



Reservoir

NYPL RESERVOIR DESIGN SYSTEM

Banner Component

Introduction

The `Banner` component is a non-modal semantic dialog used to communicate a general status event or to promote a feature, providing contextual feedback messages for typical user actions. They are displayed contextually within a page flow and they will often prompt a user to take action.

IMPORTANT: The `Banner` component is similar in form and function to the `Notification` component. However, the `Notification` component should be used for global messaging located at the top of the site, above the NYPL header, while the `Banner` component should be used for all messaging that sits within the flow of the page.

Business Drivers

As patrons interact with web apps, feedback in response to a specific user action or informative messages related to a task that needs to be completed can be important elements in effective and inclusive UX. The `Banner` component will provide a UI element with semantic coloring and iconography related to each of its variant's intended use that will allow important information to be inserted into an existing page flow. Recommendations for the banner content will ensure that a banner's intention and meaning will not be solely reliant on the coloring and iconography.

Examples that would benefit from the `Banner` component:

- Error messages on forms
- My Account page in Research Catalog

Specifications

Component Category: Feedback

Variants

Using the `type` prop, the component can be configured to render with differing colors and iconography.

Heading Element

Using the `heading` prop, the text for the heading element can be set.

If the `heading` prop is omitted or left empty, the heading element will not be rendered.

The default DS `Heading` component will be set as `level="h2"` and `size="heading6"`.

If a custom DS `Heading` component is passed, the `size` prop on the custom `Heading` component will be forced to `"heading6"` to adhere to NYPL design standards. The `level` prop will not be altered.

To align with NYPL design standards, the max width of the heading element is set at 800px.

Body Content

Using the `content` prop, the component body can be populated. The `content` prop can be set as a string or it can be composed using HTML and DS components.

To align with NYPL design standards, the max width of the component body is set at 800px.

Dismissible

Using the `dismissible` prop, the component can be configured to allow users to close/hide the banner element in the browser. When `dismissible` is true, a close X button will show in the upper right corner of the banner UI. Clicking the close X button will remove the banner element from the DOM. Refer to similar functionality in the DS Notification component.

Custom Colors

Using the `highlightColor` and `backgroundColor` props, the component can be configured to override the default variant highlight and background colors. The value of the `highlightColor` prop is used for the left border and the icon fill. The value of the `backgroundColor` prop is used for the background of the full component.

The custom color props can not be used independently. When one of the custom color props is set, the other prop must also be set.

Custom Icon

Using the `icon` prop, the component can be configured to override the default variant icons. To do so, a DS `Icon` component must be passed to the `icon` prop.

Accessibility

General

- All user interface components (i.e. buttons, links) must be accessible by keyboard.
- All user interface components must show a visible focus indicator that highlights the currently selected element. The focus indicator must not rely on color alone. If using color then there should be 3:1 contrast between the selected and non-selected state. If using an outline or similar then it should have 3:1 contrast with adjacent colors.
- All user interface components must have a meaningful accessible name (label).
- Containers must be able to scale to allow font size to increase to 200%.
- The color contrast of text to background colors should be 4.5:1.
- The color contrast of icons to background colors should be 3:1.
- Any iconography should be considered decorative. Any text content within the banner should convey all the necessary meaning regardless of the presence of an icon.
- Semantic markup should be used for the content of the banner as appropriate (e.g., headings, lists, etc).

Specific for Banner

The `Banner` component renders with an HTML `<aside>` element as its wrapper. This is an HTML landmark element that is similar to adding an attribute of `role="complementary"`. For accessibility purposes, landmark elements should not be rendered inside other landmark elements such as the `header` and `footer` landmark elements. However, adding a `Banner` component inside an HTML `main` landmark element is acceptable.

ARIA Label

Every `Banner` component must have a unique `aria-label` attribute. This is set through the `ariaLabel` prop. A unique `aria-label` value, along with the `<aside>` HTML landmark element, helps screen readers better navigate a page with multiple `Banner` components.

Icons

Icons rendered in the `Banner` component are decorative by default which means that they are hidden to screen readers. Since the "X" close icon inside the dismissible button is decorative and because there is no discernible text inside the button, an `aria-label` attribute is added to the button.

Language Patterns

When the `Banner` component is rendered, the text within the component should always include wording that aligns with the type of `Banner` that is being delivered. For example, if the banner is a warning or error, the heading or body copy within the banner should use the words "warning" or "error" and provide a clear description of the existing or potential issue.

Dynamic Notifications

If dynamically generated banners are required, a container into which the banner will be injected should be included in the DOM when the page first loads. The container element should have `aria-live="polite"` applied. The container element should always be rendered in the DOM regardless of the presence of the `Banner` component. This DOM structure will allow the newly generated content to be recognized and announced by assistive technology.

```
<div aria-live="polite">
  <Banner {...props} />
</div>
```

Resources

- [MDN Aria: complementary role](#)
- [Deque Creating Accessible SVGs](#)
- [CSS Tricks Accessible SVG Icons](#)
- [Reservoir Accessibility Guide: Dynamic Content](#)

UX Considerations and Best Practices

Semantic Variants

It is recommended to align the usage of the `Banner` component to the semantic meaning of one of the component's variants. When using the `Banner` component, the implementation should not rely on the color alone to convey meaning, as color is not a universal experience for everyone. The meaning of the heading and body copy should align with the general meaning associated with the semantic colors used in a specific variant.

Link Colors

To ensure accessibility color contrast compliance, links will always be the same color as the plain text. They will not use the standard link color. Because of this, all links within the component will be underlined.

Body Content

Body content is required for all `Banner` components. Body content should be concise, descriptive, and, if applicable, give users clear next steps. Language used should be accessible and easily understood. Tone used should be appropriate for the situation and notification. Lengthy blocks of text should be avoided - we recommend no more than 2-3 sentences. If large volumes of text are necessary, they should be broken down into short paragraphs or bullet points and/or truncated via the use of a `View More` component within the `Banner` component.

References

Figma Designs

<https://www.figma.com/file/...>

React Component Props

ariaLabel: string

OPTIONAL

Description Used to describe the component's `<aside>` HTML element.

Default *n/a*

backgroundColor: "brand.primary-05" |
 "section.blogs.primary-05" |
 "section.books-and-more.primary-05" |
 "section.connect.primary-05" |
 "section.education.primary-05" |
 "section.locations.primary-05" |
 "section.research.primary-05" |
 "section.research-library.lpa-05" |
 "section.research-library.schomburg-05" |
 "section.research-library.schwarzman-05" |
 "section.whats-on.primary-05" |
 "dark.brand.primary-05" |
 "dark.section.blogs.primary-05" |
 "dark.section.books-and-more.primary-05" |
 "dark.section.connect.primary-05" |
 "dark.section.education.primary-05" |
 "dark.section.locations.primary-05" |
 "dark.section.research.secondary-05" |
 "dark.section.research-library.lpa-05" |
 "dark.section.research-library.schomburg-05" |
 "dark.section.research-library.schwarzman-05" |
 "dark.section.whats-on.primary-05" |

OPTIONAL

Description Used to set the color of the background for the full component. Refer to how color values are defined and typed in the DS Icon component.

If `backgroundColor` is pass and `highlightColor` is NOT passed, the `backgroundColor` prop should be ignored and the following console warning should be generated:

```
NYPL Reservoir Banner: The `backgroundColor` prop has
been passed, but the `highlightColor` prop has not
been passed. Because of this, the `backgroundColor`
prop will be ignored.
```

Default *n/a*

className: string
OPTIONAL

Description CSS class name that will be added to the component's parent element.

Default *n/a*

content: string | JSX.Element
REQUIRED

Description Used to populate the body content of the component.

Default *n/a*

heading: string | JSX.Element
OPTIONAL

Description Used to populate the heading element within the component. A string value can be passed to set the text for a default `DS Heading` component, or an actual `DS Heading` component can be passed in.

If the `heading` prop is omitted, a heading UI element will not be rendered in the component.

The default `DS Heading` component should be set as `level="h2"` and `size="heading6"`.

If a custom DS `Heading` component is passed, the `size` prop on the `Heading` component should be forced to `"heading6"`.

Default

n/a

```
highlightColor: "brand.primary" |
"section.blogs.primary" |
"section.books-and-more.primary" |
"section.connect.primary" |
"section.education.primary" |
"section.locations.primary" |
"section.research.secondary" |
"section.research-library.lpa" |
"section.research-library.schomburg" |
"section.research-library.schwarzman" |
"section.whats-on.primary" |
"dark.brand.primary" |
"dark.section.blogs.primary" |
"dark.section.books-and-more.primary" |
"dark.section.connect.primary" |
"dark.section.education.primary" |
"dark.section.locations.primary" |
"dark.section.research.secondary" |
"dark.section.research-library.lpa" |
"dark.section.research-library.schomburg" |
"dark.section.research-library.schwarzman" |
"dark.section.whats-on.primary" |
OPTIONAL
```

Description

Used to set the color for the left border and icon. Refer to how color values are defined and typed in the DS `Icon` component.

If `highlightColor` is pass and `backgroundColor` is NOT passed, the `highlightColor` prop should be ignored and the following console warning should be generated:

NYPL Reservoir Banner: The `highlightColor` prop has been passed, but the `backgroundColor` prop has not been passed. Because of this, the `highlightColor` prop will be ignored.

Default *n/a*

icon: Icon
OPTIONAL

Description Custom DS `Icon` component that will override the default variant icon.

Default *n/a*

id: string
OPTIONAL

Description ID string that other components can cross reference for accessibility purposes.

Default *n/a*

isDismissible: boolean
OPTIONAL

Description Used to control whether the component can be dismissed (closed) by a user.

Default *false*

type: "informative" | "negative" | "neutral" | "positive" | "recommendation" | "warning"
OPTIONAL

| | |
|--------------------|--|
| Description | Used to control the component's semantic coloring and iconography. |
| Default | <code>"neutral"</code> |