



ESRI DEVELOPER SUMMIT

10-12 November | Berlin, Germany



ArcGIS API for JavaScript

An Introduction

Yann Cabon – Esri US - @yanncabon
Raul Jimenez – Esri ES - @hhkaos

[View Demos](#)

Why

- Interactive maps
- Widgets
- Analysis
- Lots of samples

ArcGIS for Developers [FEATURES](#) [PLANS](#) [DOCUMENTATION](#) [COMMUNITY](#) [SEARCH](#)

ArcGIS API for JavaScript

Home Guide API Reference Sample Code

Use the API

Reference the ArcGIS JavaScript API from our CDN and you are ready to get started:

```
<link rel="stylesheet" href="http://js.arcgis.com/3.14/esri/css/esri.css">
<script src="http://js.arcgis.com/3.14/"></script>
```

Your first map



```
require(["esri/map", "dojo/domReady!"], function(Map) {
    var map = new Map("map");
    map.setCenter(-118.24, 34.05);
    map.setZoom(8);
});
```

[View Overview](#)

ArcGIS Integration
Use ArcGIS.com or your own on-premises ArcGIS Server. The API provides tools to make working with both simple.

Any Screen, Any Browser
Whether it's mobile devices or desktops, the API is designed to work in a wide variety of situations.

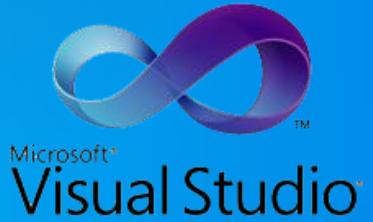
HTML5 and CSS3
Leverage the latest HTML 5 and CSS 3 standards to increase the flexibility and performance of your mapping applications.

js.arcgis.com

#DevSummit

Developer Setup

- IDE (s)
- Code Assist
- Plugins



#DevSummit



Build sample app

Widgets, Events, Graphics, Selection



#DevSummit

Get the API

- Hosted

```
<head>
  <link rel="stylesheet" href="http://js.arcgis.com/3.14/esri/css/esri.css">
  <script src="//js.arcgis.com/3.14/"></script>
</head>
```

- Download: <http://developers.arcgis.com/en/downloads>

Class: Map

[API Reference Overview](#)
[Set up a development environment](#)
[Preferred Argument Aliases](#)

esri

[basemaps](#)

[Color](#)

[config](#)

[Credential](#)

[domUtils](#)

[Graphic](#)

[graphicsUtils](#)

[IdentityManager](#)

[IdentityManagerBase](#)

[ImageSpatialReference](#)

[InfoTemplate](#)

[InfoWindowBase](#)

[kernel](#)

[lang](#)

[Map](#)

[OperationBase](#)

[request](#)

[ServerInfo](#)

[SnappingManager](#)

[SpatialReference](#)

[TimeExtent](#)

[UndoManager](#)

[AMD Module Require | Legacy Module Require]

[Constructors](#) | [CSS](#) | [Data Attributes](#) | [Properties](#) | [Methods](#) | [Events](#)

```
require(["esri/map"], function(Map) { /* code goes here */ });
```

Samples

Search for [samples](#) that use this class.

Constructors

Name	Summary
new Map(divId, options?)	Creates a new map inside of the given HTML container, which is often a DIV element.

CSS

Name	Description
esriAttribution	Represents the map attribution node. <pre>esriAttribution{ color:#B20000 !important; font-weight:bold; }</pre>

[esriAttributionOpen](#)

	Represents the map attribution node when it is open i.e., it is clicked. Use this class to define CSS rules that apply to the attribution only when it is open. <pre>.esriAttributionOpen{ font-style:italic; }</pre>
--	--

Demo

Make a map

Loading Modules

```
require([
    "esri/map",
    "dojo/domReady!"
], function(Map) {
    map = new Map("map", {
        center: [-70.6508, 43.1452],
        zoom: 16,
        basemap: "topo"
    });
});
```

Loading Modules

- Preferred arg alias

Module Path	Return Type	Preferred Arg Alias	Version Added
Folder: esri			
esri/basemaps	object	esriBasemaps	3.12
esri/Color	class	Color	3.9
esri/config	object	esriConfig	3.8
esri/Credential	class	Credential	2.5
esri/domUtils	object	domUtils	3.8
esri/graphic	class	Graphic	1.0
esri/graphicsUtils	object	graphicsUtils	3.8



#DevSummit

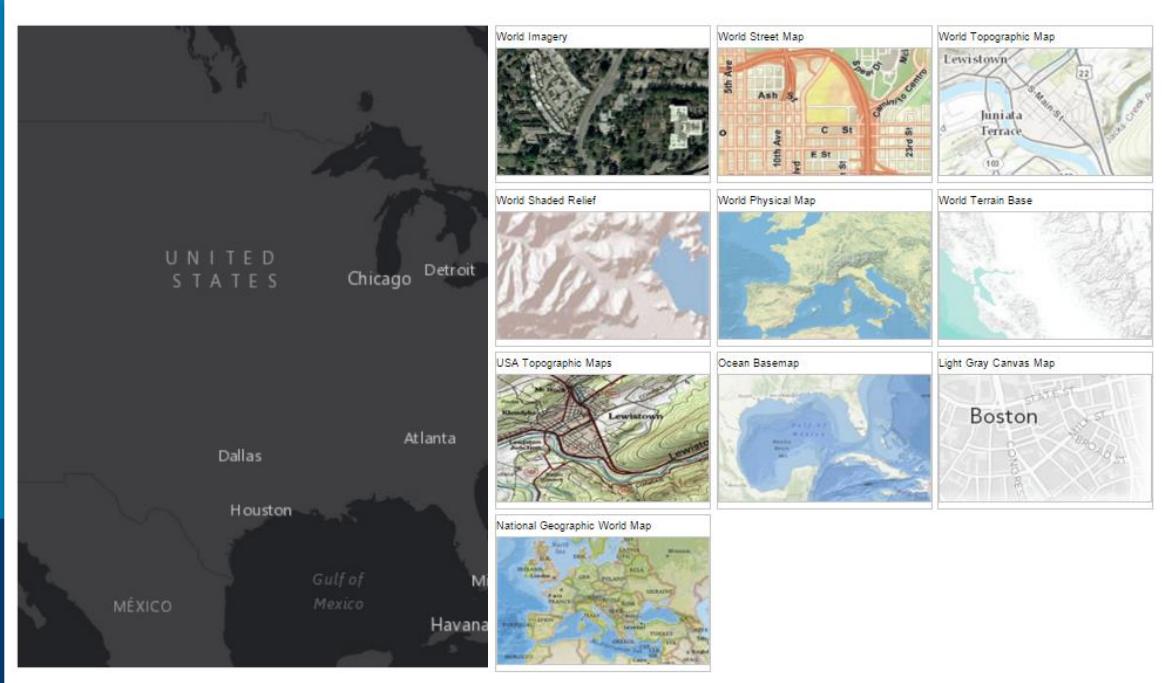
```
32 require([
33   "esri/map",
34   "esri/graphic",
35   "esri/symbols/SimpleFillSymbol",
36   "esri/layers/FeatureLayer",
37   "esri/Color",
38   "esri/renderers/UniqueValueRenderer",
39   "esri/symbols/SimpleMarkerSymbol",
40   "esri/dijit/PopupTemplate",
41   "esri/geometry/geometryEngine",
42   "esri/geometry/geometryEngine",
43   "dgrid/OnDemandGrid",
44   "dojo/store/Memory",
45   "dojo/_base/array",
46   "esri/tasks/query",
47   "esri/domUtils",
48   "dojo/dom",
49   "dojo/on",
50   "dojo/domReady!"
51 ], function(
52   Map,
53   Graphic,
54   SimpleFillSymbol,
55   FeatureLayer,
56   UniqueValueRenderer,
57   SimpleMarkerSymbol,
58   Color,
59   PopupTemplate,
60   geometryEngine,
61   geometryEngine,
62   Grid,
63   Memory,
64   array,
65   Query,
66   domUtils,
67   dom,
68   on
69 ) {
```



Scrambled Tiles



Basemaps



#DevSummit

Developer Console

The screenshot shows a browser developer console window with the "Console" tab selected. At the top, there's a warning message: "⚠ The XMLHttpRequest progress event property 'position' is deprecated. Please use 'loaded' instead." Below this, the console displays a stack trace and a detailed object inspection for a point geometry.

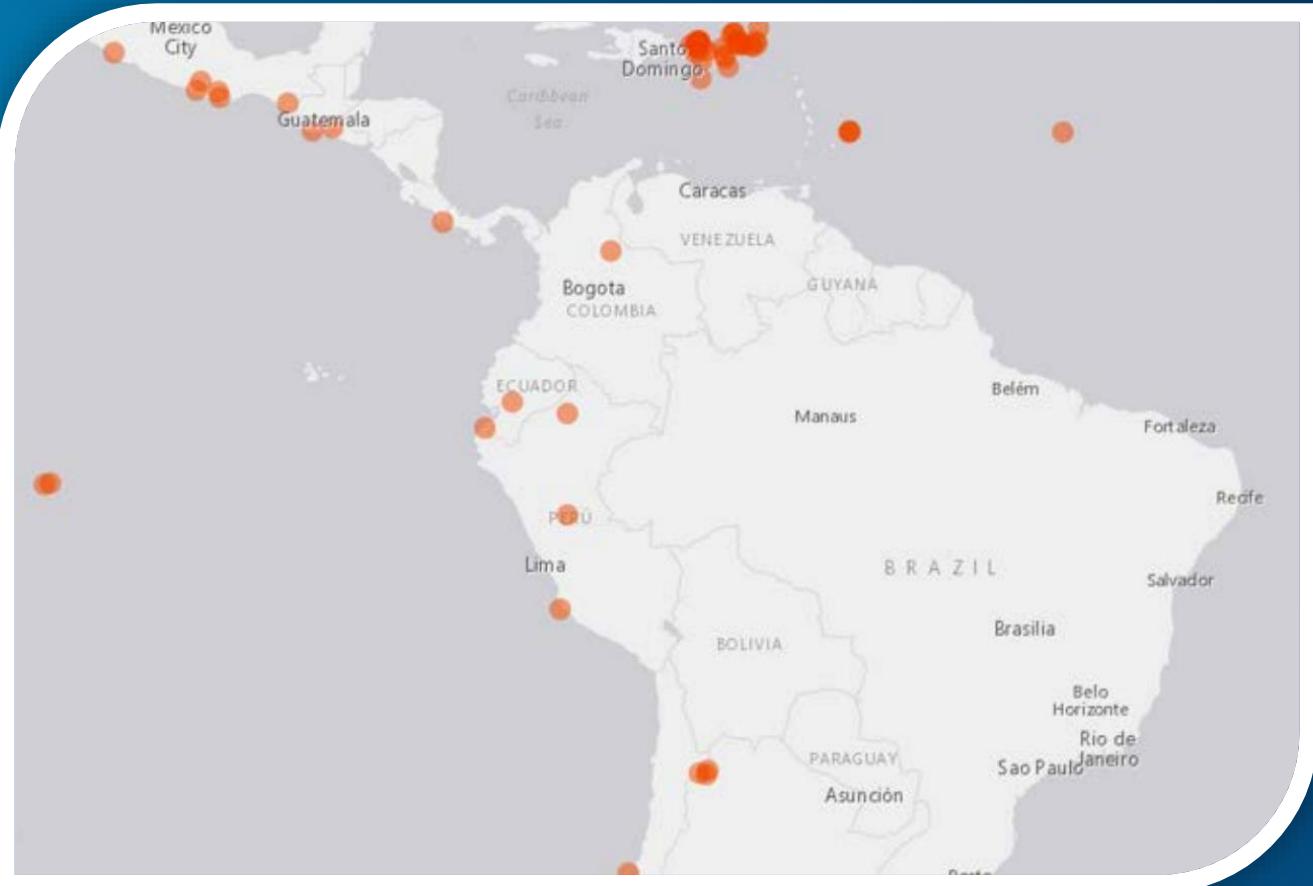
```
⚠ The XMLHttpRequest progress event property 'position' is deprecated. Please use 'loaded' instead.  
> map.extent.getCenter()  
<- ▼Object {type: "point", x: -13795824.494010163, y: 5932286.038368573, spatialReference: Object} ⓘ  
  ► spatialReference: Object  
  ► type: "point"  
  ► x: -13795824.494010163  
  ► y: 5932286.038368573  
  ▼__proto__: Object  
    ► __inherited: function n(a,c,b)  
    ► constructor: function ()  
    ► declaredClass: "esri.geometry.Point"  
    ► getInherited: function q(a,c)  
    ► getLatitude: function ()  
    ► getLongitude: function ()  
    ► inherited: function n(a,c,b)  
    ► isInstanceOf: function k(a)  
    ► normalize: function ()  
    ► offset: function (a,b)  
    ► setLatitude: function (a)  
    ► setLongitude: function (a)  
    ► setX: function (a)  
    ► setY: function (a)  
    ► toJson: function ()  
    ► update: function (a,b)  
    ► __proto__: Object  
> map.extent.getCenter().getLatitude()  
<- 46.93999999999999
```



Add Layers

- **Lots of types**

- Tiled
- Dynamic
- Graphics
- **Feature layers**
- KML
- WMS/WMTS
- Custom layers



Layer coding pattern

1. Load module
2. Create layer
3. Specify properties
4. Add layer to map

```
require([
  "esri/map",
  "esri/layers/FeatureLayer",
  "dojo/domReady!"
], function(Map, FeatureLayer) {
  ... // Map already loaded
  csv = new CSVLayer(csvUrl, {
    columnDelimiter: ",",
    latitudeFieldName: "lat",
    longitudeFieldName: "long"
  });
  map.addLayer(csv);
});
```

Demo: Add feature layer to application



Deep Dive:

- Selection ...
- Query
- Edit
- Renderer

Class: FeatureLayer

[Print this page](#)

[AMD Module Require | Legacy Module Constructors | Constants | Properties | Methods | Events]

```
require(["esri/layers/FeatureLayer"], function(FeatureLayer) { /* code goes here */ });
```

Samples

Search for [samples](#) that use this class.

Class hierarchy

```
esri/layers/Layer
|_esri/layers/GraphicsLayer
  |_esri/layers/FeatureLayer
```

Subclasses

[CSVLayer](#) [DataAdapterFeatureLayer](#) [StreamLayer](#)

Constructors

Name	Summary
new FeatureLayer(url, options?)	Creates a new instance of a feature layer object from the ArcGIS Server REST resource identified by the input URL.
new FeatureLayer(featureCollectionObject, options?)	Creates a new instance of a feature layer using a FeatureCollection object.

Constants

Name	Description

Renderers, Symbols, Visualization

BlendRenderer – Lawrence, KS by age and population size

BlendRenderer – Los Angeles housing status

BlendRenderer – Minority demographics

Class breaks renderer

Color ramp symbols

Color ramp with rotation and proportional symbols

Dot density renderer

Explore Heat Maps

Generate renderer

Opacity based unique value renderer

Proportional symbols for lines

Proportional symbols for points

Renderer using a function

Scale dependent renderer

Smart Mapping – Render class breaks with color

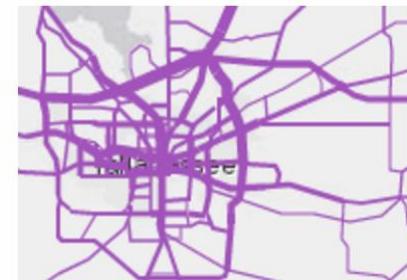
Smart Mapping – Show Data by Color

Symbol size and outline width based on map scale

Temporal renderer

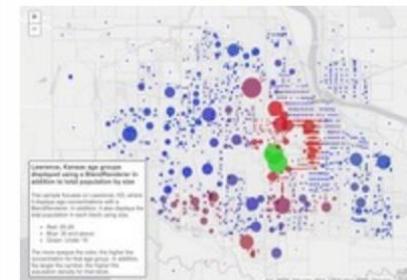
Unique value renderer

Renderers, Symbols, Visualization category: 20 samples

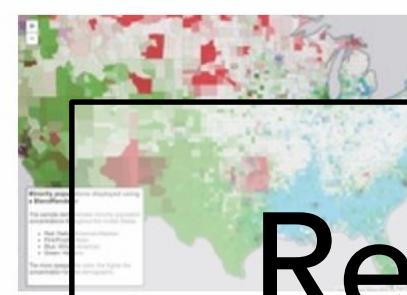


Proportional symbols for lines:

Renderer with proportional symbols representing traffic volume.

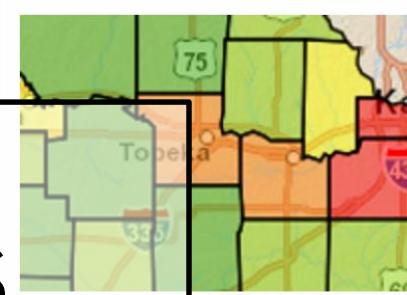


BlendRenderer – Lawrence, KS by age and population size: This sample focuses on Lawrence, KS, where it displays age concentrations with a BlendRenderer. In addition, it also displays the total population in each block using size.

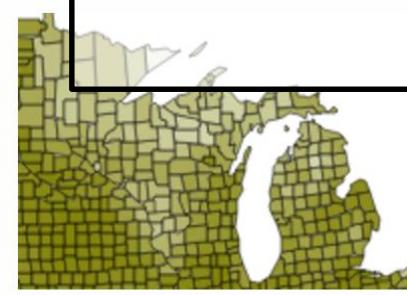


BlendRenderer – Minority demographics:

This sample demonstrates minority population concentrations throughout the United States.

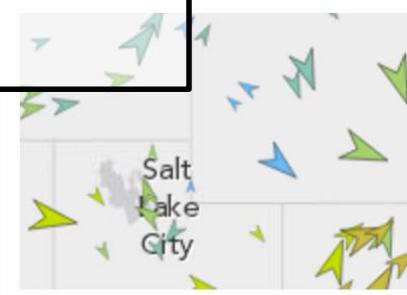


Class breaks renderer: Create a class breaks renderer to symbolize Kansas counties by population density.

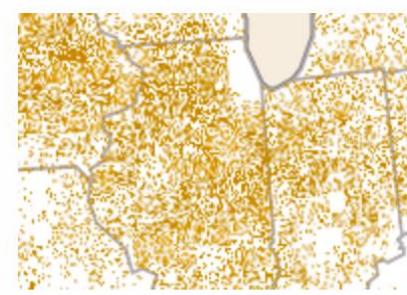


Color ramp symbols:

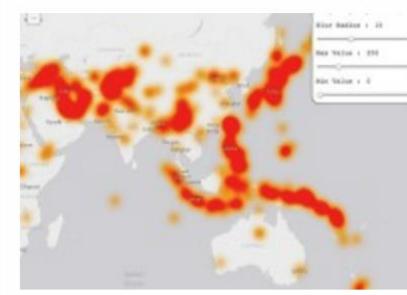
Use Renderer.setColorInfo() to render a layer with color ramp.



Color ramp with rotation and proportional symbols: Show temperature, wind speed and wind direction using a single renderer.



Dot density renderer: Create a dot density map showing corn production in the US.



Explore Heat Maps: Play with the properties to get the best heat map for a specific feature layer.

Demo: Apply renderer



[Hide Table of Contents](#)

Popups and Info Windows category: 8 samples

Popups

Samples Overview

Latest Samples

Analysis

Data Reviewer

Add interactivity
Directions and Routing

Dynamically

Information about ...
A location

Feature Layers

A feature

HTML5 - The results of a
searchMap
Mobile
Customizable

Online and Portal

Popups and Info Windows

Custom info window

Format info window content

Geoprocessing tool link in popup

Info Window Lite

Mobile popup

Popup

Popup content in side panel

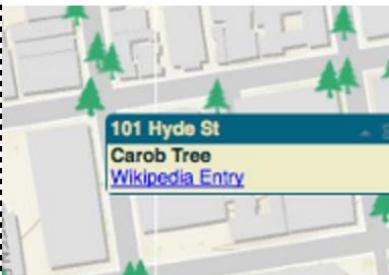
Popup with related fields

> Printing

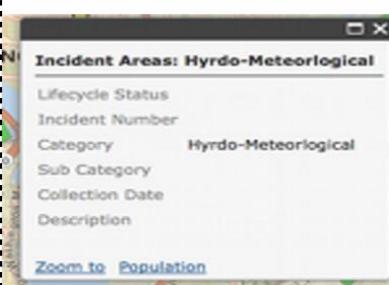
> Query and Select

> Renderers, Symbols, Visualization

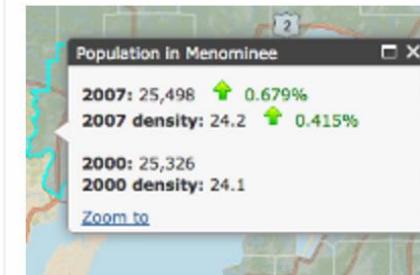
Popups



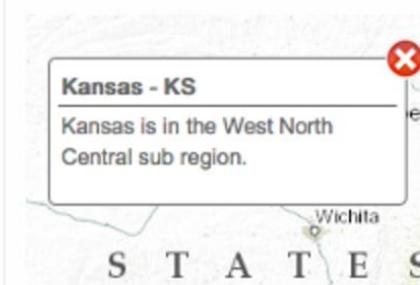
Custom info window: Use the InfoWindowBase to extend the info window create an info window with custom behavior and appearance.



Geoprocessing tool link in popup: Add a link to a geoprocessing tool to a popup window. The geoprocessing task determines the population within the currently selected area.



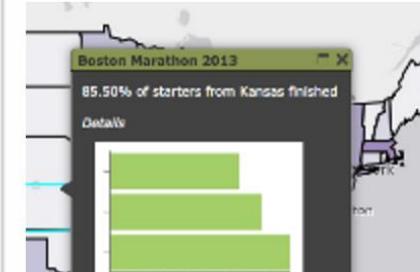
Format info window content: Define a custom formatting function that adds a gain or loss image next to the text and colors the text: red for population loss and green for population gain.



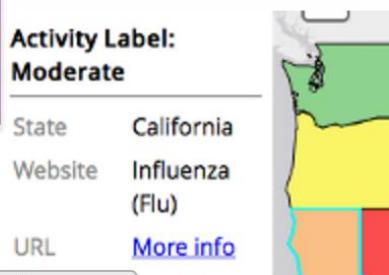
Info Window Lite: Display popup content using the InfoWindowLite



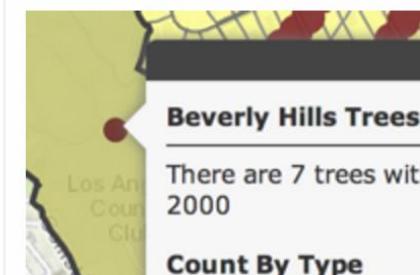
Mobile popup: Replace the map's info window with the mobile popup. The mobile popup window is designed to work well on mobile devices.



Popup: Display layer details using a popup window.



Popup content in side panel: Show feature details in a side panel.



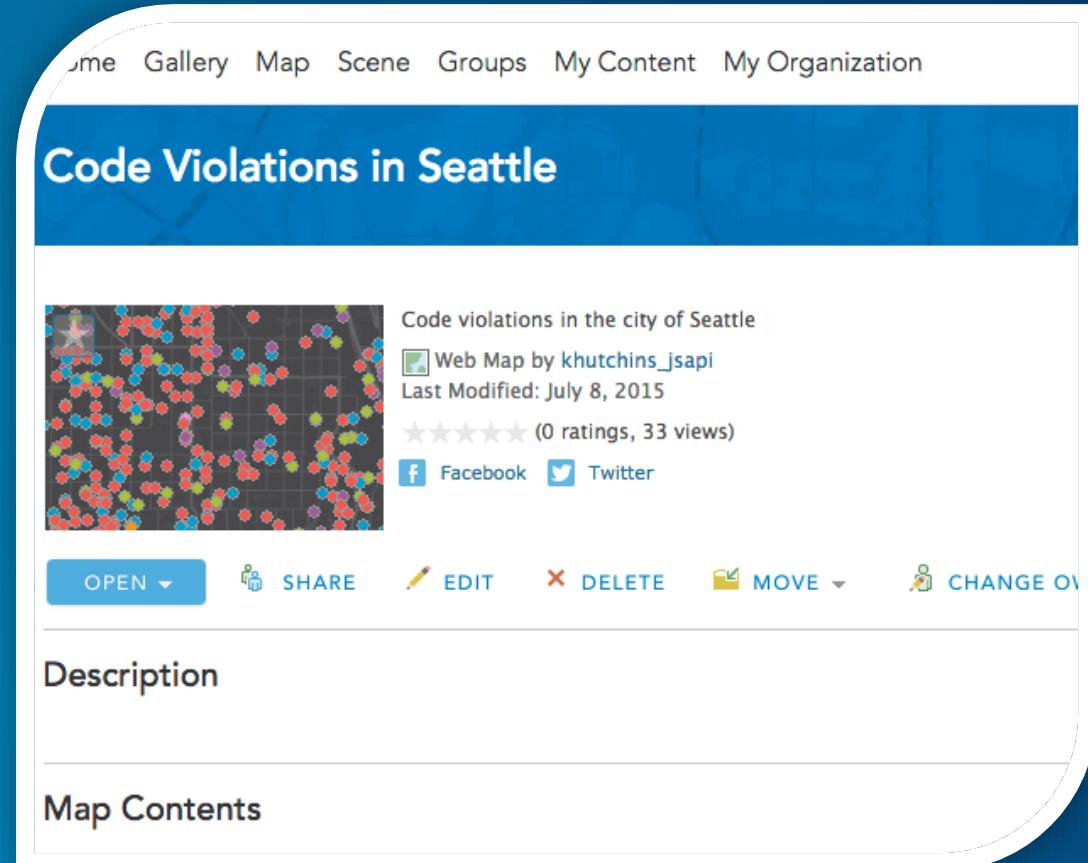
Popup with related fields: Create and use a popup template that references fields from a relationship.

Demo: Make map interactive



Tip: Reduce code by using a web map

**esri/arcgis/utils
createMap**



Widgets

- Legend,
- Directions
- Elevation
- Print
- Basemap Toggle
-

Search a country to find its ecological footprint and rating.

+

-

?

Search a country to find its ecological footprint and rating.

Spain

?

Miami, Florida, United States

Boca Raton, Florida, United States

ADD DESTINATION

BY CAR BY TRUCK WALKING

SHOW MORE OPTIONS

GET DIRECTIONS

72.36 kilometers • 54 minutes

ZOOM TO FULL ROUTE

1. Start at Miami, Florida, United States

2. Go south on N Miami Ave toward W Flagler St / E Flagler St
0.15 km 1 minute

3. Bear right onto ramp to I-95
0.13 km

4. At fork keep right on I-95 N
2.33 km 2 minutes

5. At exit 4A take ramp on the right toward Miami Beach (I-195 E)
13.89 km 11 minutes

POWERED BY Esri

Widget coding pattern

- Load module
- Create widget
- Set widget properties
- Call startup

```
require([
  ....,
  "esri/dijit/BasemapToggle",
  "dojo/domReady!"
], function(Map, BasemapToggle) {

  var toggle = new BasemapToggle({
    map: map,
    basemap: "satellite"
  }, "BasemapToggle");

  toggle.startup();
});
```



Find address or place

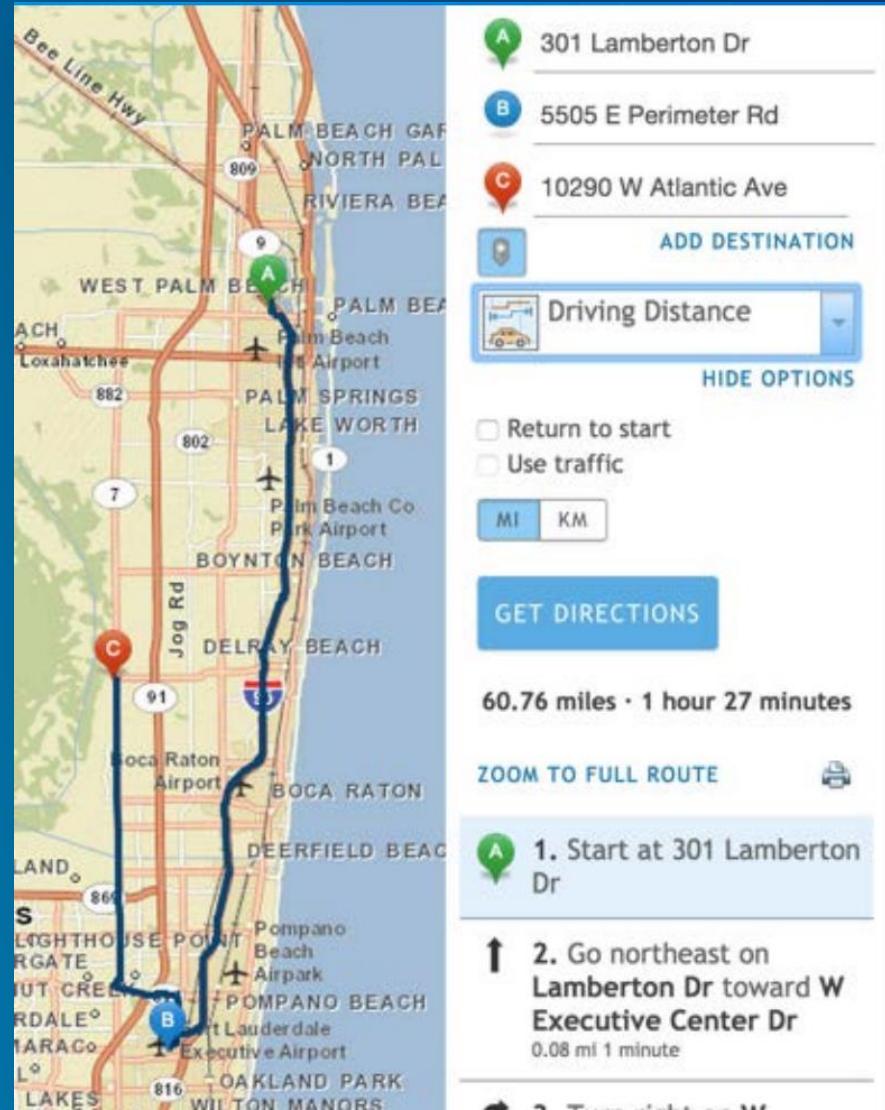


Demo: Add search Widget

Demo: More Widgets

Directions:

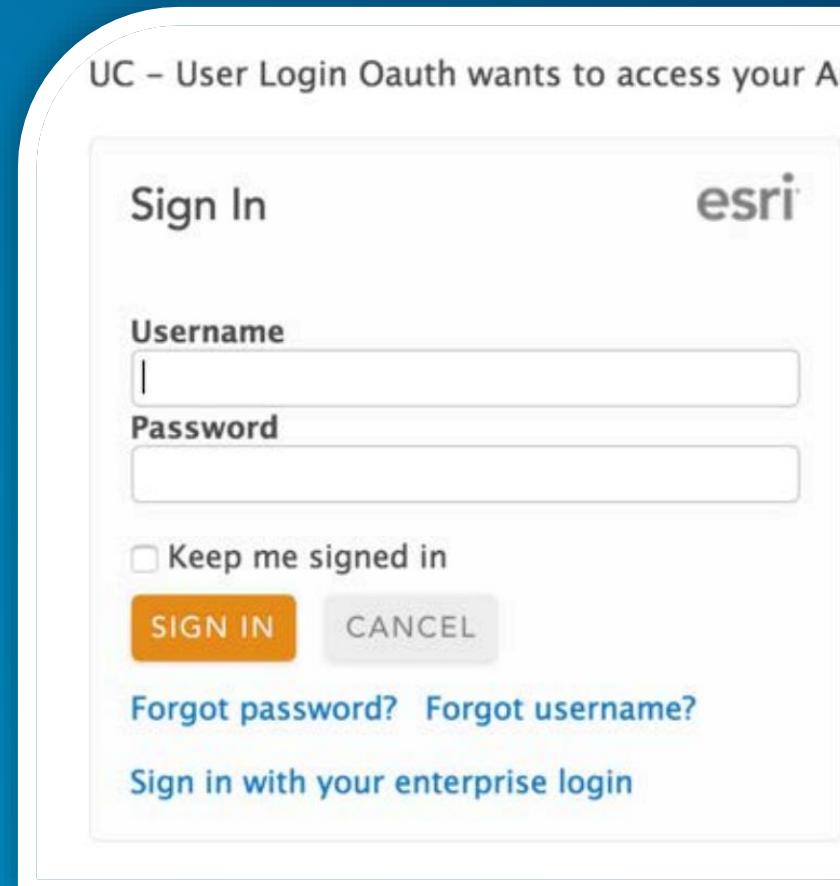
- Server
- Online



Tip: Working with secure resources

- Identity Manager
- OAuth Support
- Proxies

UC – User Login Oauth wants to access your ArcGIS Online account information



UC – User Login Oauth developed by:

 ESRI JSAPI

Team organization for the ArcGIS API for JavaScript. Apps generated by the Esri JSAPI team are examples of what you can do with the api.



Demo:
Customize
widget
appearance

Violations

- VOLUNTARY COMPLIANCE
- ENFORCED COMPLIANCE
- ADMINISTRATIVE CLOSURE
- OPEN
- CLOSED
- TRANSFERRED TO EXTERNAL AGENCY
- COMPLAINT/APPLICATN
- WITHDRAWN
- NO VIOLATION

Events

- Event handlers
 - Load page, click map, execute task, add layer
- dojo/on

```
Map.on("click", buffer);  
  
// or  
// on(map, "click", buffer);
```

Want to learn more?

- **Documentation**

- <https://developers.arcgis.com/javascript/>
- <https://dojotoolkit.org/documentation/>

- **Related Esri Training and Tutorials**

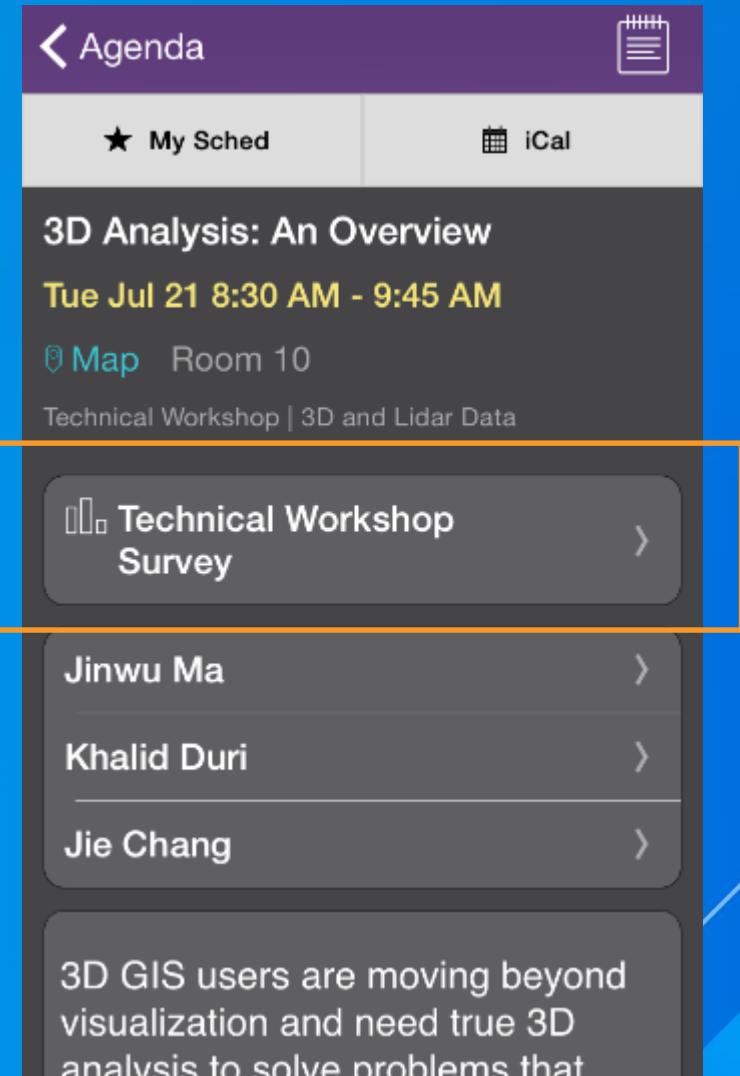
- Esri class: Building Application with the ArcGIS API for JavaScript
- Esri webinar: Data visualisation and time saving tip

- **Additional Resources**

- JavaScript online training classes: free and fee-based

Thank you...

- Please fill out the session survey in your mobile app
- Select [enter session title here] in the Mobile App
 - Use the Search Feature to quickly find this title
- Click “Technical Workshop Survey”
- Answer a few short questions and enter any comments



Thank You To Our
Gold Sponsor

con[•]terra



Understanding our world.