



ESRI DEVELOPER SUMMIT 2023

ArcGIS Maps SDK for JavaScript: Better UX with Popups

Lauren Ballantyne and Lauren Boyd

Slides available:
[https://links.esri.com/DS23-
better-ux-popups](https://links.esri.com/DS23-better-ux-popups)

```
const layer = view.map.allLayers.get(0);
view.whenLayerView(layer)
  .then(layerView => console.log(`Layer loaded`))
  // if there were problems with the layer
  .catch(console.error);
```

```
const view = new MapView({
  container: "viewDiv",
  map: map,
  environment: {
    lightings: {
      directShadowsEnabled: true
    }
  }
});
```

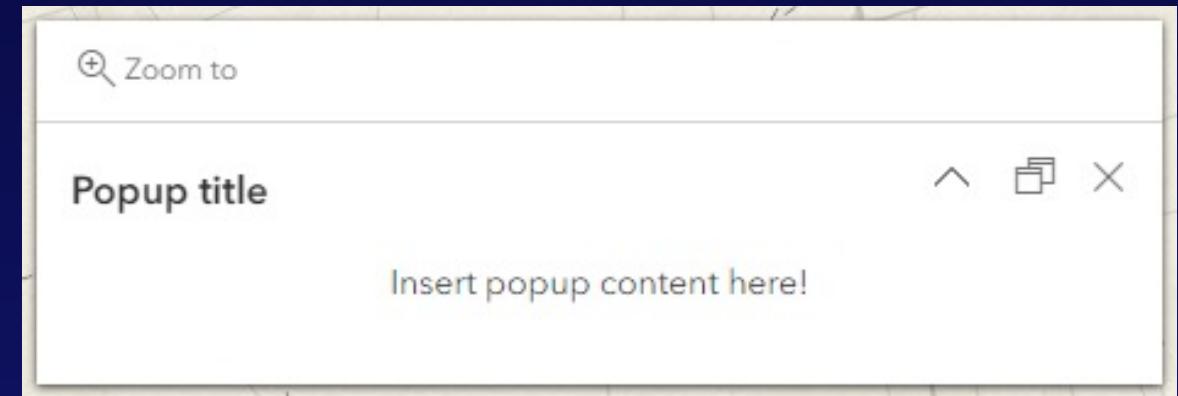
Agenda

- What are popups?
- Popup Elements
- Arcade
- Configuring popups in the Map Viewer
- Adding more to popups using JS SDK

Popups

What are they?

- Customizable
- Interactive
- Cross-platform feature
- Available for a variety of layer types



Popup Elements

- Text
- Fields list
- Attachments
- Media (Charts and Images)
- Arcade *NEW
- Related records *NEW

The image displays four separate Power BI desktop popups, each showing a different type of data visualization or record detail:

- 6.2 Magnitude Earthquake**: Shows details of an earthquake including depth (166.37 km), date (1/20/2023, 6:23 a.m.), and location (40 km WSW of Pointe-Noire, Guadeloupe). It also includes USGS PAGER ALERT and TSUNAMI WARNING sections.
- US States and Territories: Arkansas**: A table showing geographical codes: Name (Arkansas), Source of Geography Code (USS), and Universal Geography Code (AR).
- AED Location: York**: Shows a file attachment labeled "AED_Instructions.txt".
- Bird species observations**: Shows a chart of the Unemployment rate for the most recent 14 months of BLS data, with values ranging from 1 to 4. Below the chart, it says "Canada Goose" and provides behavior comments: "Canada Goose behavior comments" dated "July 20, 2022".

Arcade

What is it?

- ArcGIS's cross-platform expression language
- Create expressions to calculate new values on the fly
 - Use values
 - Use geometries
- Use FeatureSets to access data from other layers hosted in ArcGIS Online



Arcade

Use in pop-ups

- Attribute expressions
- Arcade elements

The screenshot shows the ArcGIS Arcade editor interface. On the left, the 'Run' tab displays the following JavaScript code:

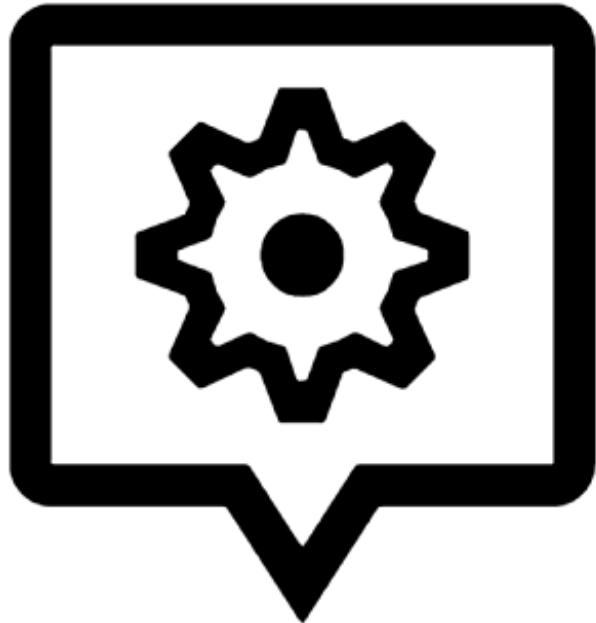
```
1  /*
2  Open the Suggestions tab and choose a template to get started creating different content types for your popups.
3  To learn more about using Arcade to create pop-up content visit:
4  https://developers.arcgis.com/arcade/guide/profiles/#popup-element
5 */
6
7 return {
8   type : 'text',
9   text : 'place your text or html here' //this property supports html tags
10 }
```

On the right, a sidebar titled 'Suggestions' is open, showing a search bar and a list of templates:

- Templates
 - Chart
 - Fields list
 - Rich text
- (x) Profile variables
- fx Functions
- Suggestions
- Help

At the bottom right of the sidebar, there is a 'Collapse' button.

```
const view = new SceneView({  
  container: "viewDiv",  
  map: map,  
  environment: {  
    lighting: {  
      directShadowsEnabled: true  
    }  
  }  
})
```



Map Viewer Popups

Lauren Ballantyne

```
  const layerView = viewer.layers.getLayer(index);  
  viewer.on("click", (event) => {  
    const layerView = viewer.layers.getLayer(index);  
    console.log(layerView);  
  })  
  // If you try to interact with the layerView, you'll get an error here
```

Popups & the JavaScript Maps SDK

Lauren Boyd

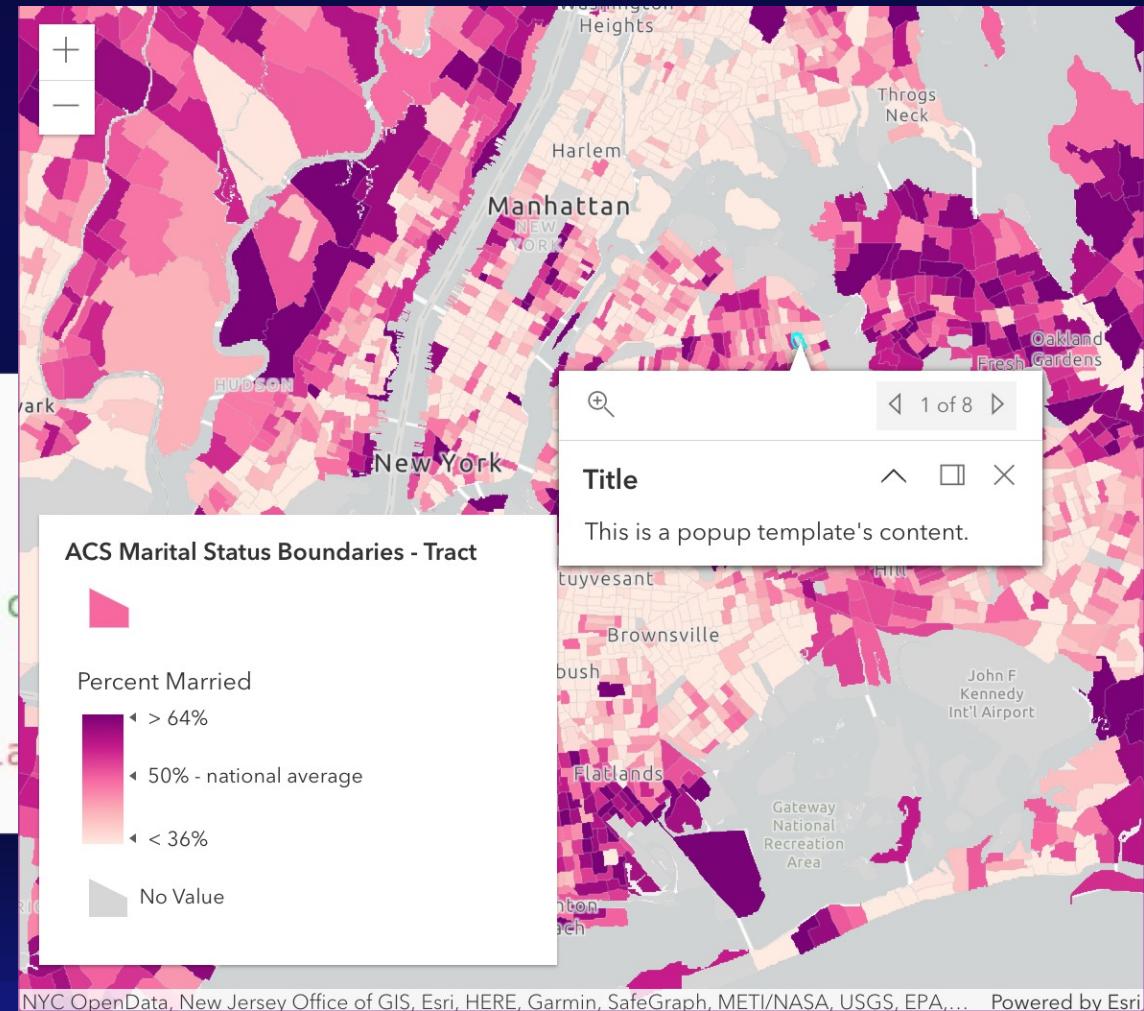
```
const layer = view.map.allLayers.get(0);
view.whenLayerView(layer)
  .then(layerView => console.log(`Layer ${layer.name} loaded`))
  // if there were problems with loading
  .catch(console.error);
```

```
const view = new View({
  container: "view",
  map: map,
  environment: {
    lightings: {
      directShadowsEnabled: true
    }
  }
});
```

PopupTemplate

- Formats and defines the popup content
- Set defaultPopupTemplateEnabled

```
1 const popupTemplate = {  
2   // autocasts as new PopupTemplate()  
3   title: "Title",  
4   content: "This is a popup template's content."  
5 };  
6  
7 featureLayer.popupTemplate = popupTemplate;
```



Content

Text Elements



```
1 // Create the TextContent element
2 let textElement = new TextContent({
3   // Set the text value to reference fields
4   text: "There are {Point_Count} trees within census block {BLOCKCE10}"
5 });
```

Content

Field Element

Beverly Hills trees by block	
Sum of species tree count	35
Count of Points	35
Common Name	NEW ZEALAND CHRISTMAS TREE
Block	3029

```
1 // Create the FieldsContent element
2 let fieldsElement = new FieldsContent({
3   fieldInfos: [
4     {
5       fieldName: "relationships/0/Point_Count_COMMON",
6       visible: true,
7       label: "Sum of species tree count",
8       format: {
9         places: 0,
10        digitSeparator: true
11      },
12      statisticType: "sum"
13    },
14    {
15      fieldName: "Point_Count",
16      visible: true,
17      label: "Count of Points",
18      format: {
19        places: 0,
20        digitSeparator: true
21      }
22    },
23    {
24      fieldName: "relationships/0/COMMON",
25      visible: true,
26      label: "Common Name"
27    },
28    {
29      fieldName: "BLOCKCE10",
30      visible: true,
31      label: "Block"
32    }
33  ]
34});
```

FieldInfo

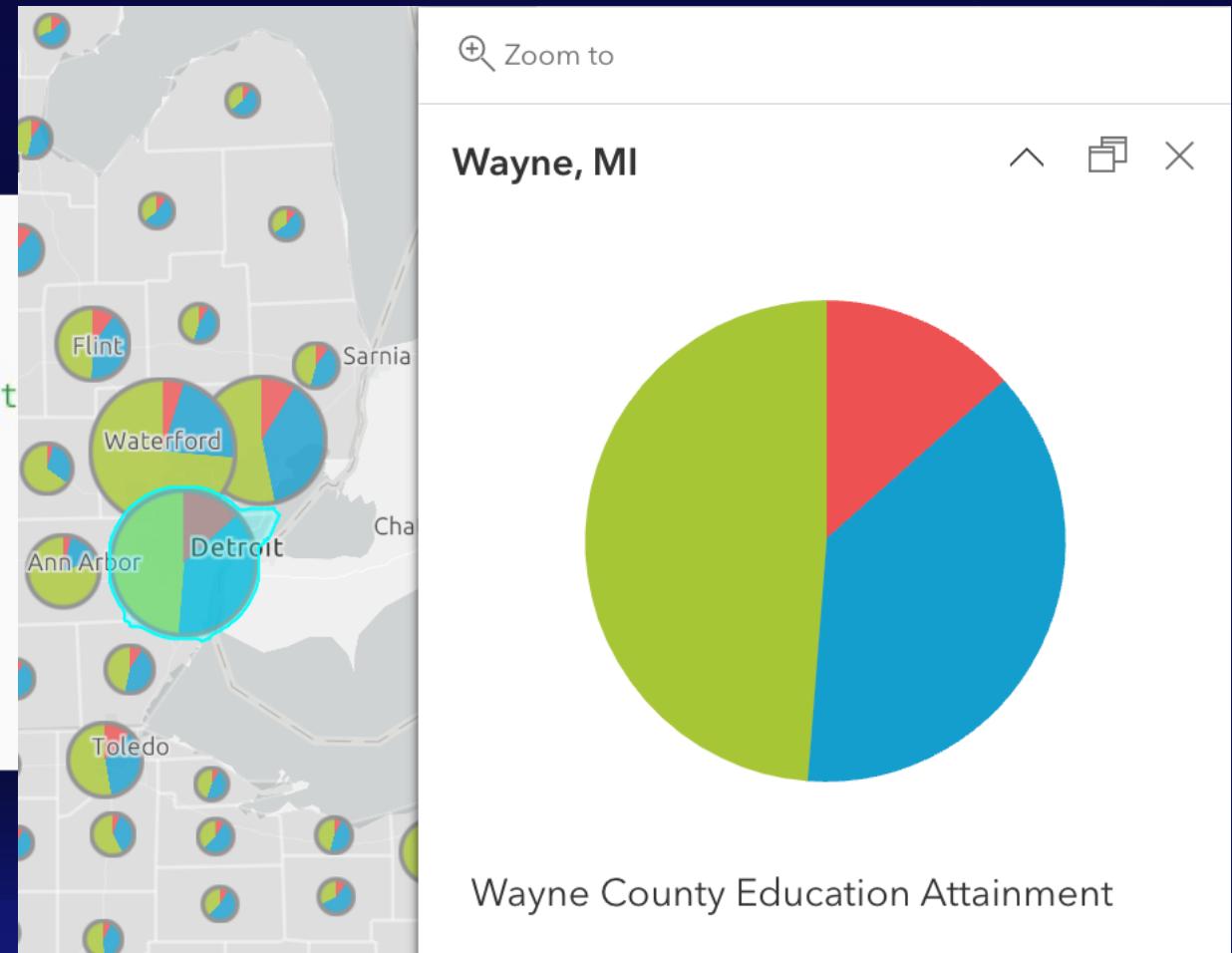
- Defines how fields participate in popup
- Set on either:
 - PopupTemplate
 - Individual FieldContent

```
1 // Format and set values for field
2 let fieldInfo = new FieldInfo({
3   fieldName: 'PROMINENCE_ft',
4   label: 'Prominence (feet)',
5   isEditable: true,
6   visible: false,
7   //autocasts to FieldInfo.Format
8   format: {
9     places: 0,
10    digitSeparator: true
11  }
12 });
13
14 // Set on popupTemplate - suggested way
15 popupTemplate.fieldInfos = [fieldInfo];
16
17 // or set on FieldContent itself
18 fieldContent.fieldInfos = [fieldInfo];
19 popupTemplate.content = [fieldContent];
```

Content

Media Elements

```
1 let mediaContent = new MediaContent({  
2   mediaInfos: [  
3     {  
4       title: '{COUNTY} County Education Attainment',  
5       type: 'pie-chart',  
6       value: {  
7         fields: ['SOMEHS_CY', 'HSGRAD_CY'],  
8       },  
9     },  
10   ],  
11 });
```



Content

Arcade Elements

- Must return a dictionary representing a popup element:
 - TextContent
 - FieldsContent
 - MediaContent

```
1 // Creates a column chart using Arcade
2 layer.popupTemplate = {
3   title: "Educational Attainment",
4   content: [
5     type: "expression",
6     expressionInfo: {
7       expression:
8         // Arcade expression that returns a popup element
9       ...
10      return{
11        type: "media",
12        attributes,
13        title: "Chart title",
14        mediaInfos: [...]
15      }
16    },
17    title: "Educational Attainment"
18  }
19];
20};
```

Content

Attachment Elements

```
1 let attachmentContent = new AttachmentsContent({  
2   displayType: "preview",  
3   title: "{StaduimName}",  
4   description: "{StadiumName} was played on in  
      {Year} and resides is {City}, {Country}."  
5 });
```

Zoom to Selected Stadium Full Extent 2 of 2

Rose Bowl (1994)

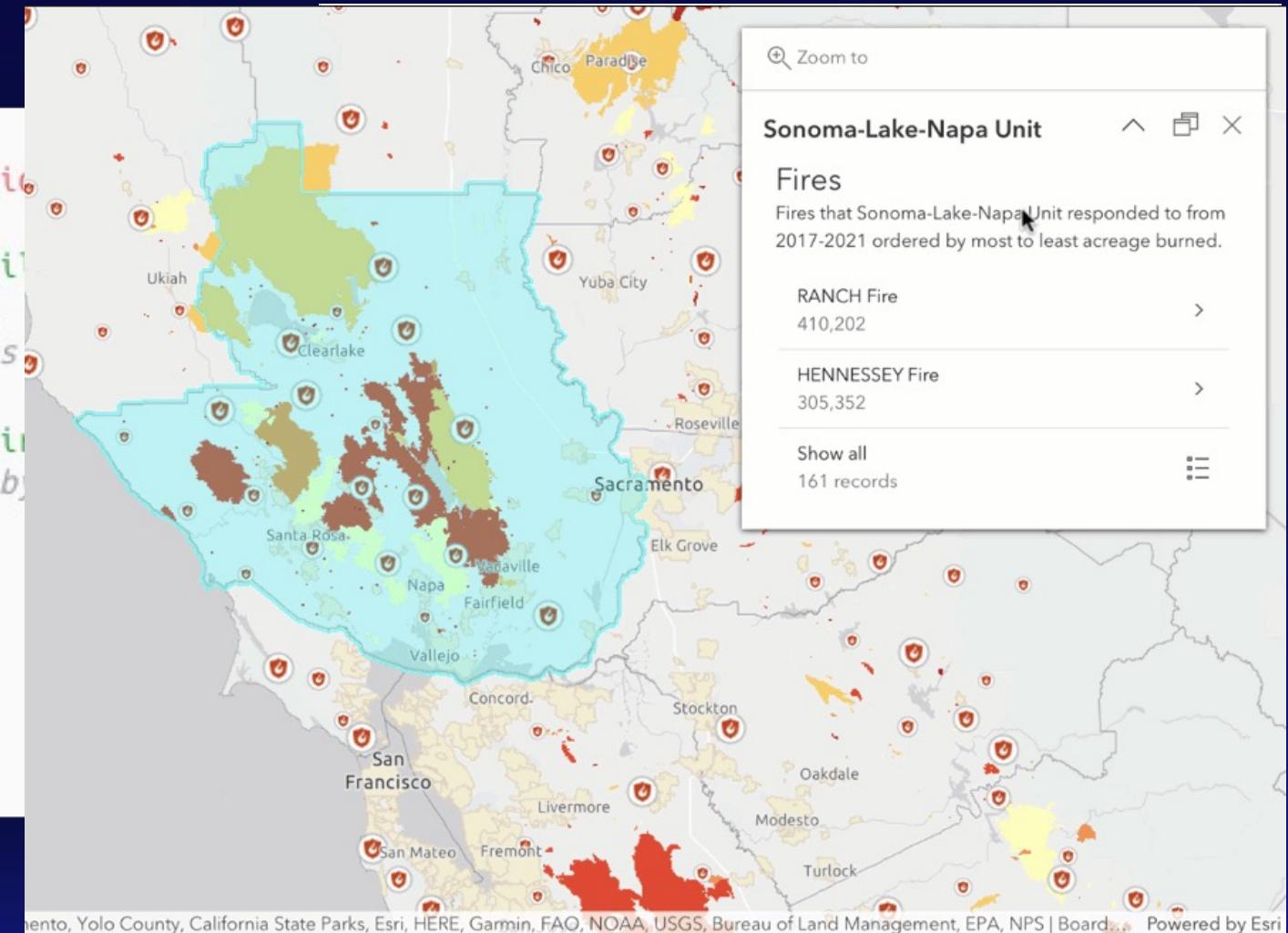
Rose Bowl was played on in 1994 and resides is Pasadena, United States.



Content

Relationship Elements

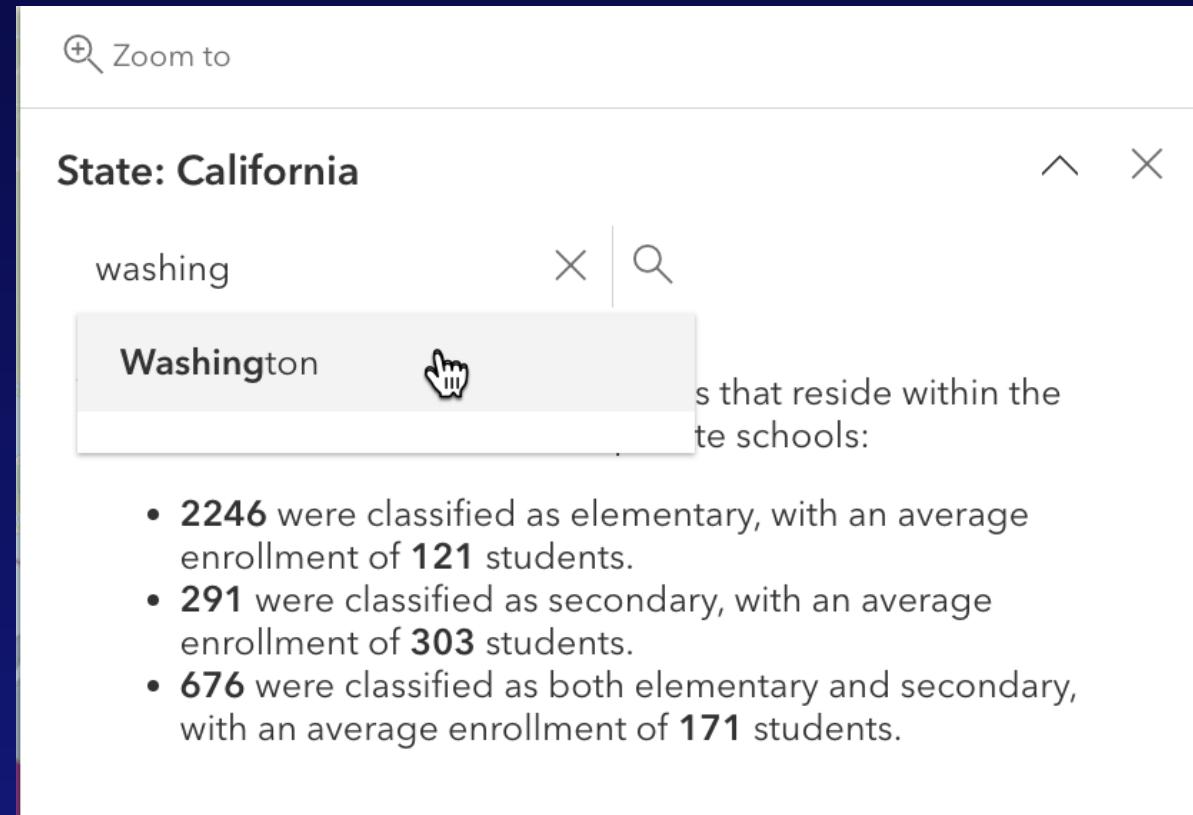
```
1 let relationshipContent = new RelationshipContent();
2   relationshipId: 1,
3   description:"All facilities for wildland fire protection that reside in {UNIT}.",
4   // Display only two related cities
5   displayCount: 2,
6   title: "Facilities for Wildland Fire Protection"
7   // Order list of related records by
8   // field in ascending order.
9   orderByFields: {
10     field: "NAME",
11     order: "asc"
12   }
13});
```



Content

JS SDK Specific

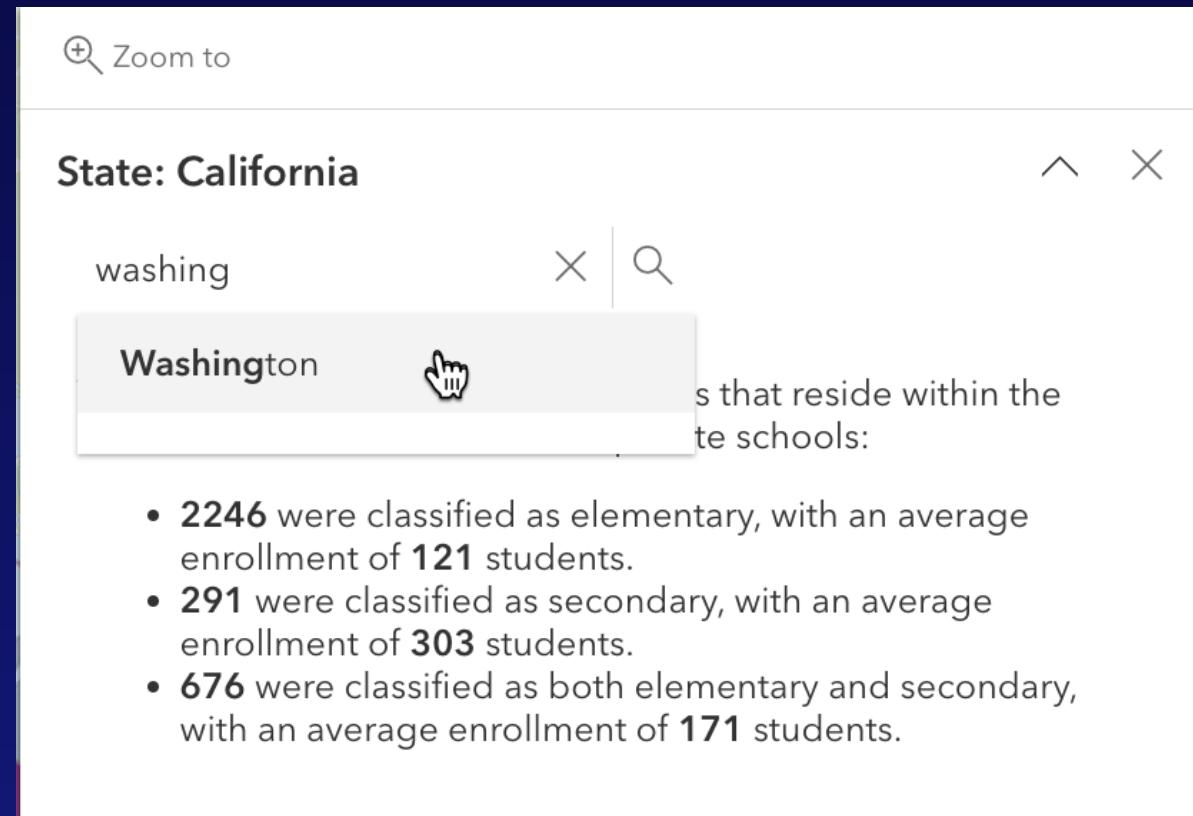
- CustomContent:
 - Displaying Widget



Content

JS SDK Specific

- CustomContent:
 - Displaying Widget
- Promise
 - Useful when calling method/executing query and want to display results



Content

JS SDK Specific

- CustomContent:
 - Displaying Widget
- Promise
 - Useful when calling method/executing query and want to display results
- Function
 - Popup may require additional processing

```
1 function formatContent(feature) {
2     // Check if the access type is open or unrestricted and set
3     // the color to green. Otherwise, set the color to red.
4     let color =
5         feature.graphic.attributes.ACCESS === "Open" || "Unrestricted"
6             ? "green"
7             : "red";
8
9     let textContent = new TextContent();
10    // Create the text content and set the text depending on which layer
11    // the feature is from and return in an array.
12    if (feature.graphic.layer.title == "Campsites") {
13        textContent.text =
14            `<b>{LABEL}</b> resides along {TRAIL}. This site is currently <span style='color: ` +
15            color + `';'>{ACCESS}</span>`;
16    } else {
17        textContent.text =
18            `'{TRAILNAME} is a <b>{MTFCC}</b> and currently <span style='color: ` +
19            color + `';'>{ACCESS}</span>.`;
20    }
21    return [textContent];
22}
```

Docking

```
1 view.popup = {  
2   dockEnabled: true,  
3   dockOptions: {  
4     // Sets which position to place the popup in the view  
5     position: "top-left",  
6     // Disables the dock button from the popup  
7     buttonEnabled: false,  
8     // Ignore the default sizes that trigger responsive docking  
9     breakpoint: true  
10    }  
11};
```

Actions

- Default actions
 - Zoom to
- Custom Actions
 - Styled with image or class

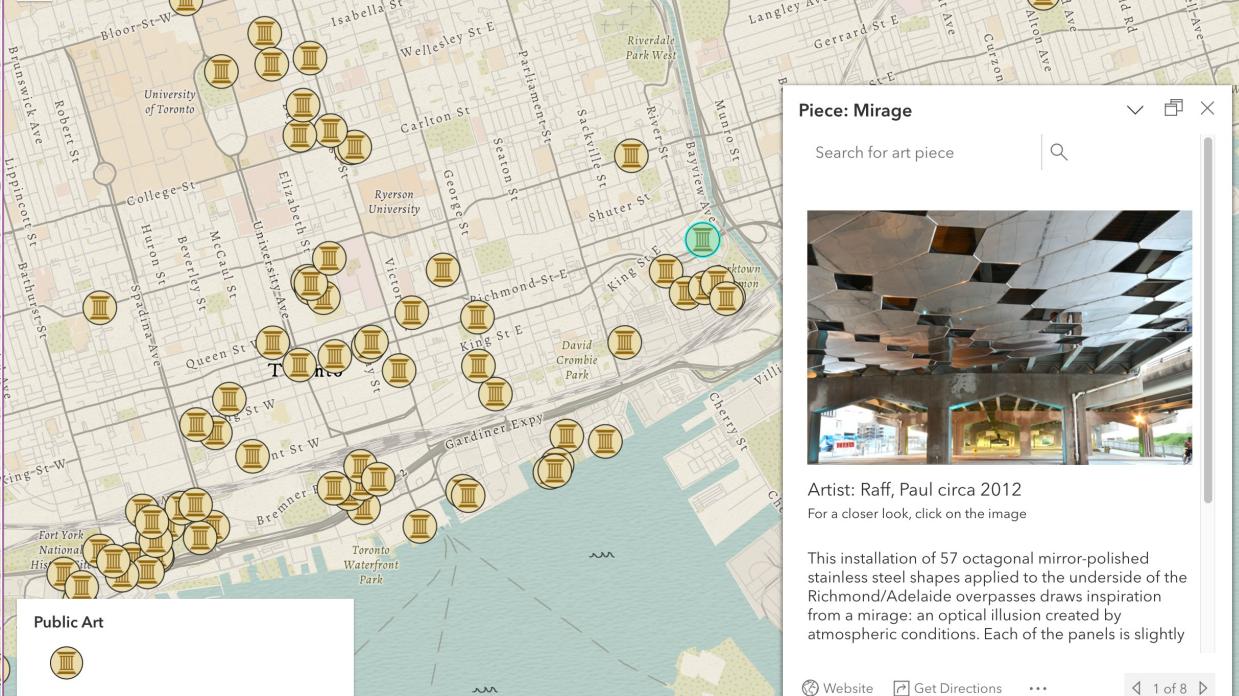
Cheshire Cat Brewery ✓ □ ×

Cheshire Cat Brewery
7803 Ralston Road
Arvada, Colorado
Phone: 1-303-431-9000

 [Brewery Info](#)  [Zoom to](#)

◀ 7 of 33 ▶

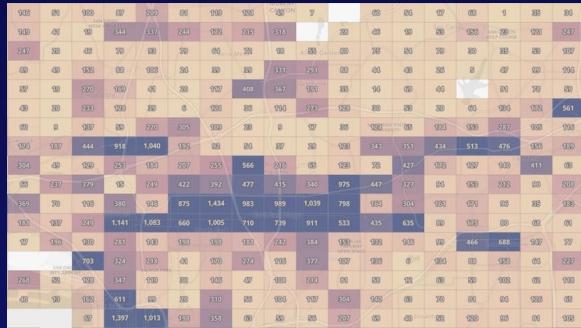
```
const view = new SceneView({  
    container: "viewDiv",  
    map: map,  
    environment: {  
        lighting: {  
            directShadowsEnabled: true  
        }  
    }  
})
```



Configuring Popups

Lauren Boyd

Resources



[How to summarize aggregate data using Arcade in popups](#)



[Rolling out related records in Map Viewer](#)



[Browsing related records with the ArcGIS API for JavaScript](#)

Slides available:
[https://links.esri.com/DS23-
better-ux-popups](https://links.esri.com/DS23-better-ux-popups)

```
const layer = view.map.allLayers.get(0);
view.whenLayerView(layer)
  .then(layerView => console.log(`Layer loaded`))
  // if there were problems with the layer
  .catch(console.error);
```

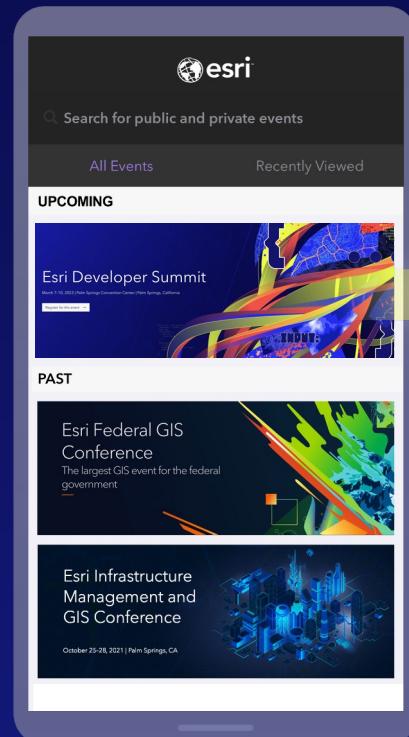
```
const view = new MapView({
  container: "viewDiv",
  map: map,
  environment: {
    lightings: {
      directShadowsEnabled: true
    }
  }
});
```

Upcoming sessions

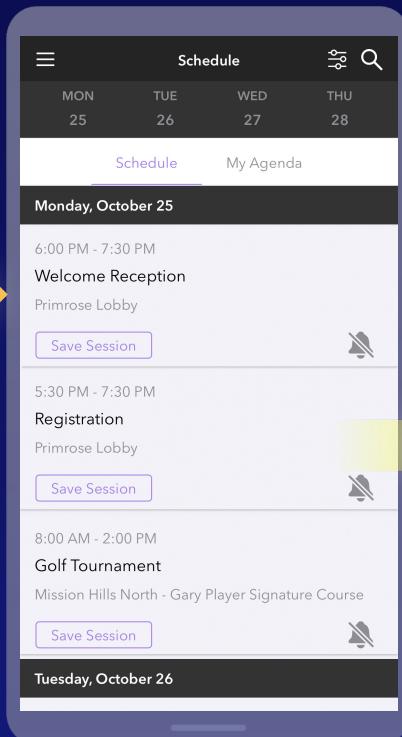
Session	Day/Time	Location
ArcGIS Maps SDK for JavaScript: Programming Patterns and API Fundamentals	Tuesday, 2:30PM – 3:30PM	Catalina Renaissance Hotel
ArcGIS Maps SDK for JavaScript: Build a Custom UI for API Widgets	Tuesday, 2:30PM – 3:30PM	Mesquite B PSCC
ArcGIS Maps SDK for JavaScript: Key Highlights of the Last Year	Tuesday, 5:30PM – 6:30PM	Primrose A PSCC
ArcGIS Maps SDK for JavaScript: Data Visualization	Wednesday, 10:30PM – 11:30PM	Catalina/Madera Renaissance Hotel
ArcGIS Online: Web Mapping with Arcade Expressions	Wednesday, 2:30PM – 3:30PM	Mojave Learning Center Renaissance Hotel
ArcGIS Maps SDK for JavaScript: Using the FeatureLayer	Wednesday, 4:00PM-5:00PM	Catalina/Madera Renaissance Hotel
ArcGIS Maps SDK for JavaScript: Bring in Data from Anywhere	Thursday, 10:30AM – 11:30AM	Mesquite C PSCC
ArcGIS Maps SDK for JavaScript: Tips and Tricks for Developing and Debugging Apps	Thursday, 1:00PM – 1:30PM	Demo Theater 3: Mesquite D-E PSCC
ArcGIS Maps SDK for JavaScript: Using Arcade with Your Apps	Thursday, 2:30PM – 3:30PM	Smoketree A-E PSCC

Please Share Your Feedback in the App

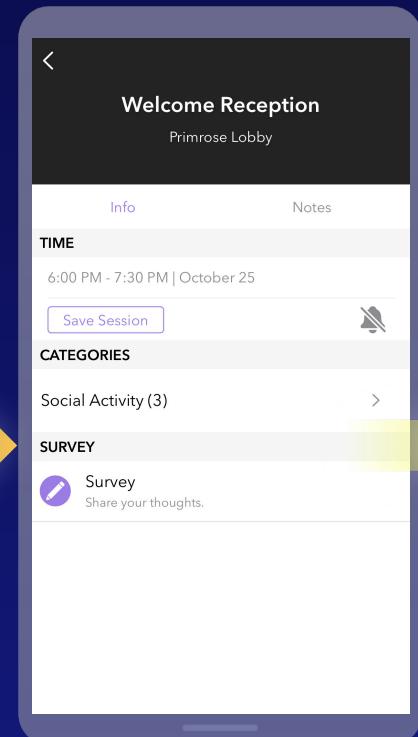
Download the Esri Events app and find your event



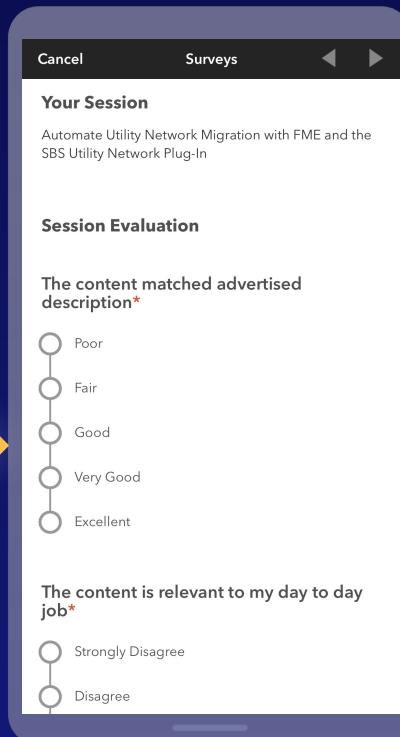
Select the session you attended



Scroll down to "Survey"



Log in to access the survey



Connect With Us On Social

And Join the Conversation Using #EsriDevSummit

-  twitter.com/EsriDevs #esridevsummit, #esridevsummit23
-  twitter.com/EsriDevEvents
-  youtube.com/@EsriDevs
-  links.esri.com/DevVideos
-  github.com/Esri
-  github.com/EsriDevEvents
-  links.esri.com/EsriDevCommunity

```
const layer = view.map.allLayers.get(0);  
view.whenLayerView(layer)  
.then(layerView => console.log(`  
// if there were problems with  
// the layer, they would appear here`))  
.catch(console.error);
```

```
const view = new MapView({  
  container: "viewContainer",  
  map: map,  
  environment: {  
    lightings: {  
      directShadowsEnabled: true  
    }  
  }  
});
```



esri®

THE
SCIENCE
OF
WHERE®

Copyright © 2022 Esri. All rights reserved.

```
const layer = view.map.addLayer(  
  view.whenLayerView(layer)   
    .then(layerView => const  
      {  
        if (there were problems)  
          catch(console.error);  
      }  
    );  
  );
```

E/SCRIPT>

```
const view = new SceneView({  
  container: "viewDiv",  
  map: map,  
  environment: {  
    lighting: {  
      directional:  
    },  
  },  
});
```

LIVE
BY
THE
CODE