

Build Custom UIs Using API Widgets

ArcGIS Maps SDK for JavaScript

Matt Driscoll

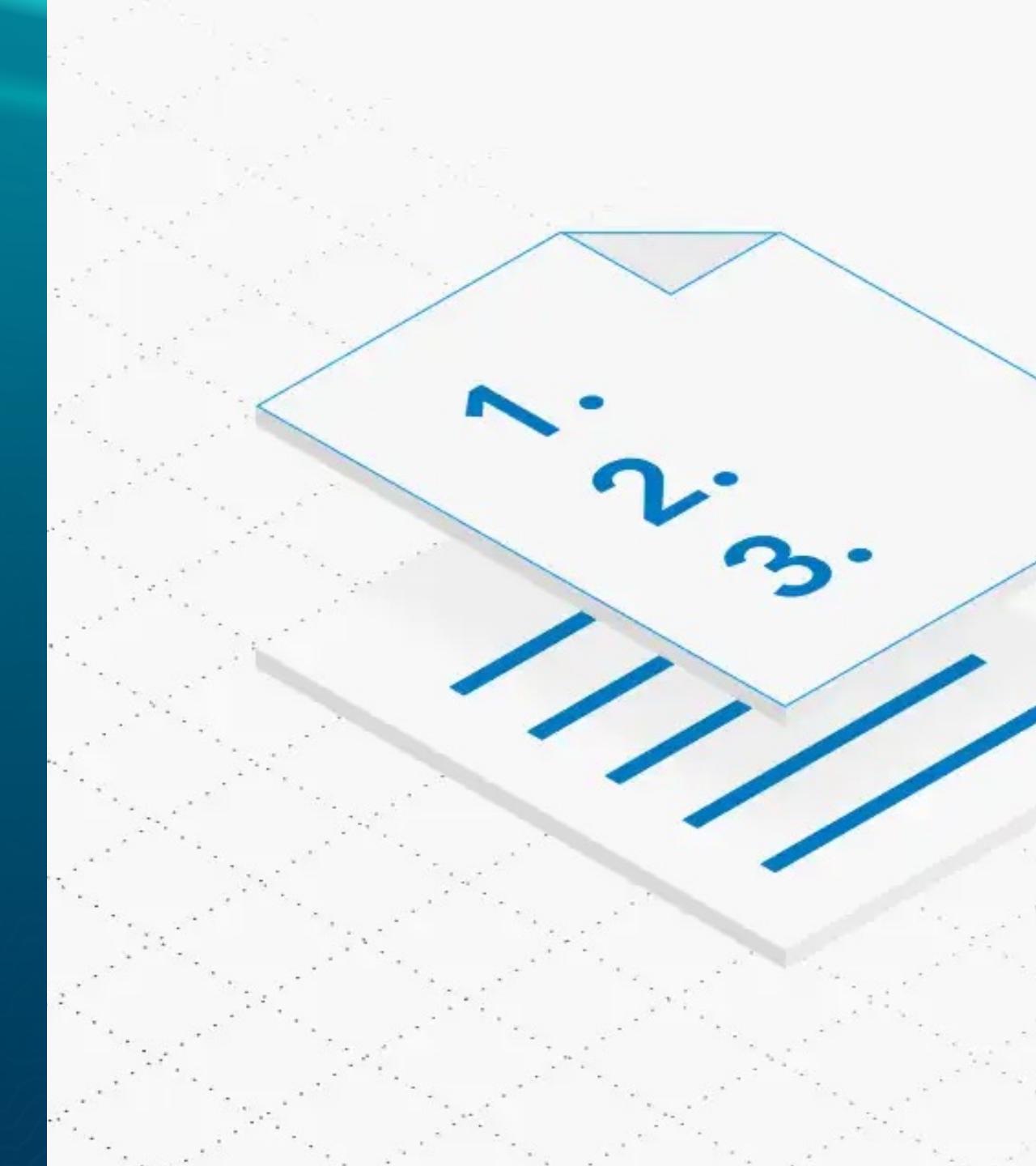
Jonathan Uihlein

```
<arcgis-map zoom="4" center="-118,34">
  <arcgis-search position="top-right" />
</arcgis-map>
```

```
<arcgis-map zoom="4" center="-118,34" />
view.goTo({
  center: [-126, 49]
})
.catch(function(error) {
  if (error.name != "AbortError") {
    console.error(error);
  }
});
```

Agenda

- Widgets
 - What are they?
 - Why use them?
 - Architecture
 - Views and View Models
 - Best practices
- Building Custom UIs
 - Calcite Design System
 - Calcite Components
 - Other frameworks, design systems, etc.
 - Demos
- Q&A



Widgets

Jonathan

```
view.goTo({  
    center: [-126, 49]  
})  
.catch(function(error) {  
    if (error.name != "AbortError") {  
        console.error(error);  
    }  
});
```

```
    queryParameters =  
QueryParameters().apply {  
    whereClause = "price > 200"  
}  
viewModelScope.launch {
```

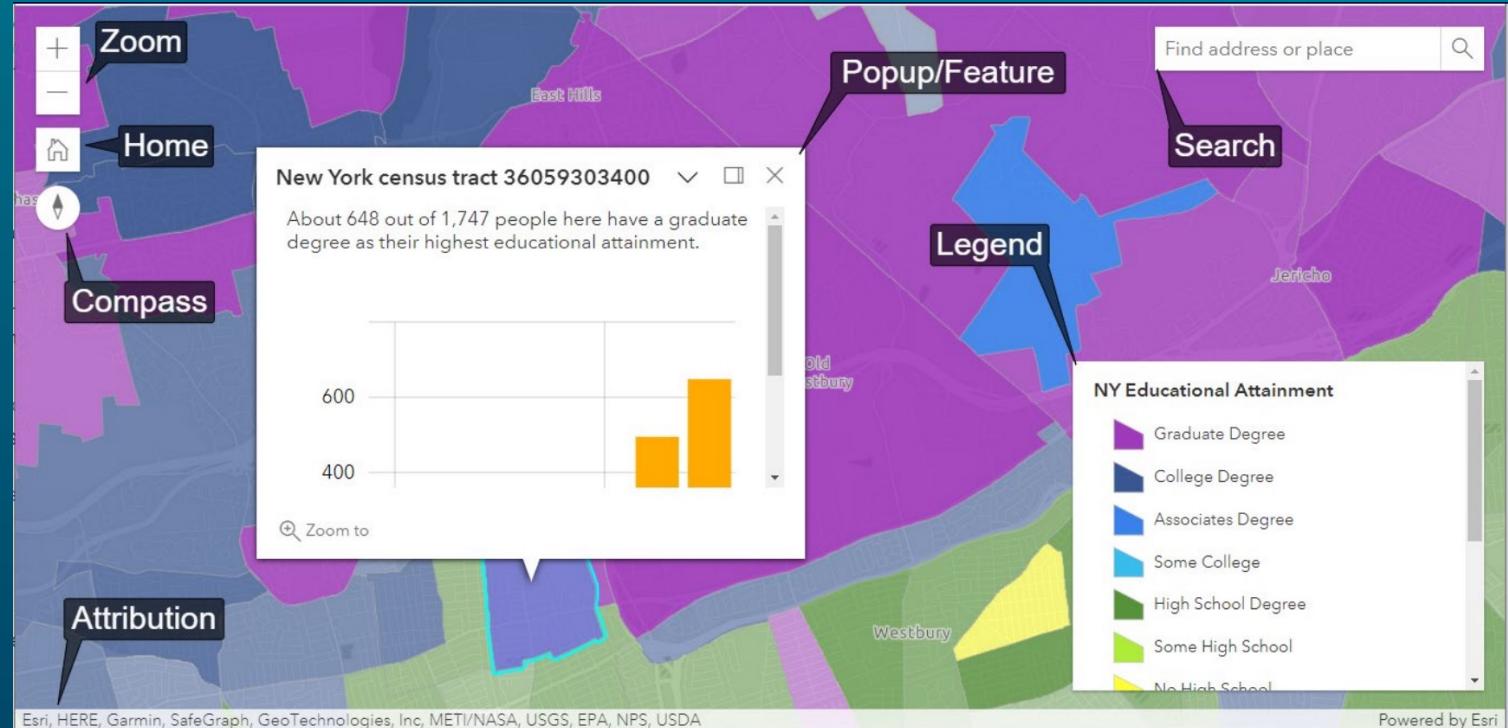
Widgets

What are they?

- Individual module with a UI
- Interactive or informative
- Perform specialized function(s)
- Stateful
- Written in TypeScript

Why use them?

- Ease of use
- Configurable
- Reusable
- Help build more complex tools and apps



Widgets

Architecture

- Widgets are comprised of:
 - views
 - viewModels
- Business logic is separate from presentation
 - view (user interface) is optional
 - viewModel logic can be used to replicate the view experience
 - Integration with other frameworks
 - Completely new UI



Widgets

Architecture

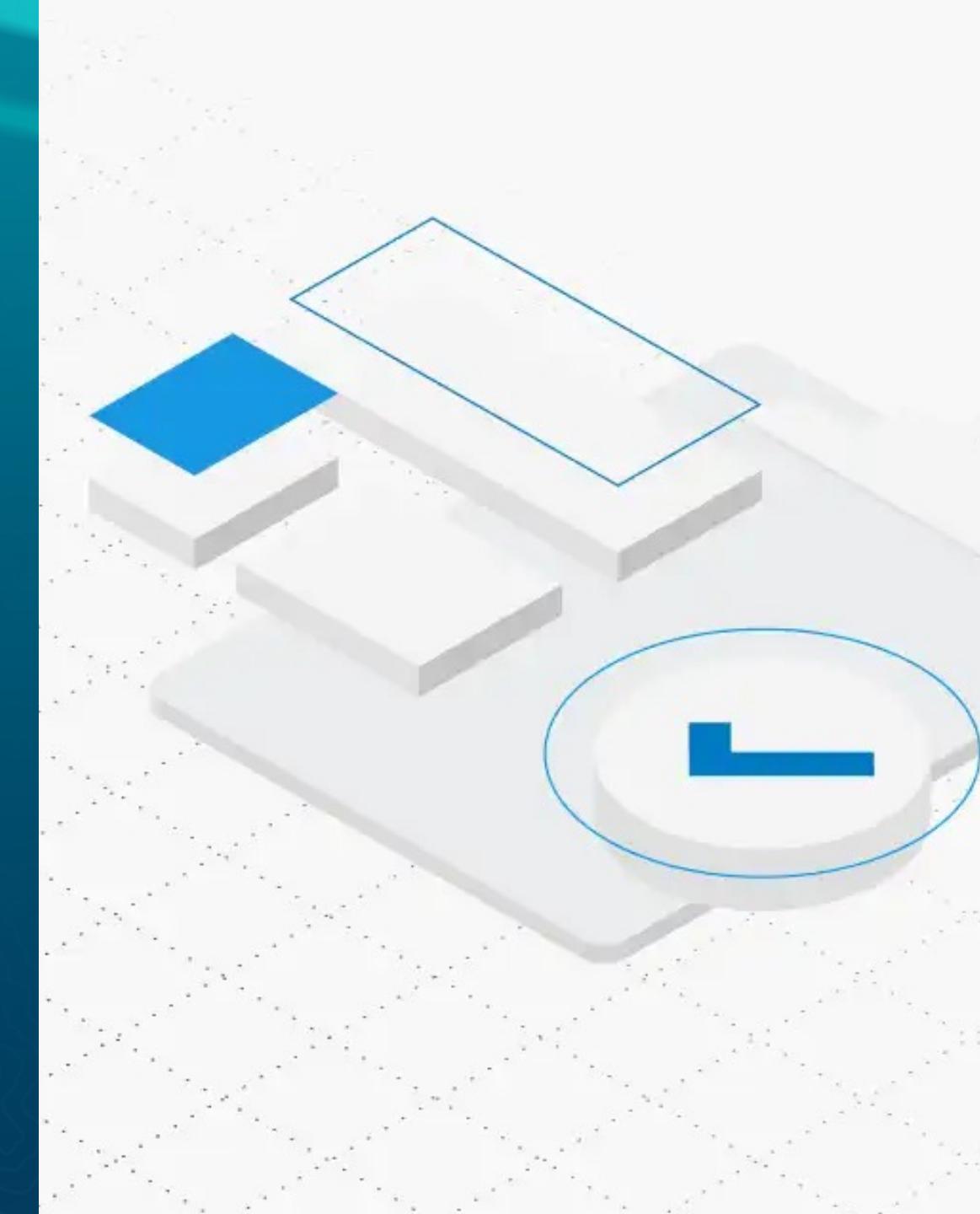
- views
 - Presentation (UI and DOM)
 - Extends [*esri/widgets/Widget*](#)
 - Example: [*esri/widgets/Search*](#)
- viewModels
 - Business logic
 - Provides APIs to support view
 - Extends [*esri/core/Accessor*](#)
 - No DOM
 - Example: [*esri/widgets/search/SearchViewModel*](#)



Widgets

views and viewModels

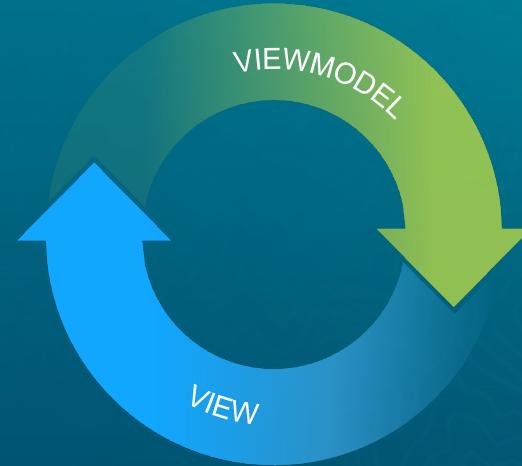
- view updates are based on:
 - User interaction
 - viewModel property changes
 - viewModel events
 - Observing state changes: [esri/core/reactiveUtils](#)
- Use viewModels to provide logic for custom views
 - Watch and modify property states
 - Use methods directly
 - Listen for events



Widgets

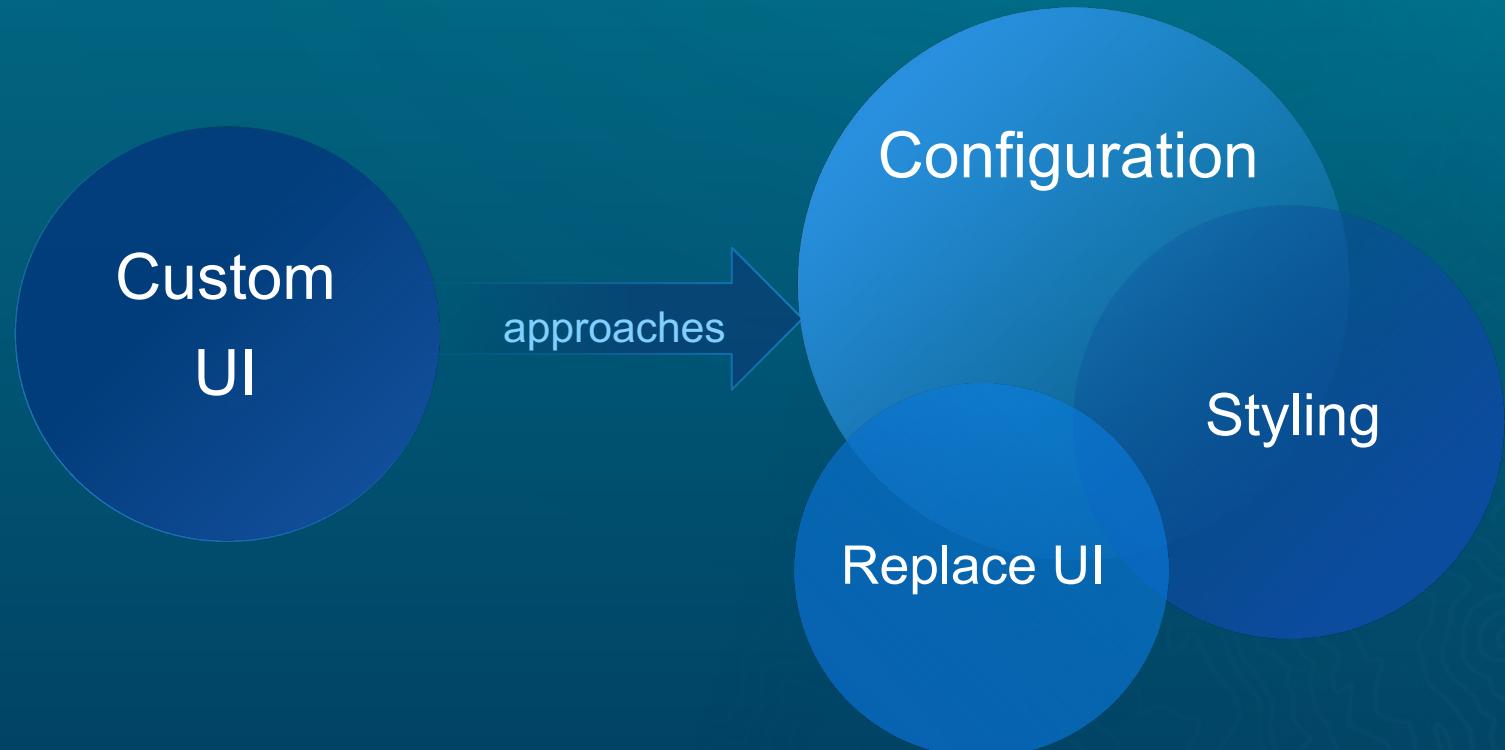
View + ViewModel in action

- View renders the state of the VM
 - Looks at properties on VM and renders accordingly
- User interacts with View (property/method)
 - Causes a change on VM or View
- View updates
 - Renders again due to changes on VM



Widgets

Approaches to customization



Widgets

Best practices for building a custom UI.

- Use widget viewModels to render custom UI
 - React to state changes and events:
[esri/core/reactiveUtils](#)
- Use your preferred
 - Framework
 - CSS Library
 - Design System



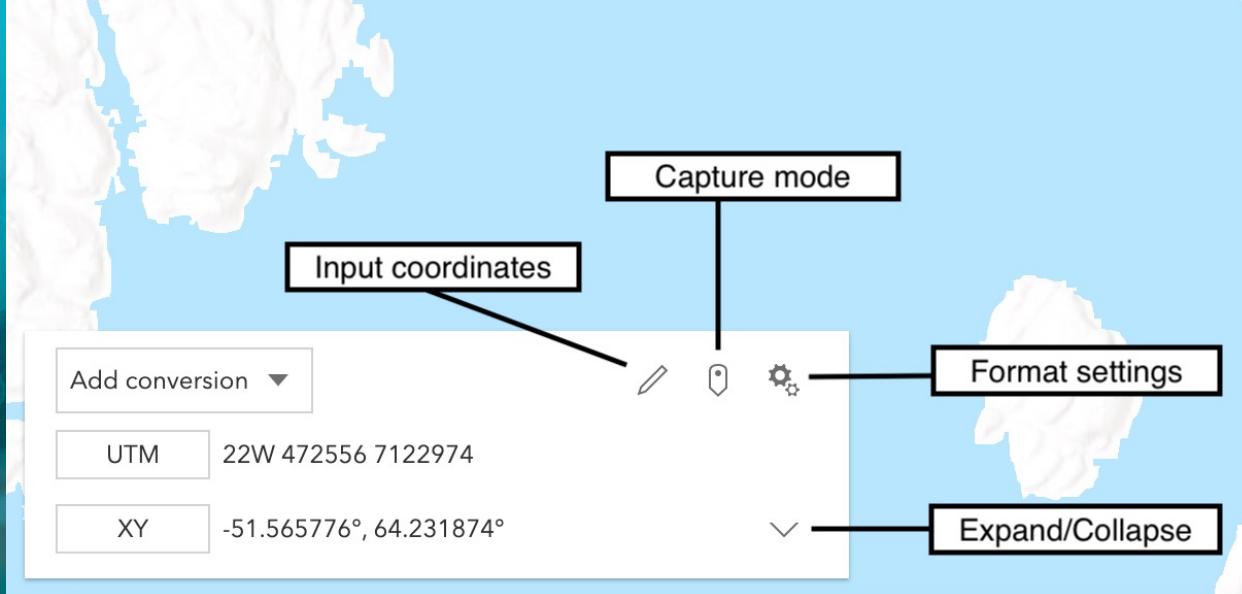
Building Custom UIs

Matt

```
view.goTo({  
    center: [-126, 49]  
})  
.catch(function(error) {  
    if (error.name != "AbortError") {  
        console.error(error);  
    }  
});
```

```
    queryParameters =  
    QueryParameters().apply {  
        whereClause = "price > 200"  
    }  
    viewModelScope.launch {
```

```
    queryParameters =  
        queryParameters().apply {  
            clause = "price > 200"  
        }  
    viewModelScope.launch {  
        val result = repository.getProducts(queryParameters)  
        products.postValue(result)
```



Coordinate Conversion Widget

Matt

Coordinate Conversion

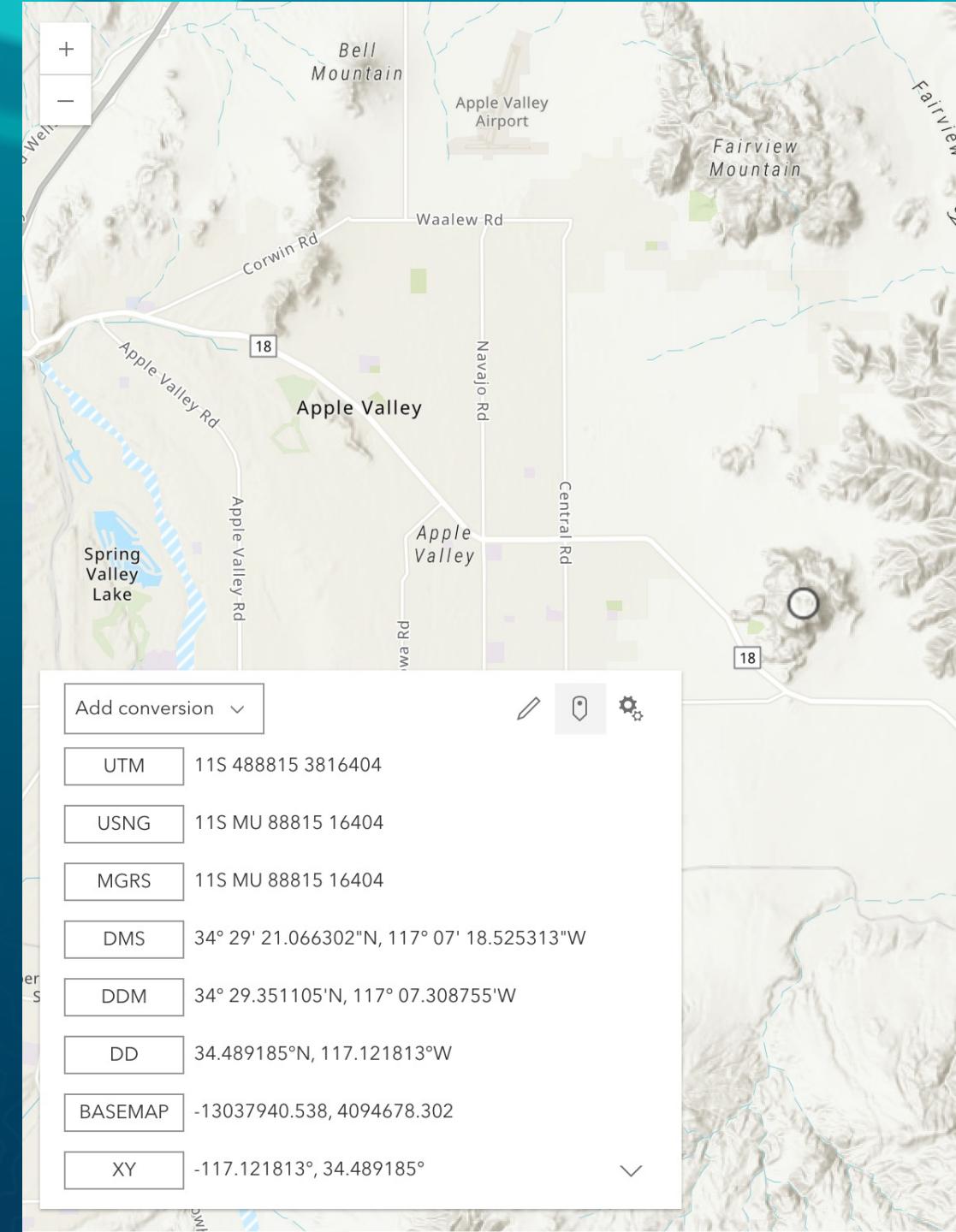
viewModel API to be used

- Properties

- *conversions* - Every *Conversion* the widget is currently displaying.
- *formats* - Every *Format* the widget is capable of displaying.
- *mode*- Current mode of the widget.
(live/capture)

- Methods

- *reverseConvert()* - Attempt to convert a string into a *Point*.



Coordinate Conversion Custom UI

Building custom UIs

- Calcite Design System

- Start design using Calcite UI Kit
- Transfer design to
 - Calcite Components
 - HTML/JS

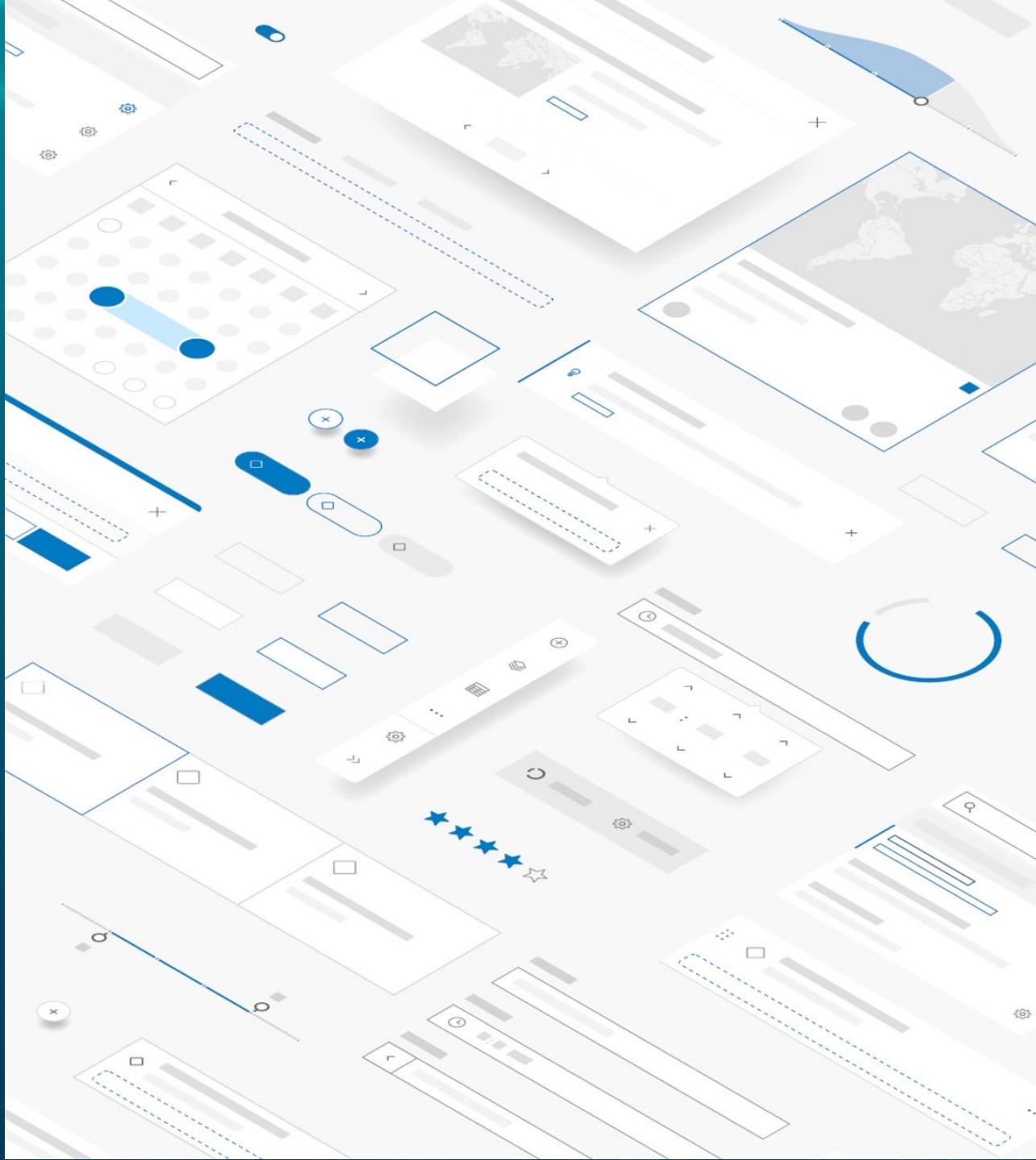
- Fluent 2 Design System

- Start design using Fluent UI
- Transfer design to
 - React JS (Create react app)
 - Fluent React Components

Calcite Design System

Collection of design and development resources

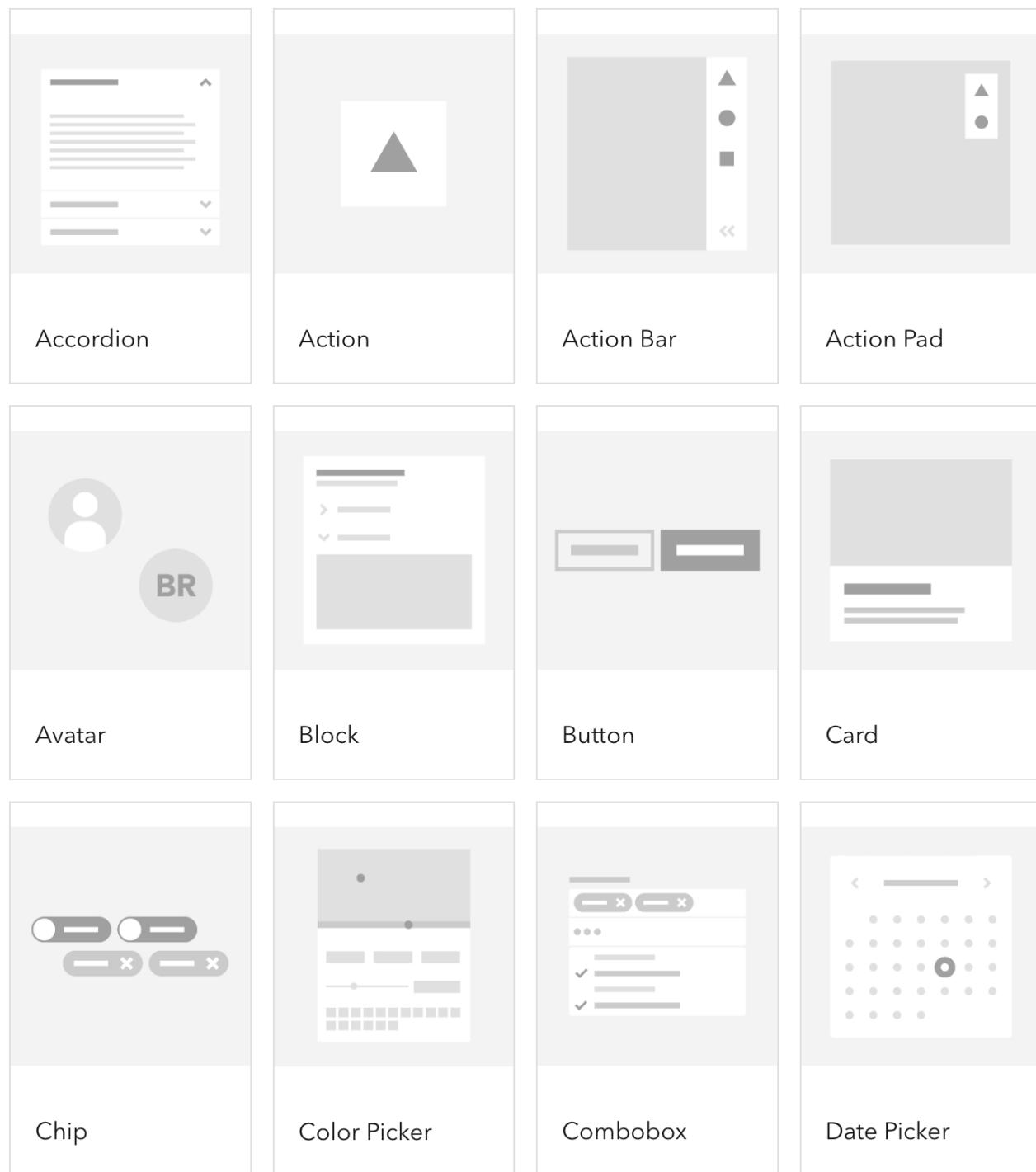
- Visual language for products
 - Design guidelines
 - Color, Spacing, Typography...
 - Graphic resources
 - Icons, symbols, Figma UI Kit
 - Interactive documentation
 - Patterns & best practices
 - Web Components
 - Tutorials
 - Sample playground
- Documentation



Calcite Components

Web Components for Calcite Design System

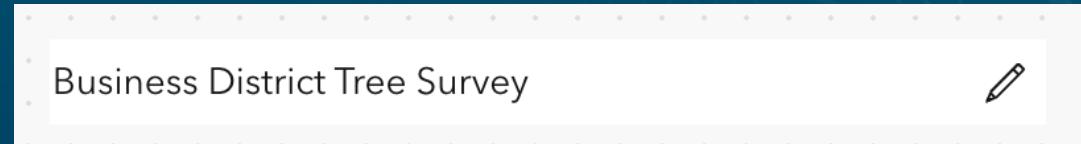
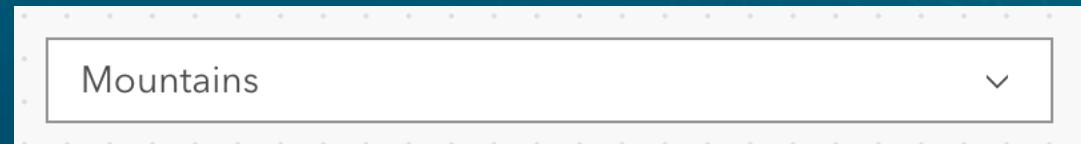
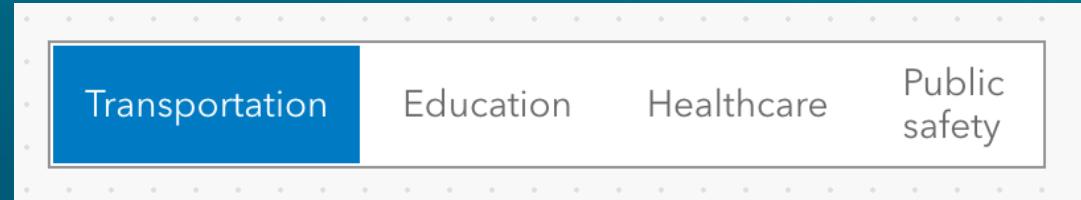
- 50+ web components
 - W3C specification
 - Consistent UX
 - Accessibility
 - Shadow DOM encapsulation
 - *slot* attribute for placing content within a component
- Easily build beautiful consistent apps
- Speed up development timelines
- Framework integration
- [Component Documentation](#)



Coordinate Conversion Design: Calcite Components

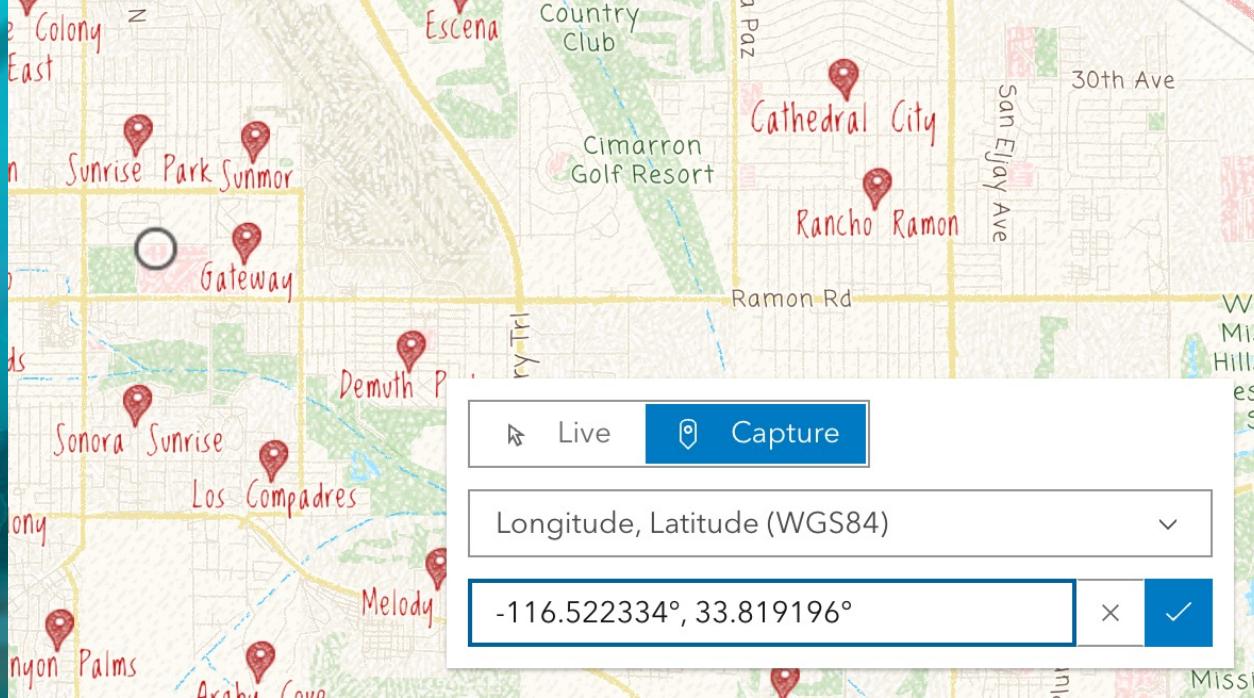
Using the Calcite UI Kit to build a custom UI

- Create design from UI Kit
- Components being used
 - Segmented control
 - Select
 - Inline Editable
 - Input



Calcite Coordinate Conversion Demo

Matt



Fluent Design System

Microsoft's design system

- Microsoft's design system
- Visual language
- Fluent UI
 - UX framework
 - React
 - Web Components
 - Windows
 - iOS
 - Etc.



Microsoft

Fluent UI

Get Started

Styles

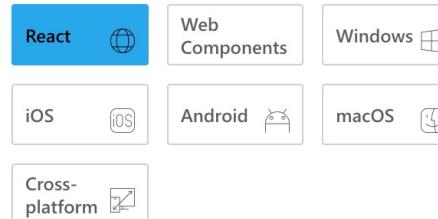
Controls

Resources

Dashboard

Introducing Fluent 2

Explore the next evolution of Microsoft's design system



Search Controls



- > Basic Inputs
- > Galleries & Pickers
- > Items & Lists
- > Commands, Menus & Navs
- > Notification & Engagement
- > Progress
- > Surfaces
- > Charts
- > Utilities
- > References

Controls

React

Check out the all new [Fluent UI version 9!](#)

Fluent UI's robust, up-to-date web controls are built on the control list to see the building blocks that are available.

Basic Inputs

Button	Checkbox
ComboBox	Dropdown
Link	Rating
Slider	SpinButton
Toggle	

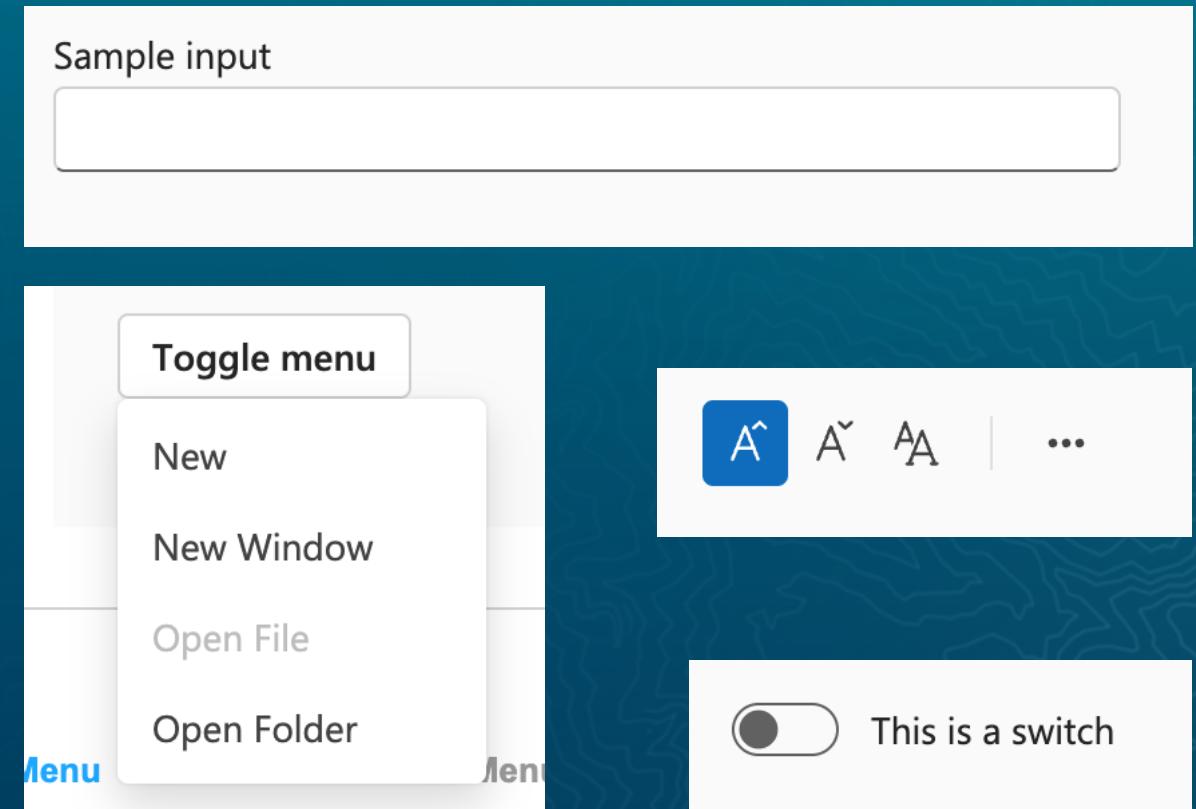
Galleries & Pickers

Calendar	ColorPicker
PeoplePicker	Pickers
TimePicker	

Coordinate Conversion Design: Fluent & React

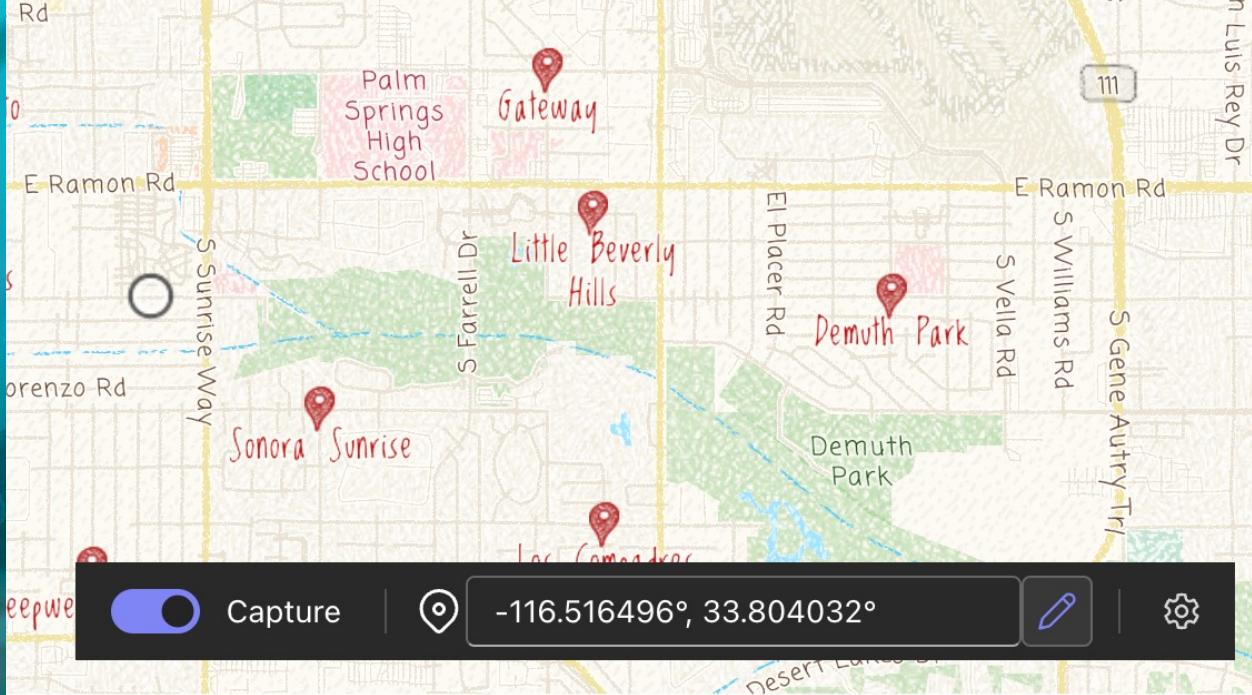
Using Fluent UI to build a custom ui

- Use Fluent UI controls to create a custom widget UI
- Components being used
 - Toolbar
 - Switch
 - Input
 - Menu
 - Toast



Fluent UI & React Coordinate Conversion Demo

Matt



```
    queryParameters =  
        queryParameters().apply {  
            cause = "price > 200"  
        }  
    viewModelScope.launch {  
  
        featureTable.queryFeatures(queryParameters)  
            .onFailure {  
                showError("Feature query failed")  
            }.onSuccess { featureQueryResult ->  
                val featureResult =  
                    featureQueryResults.firstOrNull()  
            }  
    }  
}
```

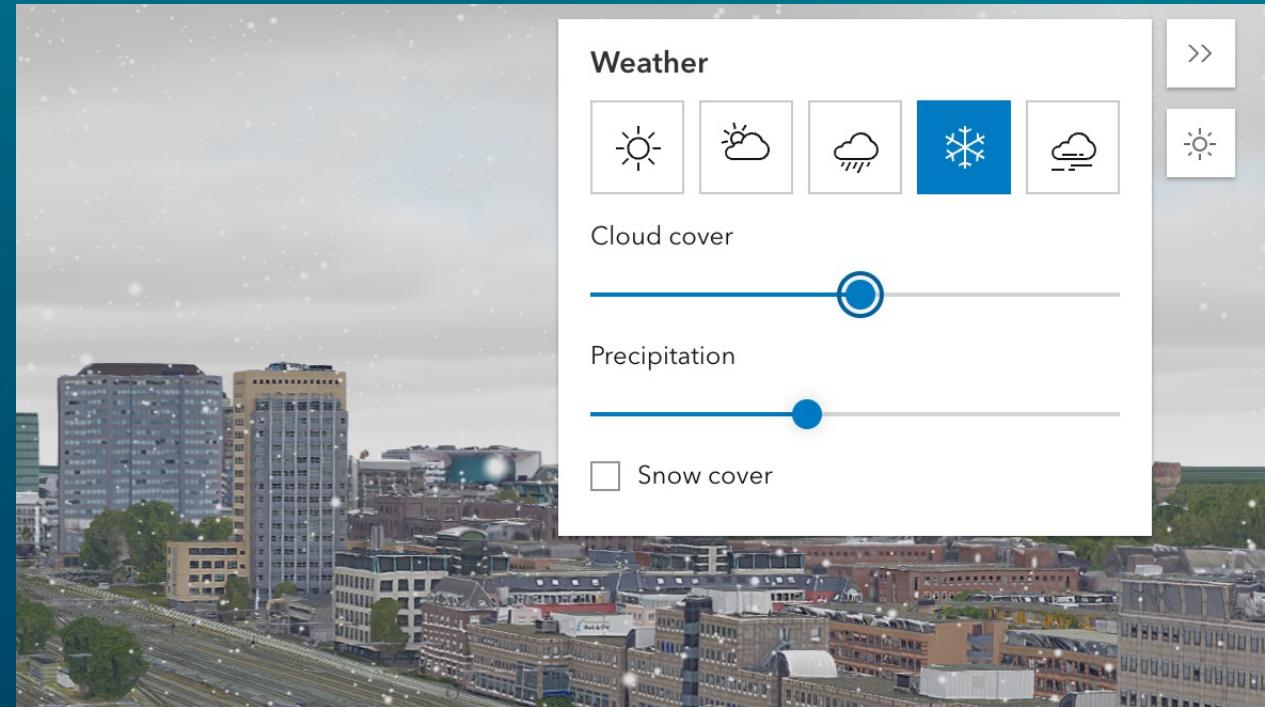
Custom Environment Widget

Jonathan

Weather Widget

Manipulate weather settings in a SceneView

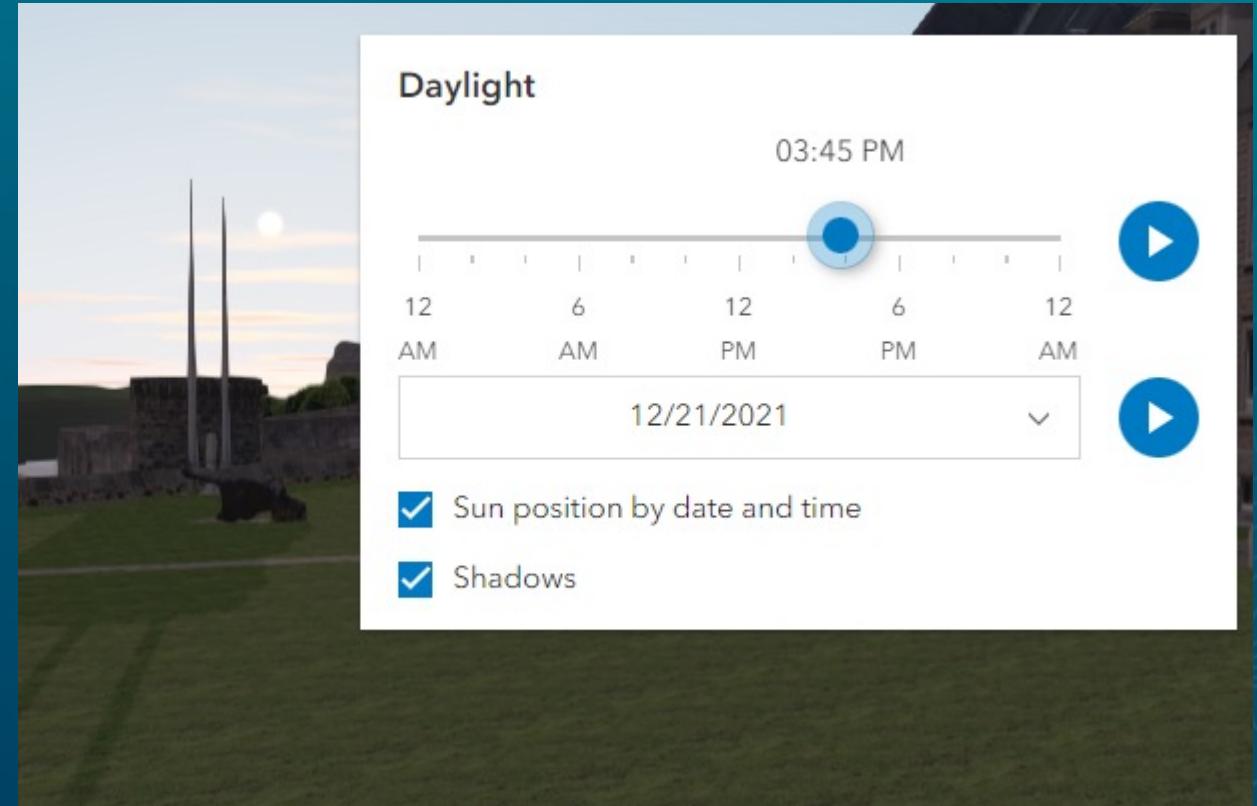
- Explore the Weather widget
- [Documentation Sample](#)
- [Widget](#)
- [ViewModel](#)



Daylight Widget

Manipulate lighting settings in a SceneView

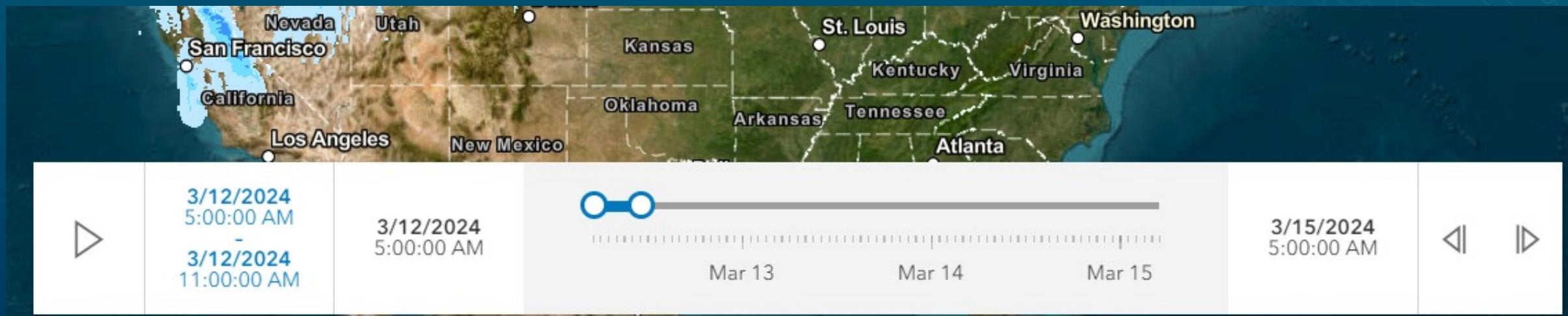
- Explore the Daylight widget
- [Documentation Sample](#)
- [Widget](#)
- [ViewModel](#)



TimeSlider Widget

Manipulate time settings in MapView and SceneView

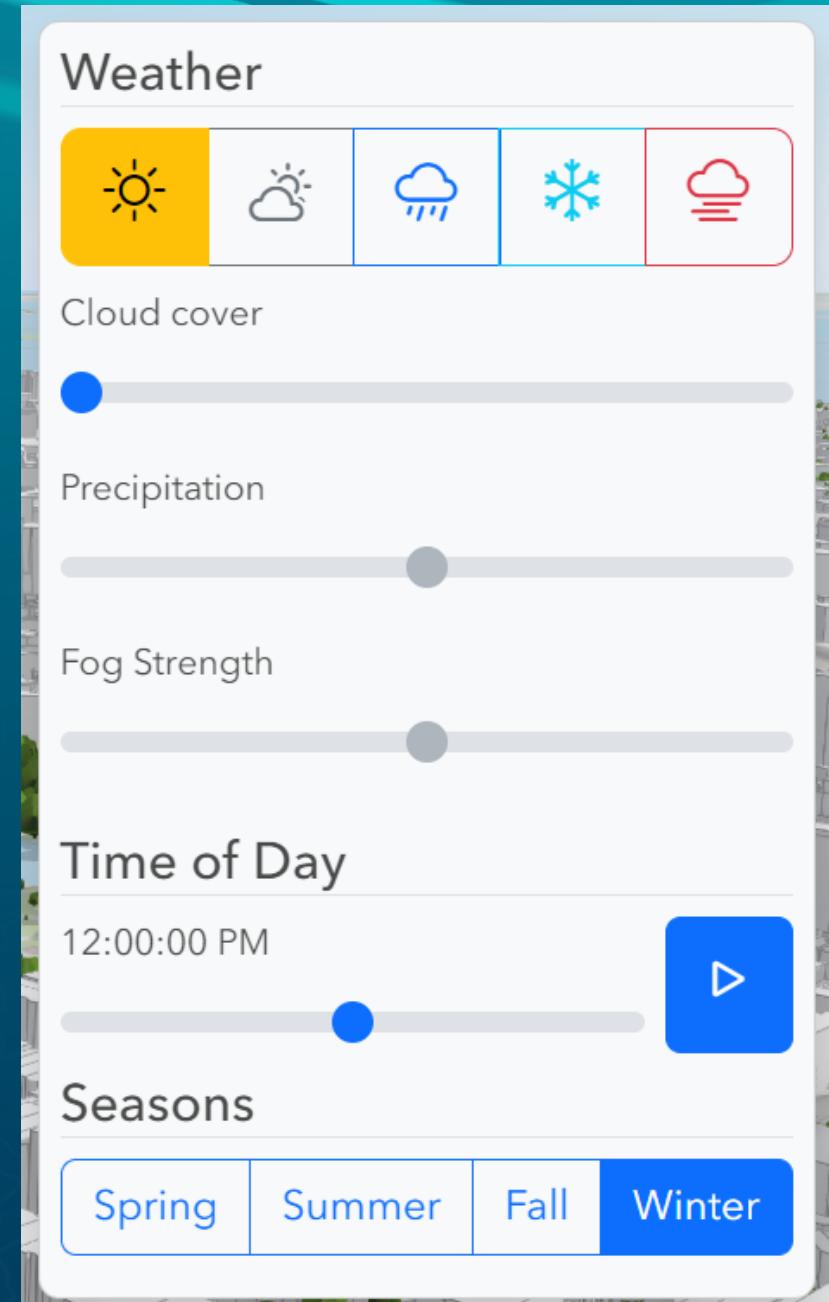
- Used internally by Daylight widget
- Built upon the base Slider widget
- Explore the TimeSlider widget
- [Documentation Sample](#)
- [Widget](#)
- [ViewModel](#)



Environment

Manipulate environment settings in a SceneView

- Scene
 - Preconfigured with time information
- SceneView
 - Environment
 - Weather
 - Lighting
- DaylightViewModel
 - currentSeason
 - dayPlaying
 - timeSliderPosition
- WeatherViewModel
 - setWeatherByType()



Bootstrap

Front-end toolkit for rapid application development

- Preconfigured CSS classes
- Visual consistency between UI components
- Share templates
- Works with larger frameworks
- [Getting Started](#)
- [Bootstrap Icons](#)
 - included separately



Getting started

Introduction

Download

Contents

Browsers & devices

JavaScript

Webpack

Parcel

Vite

Accessibility

RFS

RTL

Contribute

Customize

Overview

Sass

Options

Color

Color modes

Components

CSS variables

Get started with Bootstrap

Bootstrap is a powerful, feature-packed framework for building responsive, mobile-first projects.

Build anything—from prototype to production—using our prebuilt components and flexible utilities.

Quick start

Get started by including Bootstrap's production-ready CSS and JS files in your project. Bootstrap is designed to work without the need for any build steps. See it in practice with our [demo](#).

1. Create a new `index.html` file in your project root.

name="viewport" tag as well for proper responsive behavior.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
```

Bootstrap

What are we using?

- Components
 - Button
 - ButtonGroup
 - Input (Form)
- Utilities
 - Alignment
 - Flex
 - Spacing
- Icons

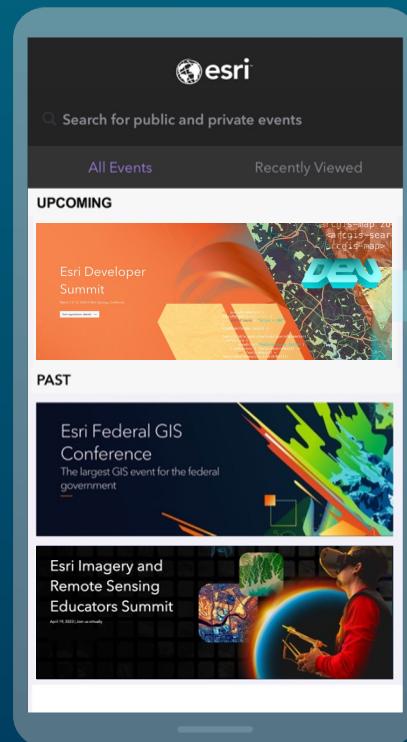
Getting started	Components
Introduction	Accordion
Download	Alerts
Contents	Badge
Browsers & devices	Breadcrumb
JavaScript	Buttons
Webpack	Button group
Parcel	Card
Vite	Carousel
Accessibility	Close button
RFS	Collapse
RTL	Dropdowns
Contribute	List group
	Modal
Customize	Navbar
Overview	Navs & tabs
Sass	Offcanvas
Options	Pagination
Color	Placeholders
Color modes	Popovers
Components	Progress
CSS variables	Scrollspy
Optimize	Spinners
	Toasts
Layout	Tooltips
Breakpoints	
Containers	
Grid	Helpers
Columns	Clearfix
Gutters	Color & background
	Colored links

Resources

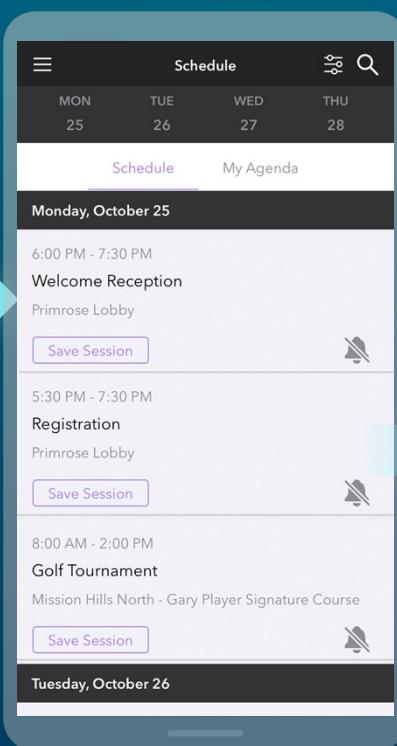
- [ArcGIS Maps SDK for JavaScript](#)
 - Weather
 - Coordinate Conversion
- [Calcite Design System](#)
- [Bootstrap](#)
- [Fluent Design System](#)
- [Create React App / React](#)
- Demo code: esriurl.com/2024-ds-custom-ui

Please Share Your Feedback in the App

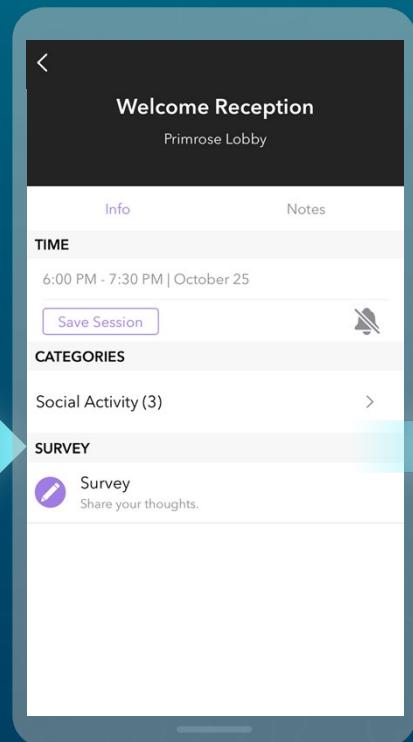
Download the Esri Events app and find your event



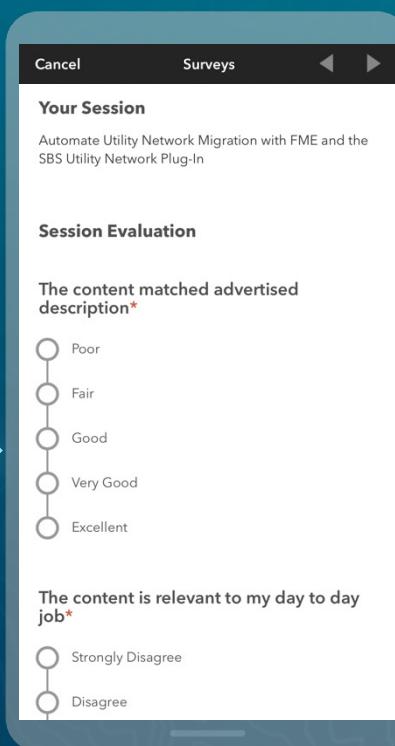
Select the session you attended



Scroll down to "Survey"



Log in to access the survey



The information shared in this presentation is
CONFIDENTIAL and **MAY NOT BE REDISTRIBUTED.**

For details on confidentiality see the Confidentiality clause in your
Esri partnering agreement or the Esri Distributor Agreement.

Connect With Us On Social

And Join the Conversation Using #EsriDevSummit2024

-  twitter.com/EsriDevs #EsriDevSummit2024
-  twitter.com/EsriDevEvents
-  youtube.com/c/EsriDevelopers
-  links.esri.com/DevVideos
-  github.com/Esri
-  github.com/EsriDevSummit
-  links.esri.com/EsriDevCommunity

```
<arcgis-map zoom="4" center="-118,34" />  
  
view.goTo({  
  center: [-126, 49]  
})  
.catch(function(error) {  
  if (error.name != "AbortError") {  
    console.error(error);  
  }  
});
```

```
// show the compass and pass the  
mapRotation state data  
Compass(rotation = mapRotation)  
    // reset the ComposableMapView's viewpoint  
rotation to point north using the  
mapViewModel  
    mapViewModel.setViewpointRotation(0.0)  
}
```



esri®

THE
SCIENCE
OF
WHERE®

```
const layerList = new LayerList({  
    view: view  
});  
  
// Add widget to the top right corner  
// of the view  
view.ui.add(layerList, "top-right")  
  
<arcgis-map zoom="4" center="-118,34"
```



esriurl.com/2024-ds-custom-ui

Copyright © 2024 Esri. All rights reserved.