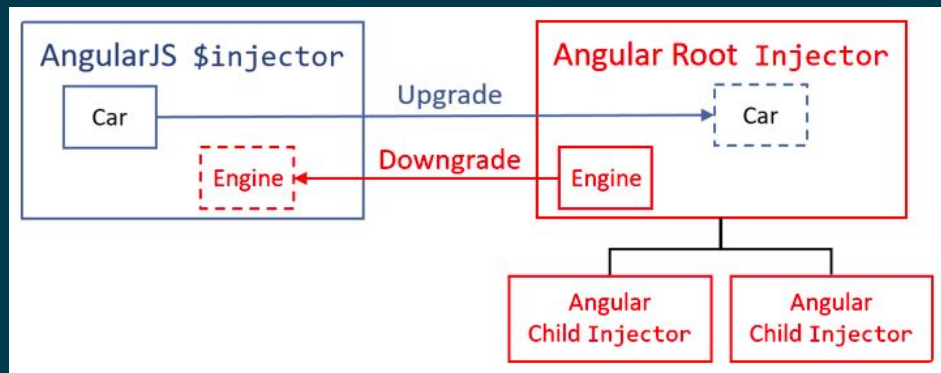- Anyone want to join the accessibility review team in Git Hub?

redhat.

Migrating from Angular-PF (Angular v1) to PF-NG (Angular v4+)

PATTERNFLY

Migrating from Angular-PF to PF-NG

Angular 5 has an ***UpgradeModule*** which runs a 'hybrid mode' where AngularJS and Angular 5 code can run in same application. Both frameworks run side by side, and can communicate with each other.

- Can incrementally or directly port AngularJS directives/components to Angular5 and run in hybrid mode. Once all is converted over, drop AngularJS (v1)
- Reports of performance hit of 20% slower in hybrid mode
  - Can be optimized using some of best practices shown here: https://github.com/mgechev/angular-performance-checklist

More Info: https://angular.io/guide/upgrade

Migrating from Angular-PF to PF-NG

Prerequisites:

- Typescript
- Module Loader: WebPack (recommended), or SystemJS or Browserify
- Must convert AngularJS Directives to AngularJS 1.5 Components
  - Directives can not be 'upgraded' using UpgradeModule
- Check for Angular 2+ replacements for your AngularJS dependencies!

Usage:

- Can *downgrade* an Angular5 (patternfly-ng!) component to run in AngularJS
- Can *upgrade* a AngularJS 1.5 Component to run in Angular5
  - Requires a small/thin Angular5 wrapper

Recommendation:

Incrementally port AnguarJS 1.5 Components and Directives to Angular 5 syntax and ***downgrade*** them to run in AngularJS. Once all is converted over, drop AngularJS

redhat.

Migrating from Angular to React

PATTERNFLY

Migrating from Angular to React

Prerequisites:

- Migrate to Redux first*
- WebPack
- Package to consume React components in your Angular app.
  - coatue-oss/react2angular
  - ngReact/ngReact

Strategy:

- Bottom-up, incremental approach (start with "leaf" or "presentational" components)
- Convert all leaves, then convert "container" level components
- Last components to be converted will be "page" level
- Convert routing and index files
- Remove Angular!

* Only necessary if you plan on using Redux in your React app

redhat.

Migrating from Angular to React

Gotchas:

- No more Angular dependency injection. Use require.
- Sharing Angular dependencies across multiple React children
- Angular two-way data binding to React one-way

Testing:

- Unit testing can often be accomplished using existing test infrastructures with the added Enzyme test library
  - Install Enzyme
  - Import enzyme and react into test spec
  - Add test to current unit test subdir, and test infrastructures like Karma will run additional react tests alongside existing Angular tests.

redhat.

RCUE Update

PATTERNFLY

RCUE Update

- https://redhat-rcue.github.io/ updated with proper screenshot of the login pattern
- Broken link to PatternFly login pattern fixed
- Login, Navbar and About modal updated to match the latest designs
- PR submitted to include RCUE with PatternFly Core as a theme, rather than a separate repo with automated updates every time PatternFly is updated.

redhat.

PatternFly Repo Team Updates

PatternFly Design
PatternFly Design Kit
Angular PatternFly
PatternFly Ng
PatternFly React

PATTERNFLY

PatternFly Design Repo Team Update

Matt Carrano

- Revised Login Page designs
  - Standard Login
  - SSO Login
  - Multi-Factor Login
- Revised Filter pattern enabled a more flexible approach
- Updates to Required Field labeling approach on forms
- Introduce Context Selector

redhat.

PatternFly Design Kit Repo Team Update

Matt Carrano and Kyle Baker

- The PatternFly Design Kit is a Sketch based template for designers.

- Create PatternFly consistent, high quality designs more quickly.

- Serves as the consistent source of truth.

- Utilizes Sketch library feature which provides reusable symbols and keeps all of your files in sync.

- GitHub repo contains a symbol, template, read me file.
    - Read instructions on how to install the library and download updates
    - Consume all or part of the kit

redhat.

Angular PatternFly Repo Team Update

Colleen Hart and Dave Taylor

- Implemented "Eyes On The Core" weekly task/meeting

- Added Support for Less to Sass Conversion - Thanks @TheRealJon

- Enhanced or fixed bugs for various components:
  - Notification Drawer
  - Empty State
  - Table View
  - Wizard

- Added new pattern components:
  - Filter Panel - Thanks @dtaylor113 @jcaianirh @rhamilito
  - Modal Overlay - Thanks @nicolethoen
  - Datepicker - Thanks @nicolethoen

- Big thank you to all of our contributors

- Stay tuned for an upcoming community survey! Previous survey results available

redhat.

PatternFly Ng Repo Team Update

Cat Robson and Dan Labrecque

- Table implementation with ngx-datatables (thanks @dlabrecq!)
  - Supports patternfly pagination, toolbar, and empty state
  - Drag and drop support
  - Full table view pattern
  - Expansions & Group headers
- App launcher (thanks @akoserwal!)
- Updated demo site with clearly separated components, pipes, directives, and services (thanks @dlabrecq!)
  - http://www.patternfly.org/patternfly-ng
- ES6 modules can now be imported individually (thanks @dlabrecq!)
  - import { NotificationModule } from 'patternfly-ng/notification';

redhat.

PatternFly React Repo Team Update

Roxanne Hoover, Patrick Riley and Jeff Phillips

- New Components! So many we can't list them <u>all</u>...
  - Notification Drawer, Cards, Breadcrumbs, Tables Galore
  - Thank you to all our contributors. Teamwork makes the dreamwork!
- Keepin' it Fresh with React 16.3 Upgrades
- Tooling Updates:
  - Now using Enzyme for testing
  - Incorporating Yarn for package management
- Investigating:
  - MonoRepos
  - Structural Patterns
- <u>Next PatternFly React Community Meeting in May...Stay tuned!</u>

redhat.

PatternFly Next Update

Check out the [PatternFly-Next demo recording](#) and then [give us your feedback](#).
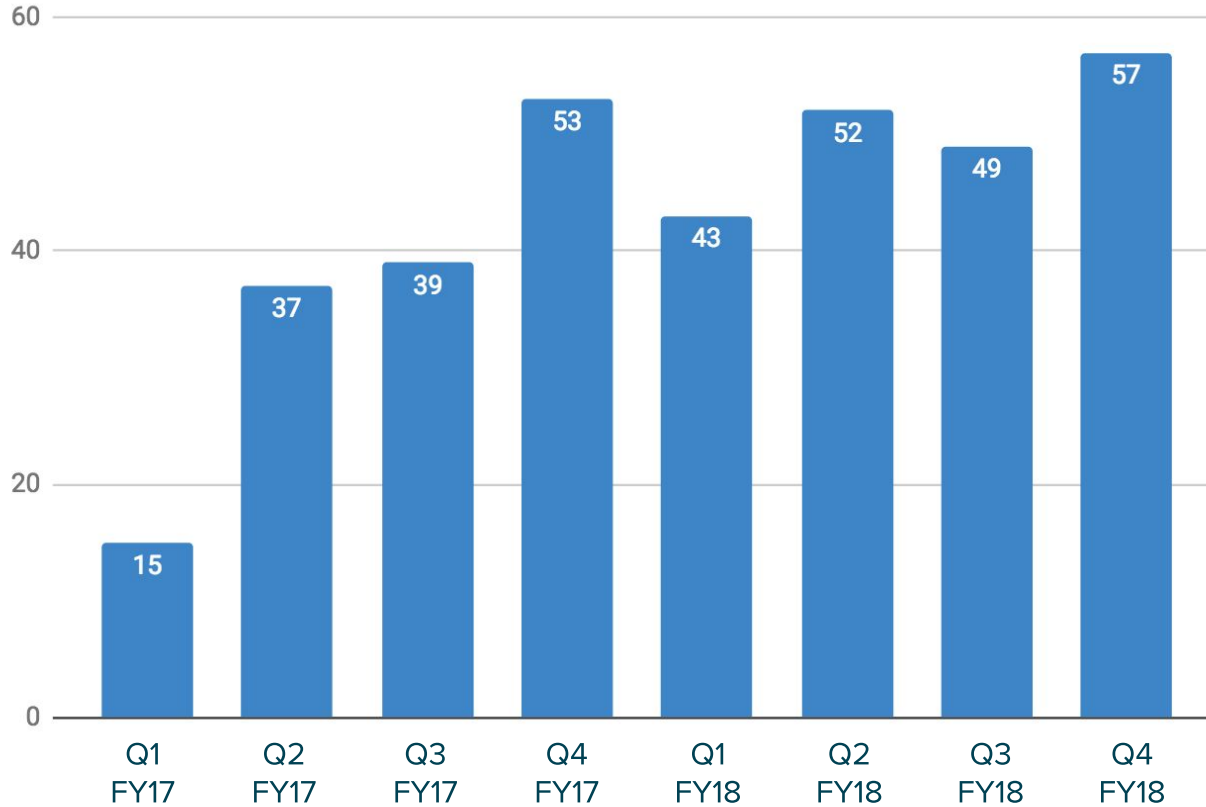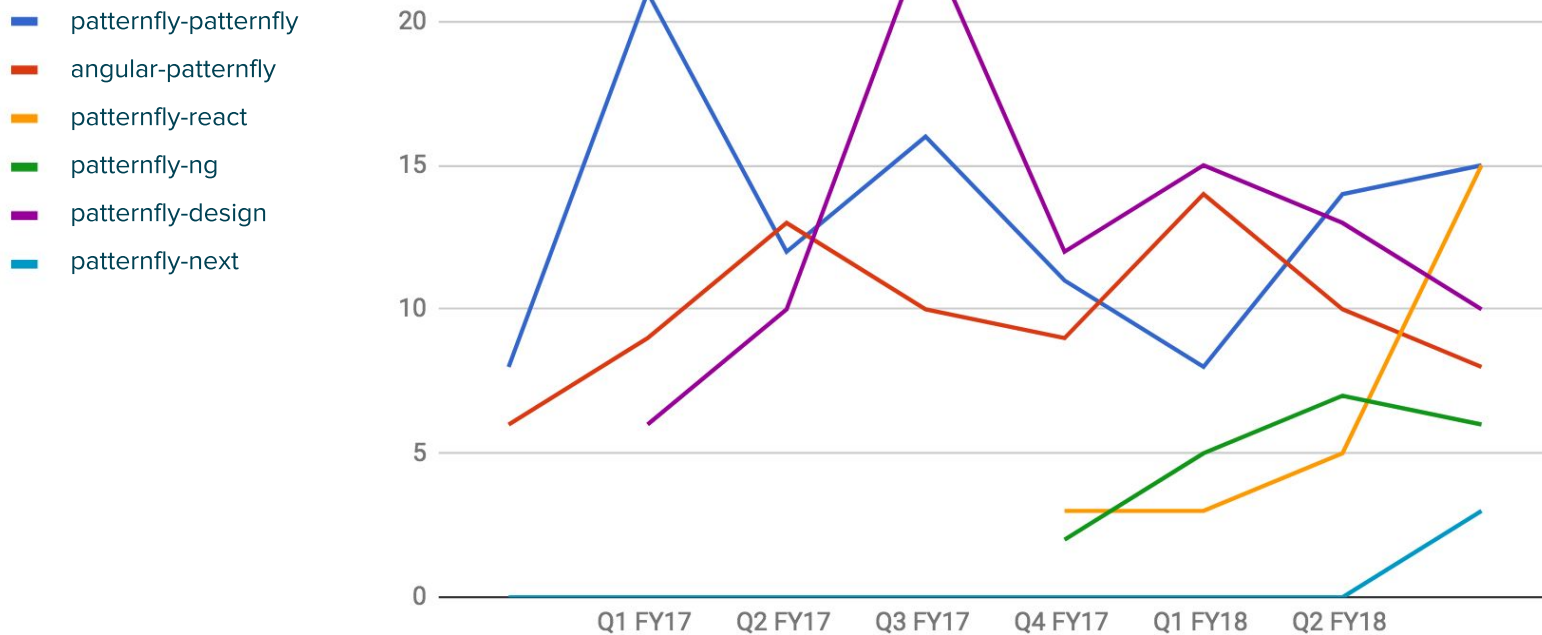
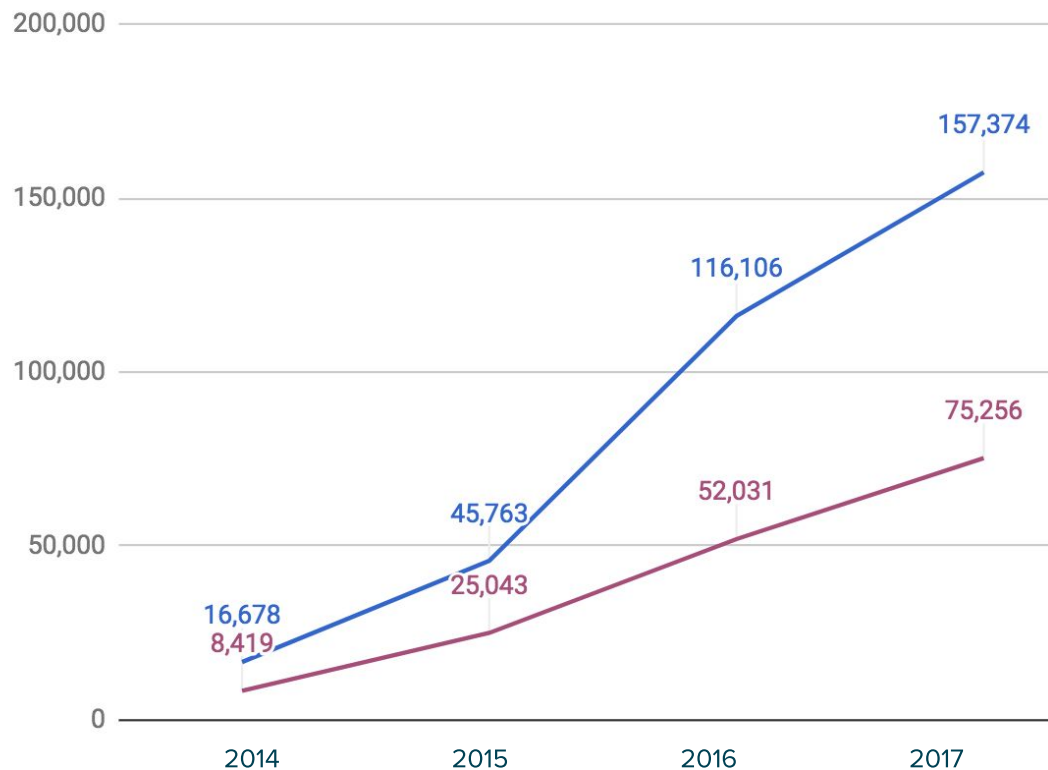PATTERNFLY

PatternFly Metrics

Leslie Hinson

PATTERNFLY

Total Quarterly Contributors

Quarterly Contributors Breakdown by Repo

- patternfly-patternfly
- angular-patternfly
- patternfly-react
- patternfly-ng
- patternfly-design
- patternfly-next

Traffic on the PatternFly Site

Sessions: Total number of Sessions within the date range. A session is the period time a user is actively engaged with your website.
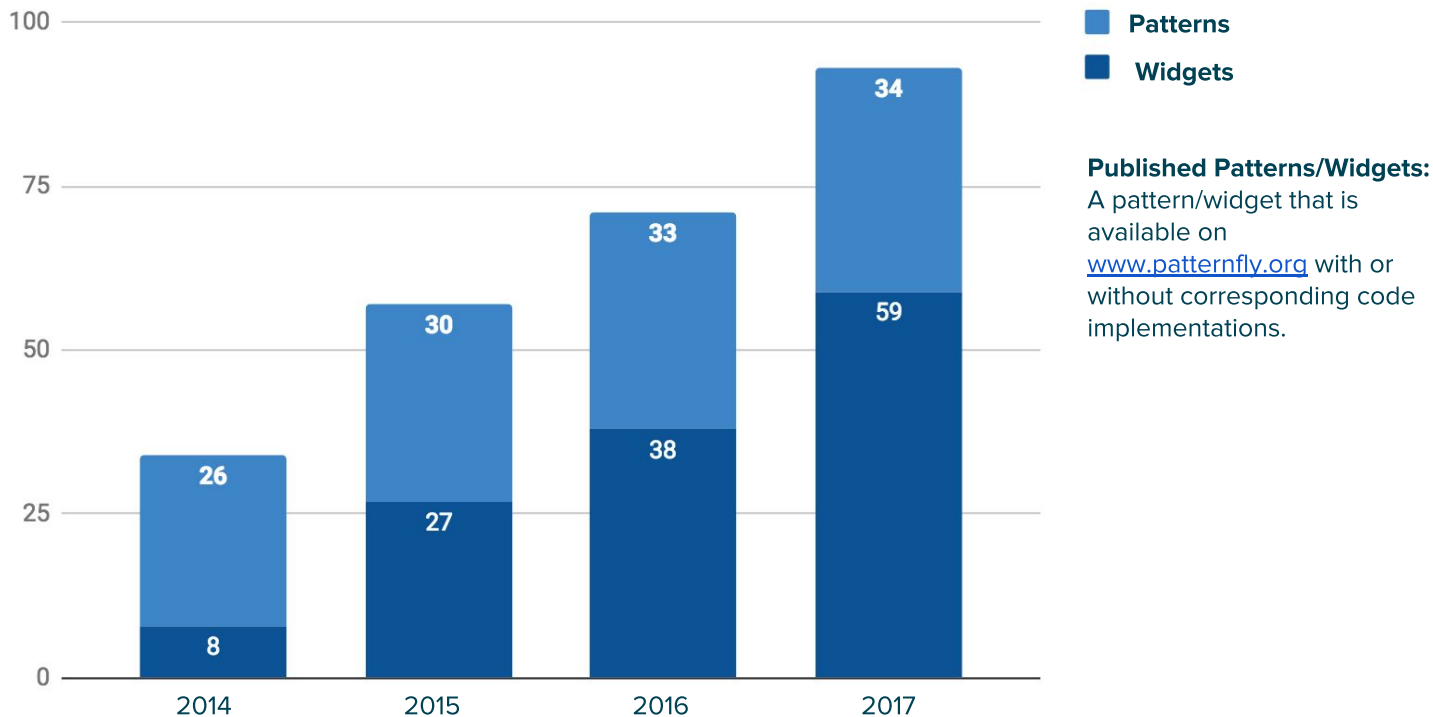
Users: Individuals that have initiated at least one session during the date range

The PatternFly blog sessions and users are included in these totals.

Average sessions per month

- 2014: 1,390/month
- 2015: 3,813/month
- 2016: 9,676/month
- 2017: 13,115/month

Published Patterns and Widgets

Patterns
Widgets

Published Patterns/Widgets:
A pattern/widget that is available on www.patternfly.org with or without corresponding code implementations.

REQUESTS/QUESTIONS?