

ArcGIS Maps SDK for JavaScript: Programming Patterns and API Fundamentals

Lauren Boyd and Jose Banuelos

2023 ESRI USER CONFERENCE

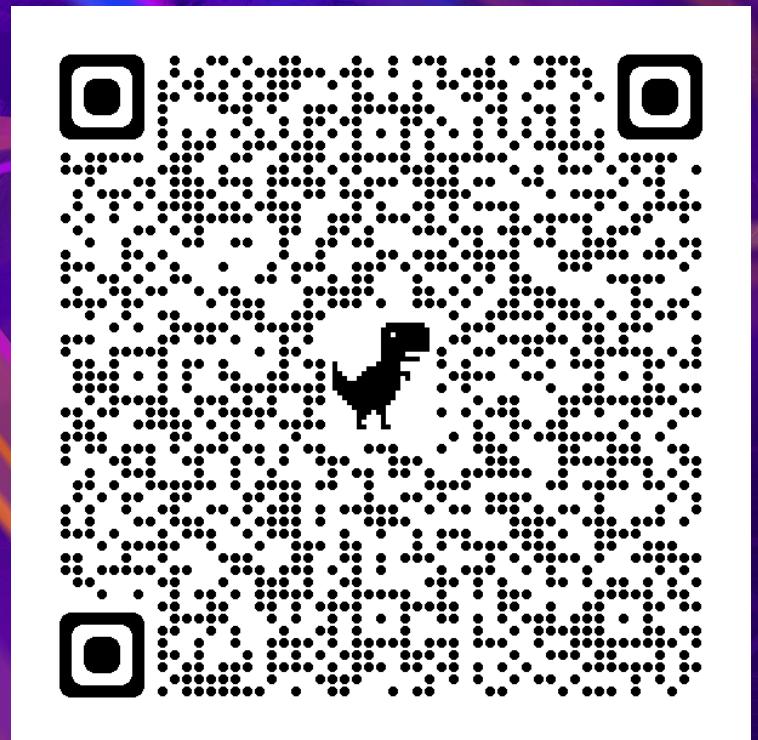


Agenda

- Accessing the SDK
- Maps, Webmaps, and Views
- Layers and Visualization
- Client and server queries
- Programming Patterns
- Widgets and ViewModels

Presentation Resources

<https://links.esri.com/UC2023/programming-patterns>



Accessing the SDK

Jose Banuelos

Get Started

The screenshot shows the ArcGIS Developers website with a dark blue header. The header includes the ArcGIS Developers logo, navigation links for Documentation, Features, Pricing, and Support, a search bar, and a sign-in button.

The main content area has a light gray background. On the left is a sidebar with a search bar and a list of links including Overview, Key features, Get started (which is highlighted with a purple dot), Install and set up, Release notes, FAQ, Community, Tutorials, Core concepts, Visualization, Building your UI, Working with ArcGIS Online and Enterprise, Developer tooling, and Migrating from 3.x.

The main content area features a large heading "Get started" and a numbered list:

- 1 | [Install and set up](#) the API
- 2 | [Get an API key](#) (access services)
- 3 | Start the [Display a map \(2D\)](#) tutorial

Below this, a section titled "1. Install and set up" provides information about bringing the ArcGIS Maps SDK for JavaScript into an app, mentioning AMD modules via ArcGIS CDN and ES modules for local builds, and linking to the "Install and set up" tutorial.

A code block at the bottom shows the following script:

```
<link rel="stylesheet" href="https://js.arcgis.com/4.25/esri/themes/light/main.css">
<script src="https://js.arcgis.com/4.25/"></script>
```

On the right side of the content area, there's a sidebar titled "On this page" with the same three items from the numbered list. Below that is a "Was this page helpful?" section with "Yes" and "No" buttons.

Asynchronous Module Definition (AMD)



```
<link rel="stylesheet" href="https://js.arcgis.com/4.27/esri/themes/light/main.css" />
<script src="https://js.arcgis.com/4.27/"></script>
```

Loading Classes via AMD



Array of ordered namespace
strings for each class

Class loaded into
arguments of callback

```
1 require(["esri/Map", "esri/views/MapView"], (Map, MapView) => {  
2   // The application logic using `Map` and `MapView` goes here  
3   const map = new Map();  
4   ...  
5 });
```

ECMAScript Modules (ESM)

ES modules via NPM

```
npm install @arcgis/core
```



```
import Map from "@arcgis/core/Map.js";
import MapView from "@arcgis/core/views/MapView.js";

const map = new Map({
  basemap: "topo-vector"
});
```

Constructors, properties and methods

Classes have a constructor; property values can be set by passing parameters to the constructor

```
const map = new Map({  
  basemap: "topo-vector"  
});
```

Then the class's properties and methods are available



```
// sets the basemap property to "streets-vector"  
map.basemap = "streets-vector";
```

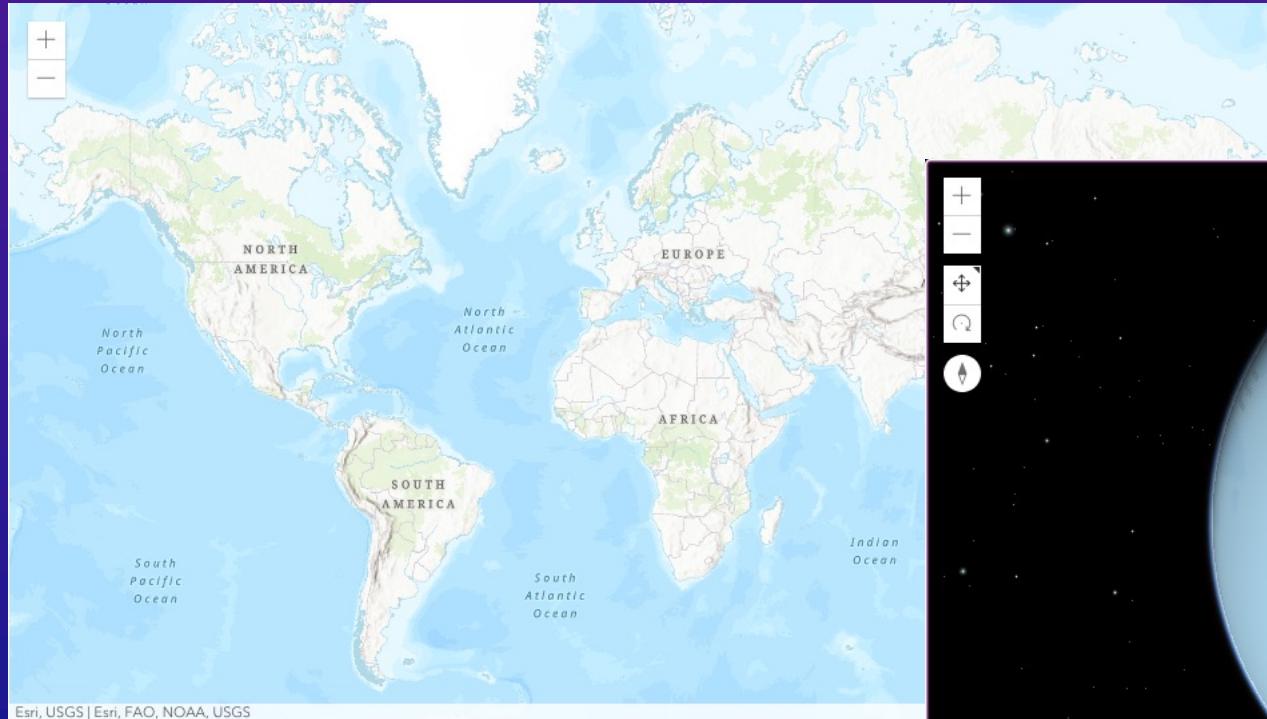


```
// destroys the map and any associated resources  
map.destroy();
```

Maps and Views

Lauren Boyd

MapView & SceneView



Map and Views

```
1 const map = new Map({
2   basemap: "topo-vector",
3   // World elevation service
4   ground: "world-elevation"
5 });
6
7 const mapView = newMapView({
8   map: map,
9   container: "viewDiv"
10 });
11
12 const sceneView = newSceneView({
13   map: map,
14   container: "viewDiv"
15 });
```

Basemaps

- Basemaps for developers

- For use with API keys

- Basemaps for ArcGIS Organization users

- For use with ArcGIS Online/ArcGIS Enterprise users

```
1 const map = new Map({  
2   /*  
3    For ArcGIS Organizational Accounts:  
4    "satellite", "hybrid", "terrain", "oceans", "osm",  
5    "dark-gray-vector", "gray-vector", "streets-vector",  
6    "topo-vector", "streets-night-vector", "streets-relief-vector",  
7    "streets-navigation-vector"  
8   For use with API keys:  
9   "arcgis-imagery", "arcgis-imagery-standard", "arcgis-imagery-labels",  
10  "arcgis-light-gray", "arcgis-dark-gray", "arcgis-navigation",  
11  "arcgis-navigation-night", "arcgis-streets", "arcgis-streets-night",  
12  "arcgis-streets-relief", "arcgis-topographic", "arcgis-oceans",  
13  "osm-standard", "osm-standard-relief", "osm-streets",  
14  "osm-streets-relief", "osm-light-gray", "osm-dark-gray",  
15  "arcgis-terrain", "arcgis-community", "arcgischarted-territory",  
16  "arcgis-colored-pencil", "arcgis-nova", "arcgis-modern-antique",  
17  "arcgis-midcentury", "arcgis-newspaper", "arcgis-hillshade-light",  
18  "arcgis-hillshade-dark", "arcgis-human-geography", "  
19  arcgis-human-geography-dark"  
20  */  
21  basemap: "topo-vector",  
22  // World elevation service  
23  ground: "world-elevation"  
24 })
```

WebMaps and WebScenes

WebMap

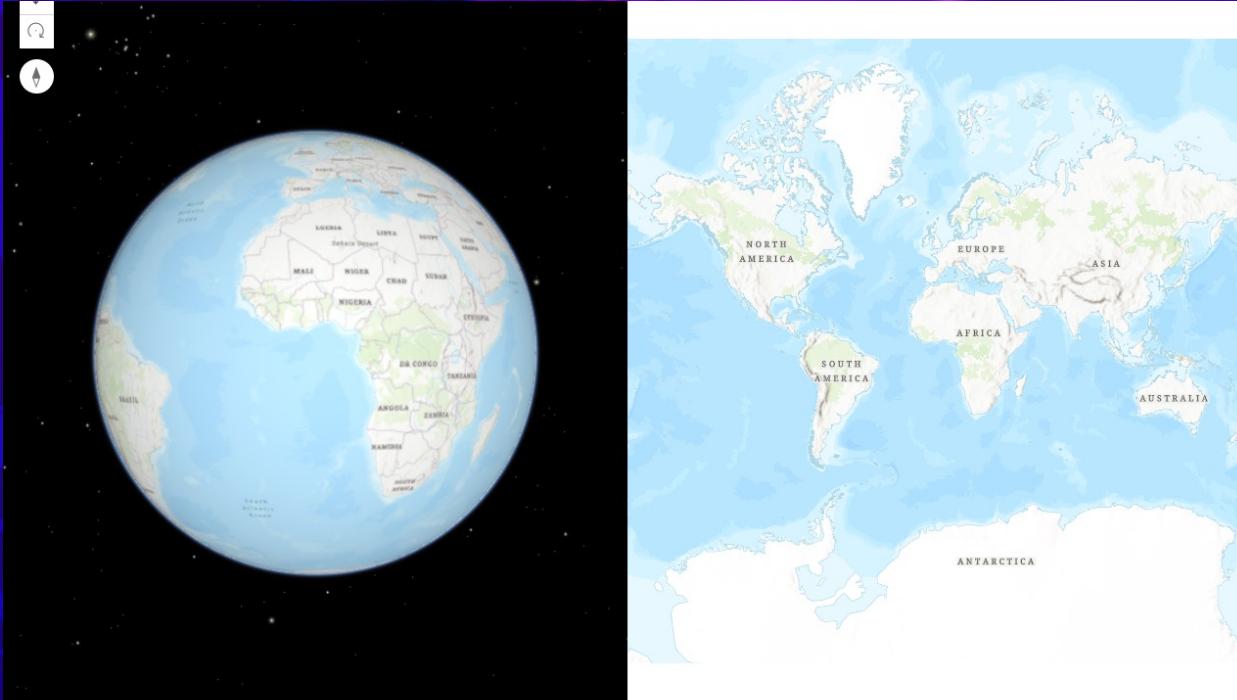
```
1 const webmap = new WebMap({
2   portalItem: {
3     id: "e691172598f04ea8881cd2a4adaa45ba",
4     // If using a webmap from Enterprise, set the portal property
5     // portal: {
6     //   url: "https://myHostName.esri.com/portal"
7     // }
8   },
9 });
10
11 // Add the webmap to the MapView
12 const view = newMapView({
13   map: webmap,
14   container: "viewDiv"
15 });
```

WebScene

```
1 const scene = new WebScene({
2   portalItem: {
3     id: "e691172598f04ea8881cd2a4adaa45ba",
4     // If using a scene from Enterprise, set the portal property
5     // portal: {
6     //   url: "https://myHostName.esri.com/portal"
7     // }
8   },
9 });
10
11 // Add the webmap to the MapView
12 const view = new SceneView({
13   map: scene,
14   container: "viewDiv"
15 });
```

Maps, WebMaps, and Views

[Demo](#)



Layers and Visualization

Lauren Boyd

Layers

Create a layer from a URL or portal item

```
1 const layer = new FeatureLayer({
2   // Autocasts to esri/portal/PortalItem
3   // and defaults to ArcGIS Online
4   portalItem: {
5     id: "8444e275037549c1acab02d2626daaee"
6     // portal: {
7     //   url: "URL to Portal (if not ArcGIS Online)"
8     // }
9   }
10 });
11 map.add(layer);
```

Visualization

Visualization Style	Renderer
Location Style	Simple Renderer
Data Driven Style	Unique Types , Class Breaks , Visual Variables , Time , Multivariate , Predominance , Dot Density , Relationship , Smart Mapping , Pie Charts
High Density Data	Clustering , Binning , Heatmap , Opacity , Bloom , Aggregation , Thinning , Visible Scale Range
3D Visualization	Global and local scenes , terrain rendering , cities in 3D , Visualizing points with 3D symbols

Renderers

- Renderers define how to visually represent features in the layer

```
1 const layer = new FeatureLayer({  
2   url: "SERVICE URL",  
3   renderer: {  
4     type: "simple", // autocasts as new SimpleRenderer()  
5     symbol: {  
6       // define symbol properties  
7       type: "simple-marker", // autocasts as new SimpleMarkerSymbol()  
8       outline: {  
9         color: [0, 246, 84, 1]  
10      },  
11      size: 14,  
12      color: [255, 255, 255, 0]  
13    }  
14  }  
15 })
```



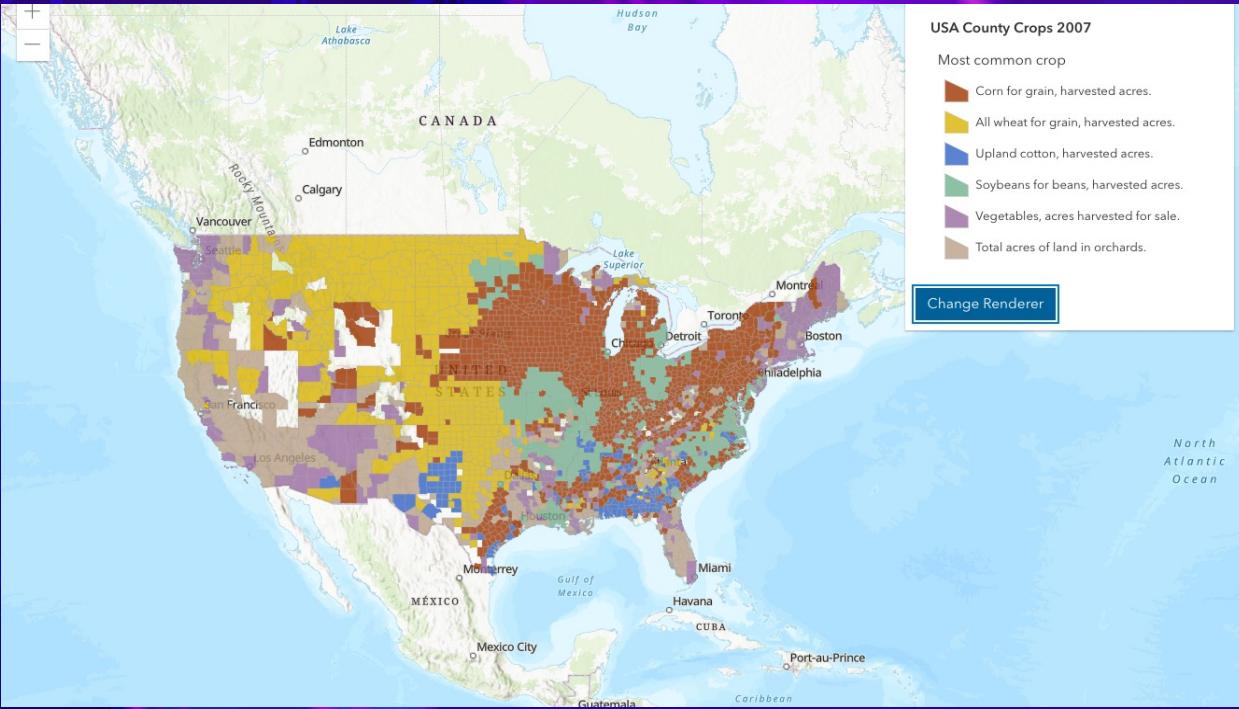
Smart Mapping

- Easy way to generate visualizations
- Focus on:
 - One visualization method
 - Subset of attributes
 - Specific Area



```
1 let layer = new FeatureLayer({
2   url:
    "https://services.arcgis.com/V6ZHFr6zdgNZuVG0/arcgis/rest/services
     /counties_politics_poverty/FeatureServer/0"
3 });
4
5 // simple visualization to indicate features with a single symbol
6 let params = {
7   layer: layer,
8   view: view
9 };
10
11 // when the promise resolves, apply the renderer to the layer
12 locationRendererCreator.createRenderer(params)
13   .then(function(response){
14     layer.renderer = response.renderer;
15   });

```



Smartmapping

Demo

Widgets and ViewModels

Jose Banuelos

Widgets

- Professionally Designed and tested building blocks
- Responsive, Accessible, and Localized
- Out of the box



```
const widget = new SomeWidget({  
  view,  
  ...otherOptions  
})
```



Weather Demo

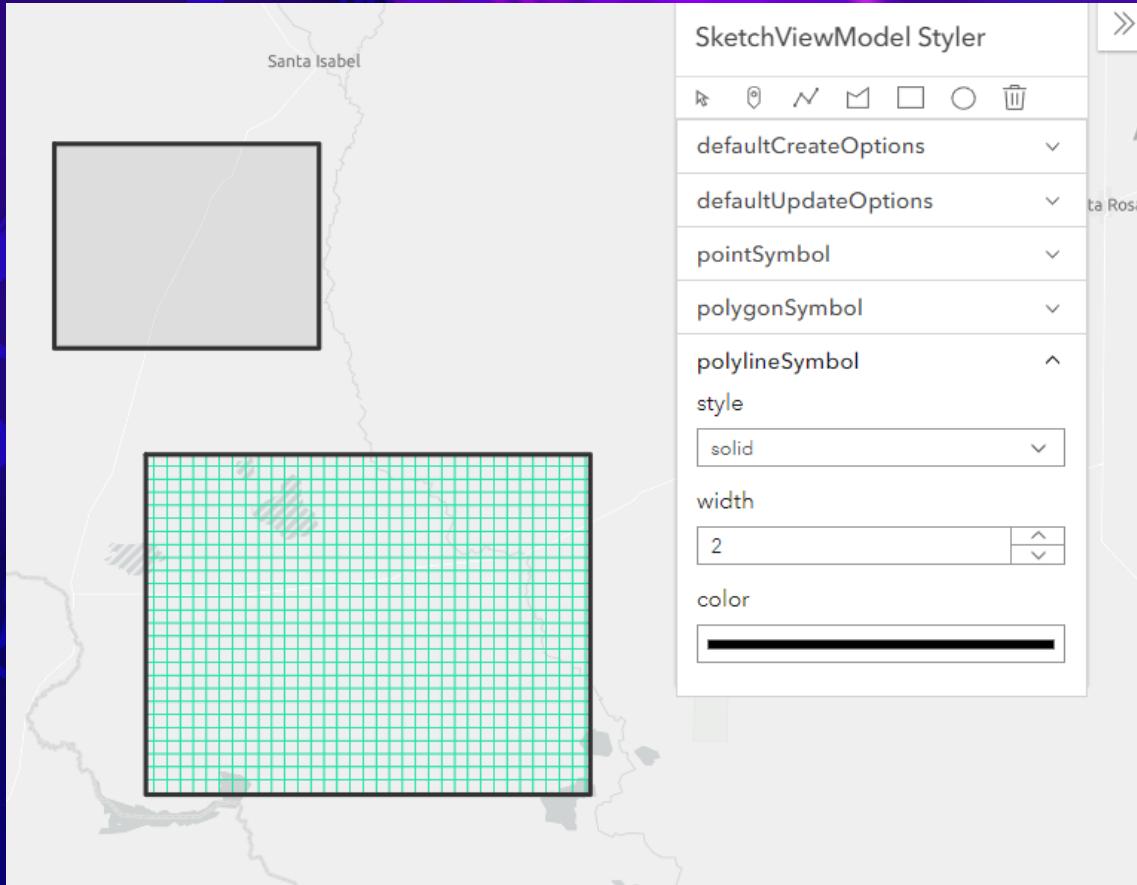


Editor

Demo

ViewModels





ViewModels

Demo

Client and Server Queries

Jose Banuelos

LayerView

- Renders features in the view
- Client-side querying, filtering, highlighting, and effects



```
let featureLayerView;  
  
view.whenLayerView(featureLayer).then(layerView => {  
  featureLayerView = layerView;  
});
```

Creating and using a query

- Using **createQuery()**
 - Maintains filters and definition expressions
- **new Query()**
 - Clean query object

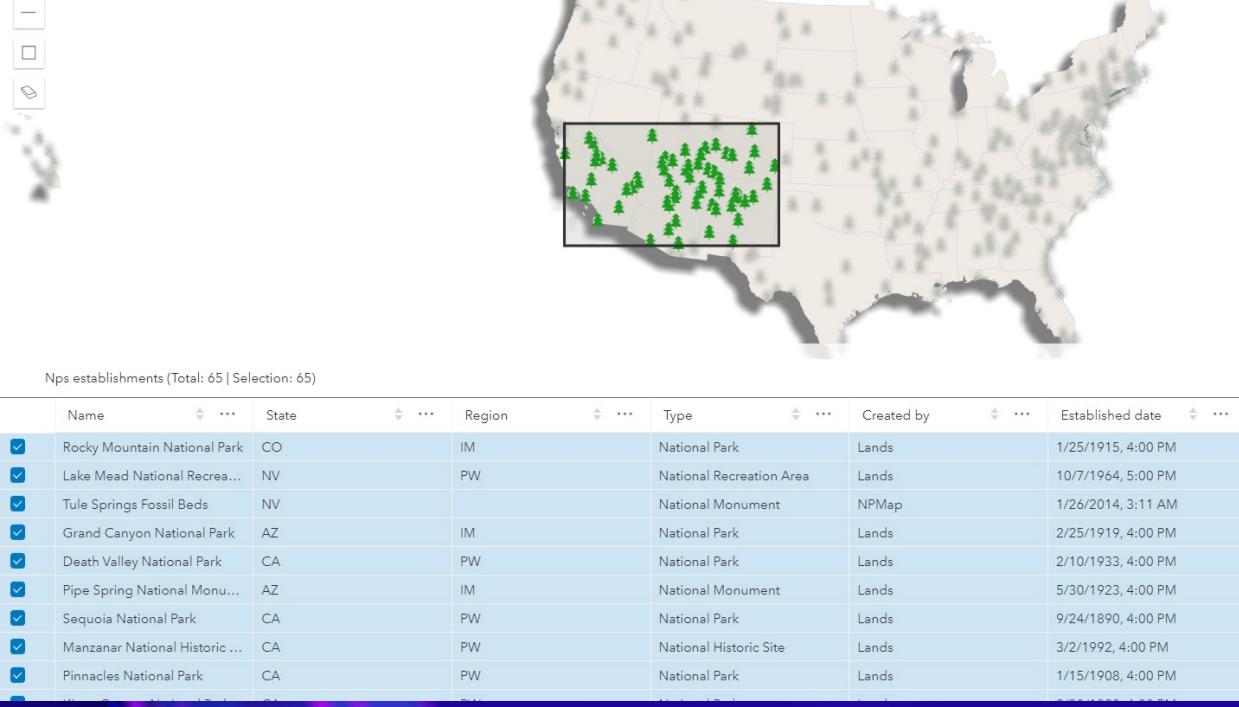
```
// construct a query with a where clause
const query = featureLayer.createQuery();
query.where = "Tree_Height < 60";

// query the FeatureLayer for features with "Tree_Height" < 60 ft
const treesUnder60 = await featureLayer.queryFeatures(query);
// map through the results and assign the objectIds to a variable
const treeOIDs = treesUnder60.features.map((feature) => feature.attributes.ObjectId);

// highlight the queried features
layerView.highlight(treeOIDs);
```

Should I use a layer query or a layer view query?

Query	LayerView	Layer
Performance	X	
Geographic precision		X
Query every feature		X



Client-side query

Demo

Programming Patterns

Lauren Boyd

Autocasting

```
1 require(["esri/Color", "esri/Graphic", "esri/symbols/SimpleLineSymbol",
2   "esri/symbols/SimpleFillSymbol"],
3   (Color, Graphic, SimpleLineSymbol, SimpleFillSymbol) =>
4   const graphic = new Graphic({
5     symbol: new SimpleFillSymbol({
6       color: new Color([173, 216, 230, 0.2]),
7       outline: new SimpleLineSymbol({
8         color: new Color([255, 255, 255]),
9         width: 1
10      })
11    });
12 });


```

symbol [Symbol](#) [autocast](#)

```
require(["esri/Graphic"], (Graphic) => {
  const graphic = new Graphic({
    symbol: {
      type: "simple-fill",
      color: [173, 216, 230, 0.2],
      outline: {
        color: [255, 255, 255],
        width: 1
      }
    }
  });
});


```

Promises

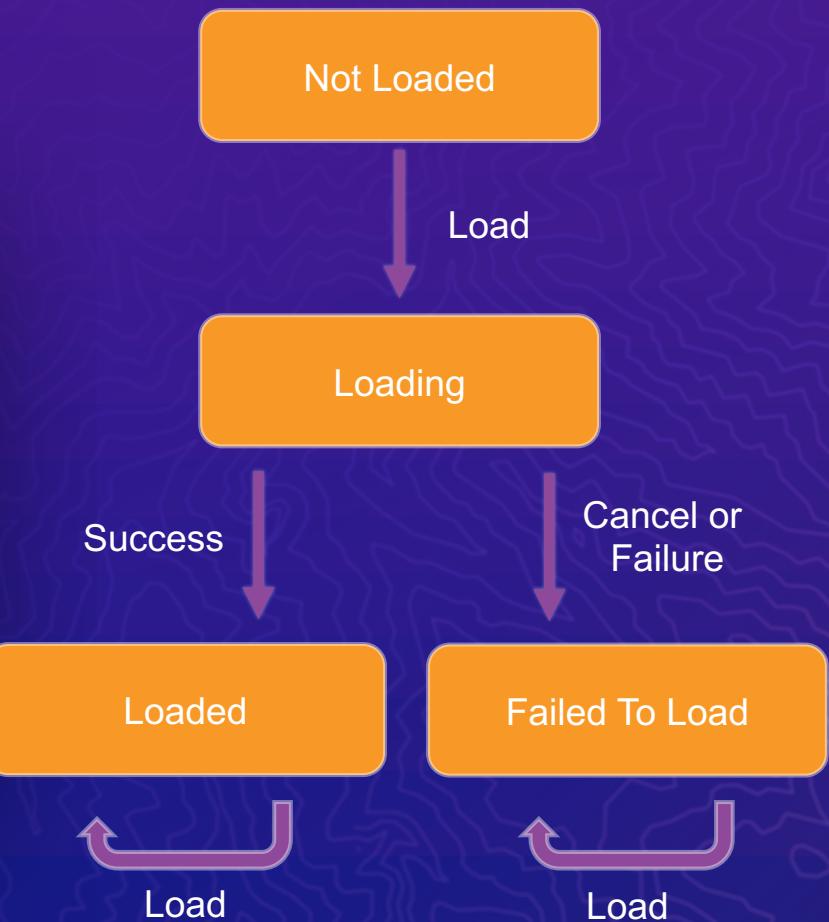
Promise = representation of a future value returned from an asynchronous task



```
1 query.executeQueryJSON(queryUrl, {  
2   where: "POP > 1000000"  
3 }).then((results)=>{  
4   // Do something with the returned features  
5   const features = results.features;  
6 }, (error)=>{  
7   // will print error in console, if any  
8   console.log(error);  
9 });
```

Loadable

```
1 // Create a FeatureLayer object from a portal item
2 const layer = new FeatureLayer({
3   portalItem: {
4     id: "d257743c055e4206bd8a0f2d14af69fe"
5   }
6 });
7
8 // Load the layer to change the popupTemplate's outFields
9 layer.load().then(()=>{
10   layer.popupTemplate.outFields = ['Colleges']
11 });
12
```



Reacting to Changes

- Watch when values change
- React when expressions become truthy
- Add/remove event listeners for a given event

```
1 // React when an event occurs
2 // Watch for a when the view changes and add a
3 // click event listener
4 reactiveUtils.on(
5   () => view,
6   "click",
7   (event) => {
8     console.log("Click event emitted: ", event);
9   }
10 );
```

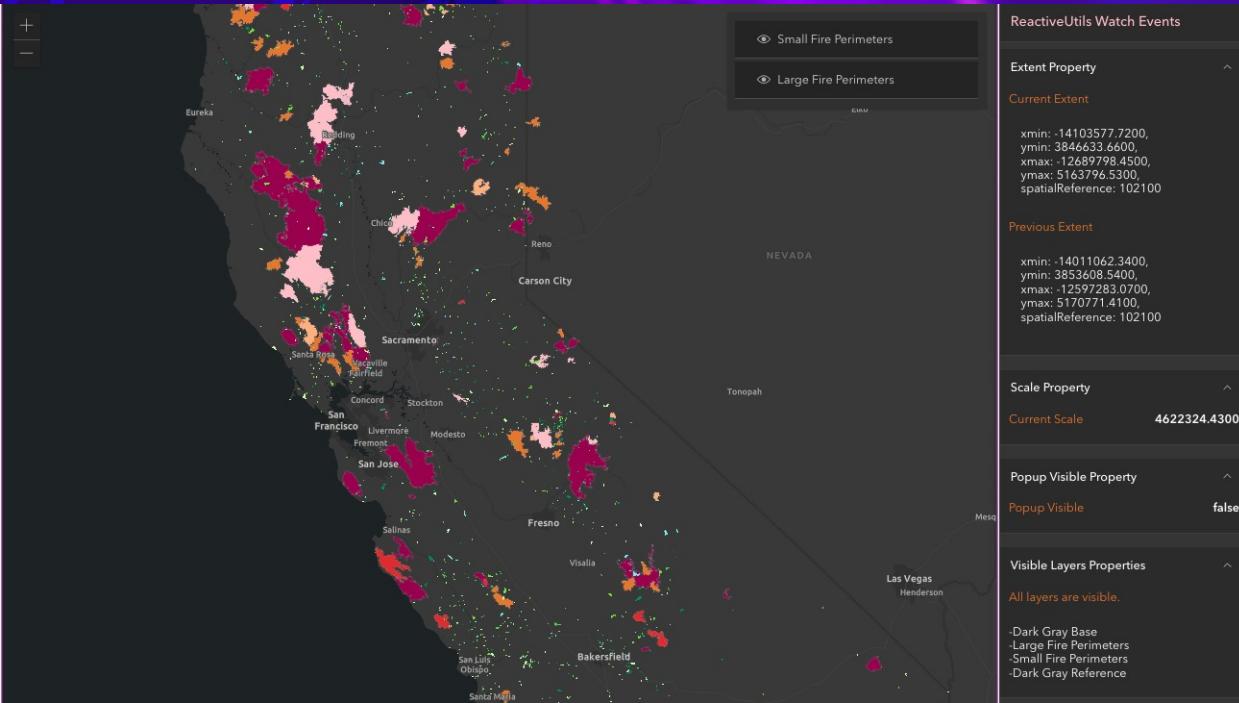
Reacting to Changes



```
1 // React to multiple property changes
2 reactiveUtils.watch(
3   () => [view.stationary, view.scale],
4   ([stationary, scale]) => {
5     console.log(`View is stationary: ${stationary} and scale is ${scale}`);
6   });
7
8 // React to Collection changes
9 reactiveUtils.watch(
10  () => view.map.layers((layer) => layer.title),
11  (titles) => {
12    console.log(`Layer titles changed! New titles: ${titles}`);
13  });

```

- Combine values from multiple sources
- Watch for changes within a Collection



reactiveUtils

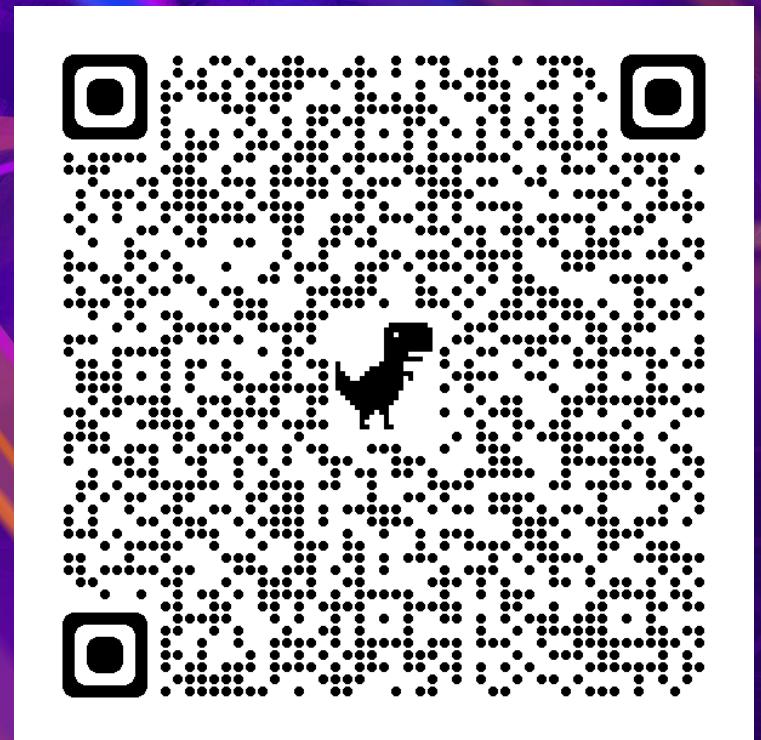
Demo

Where can I learn more?

- [SDK Documentation](#)
- [Programming Patterns Guide](#)
- [Esri Blogs](#)
- [Esri Community](#)

Presentation Resources

<https://links.esri.com/UC2023/programming-patterns>

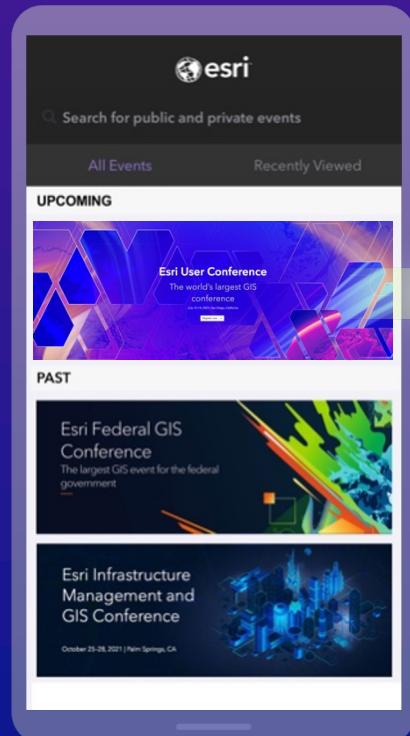


Upcoming Sessions

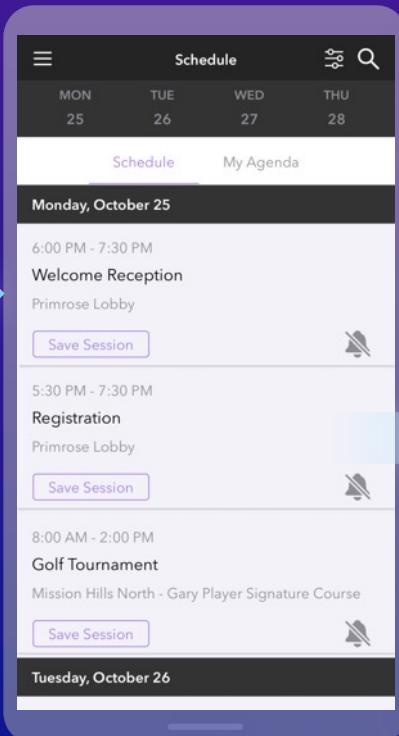
Session	Day/Time	Location
ArcGIS Maps SDK for JavaScript: Data Visualization	Thursday, 10:00AM – 11:00AM	Room 2
ArcGIS Maps SDK for JavaScript: Building 3D Web Apps	Thursday, 11:30AM – 12:15PM	Expo Demo Theater 13
Building Web Apps with Calcite Design System	Thursday, 1:00PM – 2:00PM	Room 14 B
ArcGIS Maps SDK for JavaScript: What's New and the Road Ahead	Thursday, 2:30PM – 3:30PM	Room 1 AB

Please Share Your Feedback in the App

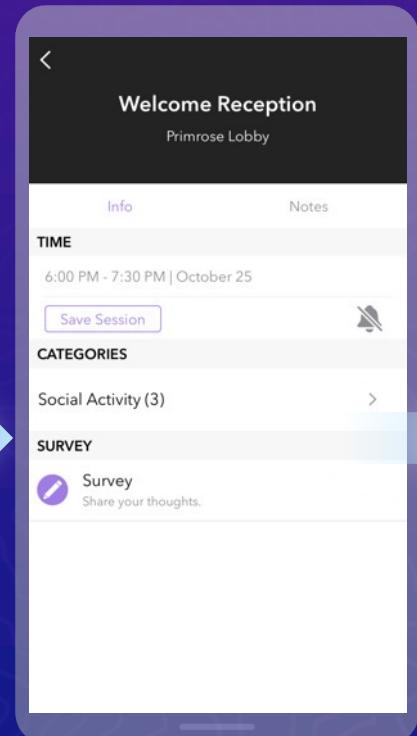
Download the Esri Events app and find your event



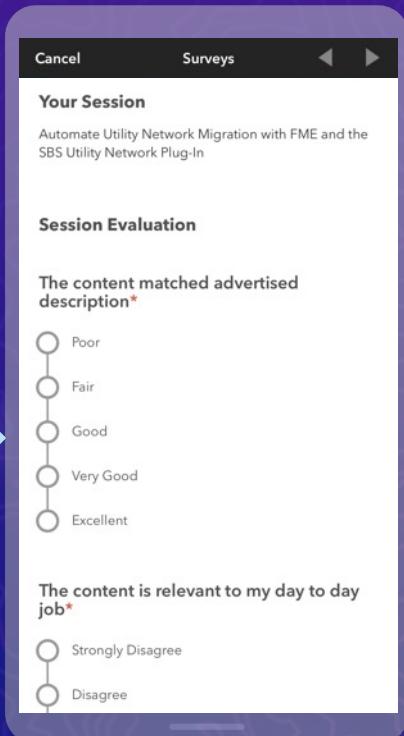
Select the session you attended



Scroll down to "Survey"



Log in to access the survey





esri®

THE
SCIENCE
OF
WHERE®