


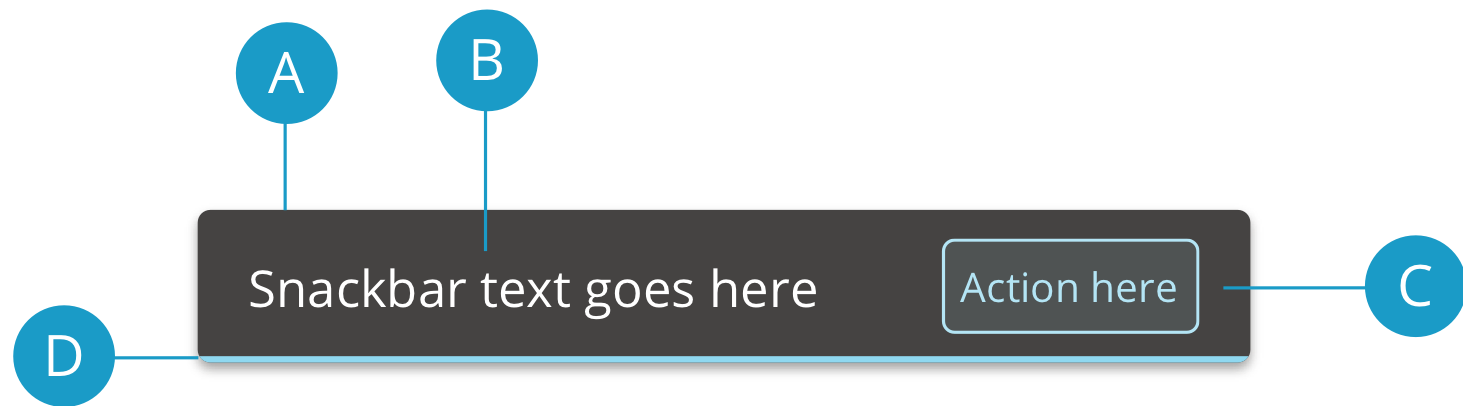
[Draft] Snackbar



Snackbars provide a temporary alert that is unobtrusive to the user.

Created	August 20, 2020
Last Updated	August 20, 2020
Contributors	Christopher Frydryck , Steve Kurtz , Lisa Martin , Sean Connolly .
Status	Experimental 
Languages	React.js
Current Version	Version 1.0.0

Anatomy



Progress Bar Not In Use

Whoops! Our progress bar needs a little tweaking before we release to the team. All of our other guidelines are up to date, but our progress bar feature should be coming in the future! Stay tuned via our Slack channel.

A. Container

The width and height of the snackbar alert that contains all content within it.

B. Label

Text that provides context for the snackbar's purpose and alerts the user.

C. Call to Action (*Optional*)

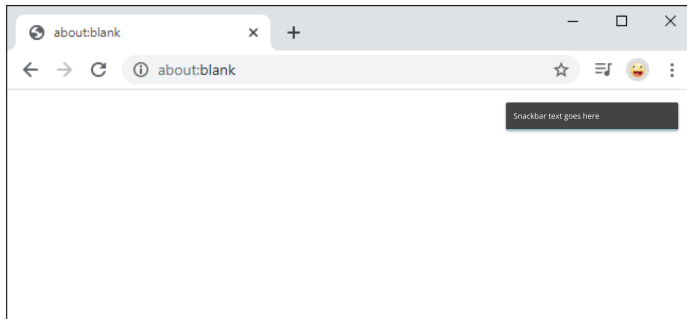
D. Progress Bar

A sliding bar that showcases how much time is left before the alert will dismiss itself.

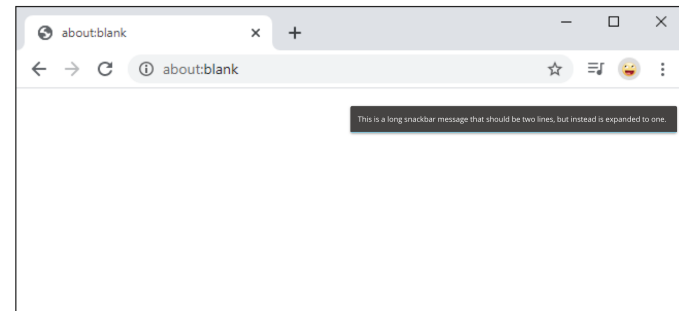
Container

Width

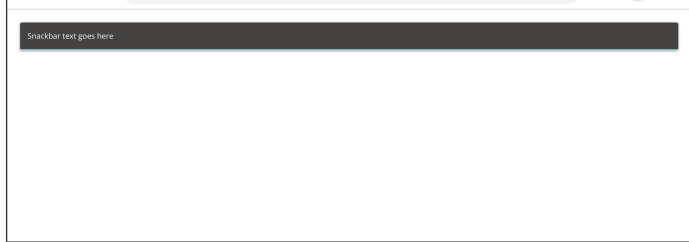
The width of a Snackbar is flexible and should be adjusted at designer's discretion. The width of the container can allow content to skip to a second line.



Do: Adjust the width of a Snackbar to fit the content and action if present.



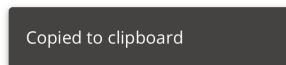
Don't: Stretch a Snackbar for the sake of fitting content to one line.



Don't: Stretch a Snackbar to full width on desktop or tablet.

Color

The container color can help guide users to understand further the urgency or type of message that appears before they even read the label. Green is a confirmation or successful color, yellow is a warning, and red is destructive or an error.



Do: Use the default Snackbar color for system triggered alerts or for alerts that are informative only.

Don't: Use an error red color if the message is not destructive.

Removed four items from the list

Copied to clipboard

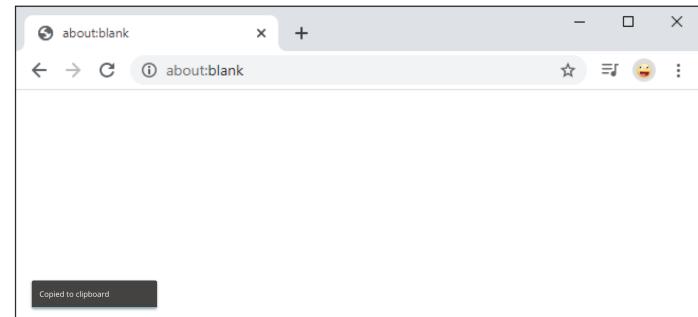
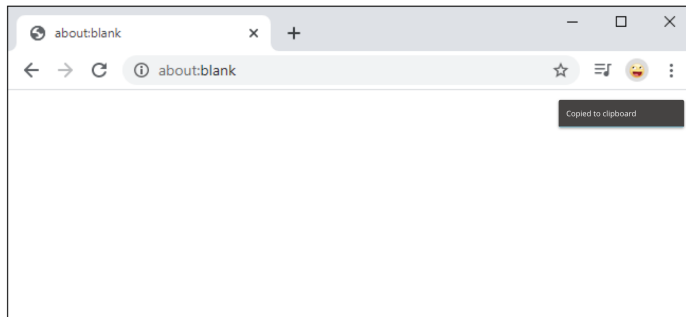
Don't: Use colors that don't correctly portray the message's purpose

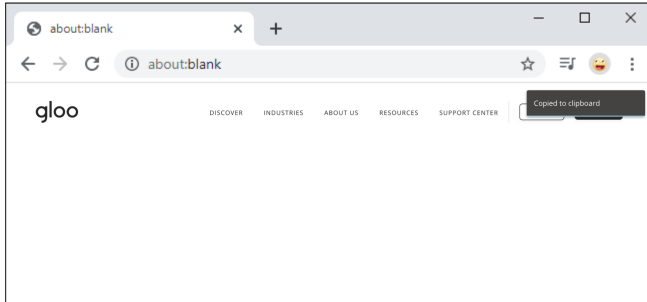
Don't: Add a transparency to the container.

Positioning

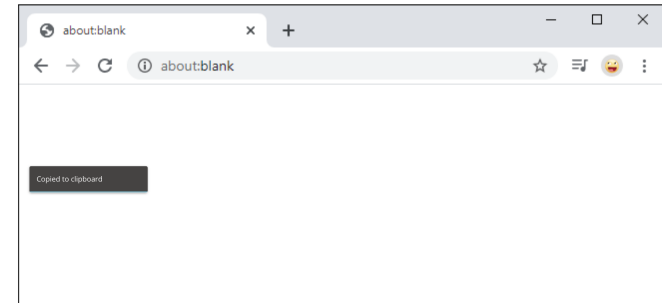
Snackbars should overlay all other content and position themselves by default in the top right of the viewport for desktop and tablet. For mobile, Snackbars can be full width at the bottom of the viewport.

In the event that pinning to the top right of the screen will cover content or will blend in with the design, Snackbars can be pinned to any corner, or center aligned at the top of bottom of the viewport. This should be done sparingly and at designer's discretion.





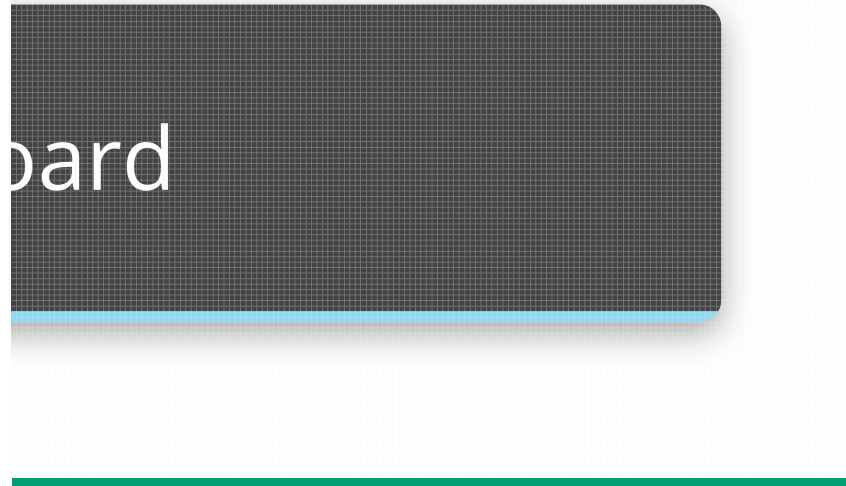
Don't: Cover actionable content or navigational components with your Snackbar.



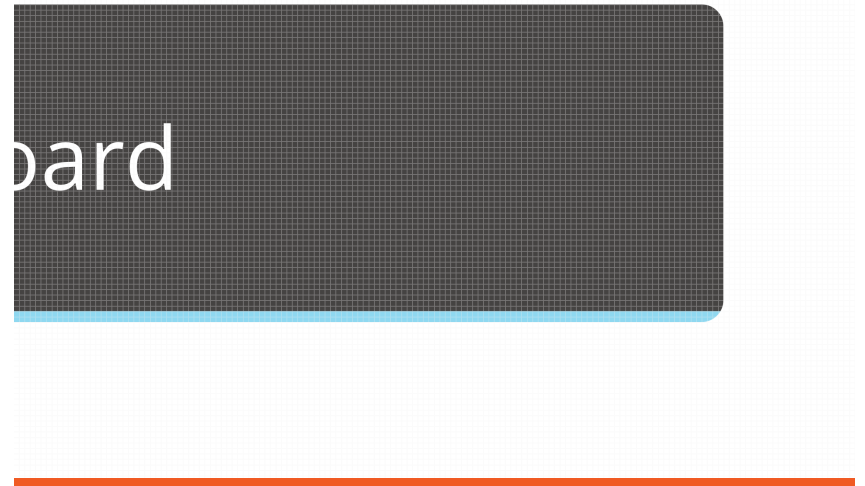
Don't: Pin the Snackbar to a custom location in the viewport.

Elevation

Elevation helps a Snackbar appear above all other content and not a permanent or fixed part of the experience in a product.



Do: Provide elevation to a Snackbar.



Don't: Remove elevation from a Snackbar.

Label

Language

Labels in Snackbars should be written in basic sentence structure and avoid using slang, unfamiliar abbreviations or error codes that don't bring value to the user.

Link copied to clipboard

Error saving settings. Error code:
XC9928373_1643

Do: Use proper sentence structure that is concise and meaningful to a user.

Don't: Provide content that is not valuable to a user or contains content that is obscure.

Height

Snackbars should be short, simple and clear messages that are scannable and don't take away from an experience. To help guide designers to making messages clear, we want to ensure that most cases of snackbars don't have more than two lines of content

Snackbar labels can be two lines if needed to fit content

Snackbar labels should avoid going to three or more lines, but can be unavoidable with foreign languages or compliance requirements

Do: Break long labels to a second line.

Caution: Extending a Snackbar past two lines may warrant a higher priority alert.

Snackbar labels should avoid going to three or more lines, but can be unavoidable with foreign languages or compliance requirements

Don't: Extend a snackbar for the sake of fitting all the content on one line.

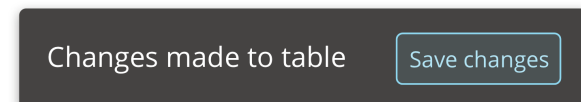
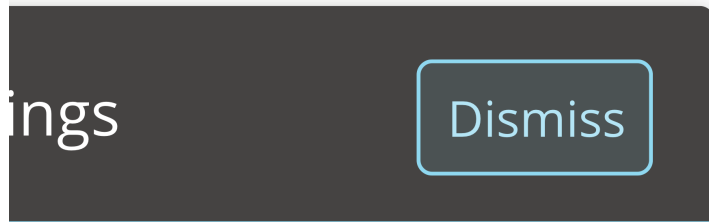
Icons

Because icons are temporary, we want to avoid the usage of any icons that may confuse the user or hinder the swiftness of understanding the label.

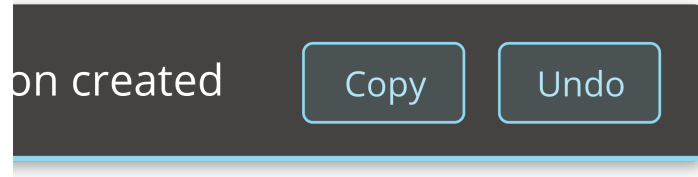
Call To Action

Usage

Call to action buttons in a Snackbar should bring action to the label provided.



press.



Don't: Use more than one call to action in a Snackbar.

States

Snackbars can feature different styling depending on the message being shown to the user. All Snackbars are a low priority message, but can also showcase messages that still are useful or even actionable.

Default

The default message type of a Snackbar.



Lost connection to internet

Retry

Success

The state of a Snackbar when a message needs to portray that an action was completed as expected by the user.



Added 4 new members to the group

Warning

The state of a Snackbar when a message needs to portray that an action was not completed as expected by the user, but was not destructive.



Unable to save changes to form

Error

The state of a Snackbar when a message needs to portray that a destructive action was performed regardless of if the user intent was to do so.



Deleted 4 entries from the table

Undo

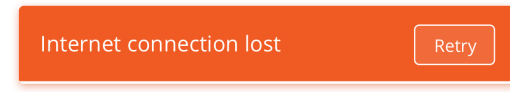
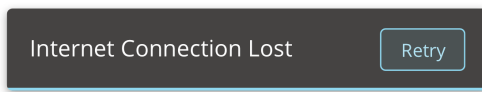
Behavior

Trigger

Snackbars require some sort of trigger in order to appear for a user. Otherwise, they would be useless messages that have no contextual value to the user.

In Dan Saffer's book *Microinteractions*, Dan talks about two types of triggers, manual triggers and system triggers. Manual triggers are actions that occur due to a user's interaction with the software. For instance, a user guides a volume slider along its linear path and in turn the volume increases. The user is manually triggering an action that performs.

Due to the nature of system triggers being out of the control of the user, we want to avoid destructive actions being taken place by the system. Therefore, it should be known that an error Snackbar is bad practice with system triggers, and should only occur if a user performs a destructive action.

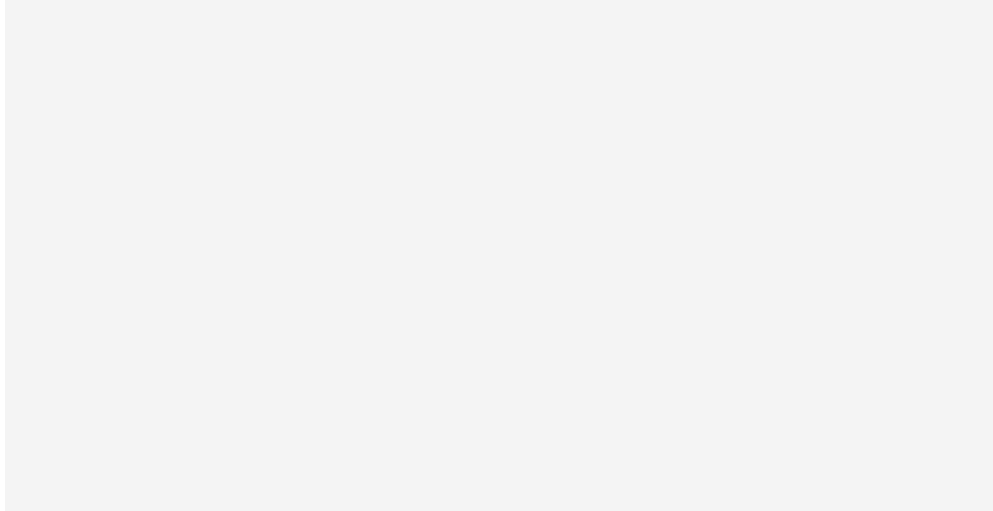


Do: Use any state of Snackbars to portray an action for a manual trigger.

Don't: Use an error state for Snackbars triggered by the system. Error state should always showcase a destructive action.

Animation

Snackbars should slide in from off the screen into the view of the user, and should also feature a reasonable timeout duration that is displayed as a slider on the bottom of the Snackbar.

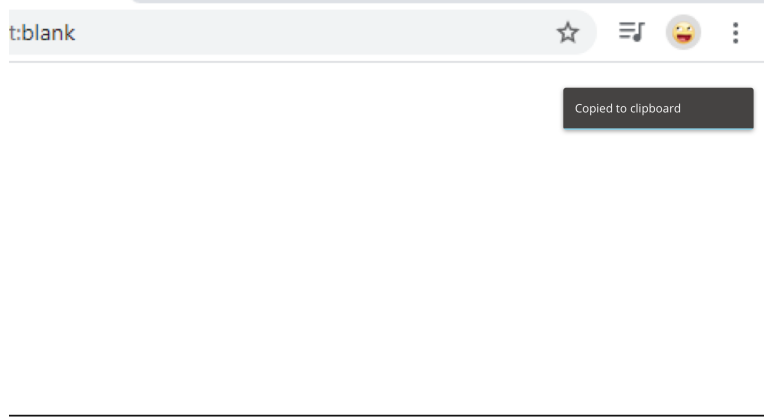


Auto Hide Duration

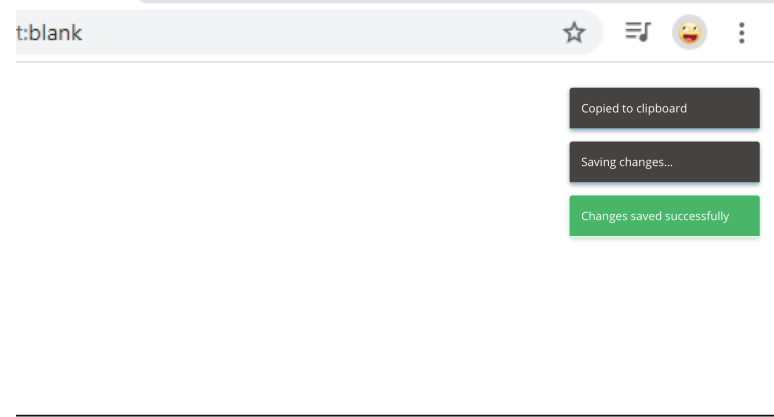
Snackbars are a low priority message that should auto dismiss. We recommend a duration that is a minimum of three seconds, and is dependent on the length of the message. For any message that is longer than 30 characters (including spaces and punctuation), add 100ms per additional character to the length of the time before a Snackbar dismisses itself.

Quantity & Dismiss

Snackbars should only show one at a time, and should dismiss the current Snackbar if another action that warrants a Snackbar is waiting to fire.



Do: Show one Snackbar at a time and dismiss the current snackbar for the next one in the queue when an action is performed.



Don't: Layer or stack snackbars on eachother to showcase multiple messages at once.

Hover

Upon hovering a Snackbar, the timeout function pauses and will prevent dismissing the Snackbar to allow a user to interact without the Snackbar disappearing automatically.


API

Snackbars give a small message that guide users or inform them at a low priority level.

Try using Snackbars when a user takes an action that needs additional information to clarify.

```
const handleClick = () => {  
  showSnackbar({ message: "My message", color: "success" })  
}
```

Props

 gloo-react-components.herokuapp.com
Storybook

 Updated 27 days ago

Storybook Demo

To see a Snackbar in action, view it [here in Storybook](#).

Two Lines - No Button

Snackbar labels can also be two lines and would caution going past two

Snackbar labels can also be two lines and would caution going past two

Snackbar labels can also be two lines and would caution going past two

Two Lines - Button

 **readme**

Setup

JavaScript

```
// Recommended
import {useSnackbar} from 'Gloo-DS'

// Alternative
const {useSnackbar} = require('Gloo-DS');
```

Usage