

HOW TO USE THIS TEMPLATE

Fill in the parts that are relevant, delete those that aren't. Not all sections are going to apply to every component.

See the [sidebar design doc](#) for a filled-in example.

Angular Material

<Component Name>

Author: <Name> <Email>
Date: <Date, formatted March 15, 2016>

Goals

Talk about what you're building why you're building it

References

Link to relevant materials, such as UX spec, related articles, etc.

Background

Talk about the current implementation status or any other context that affects the design / implementation.

Requirements

- Talk about behaviors that the component must have
- Talk about any performance constraints
- Talk about how the component should interact must behave itself WRT other components / the rest of the app.
- Talk about how the component should behave on different screen sizes

Case studies

Talk about one or more applications that use this (or a very similar) component. Case studies would usually include something novel or noteworthy that will inform the design. For example, you may reference App X which has a view that would use 10,000 instances of this component and thus have extremely high performance needs. Don't go overboard with different cases.

Existing implementations

Talk about versions of this component that exist in other libraries, looking at capabilities, public API, implementation approach, accessibility, and performance. Screenshots and/or API examples are very helpful.

For each thing looked at, analyze:

What does this do *well*?

What could this do *better*?

Some libraries to look at are: [Angular Material 1](#), [Polymer](#), [Material Design Lite](#), [Kendo UI](#), [Bootstrap](#), [Angular UI Bootstrap](#), [Materialize](#)

Technical considerations

Think of this section as "implementation requirements". You should cover any platform / browser constraints and quirks that will impact the implementation. For example, on the web platform, mentioning that the implementation should take advantage of CSS transform's GPU accelerated rendering as much as possible.

Proposed implementation

This is the main content of your design doc. Talk about that you're actually to implement. Description should not be so detailed as to include large swaths of code, but should give the reader a good understanding of what you're going to code. Include examples.

Proof of concept

If you created a proof-of-concept demo, link it here. This is a good place to use something like JsBin, CodePen, Plunker, etc.

API

Talk primarily about the public API of the components (both in templates and in component code). The following is an example API format that you can re-use.

NOTE: We recommend using a Google Docs addon to do syntax highlight in this section

<md-example>

Constraints

It is invalid to create an `<md-example>` without a value set.

It is invalid to create an `<md-example>` with more than one `md-example-child` within.

Component

```
class ExampleViewer {  
  
  /** Optional, default = "start". The side on which the sidenav will appear */  
  @Input() align: 'start' | 'end';  
  
  /** Whether the example is open. */  
  @Input() opened: boolean;  
  
  /** Emitted when the example is opened. */  
  @Output() opened: EventEmitter<void>();  
  
  /** Emitted when the example is closed. */  
  @Output() closed: EventEmitter<void>();  
  
  /** Open the example */  
  open(): void;  
}
```

Accessibility

You should talk about all accessibility concerns here. See [this article](#) for a good overview of accessibility concerns.

If the component is one listed in [the WAI-ARIA Authoring Practices doc](#), be sure to address the practices and guidance given there.

How to use the component with a keyboard only

This includes having keyboard shortcuts for all of the actions and managing focus.

How a screen-reader user will interact with the component

This includes providing semantic meaning, text alternatives, labelling, and announcements (using ARIA). The overall user-experience with a screen-reader should be well understood.

How the component will work without color

Are there any colors being used that would not be discernable by colorblind users?

Note: there are some Chrome extensions that attempt to simulate what a color-blind user would see: [SEE](#), [Daltonize](#)

How the component will enforce any a11y requirements

Some components may require the user to provide messages, labels, or additional keyboard shortcuts. These requirements should be strongly enforced in dev mode.

Internationalization

You should talk about any internationalization concerns here. This includes, but is not limited to:

- How any messages embedded in the component (such as aria labels) will be translated.
- Formatting for numbers, dates, times, or currency.
- RTL vs. LTR layout (see the [Material Design bidirectionality guide](#))

Mobile

Talk about any concerns for different device sizes or for touch devices. This can include:

- Are there any **hover** or **mouseover** type interactions? How will this work on mobile?
- Will the component need to respond to different viewport sizes?
- Does the component employ any gestures?

Performance considerations

Talk about any performance related topics (may also include memory use topics). This may include animation performance (60 fps) concerns, page weight, scrolling performance, load-time, etc.

Other approaches considered

Briefly talk about other approaches that were considered and why they were not chosen.

Migration from 1.x

Describe a strategy for migrating to this component for an app currently using the Angular 1 version.

Roadmap and timeline

Talk about how the work can be split up into different milestones and the timing for completion.