Developing React Web Parts

Agenda

- Designing and Developing React Web Parts
- Web Part Properties versus React Component State
- Leveraging the Office UI Fabric React Library
- Developing Web Parts using the SharePoint REST API
- Designing Web Parts to Manage SharePoint Lists
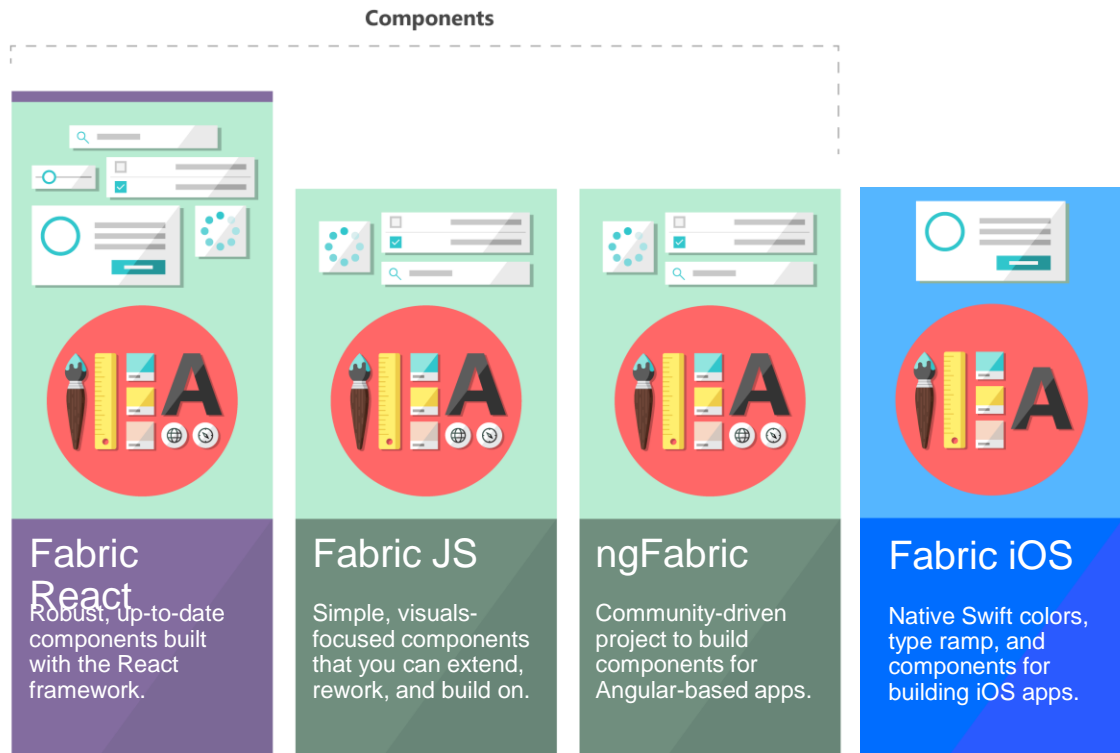
What is the Office UI Fabric?

- Office UI Fabric is responsive, mobile-first, front-end style framework
  - Built by Microsoft to style Office 365, OneDrive and SharePoint sites
  - All about styling instead of JavaScript
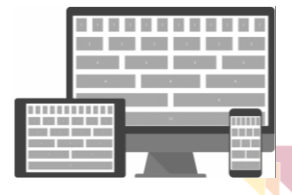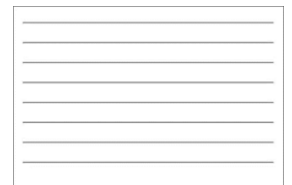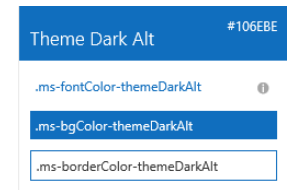 - Can be used by 3rd party developers

Components

Fabric Core

Core elements of the design language including icons, colors, type, and the grid

or

Fabric React

Robust, up-to-date components built with the React framework.

Fabric JS

Simple, visuals-focused components that you can extend, rework, and build on.

ngFabric

Community-driven project to build components for Angular-based apps.

Fabric iOS

Native Swift colors, type ramp, and components for building iOS apps.

Fabric Core styling

- Fonts and typography
  - Segoe font family + type ramp
  - Official Office 365 iconography

- Color
  - Official Office 365 color palette

- Branded assets
  - Product symbols + product filetype symbols

- Animations
  - Official Office 365 selection of easings and animations

- Responsive grid
  - Tailored to Office 365 silhouettes

Styles

- The Office UI Fabric provides styles for..
    - Typography
    - Color
    - Icons
    - Animations
    - Responsive Grid
    - Localization

Typography

- Base font classes
  - Fabric includes 10 base font classes
  - Each base class sets a default size, weight, and color.

Class	Size	Weight	Color
.ms-font-su	42px	Segoe UI Light	ms-color-neutralPrimary
.ms-font-xxl	28px	Segoe UI Light	ms-color-neutralPrimary
.ms-font-xl	21px	Segoe UI Light	ms-color-neutralPrimary
.ms-font-l	17px	Segoe UI Semilight	ms-color-neutralPrimary

Typography

- Helper font classes
  - There are helper font classes to change the text weight.

Class	Weight
.ms-fontWeight-light	Light
.ms-fontWeight-semilight	Semilight
.ms-fontWeight-regular	Regular
.ms-fontWeight-semibold	Semi Bold

Color

- Includes 9 theme colors and 11 neutral colors.
  - Helper classes for text, border, background, and hover states.
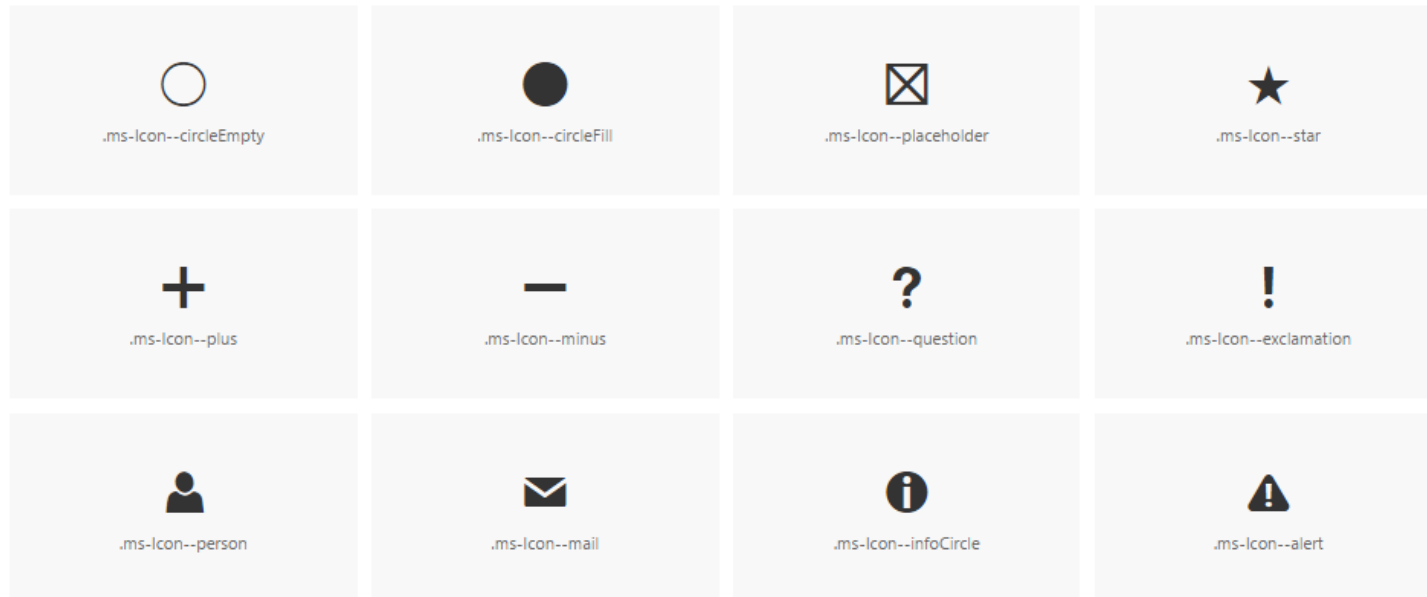 - Color classes act as hooks into the Office 365 theming system

Icons

- Fabric uses a custom font for its iconography.
  - Font contains glyphs you can scale, color, and style

○	●	⊠	★
.ms-Icon--circleEmpty	.ms-Icon--circleFill	.ms-Icon--placeholder	.ms-Icon--star
+	−	?	!
.ms-Icon--plus	.ms-Icon--minus	.ms-Icon--question	.ms-Icon--exclamation
👤	✉	ⓘ	⚠
.ms-Icon--person	.ms-Icon--mail	.ms-Icon--infoCircle	.ms-Icon--alert

Icons

- To use the icons, combine the base **ms-Icon** class with a modifier class for the specific icon.

```
<i class="ms-Icon ms-Icon--mail" aria-hidden="true"></i>
```

.ms-Icon--xCircle

.ms-Icon--mailOpen

.ms-Icon--people

.ms-Icon--bell

.ms-Icon--calendar

.ms-Icon--scheduling

.ms-Icon--event

.ms-Icon--folder

Dialog animations

- When choosing a motion for dialogs, consider the origin and tone of the content.

- For a warning or error dialog, a quick fade in might be appropriate.

- If the dialog appears when a user chooses an item to get more information, a scale-up might be appropriate.

CSS Class	Description	Cubic Bezier	Timing
ms-u-scaleUpIn100	Scale up to 100% and fade in	0.1, 0.9, 0.2, 1	0.167s

Responsive Grid

- ## Fabric comes with a mobile-first, responsive grid
    - ### Based on 12 column grid
    - ### Used to create flexible layouts

```html
<div class="ms-Grid">
    <div class="ms-Grid-row">
        <div class="ms-Grid-col ms-u-sm6 ms-u-md4 ms-u-lg2">First</div>
        <div class="ms-Grid-col ms-u-sm6 ms-u-md8 ms-u-lg10">Second</div>
    </div>
</div>
```

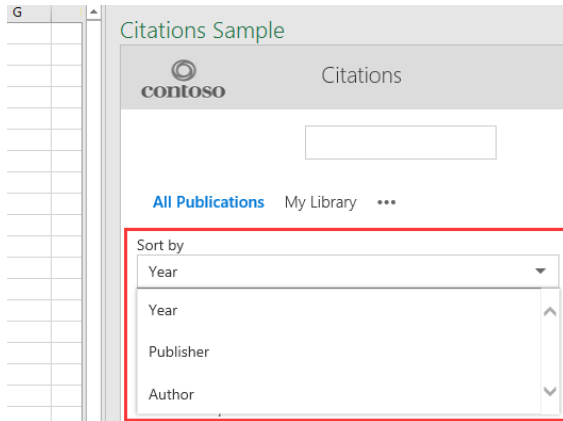# Office UI Fabric Components

- Fabric's components are building blocks for UI
  - All JavaScript is presentational to explain behavior.

SearchBox


```html
<div class="ms-SearchBox">
    <label class="ms-SearchBox-label">
        <i class="ms-SearchBox-icon ms-Icon ms-Icon--search"></i>
        Search publications
    </label>
    <input class="ms-SearchBox-field">
    <button class="ms-SearchBox-closeButton"><i class="ms-Icon ms-Icon--x"></i></button>
</div>
```

Dropdown


```html
<div class="ms-Dropdown" tabindex="0" style="width:80%">
    <i class="ms-Dropdown-caretDown ms-Icon ms-Icon--caretDown"></i>
    <select class="ms-Dropdown-select">
        <option>Year</option>
        <option>Publisher</option>
        <option>Author</option>
    </select>
</div>
```
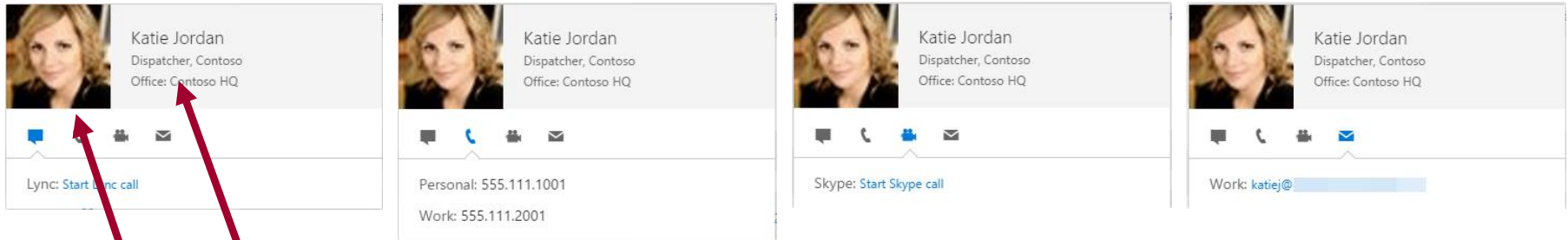
Persona Card

Bind Data With Server Side or Client Side Code

Property Management Server Side example - https://github.com/OfficeDev/Property-Inspection-Code-Sample

```
<div class="details-block">
    @foreach (var item in Model.members)
    {
        <div class="content-row">
            <div class="ms-Persona ms-Persona ms-Persona--square ms-Persona--xs">
                <div class="ms-Persona-imageArea">
                    <img class="ms-Persona-image" src="@Url.Action("UserPhoto", "Photo", new { userId = item.id })">
                </div>
            <div class="ms-Persona-details">
            <div class="ms-Persona-primaryText">
                @item.displayName
            </div>
```