




2022 ESRI USER CONFERENCE

ArcGIS API for JavaScript: Web Editing

Heather Gonzago, Jose Banuelos, and Larry Young



Demos and slides
available at

Session Agenda

- What are you looking for when editing?
- What's currently out there?
- Goals for Web Editing with JavaScript API
- Workflows and Examples

What are you looking for when editing?

- Historically the approach has been targeted apps
 - Addressing specific workflows
 - Simplicity – bottom line is make it easy for end user
 - Organizations struggling to maintain trained staff, so simple becoming even more important
- Getting some requests that are more generic (i.e., just give us everything)
 - Complete ArcGIS Online based solution for managing parcels, utilities, addresses, etc.
 - Organizations looking for easier deployment options. Don't want to have to update software on every machine
- Organizations also looking to go all in on cloud
 - Everything in ArcGIS Online
 - All view, query, editing and analysis through web (and sometimes mobile) applications

What are you looking for when editing?

- Users are looking for:
 - Usability for non-GIS people
 - Add features with minimal clicks
 - Add multiple features with a single sketch
 - Snapping
 - Configured on back end with fewer options for end user
 - Auto-closure of polygons (don't let me create slivers/gaps)
 - Create polygon for whole area and then chop up instead of adding the pieces one by one and hope you don't get gaps
 - Edit related records
 - Copy/paste for simplicity of creating new features
 - Split and Merge tools with attribute update
 - Attribute rule/Field expression capabilities

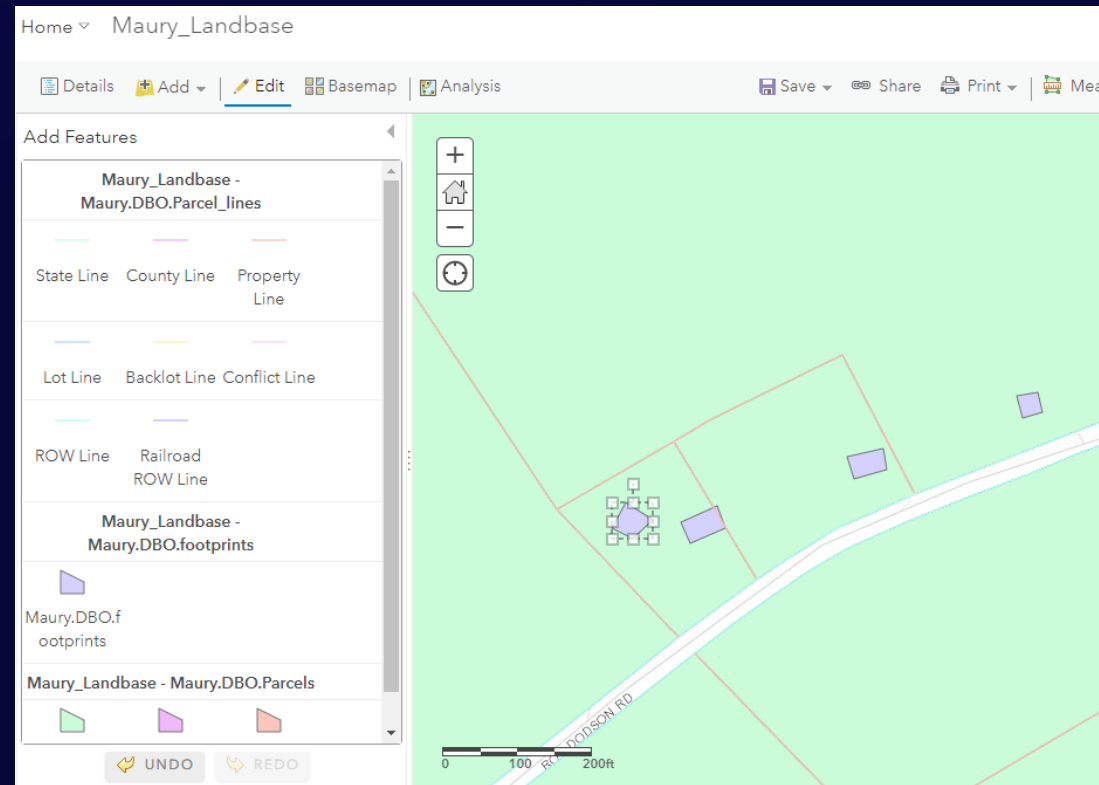
What are you looking for when editing?

- What do you want?
 - Targeted apps or everything?
 - Applications or widgets?
 - Configuration at back end or by end user?
 - Task frameworks or sets of tools/widgets?

What's currently out there?

- Classic Map Viewer Editor

- Add features
- Edit attributes
- Add/update attachments



What's currently out there?

- Editor Widgets for Web App Builder
 - Edit widget
 - Specify editable layers
 - Add features via templates
 - Snapping
 - Smart Editor widget
 - Configure editable fields
 - Automatically calculate fields
 - Global attribute values (set once and apply to all new features)

The screenshot displays two overlapping windows from the Web App Builder interface. The background window is the 'Smart Editor' for a feature template, showing fields for 'Lifecycle Status' (set to 'In Service'), 'Install Date' (set to 'Current'), 'Material' (set to 'Ductile Iron - DIP'), and 'Diameter' (set to '6"'). A checkbox at the top is checked for 'Use preset values (new features only)'. The foreground window is a 'Select features to copy' dialog. It contains a red warning message: 'Creating multiple features using this functionality will save all the new features immediately. Field matching is not supported when creating a new multi-part feature.' Below the message, there are three checked items: 'Lateral (2/2)' with two sub-items 'Residential', and 'Collector Main (1/1)'. At the bottom of the dialog are four buttons: 'Create Features', 'Create 1 Multi-...', 'Apply Field Matching' (which is highlighted with a green border), and 'Cancel'.

☒ Use preset values (new features only)

Lifecycle Status
In Service

Install Date
Current

Material
Ductile Iron - DIP

Diameter
6"

Select features to copy
Creating multiple features using this functionality will save all the new features immediately. Field matching is not supported when creating a new multi-part feature.

☒ Lateral (2/2)
☒ Residential
☒ Residential

☒ Collector Main (1/1)

Create Features Create 1 Multi-... **Apply Field Matching** Cancel

What's currently out there?

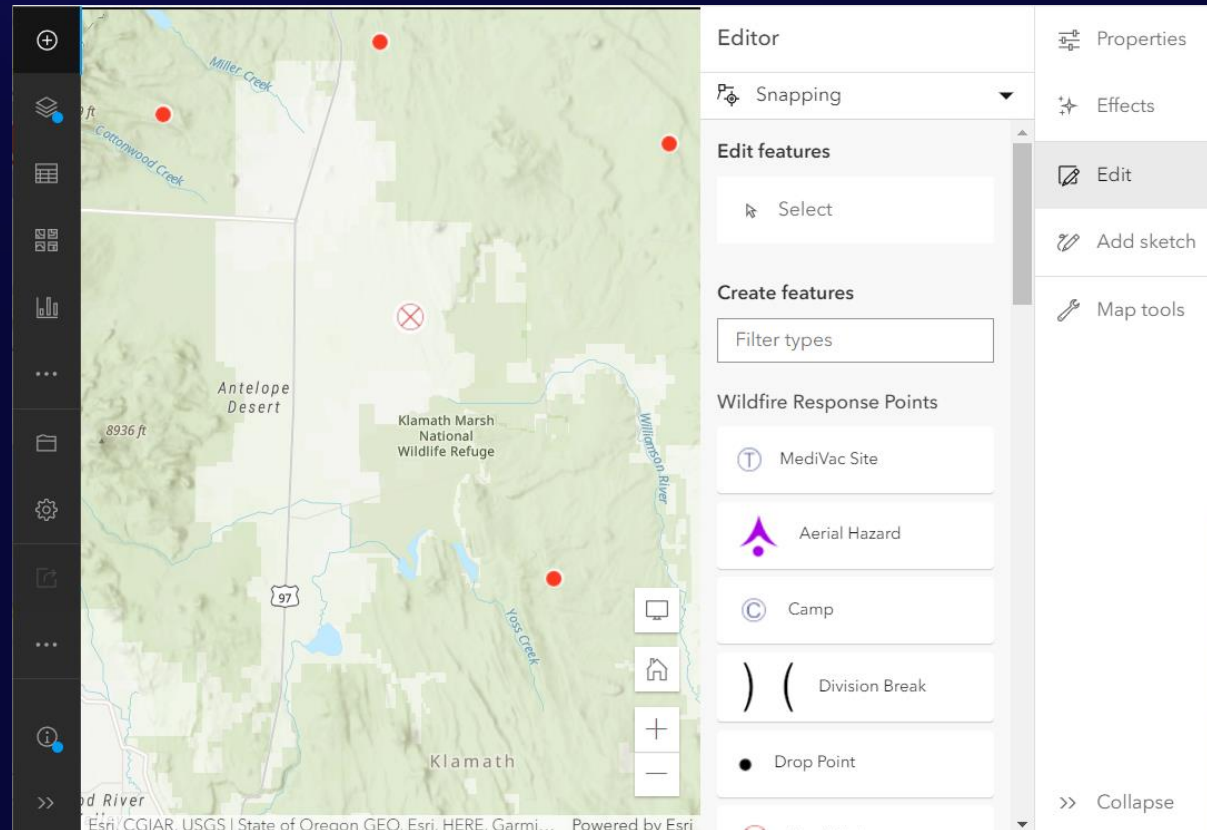
- SceneViewer Editor

- Work with 2D and 3D features
- Move, scale, and rotate geometries, as well as edit vertices
- Configurable attribute editing (choose fields to edit, arrange in groups, etc.)



What's currently out there?

- REST API
- ArcGIS JavaScript API
 - FeatureLayer.applyEdits()
 - Editor widget!
- Experience Builder
- Map Viewer




What's currently out there?

- Solutions

- Utility Management solutions (Stormwater, Water, and Electric Distribution)
- Parcel Management solution
- Park Asset Management

- Partner/Distributor Apps

- ArcGIS Marketplace
 - Search for “editing” or “web editing”



Parcel Drafter

This solution can be used to enter metes and bounds descriptions, verify deeds and documents, and confirm survey information prior to submission.

[Learn more →](#)

Goals for Web Editing and JavaScript API

Configurable Editing Capabilities

- There is a need for a configurable editing experience
 - Apps (Viewer, more?), widgets (Experience Builder and API), and API capabilities where users can configure focused editing experiences
 - Supports a configurable task-based workflow
- Framework to support advanced editing workflows for utility networks, parcel fabrics, geodatabase topologies, etc.

Goals for Web Editing and JavaScript API

2nd half of 2022

- Rule based snapping
- Version Management experience
- Undo/Redo
- Async support for form fields using Arcade expressions , ie. \$featureSet support
- Tooltip feedback integration
- Better support for server-side Attribute Rules
 - Updating other layers based on an edit from the client

Goals for Web Editing and JavaScript API

Long Term

- Support for curves
 - Display and editing
- Display Filters on SubtypeGroupLayers
- Batch attribute editing
- Group and Preset feature templates
 - Create multiple features at one time
- Dimensions & Annotations
- Editing related records
- Support for auto save while editing
- Planar Topology and shared edge editing
 - Rubberband style editing where moving one features moves other features

Goals for Web Editing and JavaScript API

Long Term

- Integrated Utility Network Editing
 - Associations
 - Creating associations
 - Structural attachments
 - Connectivity associations
 - Containment associations
 - Tools for building Utility network topology (validation) and tracing
 - Tools for validating subnetworks
- Editing of Parcel Fabric

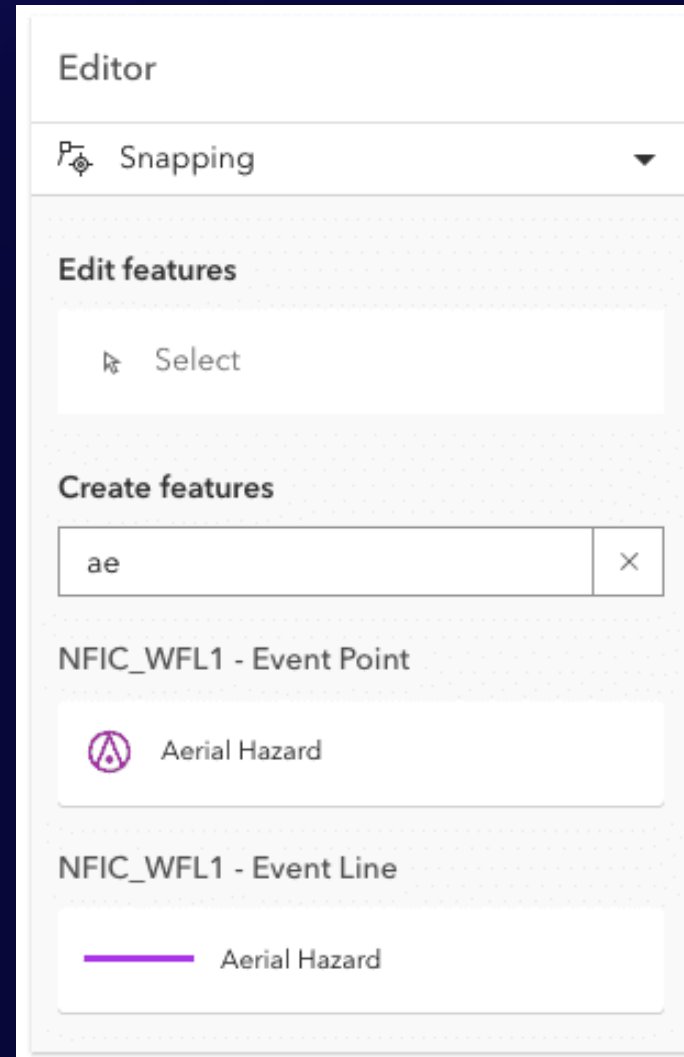
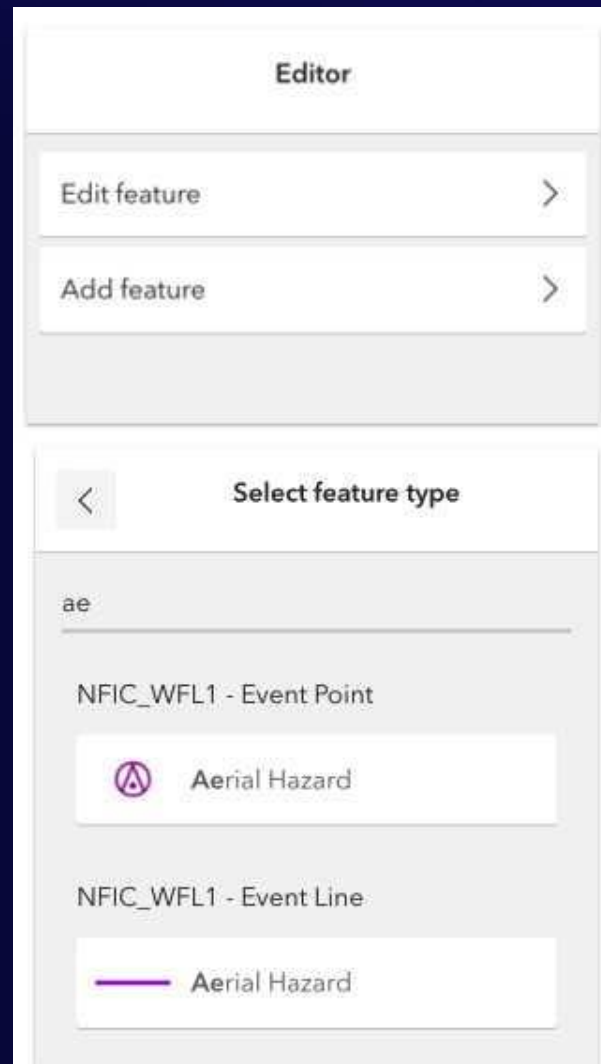
JavaScript API Workflows and Examples



Where to begin?

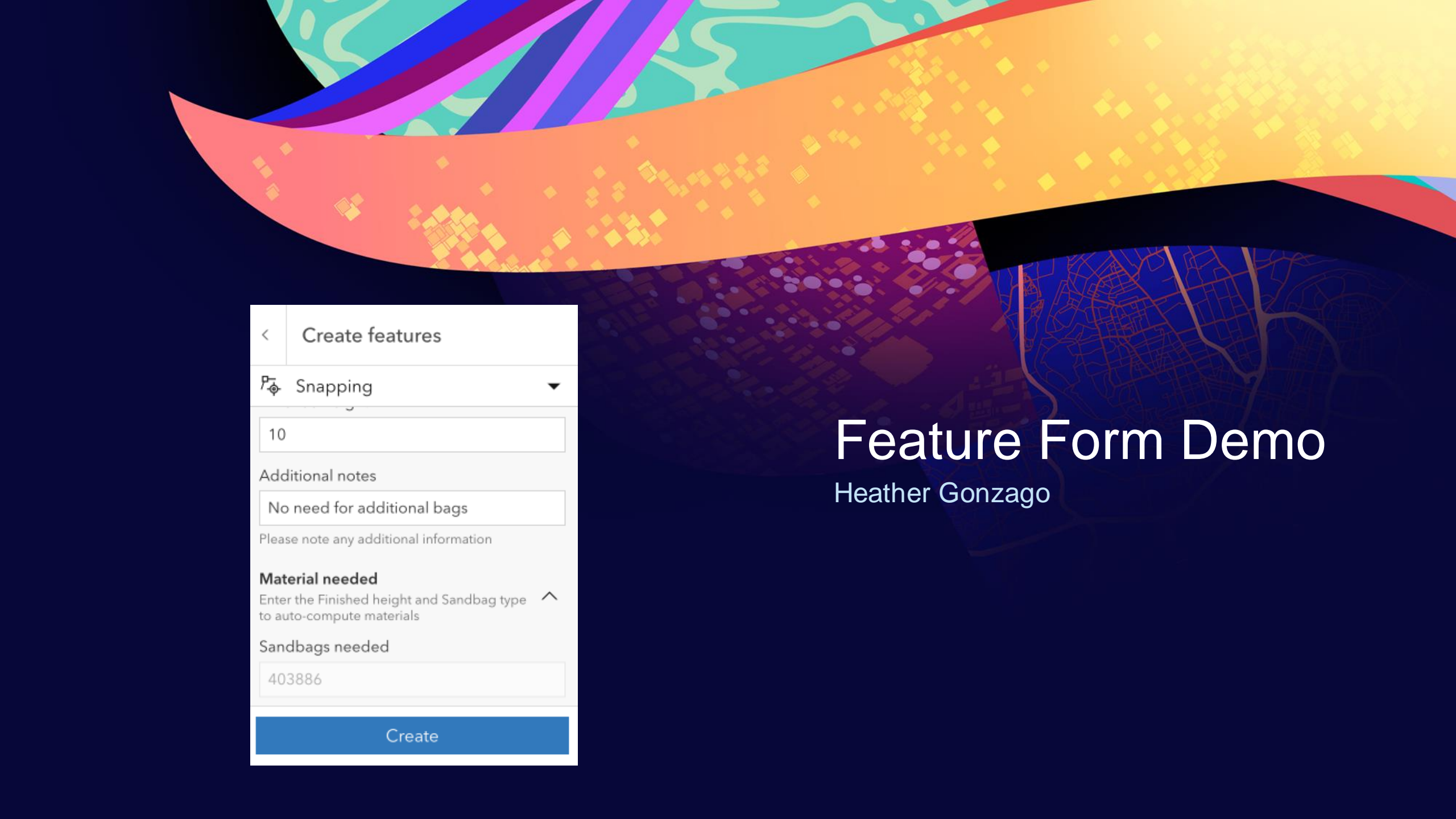
- New to the API or just need a refresher?
- SDK documentation
 - Release notes
 - API reference
 - Samples

Updated Editor UI with additional capabilities



Feature Form

- Displays attributes of a feature
- Configure input fields for attribute editing
- Format via the FormTemplate
- Support for calculated Arcade expressions
 - Field visibility displays attributes of a feature.
 - Required expressions
 - Field value expressions
- Configure form in webmap and read automatically
- Configure form in saved layer and read automatically



< Create features

Snapping ▾

10

Additional notes

No need for additional bags

Please note any additional information

Material needed

Enter the Finished height and Sandbag type to auto-compute materials ^

Sandbags needed

403886

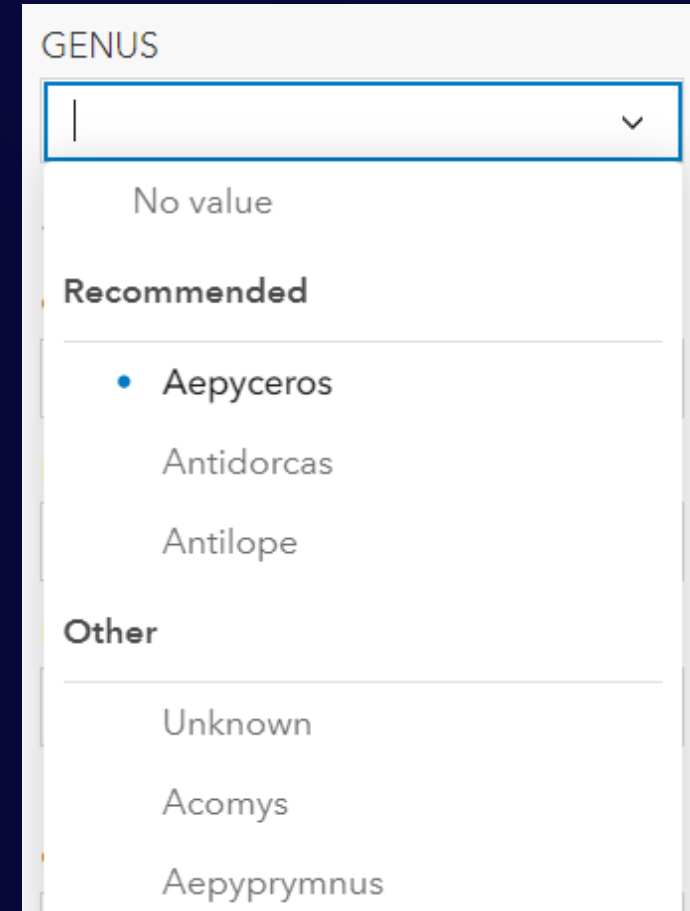
Create

Feature Form Demo

Heather Gonzago

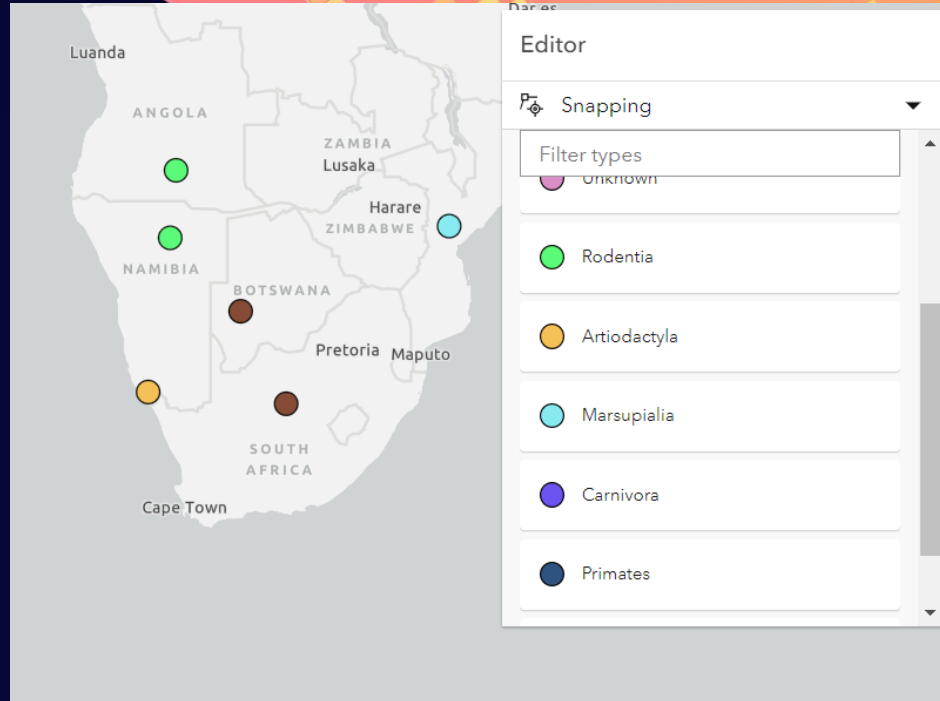
Contingent Values

- A field value becomes dependent on the values in one or more other fields.
- Restrict the list of valid inputs
- Helps preserve data integrity



The image shows a screenshot of a web form with a dropdown menu labeled 'GENUS'. The dropdown is open, displaying a list of options. The first option is 'No value'. Below it is a section header 'Recommended' followed by three options: 'Aepyceros' (which is selected with a blue dot), 'Antidorcas', and 'Antilope'. Below this is another section header 'Other' followed by three options: 'Unknown', 'Acomys', and 'Aepyprymnus'.

Category	Value
	No value
Recommended	Aepyceros
	Antidorcas
	Antilope
Other	Unknown
	Acomys
	Aepyprymnus

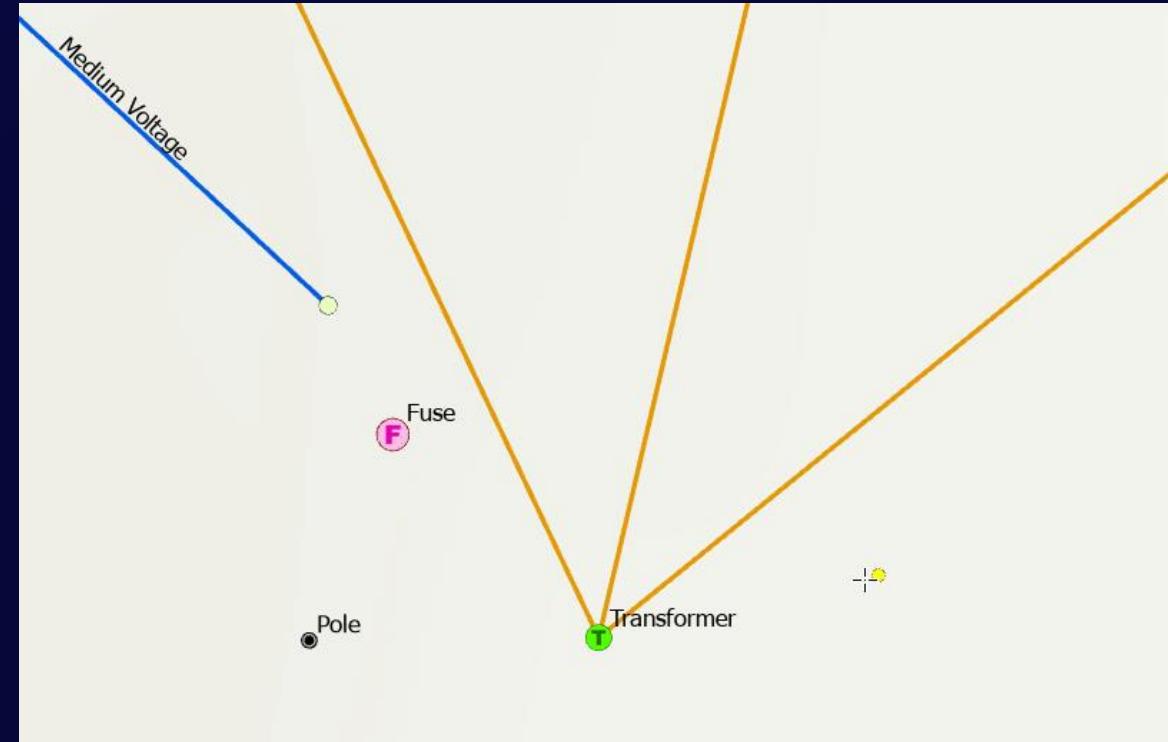


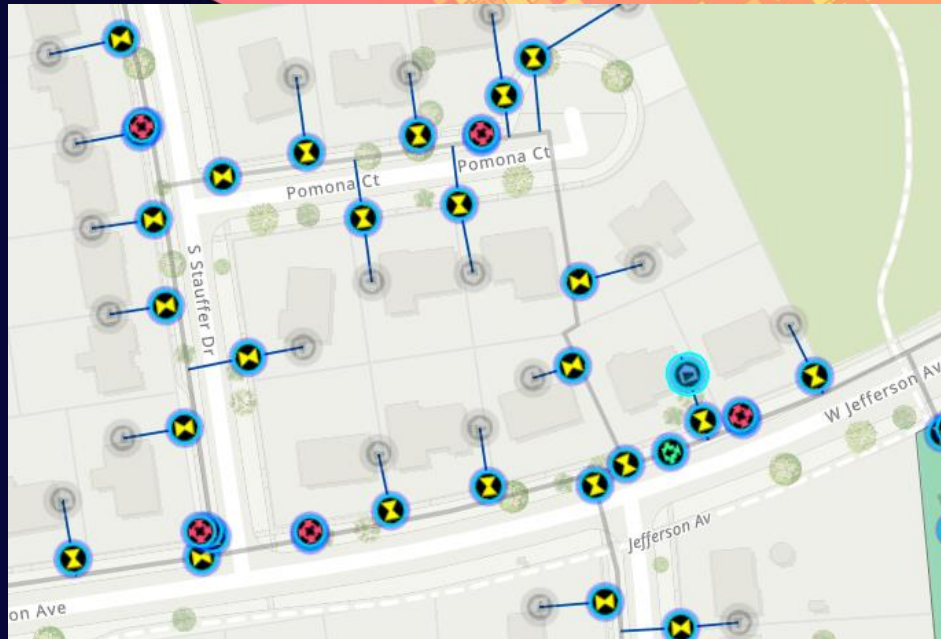
Contingent Values Demo

Jose Banuelos

Network Snapping Rules

- Connectivity rules define which features can be geometrically coincident or associated.
- There are three types of connectivity rules:
 - Junction-junction
 - Junction-edge
 - Edge-junction-edge






Network Snapping Rules with Editor Demo

Jose Banuelos

Q & A

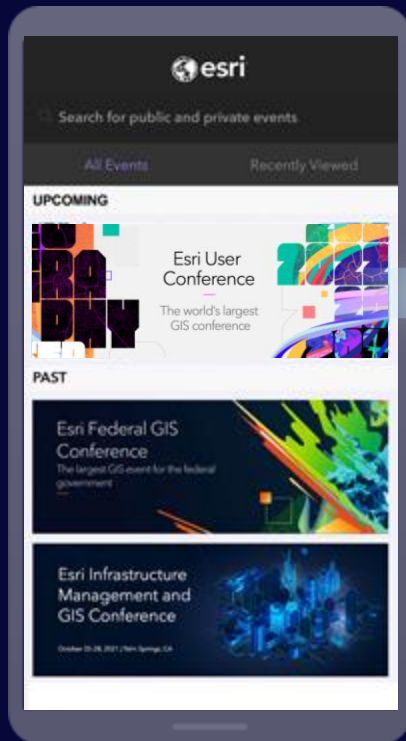


The background features a dark blue field with several overlapping, wavy, translucent bands in shades of orange, yellow, and purple. These bands are decorated with a pattern of small, glowing yellow squares. Below the bands, there are faint, stylized map patterns in purple and blue, showing street grids and geographical features.

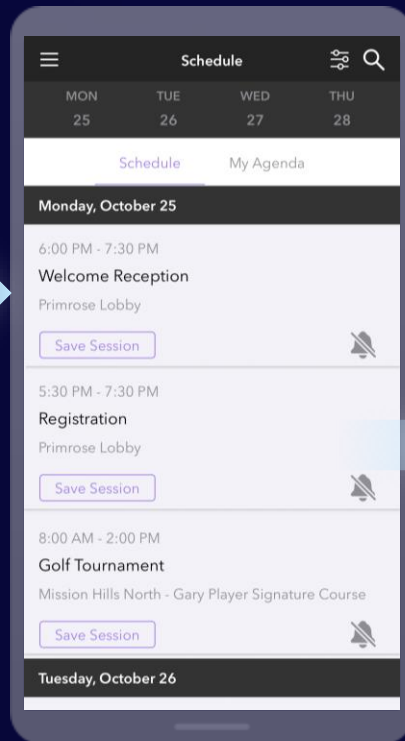
Demos and slides
available at

Please Share Your Feedback in the App

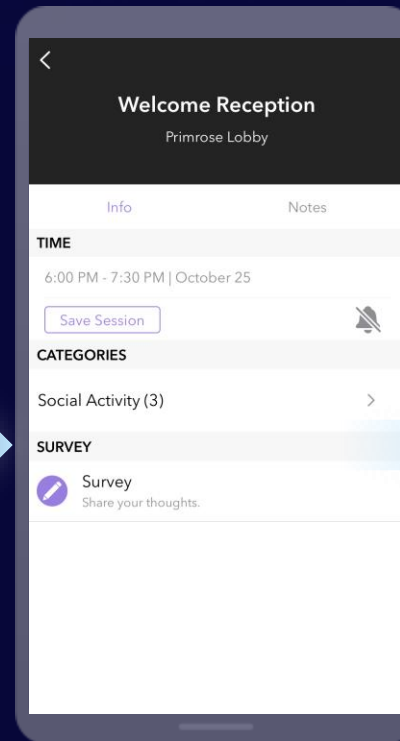
Download the Esri Events app and find your event



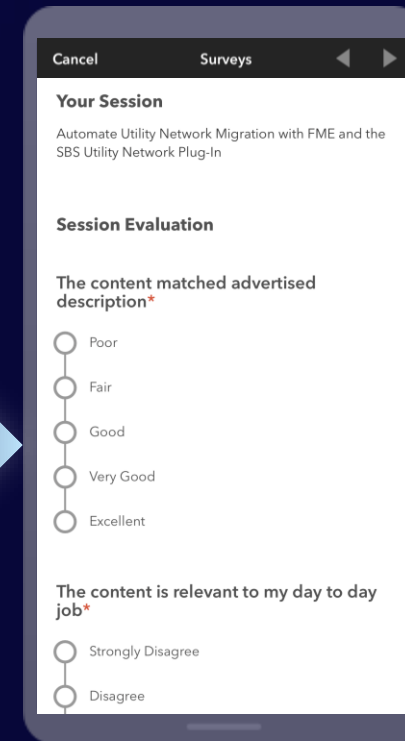
Select the session you attended



Scroll down to "Survey"



Log in to access the survey





esri[®]

THE
SCIENCE
OF
WHERE[®]