

# Loading & Waiting

---

*[Design System Image]*

Oftentimes, your application will need to load data from (or send data to) a remote location with some amount of latency. When this happens, you will need to provide some indication to the user that something is happening. Different situations may call for different styles of indicator, explained below.

*[MaterialDesignDescription Component - Interactive React component]*

*[TOC Component - Interactive React component]*

---

## Variations

---

There are a number of different ways to indicate to a user that something is happening / loading behind the scenes. Different variations are appropriate in different situations.

## Placeholder Shapes / Skeleton

*[Design System Image]*

When loading data for the first time, you do not want to show an empty screen while you are fetching the data. Instead, you should show placeholder shapes ("skeletons") that mimic the data that will eventually be loaded. This will give the impression that your application is faster and gives users an indication of what kind of data is being loaded.

*[DemoCard Component - Interactive React component]*

The skeleton should match as closely as possible the real `_structure_` of the data, using shapes like rectangles and circles. The data `_length_`, however, may not be available at the time of loading, and it is likely that your skeleton ended up looking slightly different from your real data. This is OK; your mission is only to set up users' expectations.

*[Design System Image]*

## Spinner Overlays

Spinner overlays make use of a semi-transparent overlay with a large, centered spinner. This variation of indicators is useful when you want to block user interaction with the screen (or a section of the screen) while processing occurs.

*[Design System Image]*

*[DemoCard Component - Interactive React component]*

## Contextual Spinners

Contextual spinners can be used as an alternative to spinner overlays in certain situations. The most common application of contextual spinners is during [Form](#) submission / verification. If you need to wait for a server response to validate inputs in a form (after clicking the submit button), you should disable the submit button and replace its label with a spinner. If the verification succeeds, the application should proceed to the next screen. If verification fails, the appropriate error messages should be displayed and the submit button label restored.

*[Design System Image]*

*[DemoCard Component - Interactive React component]*

## Progress Bars

*[Design System Image]*

Progress bars should be used sparingly in your application. They should only be used in situations where progress is discrete / measurable. For example, it could be used to indicate progress when uploading a file to a server.

*[DemoCard Component - Interactive React component]*

*[DemoCard Component - Interactive React component]*

---

## Behaviors

---

### Real-time Data (Polling / Pub-Sub)

If you are showing real-time data (either via polling or a push mechanism from the server), there is no need to show loading indicators. If you wish, you can display a small indicator when data is being refreshed or mention somewhere on the screen when the data was last refreshed so users can identify if the data is stale. You should avoid showing a loading indicator if your data is refreshed more than once per minute.

*[Design System Image]*

## Manual Refresh

---

If your users can manually refresh data on the screen, you may wish to show the placeholder elements (as with initial load), or simply refresh the data behind the scenes without indication (as with real-time). Ultimately, this should depend on how long it takes to refresh the data - if it takes more than one second to load the data, you should use an indicator.

---

## Design Specifications

---

Context plays an important role in sizing your indicators. If you are using them in a small container, you may need to scale down from the recommended size.

*[Design System Image]*

---

## Developers

---

### Placeholder Shapes / Skeleton

There are a number of libraries available to help you create placeholder shapes, such as:

**Angular:**

- [@netbasal/ngx-content-loader](#)
- [placeholder-loading](#)

**React:**

- [placeholder-loading](#)
- [react-content-loader](#)
- [@mui/material/Skeleton](#)

**React Native:**

- [rn-placeholder](#)

## Progress Indicators

Additionally, you should use the following progress / spinner components:

**Angular** - Angular Material:

- [Progress Spinner](#)
- [Progress Bar](#)

**React** - MUI:

- [Circular Progress](#)
- [Linear Progress](#)

**React Native** - React Native Paper:

- [Activity Indicator](#)
- [Progress Bar](#)