Power Apps: Create a reusable component for canvas apps

Components are reusable building blocks for canvas apps so that app makers can create custom controls to use inside an app, or across apps using a component library. Components can use advanced features such as custom properties and enable complex capabilities.

Let's break it down further to reusable components. We need to make the effort worthwhile by having components that we can reuse. We don't want to be creating more screens in the future by copying basic components like labels and buttons, we want reusable components that can be used across the app. That way, we minimize the amount of effort required when we want to change something. So, looks like we need a header component for both the Home screen and the Drilldown screen. Let's call them HomeHeader and DrilldownHeader.

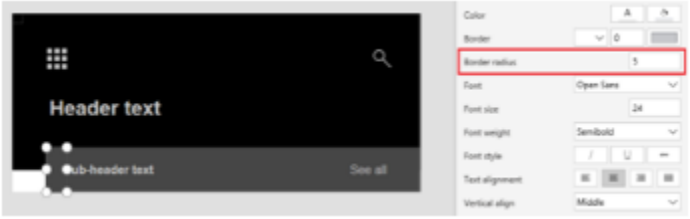
Components to create

- HomeHeader
- DrilldownHeader

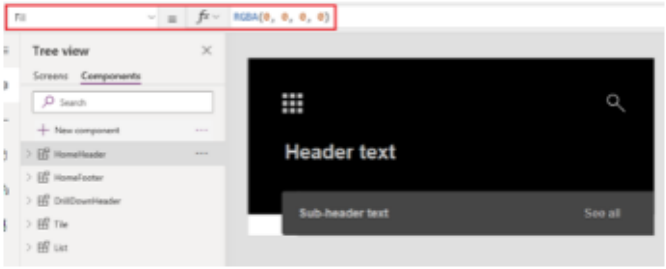
HomeHeader Components

When creating the HomeHeader component, use a disabled button to get the rounded corner and make the component transparent.

To make the rounded corner, set the button border radius to 5



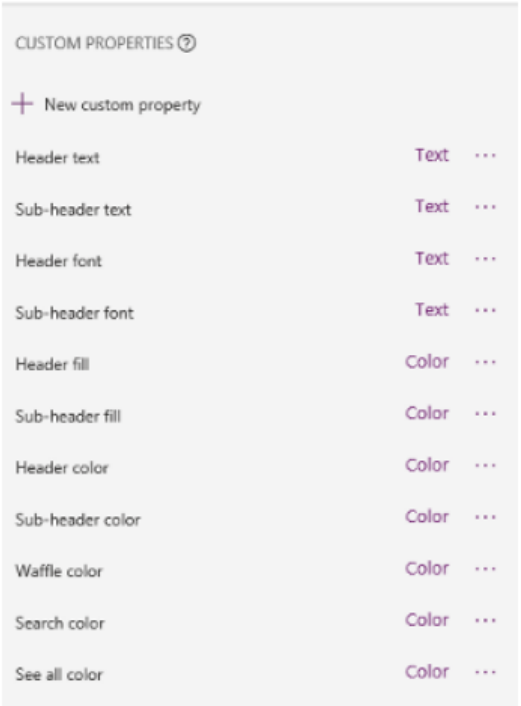
To make the component transparent, set the fill property to RGBA(0, 0, 0, 0).



Next we create some custom properties so that we can customize the component when we use it

- Header text
- Sub-header text
- Header font
- Sub-header font
- Header fill
- Sub-header fill
- Header color
- Sub-header color
- Waffle color
- Search color
- See all color

All of these are of input property type. Please refer to the image below for the data type.

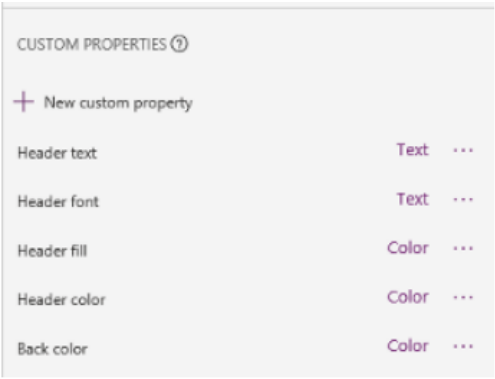


DrilldownHeader Components

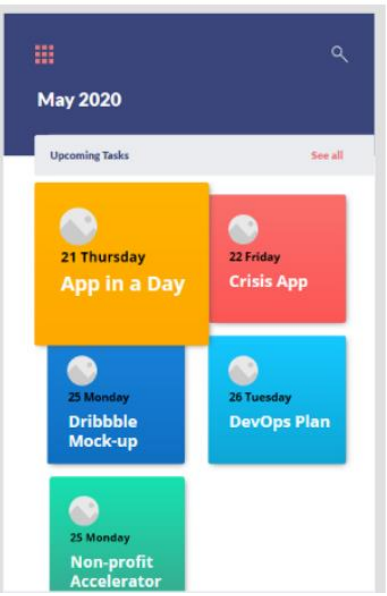
For our DrillDownHeader Component, we will also create some custom properties so that we can customize the component when we use it in our app like the other previous components.

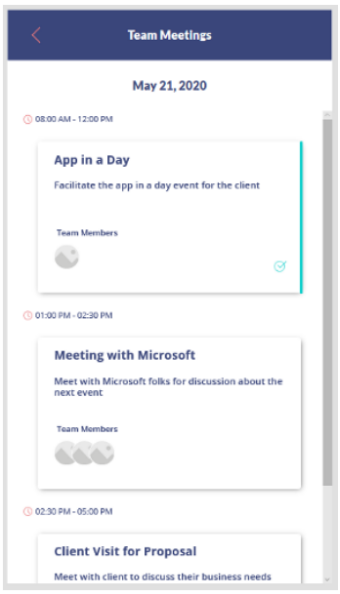
- Header text
- Header font
- Header fill
- Header color
- Back color

All of these are of input property type. Please refer to the image below for the data type.



After arranging our components on their respective screens and passing values to their custom properties, here's our end result.





Current limitations

- **Known issue**: locales that use non-English (United States) number or comma separators in formulas may receive an error when inserting a component. This issue has been resolved and a fix will roll out worldwide within two weeks (depends on your region).To mitigate this before then, you can revise the properties like Fill (of type Color) for your locale. For example, someone in Germany might revise the commas in a Fill property like this: RGBA(0, 0, 0, 0) -> RGBA(0; 0; 0; 0)
- Images: it is not yet possible to package media files when importing a component, but this feature is coming soon.
- Collections: using a collection within a component is not yet fully supported, but you can use variables today.
- Data sources: data sources are not saved with components. Forms and data-grids are disabled due to this reason.
- Nesting: components cannot be inserted into galleries, forms, and data cards today. Likewise, forms and data-grids cannot be embedded into components.