



Practical Guide to Building 2D Web Apps

Heather Gonzago and Kelly Hutchins



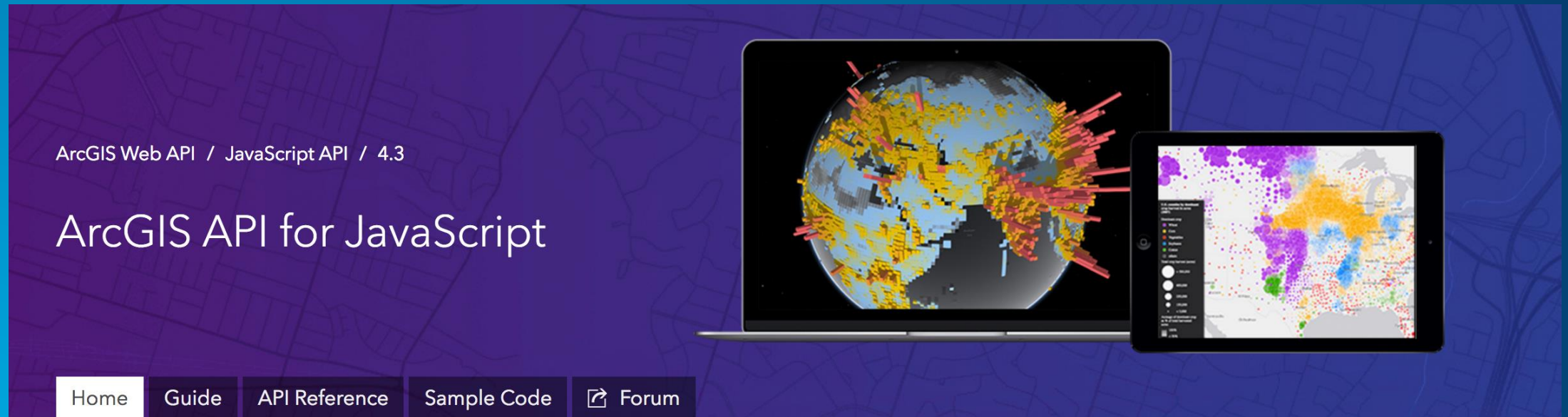
Agenda

- Setup
- First Steps
- Working with layers
- Symbols and Renderers
- Make your map interactive
- Widgets and styling

Presentations accessible on GitHub

- Focusing on version 4.0 for this presentation
- Concepts are similar between 3.x and 4.0
- <https://github.com/hgonzago/DevSummit-presentations>

Tour of the SDK



Which API version should I use?

3.x module name	4.x	Note	Version
basemaps	Not planned	See Map.basemap for list of named basemaps. Use esri/Basemap to easily create custom basemap.	4.0
Color	Color		4.0
config	config	See table for details.	4.0
Credential	Credential	Moved to <code>esri/identity</code> folder.	4.0
domUtils	Not planned		n/a
graphic	Graphic	Renamed to <code>Graphic</code> - note the uppercase <code>G</code> . See table for details.	4.0
graphicsUtils	Coming soon		
IdentityManager	IdentityManager	Moved to <code>esri/identity</code> folder.	4.0
IdentityManagerBase	IdentityManagerBase	Moved to <code>esri/identity</code> folder.	4.0
ImageSpatialReference	Coming soon		

Developer Setup

- Choose an IDE
- Code assist
- Helpful tools



Esri / jsapi-resources

Unwatch 66 Star 173 Fork 144

<> Code Issues 7 Pull requests 0 Projects 0 Wiki Pulse Graphs

A collection of useful resources for developers using the ArcGIS API for JavaScript.

107 commits 2 branches 0 releases 11 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

odoe committed on GitHub Merge pull request #58 from tomwayson/frameworks Latest commit 4b7cd5b 17 days ago

3.x	Use npm instead of typings to install arcgis-js-api.d.ts file (#49)	2 months ago
4.x	workaround for TS support in atom	a month ago
frameworks	move framework first examples to own pages and add react examples	18 days ago
jshint	Set 'laxbreak' to true	2 years ago
oauth	Update README.md (#36)	6 months ago
README.md	Reorg'ed to support 4.x	10 months ago
license.txt	Create license.txt	3 years ago

Get the API

- [CDN](#)
- Custom builds
- [Download API](#)

```
<link rel="stylesheet" href="https://js.arcgis.com/4.3/esri/css/main.css">  
<script src="https://js.arcgis.com/4.3/"></script>
```


JSAPI-Resources

- Collection of resources to aid development in the JSAPI
- Some of these include:
 - JSHint file
 - Helps to detect errors and potential problems in code
 - Typescript definition file
 - Provides code assist in some IDE's such as WebStorm and Visual Studio
 - Build tools, e.g. Bower for custom builds of the API
 - Callback page for popups using OAuth2 for security

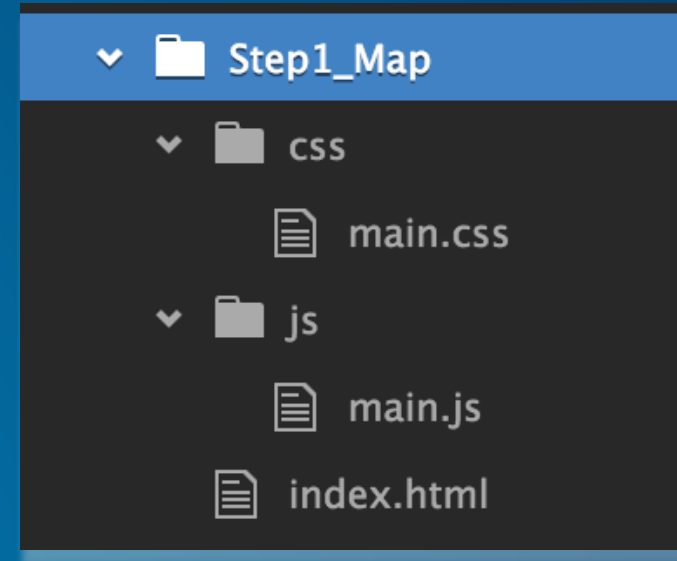
CSS

- Main.css
- View.css
- Themes
- Custom CSS(SASS)

```
<link rel="stylesheet" href="https://js.arcgis.com/4.3/esri/css/main.css">
```

First steps

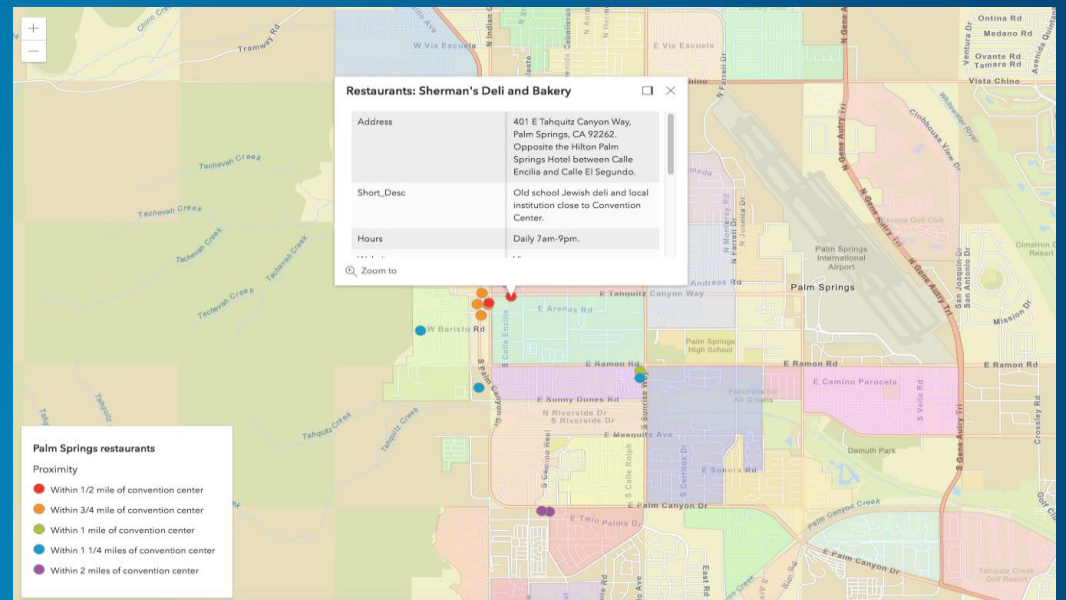
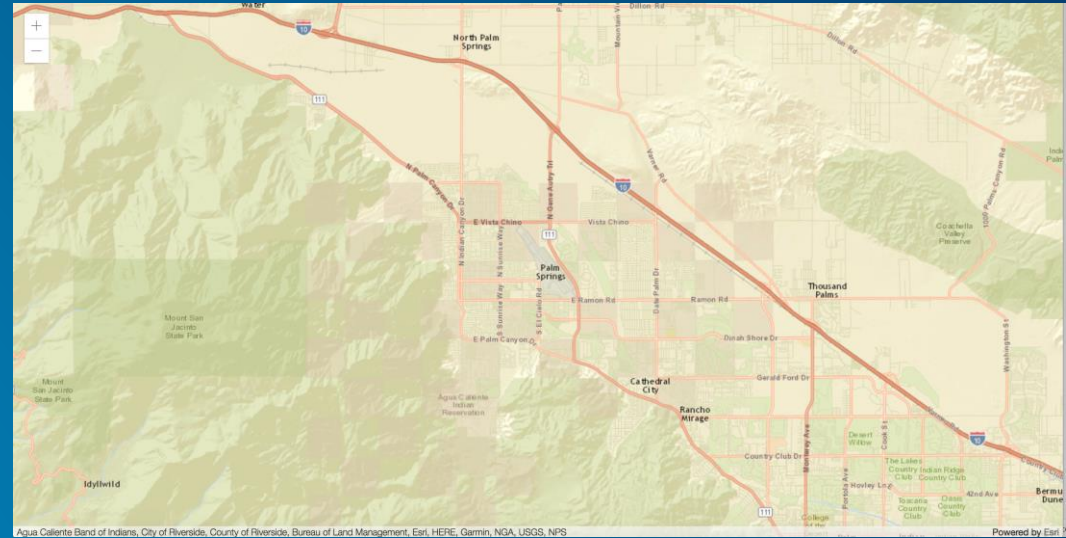
- How will application be written?
- Separate files or one combined file?
- Demos shows separate HTML, CSS, and JS files
- Reference JSAPI and any additional references, e.g. CSS, any other libraries in markup



```
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="initial-scale=1,maximum-scale=1,user-
scalable=no">
  <title>Step 1: Create a map</title>
  <link rel="stylesheet" href="https://js.arcgis.com/4.3/esri/css/main.css">
  <link rel="stylesheet" href="css/main.css">
  <script src="https://js.arcgis.com/4.3/"></script>
  <script src="js/main.js"></script>
</head>
```

Make a map

```
var map = new Map({  
  basemap: "streets"  
});
```



Map View

```
view = new MapView({  
  container: "viewDiv",  
  map: map,  
  zoom: 12,  
  center: [-117.168, 32.776]  
});
```

Add layers

- Layer types
 - FeatureLayer
 - MapImageLayer
 - ImageryLayer
 - TileLayer
 - VectorTileLayer
 - ... and more



Layer coding pattern

1. Load module
2. Create layer
3. Set properties
4. Add to map (or scene)

```
require(["esri/layers/FeatureLayer"], function(FeatureLayer){  
    // points to the states layer in a service storing U.S. census  
    var f1 = new FeatureLayer({  
        url: "https://sampleserver6.arcgisonline.com/arcgis/rest/se  
    });  
    map.add(f1); // adds the layer to the map  
});
```

Working with Properties

- All properties for each class can be set in its constructor
- No need to explicitly call Get/Set for properties

```
view.center = [ -100, 40 ];  
view.zoom = 6;  
map.basemap = 'oceans';
```

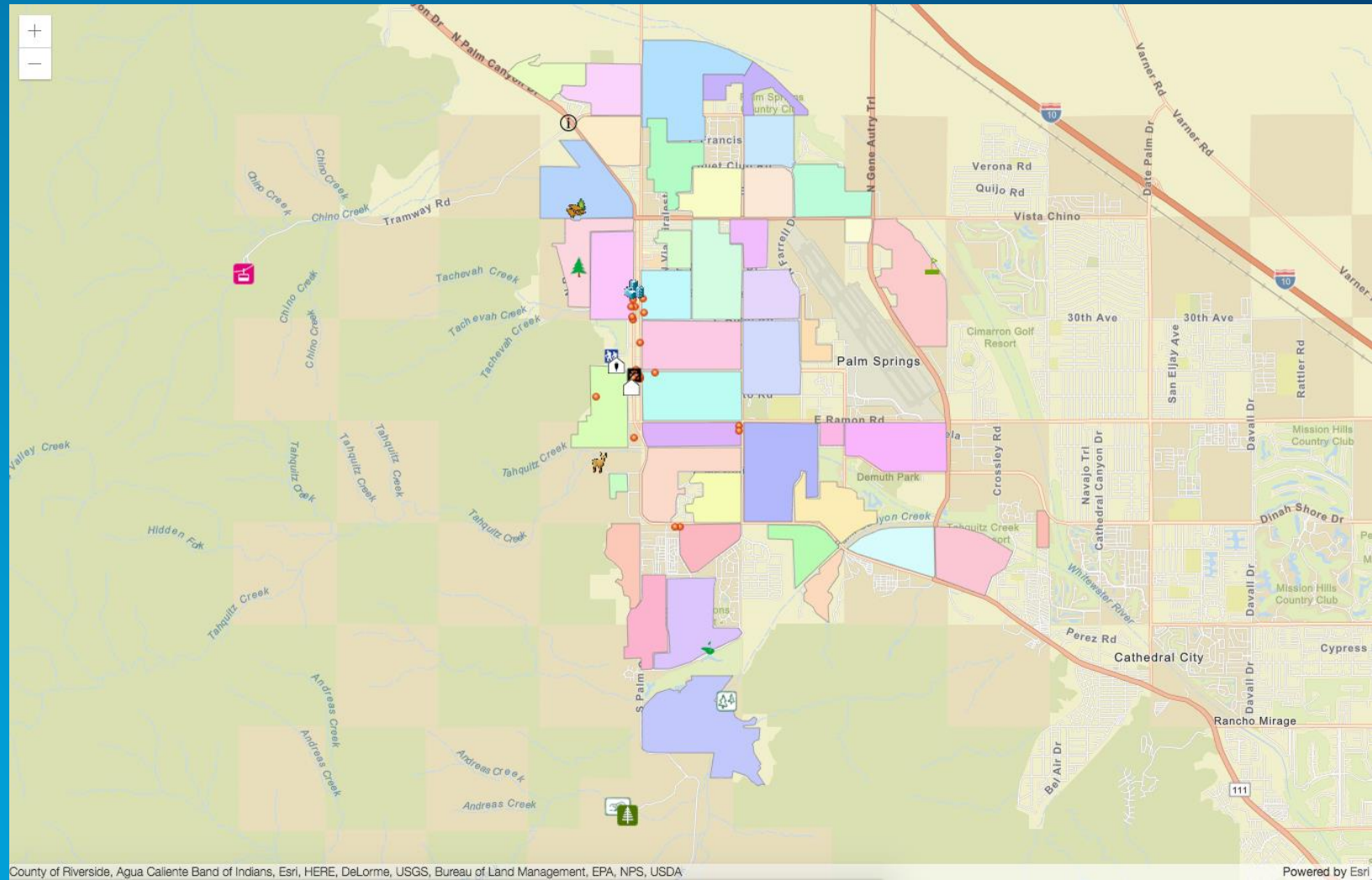
```
var map = new Map({  
  basemap: "streets-vector",  
  layers: [hoods, foodLayer]  
});
```

- Watch for property changes

```
layer.watch("loadStatus", function(status) {/**/});
```

- Use [esri/core/watchUtils](#) to aid in watching property changes

Demo: Add layer to sample app



Renderers

- Define a set of symbols to use for the layer
- Set rules on how the symbols should be used
- Basic coding pattern

```
var layerRenderer = new UniqueValueRenderer();  
var featurelayer = new FeatureLayer({  
    url: "<url to featurerlayer>",  
    // set renderer  
    renderer: layerRenderer  
});
```

Symbols

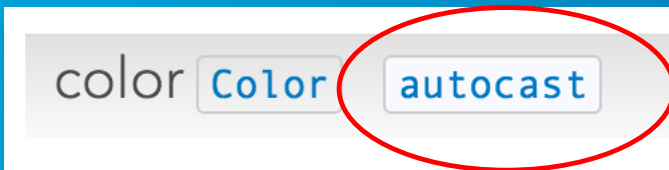
```
// Symbol for restaurants within 1/2 mile of convention center
var halfMileSymbol = new SimpleMarkerSymbol({
  size: 14,
  color: "#ff3323",
  width: 7,
  outline: { // Autocasts as new SimpleLineSymbol()
    color: [255, 255, 255, 0.50], // Autocasts as new Color()
    width: 2
  }
});
```

- Renderers make use of symbology
 - Points, lines, and polygons
- Pass the symbol to the renderer

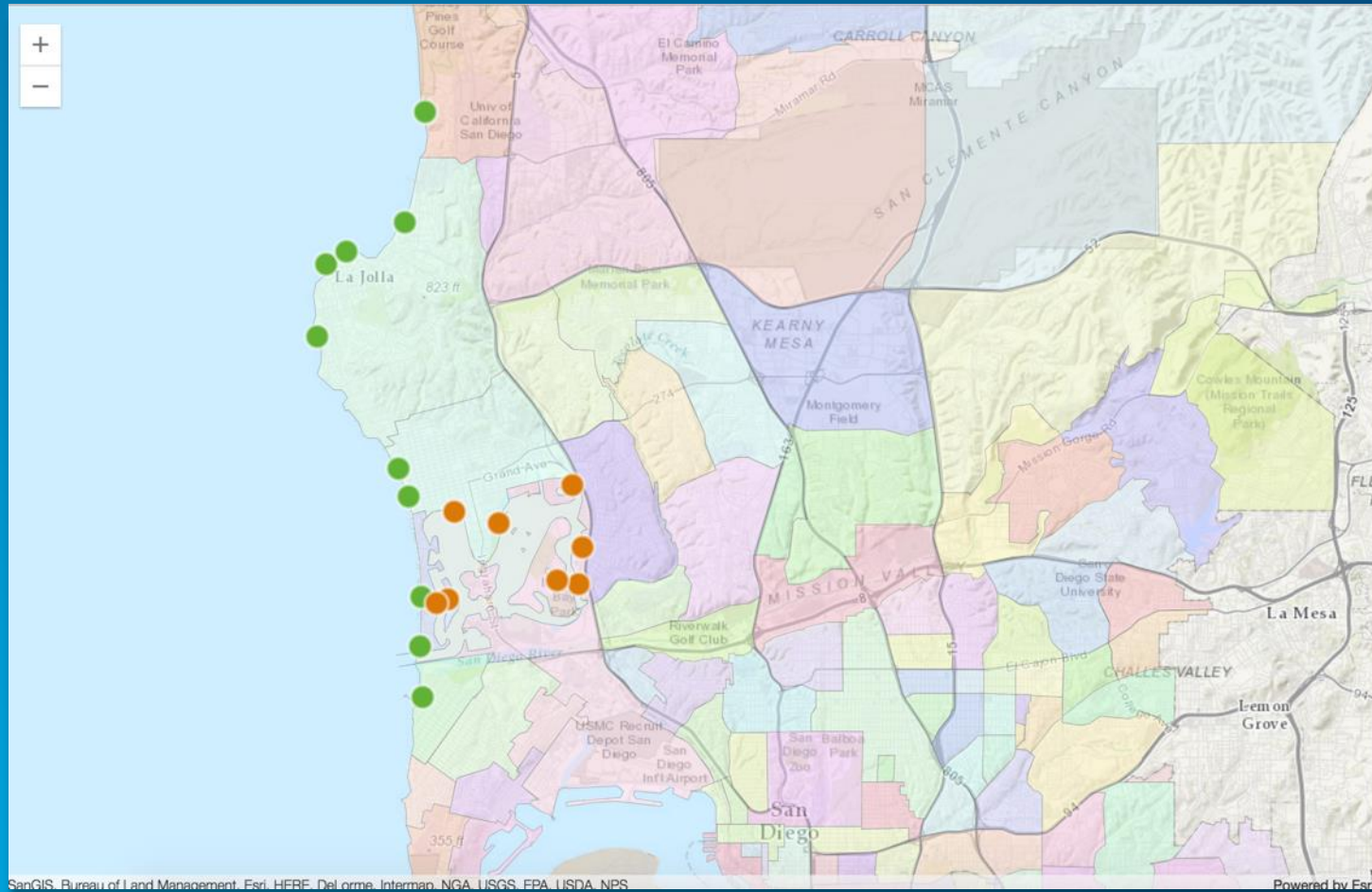
```
var foodRenderer = new UniqueValueRenderer({
  defaultSymbol: halfMileSymbol,
  defaultLabel: "Within 1/2 mile of convention center",
  field: "Proximity",
  uniqueValueInfos: [{
    value: "0.5", //attribute value for features within 1/2 mile of CC
    symbol: halfMileSymbol,
    label: "Within 1/2 mile of convention center"
  }
]);
```

Autocasting

- Allows you to pass in a property without having to specifically require it in the code.
 - Efficient
 - Less code
 - Less overhead
- Not sure if a property can use this functionality?
 - Check property's API reference for Autocast label

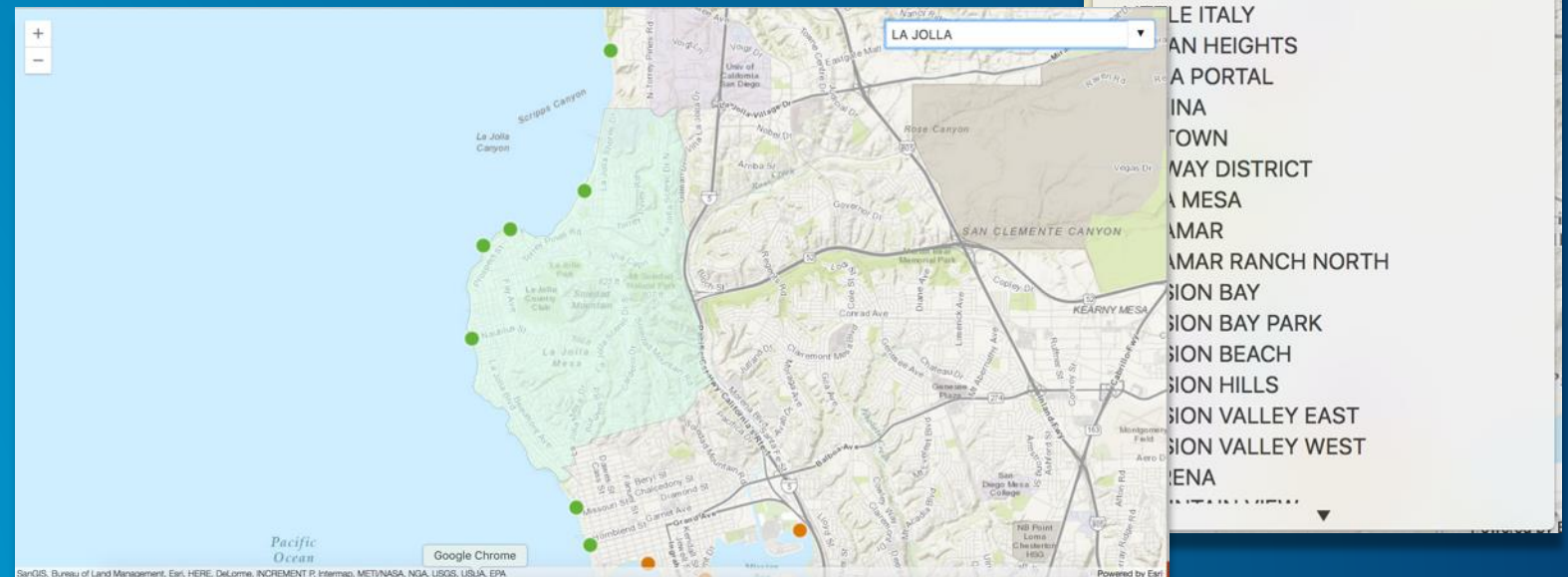
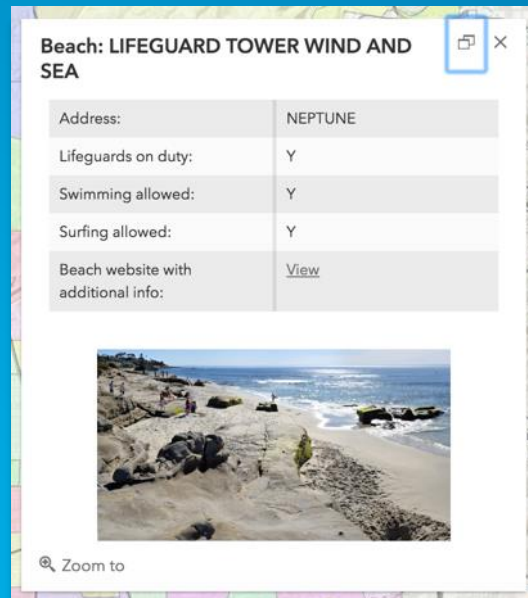


Demo: Update feature layer renderer



Interactive Map

- Make map interactive
 - Popups
 - Query
 - Filter out features based on a definition expression




Popups

- Responsive widget
- Information about
 - Clicked location
 - Feature
 - Search results
- Customizable

Beach: LIFEGUARD TOWER WIND AND SEA

Address:	NEPTUNE
Lifeguards on duty:	Y
Swimming allowed:	Y
Surfing allowed:	Y
Beach website with additional info:	View



Zoom to

FeatureLayer PopupTemplate

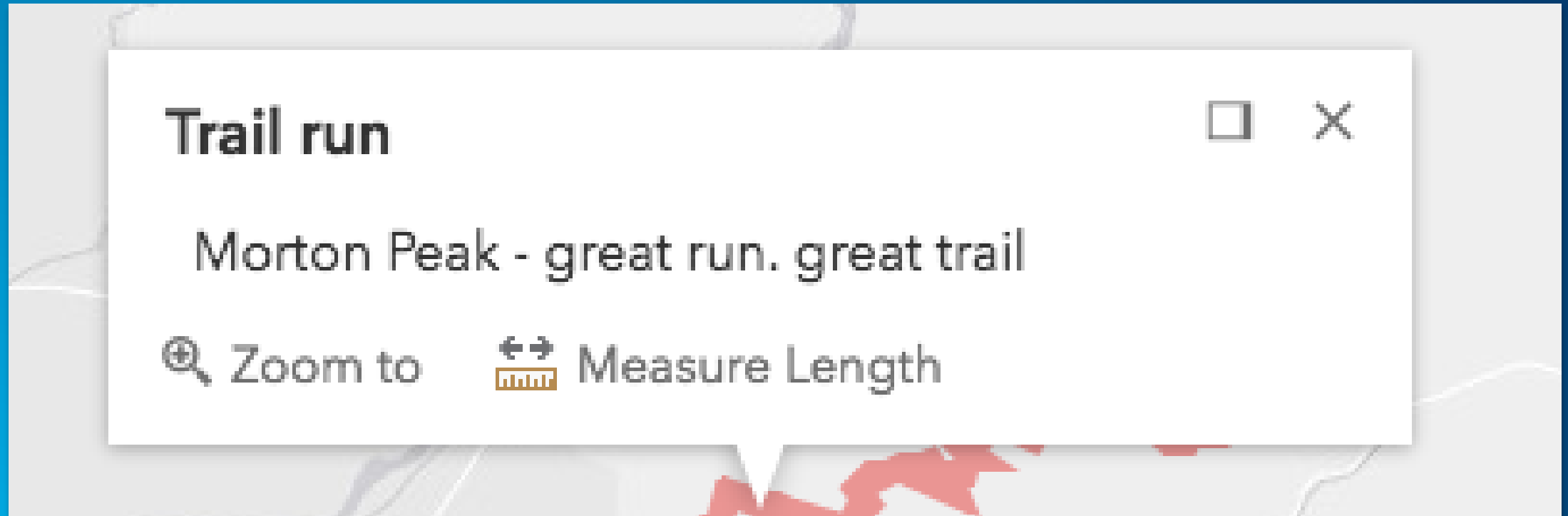
```
// Create the popuptemplate
var popupTemplate = new PopupTemplate({
  title: "<b>Beach: {NAME}</b>",
  // Specify the content, first set the display fields
  content: [{
    type: "fields",
    fieldInfos: [{
      fieldName: "ADDRESS",
      visible: true,
      label: "Address: "
    }], {
```

```
var beaches = new FeatureLayer({
  url: "<URL to featurelayer>",
  // Step 4: Specify the outfields
  outFields: ["*"],
  popupTemplate: popupTemplate,
  // set renderer
  renderer: beachRenderer
});
```

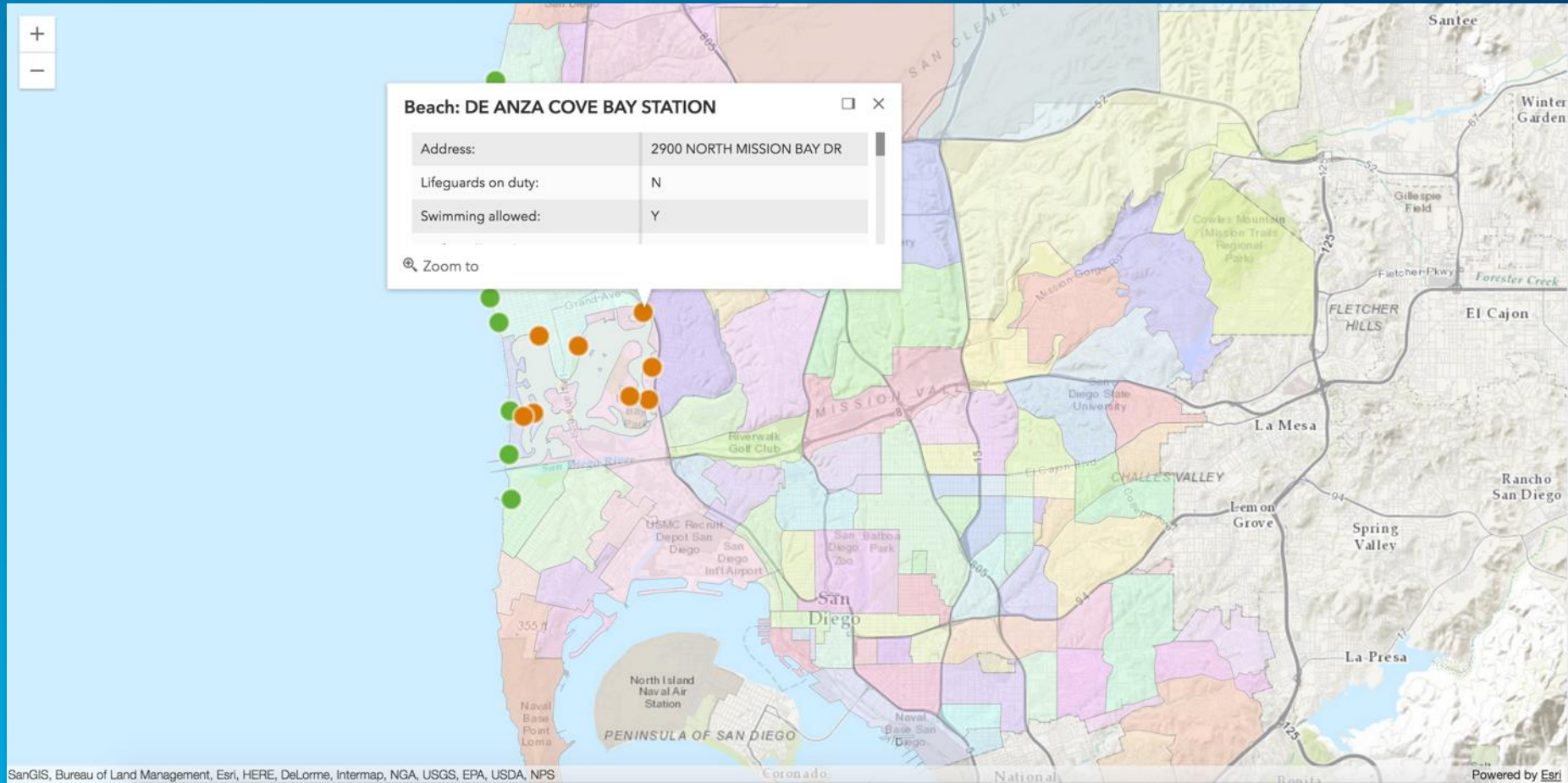
```
popup: {
  dockEnabled: true,
  dockOptions: {
    buttonEnabled: true,
    position: "bottom-right"
  }
}
```

- View has associated popup
- Popup content is set via PopupTemplate
- FeatureLayer has popupTemplate
- Popup's dockOptions

Popup: Custom Actions

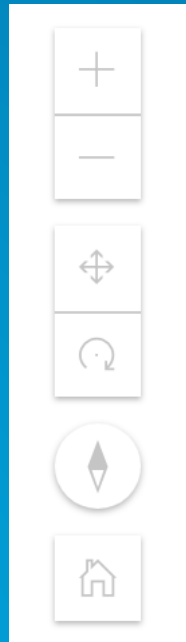
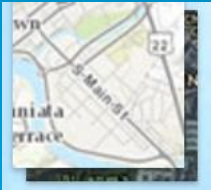


Demo: Add a popup to the map



Widgets

- Responsive widgets
- Customizable



Export

Layout

Map Only

Title

Title of file

Page setup

Letter ANSI A Landscape

File format

PDF

Advanced Options

Export

Exported Files

NY Educational Attainment

Predominant Educational Attainment

- Graduate Degree
- College Degree
- Associates Degree
- Some College
- High School Degree
- Some High School
- No High School
- Tie

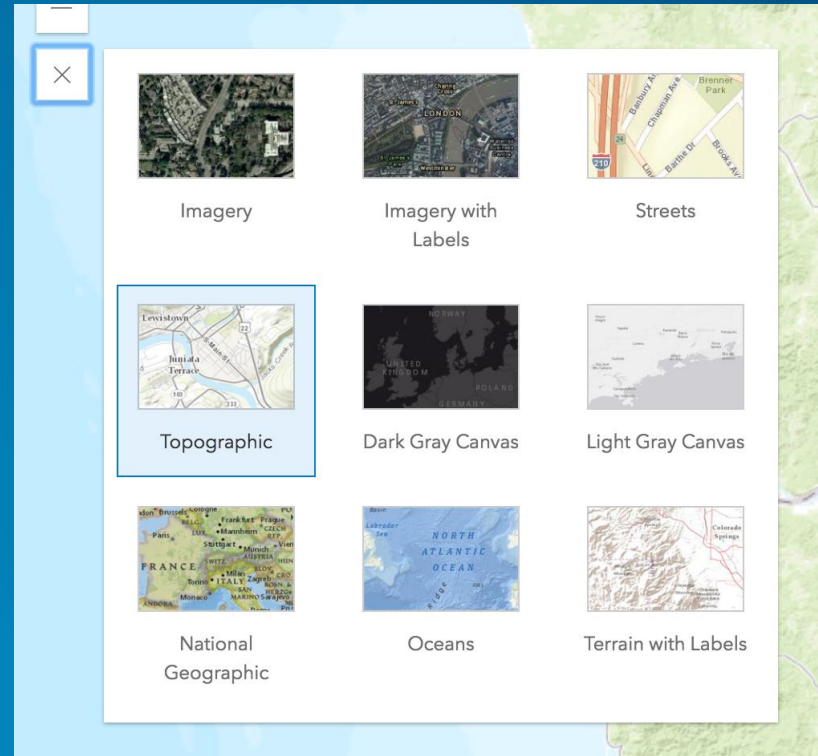
View UI

- **Position widgets**
 - **Add**
 - **Move**
 - **Remove**

```
view.ui.add([ searchWidget, "infoDiv" ], "bottom-left");
```

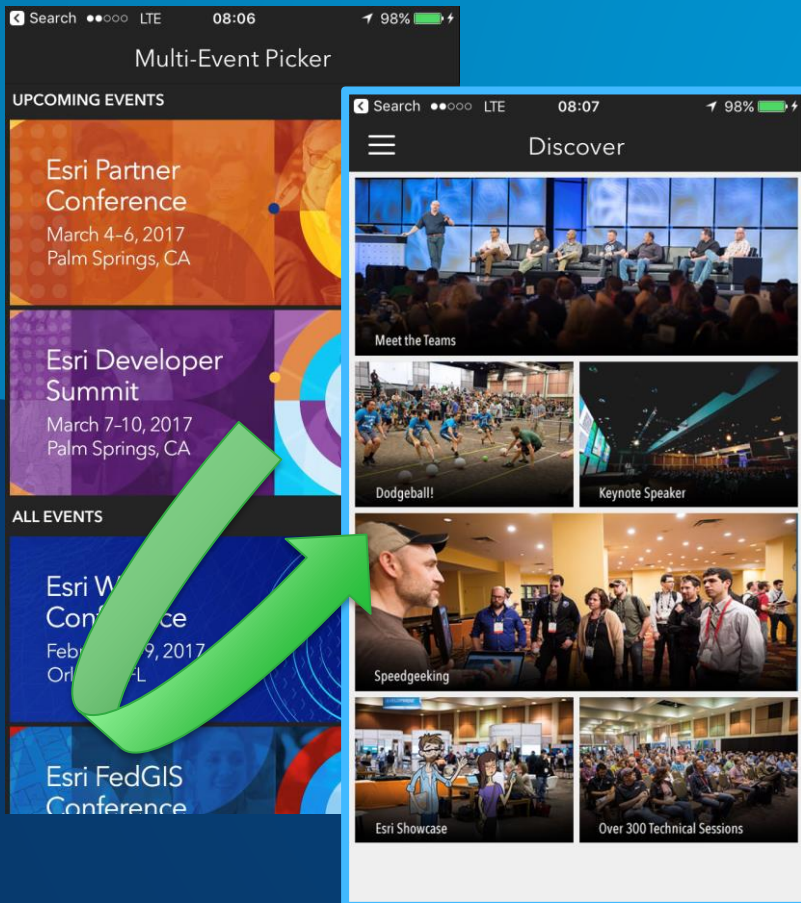
Expand

- Expandable container

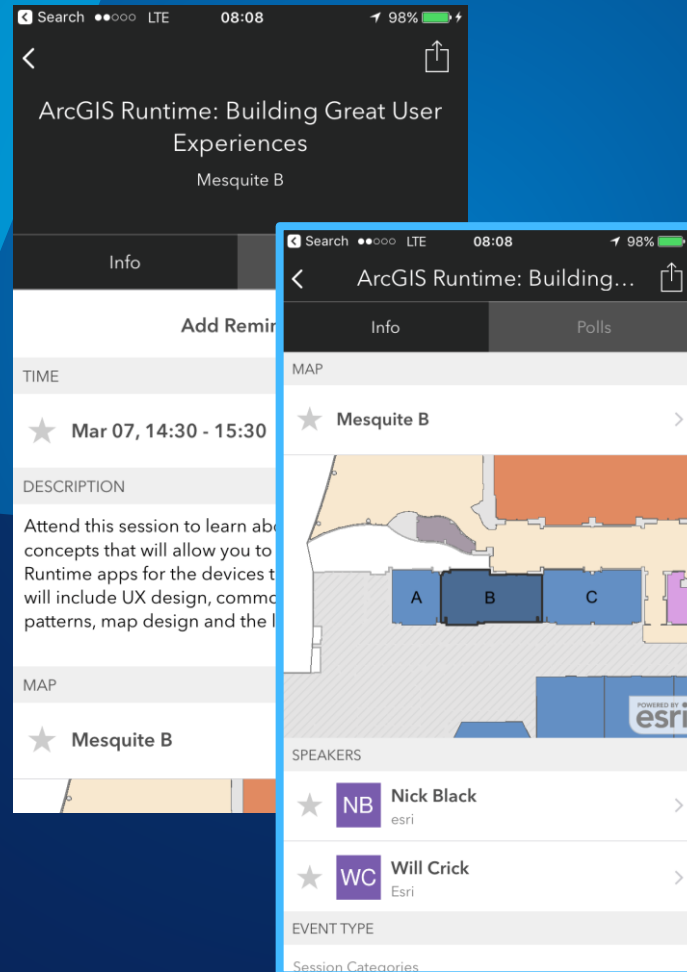


Please Take Our Survey!

Download the Esri Events app
and go to DevSummit



Select the session you attended



Scroll down to the
“Feedback” section

Complete Answers,
add a Comment,
and Select “Submit”

The screenshot shows the 'Feedback' section of the Esri Events app. It contains five Likert scales for rating the session. The first scale is for 'The title and description were consistent with the content presented'. The second scale is for 'The presentation was well organized and presented clearly'. The third scale is for 'The presenter exhibited strong public speaking skills'. The fourth scale is for 'The session provided information or techniques I can apply right away'. The fifth scale is for 'I would recommend this session to a colleague'. Below the scales is a 'Comments' section with a text input field. At the bottom, there is a 'Submit Feedback' button circled in red, and a 'Notes' section with a text input field and an 'Email Notes' button.



esri

THE
SCIENCE
OF
WHERE

Did you know?

For More Templates, Sample Files, and Icons See

<https://compass.esri.com/resources/presentations/Pages/Main.aspx>

New Feature

Footers have been added to this template. To learn about using footers for PowerPoint 2010:

<https://support.office.com/en-US/article/Add-a-header-or-footer-882EFCEA-35CD-4B68-AC0B-041AE1BA7099>


For PowerPoint 2013:

<https://support.office.com/en-US/article/Video-Add-headers-and-footers-to-a-presentation-C76A96AA-6622-442E-8B0A-ED75BBAE22B3>

Videos not working?

For video in PowerPoint, animated GIF is recommended. AVI and WMV (native PC files) will work best on a PC platform. H.264 MP4 will work only on PowerPoint 2013 and later.

Aspect Ratio Test



If this shape does not appear as
a perfect circle, adjust the aspect
ratio of your display until it does.

Try the resolution 1920x1080 for
16:9 displays.