

```
<arcgis-map zoom="4" center="-118,34">  
  <arcgis-search position="top-right" />  
</arcgis-map>
```

ArcGIS Maps SDK for JavaScript: Web Editing

Jonathan Uihlein

Jose Bañuelos

```
<arcgis-map zoom="4" center="-118,34" />
```

```
view.goTo({  
  center: [-126, 49]  
})  
.catch(function(error) {  
  if (error.name !== "AbortError") {  
    console.error(error);  
  }  
});
```

2024 ESRI DEVELOPER SUMMIT

Agenda

- Prep data for editing
- Available components for editing
- Editor
- Form Template & Smart Forms
- Custom editing: `Layer.applyEdits()`
- Custom editing: `FeatureTable`
- Custom editing: Branch-versioned editing

Slides & Demos available:



[https://github.com/banuelosj/
DevSummit-presentation/tree/
main/2024/web-editing](https://github.com/banuelosj/DevSummit-presentation/tree/main/2024/web-editing)


```
view.goTo({
  center: [-126, 49]
})
.catch(function(error) {
  if (error.name !== "AbortError") {
    console.error(error);
  }
});
```

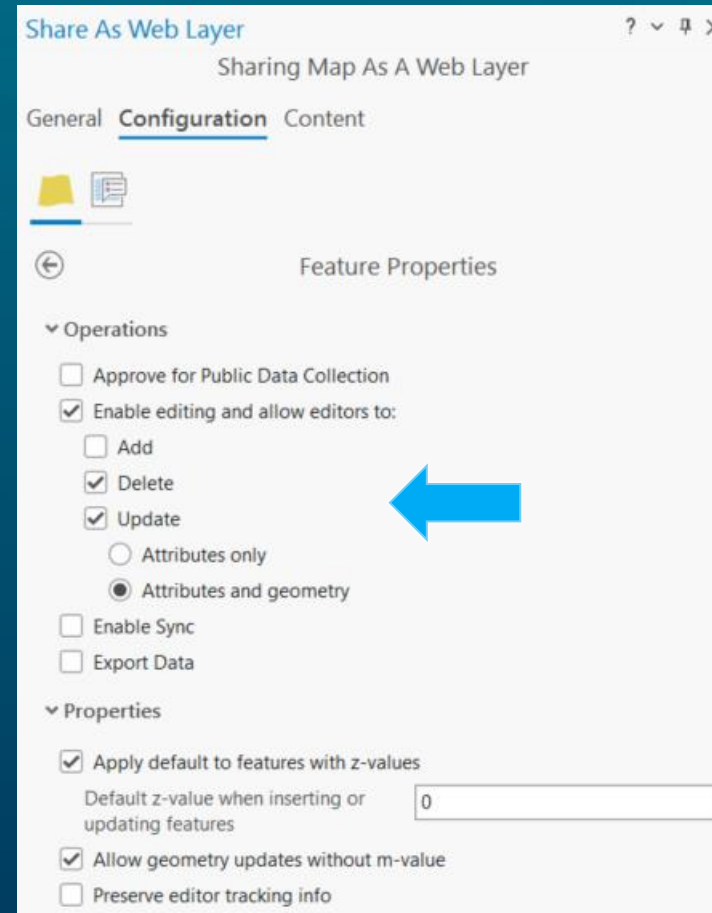
Prep your data for editing

Jose Bañuelos

```
queryParams =
  queryParams().apply {
    where{ouse = "price > 200"
  }
viewModelScope.launch {
```

Enable editing in Pro and ArcGIS Online

- Publish a feature service from Pro
- Configure the operations



The screenshot shows the 'Share As Web Layer' dialog box with the 'Configuration' tab selected. The 'Feature Properties' section is expanded, showing the 'Operations' group. A blue arrow points to the 'Update' checkbox, which is checked. Below it, the 'Attributes and geometry' radio button is selected.

Share As Web Layer

Sharing Map As A Web Layer

General **Configuration** Content

Feature Properties

Operations

- ☐ Approve for Public Data Collection
- ☒ Enable editing and allow editors to:
 - ☐ Add
 - ☒ Delete
 - ☒ Update
 - ☐ Attributes only
 - ☒ Attributes and geometry
- ☐ Enable Sync
- ☐ Export Data

Properties

- ☒ Apply default to features with z-values
 - Default z-value when inserting or updating features:
- ☒ Allow geometry updates without m-value
- ☐ Preserve editor tracking info

Enable editing in Pro and ArcGIS Online

- Configure allowed operations in Online
- Similar to Pro

Feature layer (hosted)

Editing

☒ Enable editing

☐ Keep track of changes to the data (add, update, delete features).

☐ Keep track of who edited the data (editor name, date and time).

☐ Enable Sync (required for offline use and collaboration).

• Who can edit features?

Share the layer to specific groups of people, the organization or publicly via the Share button on the Overview tab. This layer is currently shared with: Everyone (public)

• What kind of editing is allowed?

☒ Add


☒ Delete

☒ Update

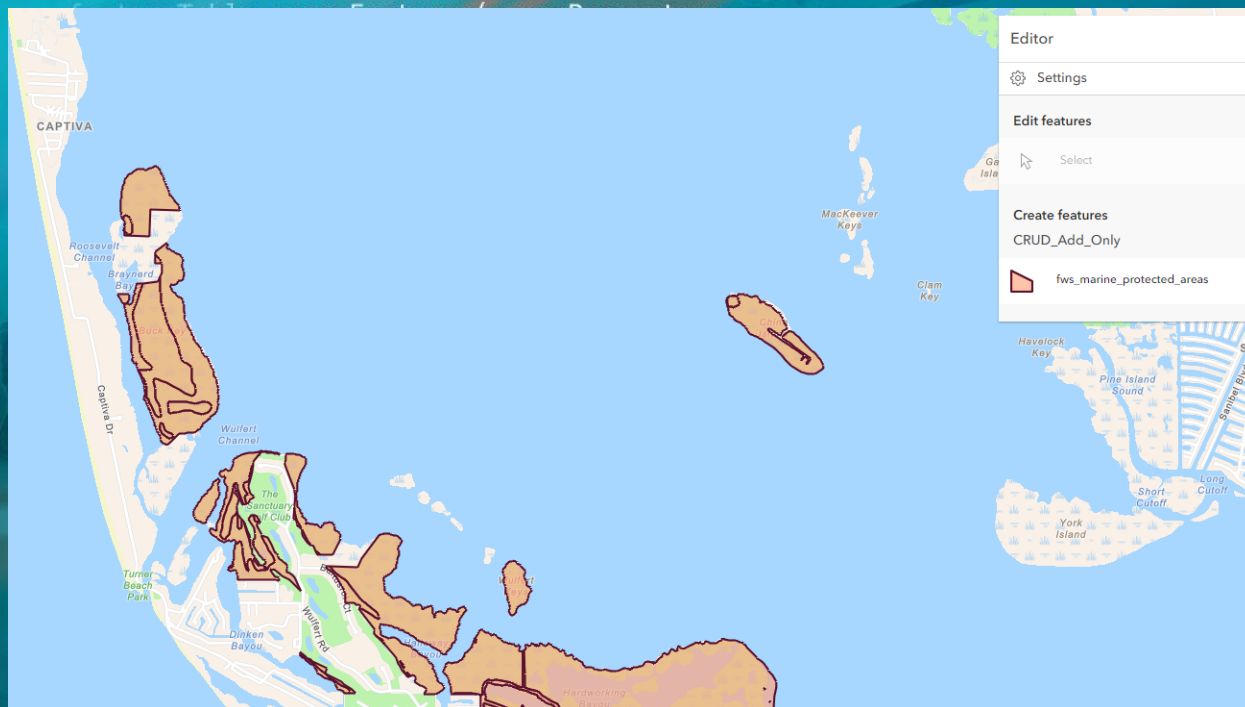
☐ Attributes only

☒ Attributes and geometry

[Manage geometry updates](#)



```
const queryParameters =  
  queryParameters().apply {  
    filter = "price > .200"  
  }  
viewModelScope.launch {
```



Editor: Reacting to edit configurations

Jose Bañuelos

Out-of-the-box components for editing

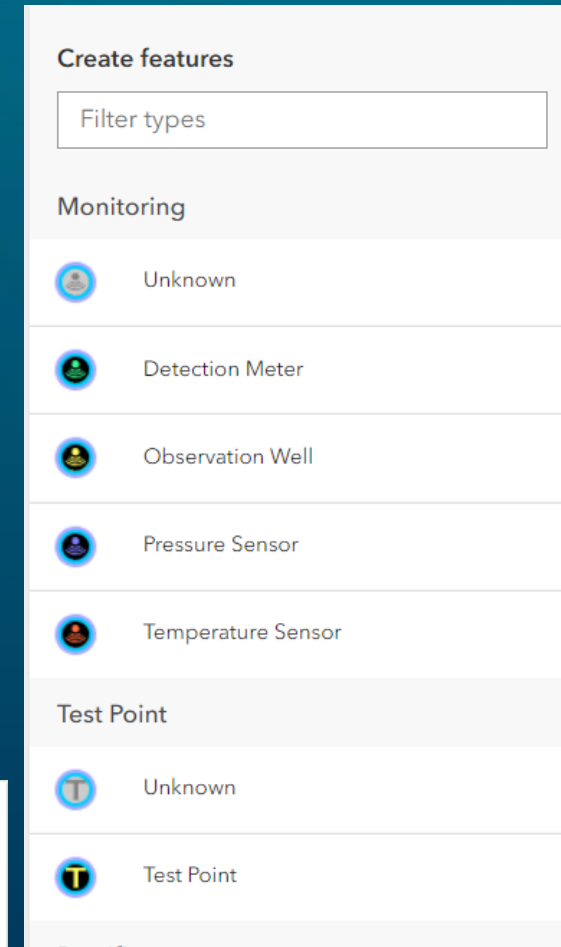
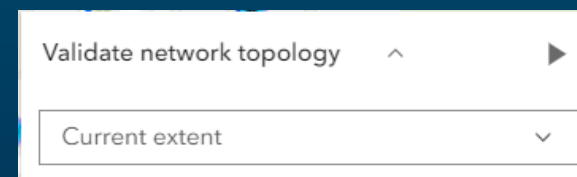
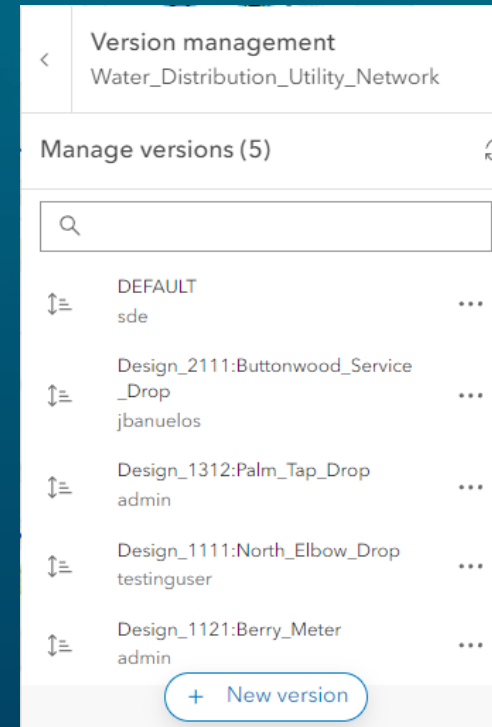
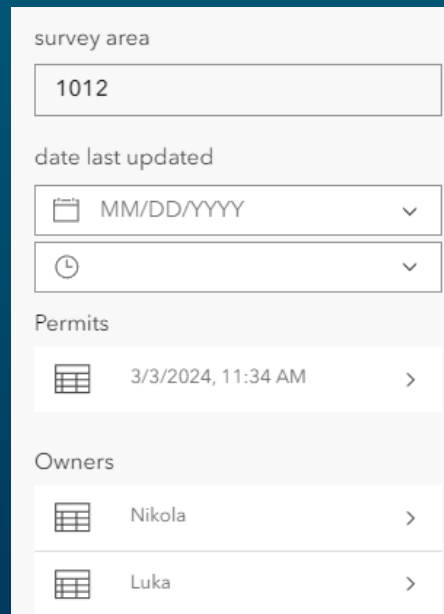
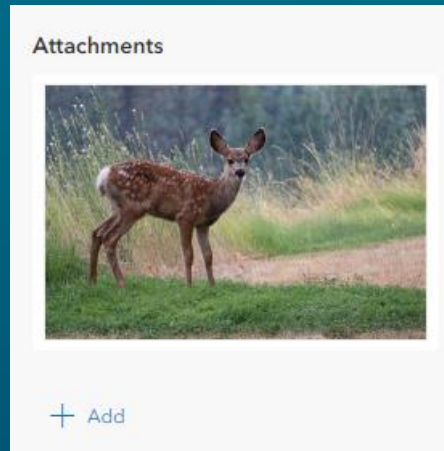
Jonathan Uihlein

```
view.goTo({  
  center: [-126, 49]  
})  
.catch(function(error) {  
  if (error.name !== "AbortError") {  
    console.error(error);  
  }  
});
```

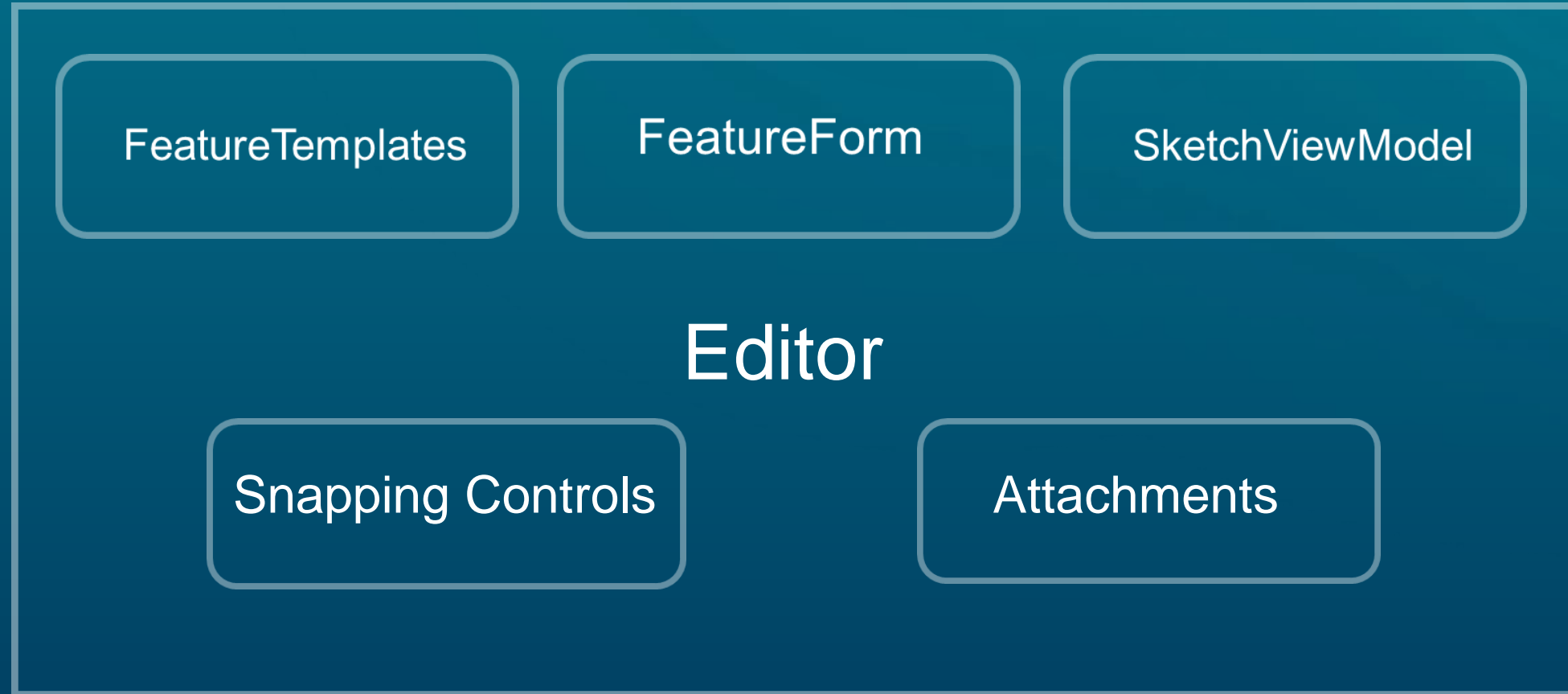
```
queryParams =  
queryParams().apply {  
  where{ouse = "price > 200"  
}  
viewModelScope.launch {
```


Available JS SDK widgets (and components)

- Attachments
- Editor
- FeatureForm
- FeatureTable
- FeatureTemplates
- SnappingControls
- Sketch
- VersionManagement
- UtilityNetworkValidateTopology



Editor building blocks



Editor

Jonathan Uihlein

```
view.goTo({  
  center: [-126, 49]  
})  
.catch(function(error) {  
  if (error.name !== "AbortError") {  
    console.error(error);  
  }  
});
```

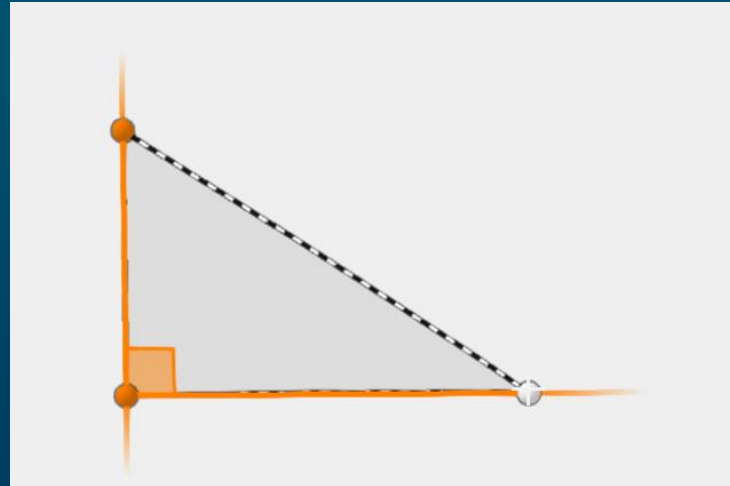
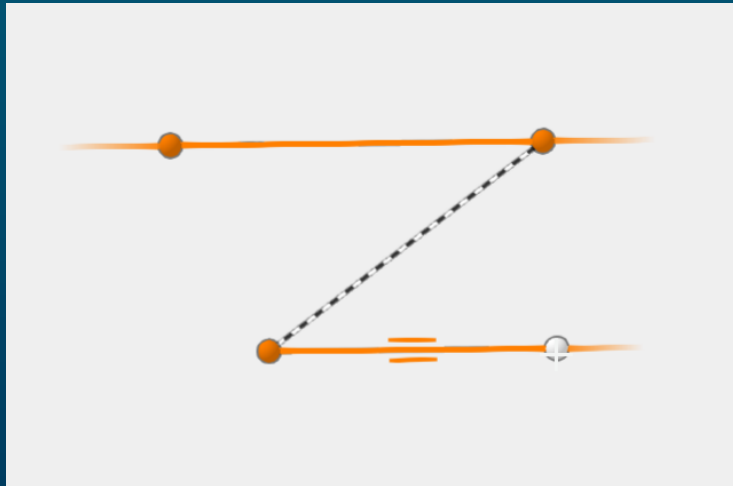
```
queryParams =  
queryParams().apply {  
  where{ouse = "price > 200"  
}  
viewModelScope.launch {
```


Editor

- Out-of-the-box
- Attribute, geometry edits
- Add, Update, Delete features, attachments
- Snapping
- Tooltips and Editing constraints
- Arcade support
- Contingent values
- Related records
- 2D / 3D
- FormTemplate & SmartForms*

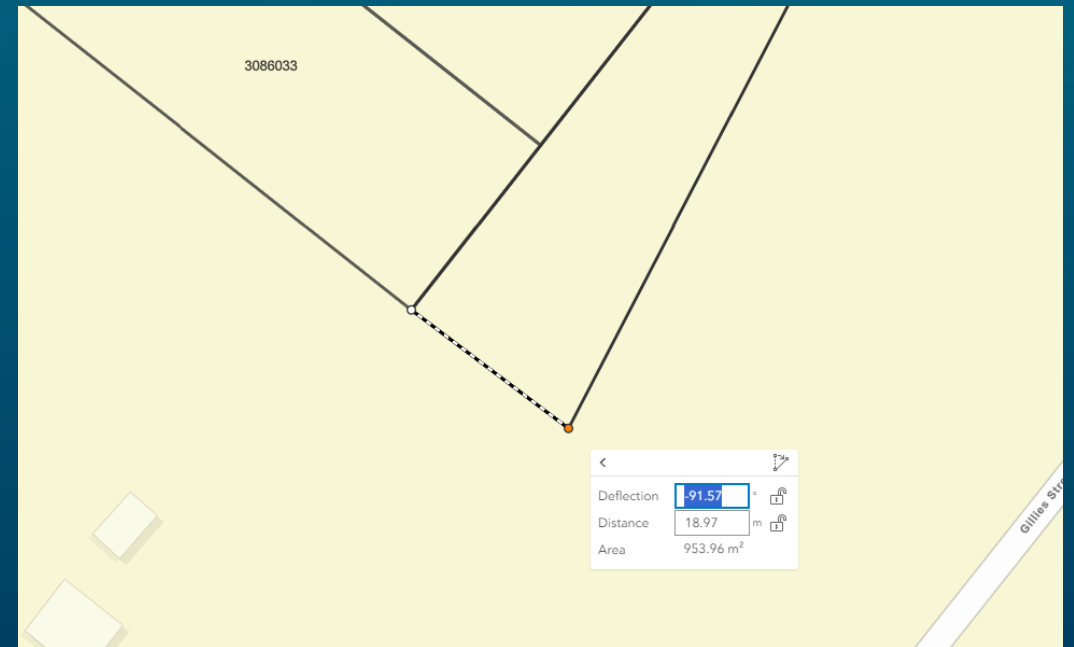
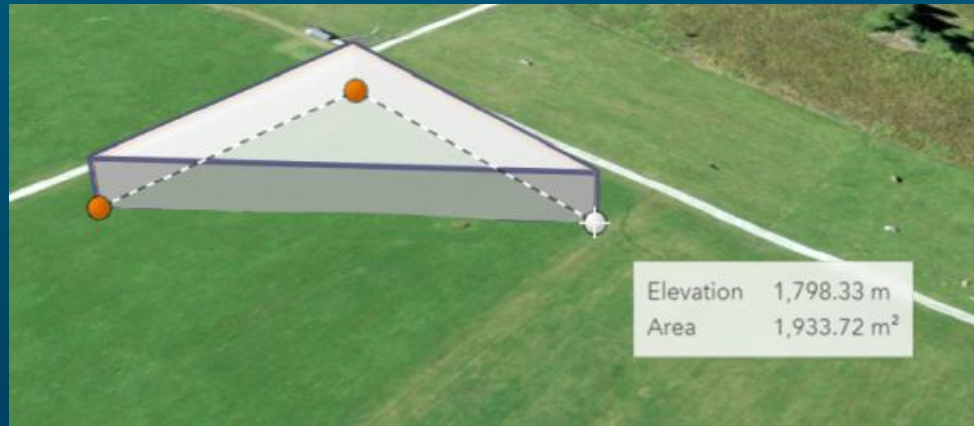
Snapping

- Geometry Guides
- Feature snapping
- Recent support for SubtypeGroupLayer
- UtilityNetwork rule based snapping
- Ctrl shortcut key for toggling

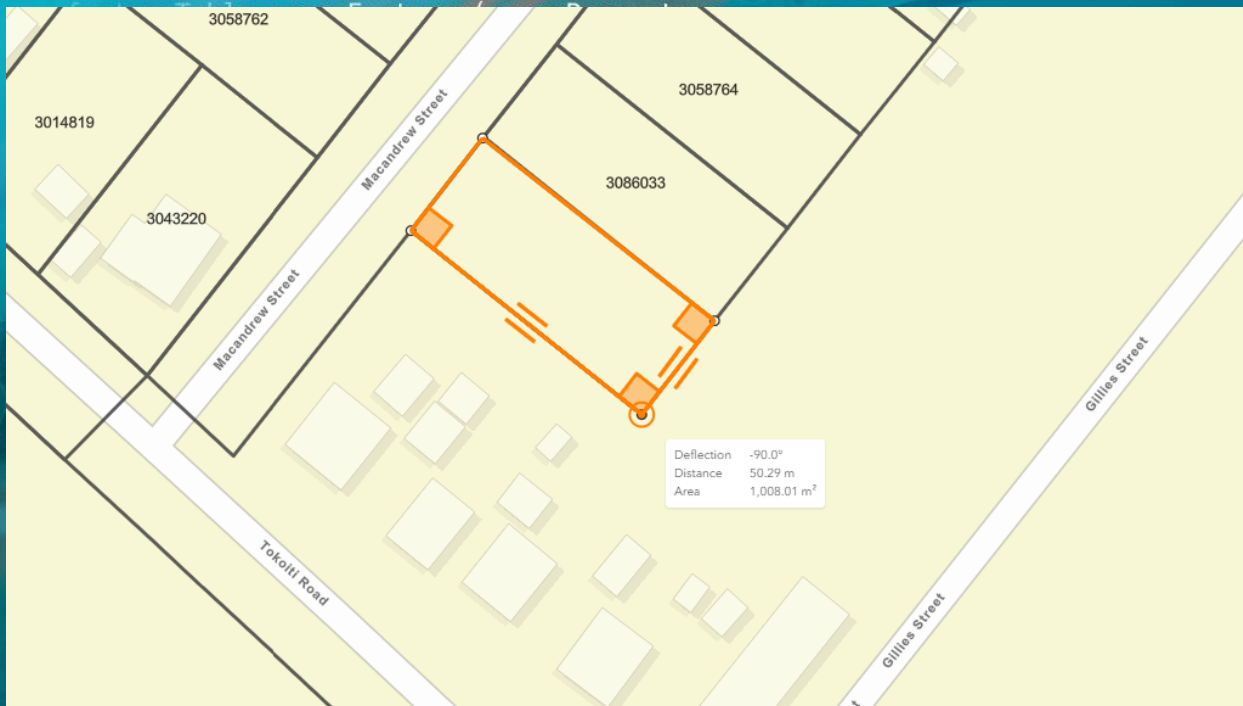


Tooltips & editing constraints

- Tooltips display distance, area, elevation, etc.
- Editing constraints provide a user interface to set the angle, distance, elevation, etc. while actively drawing.
- Tooltips need to be enabled + Tab key for getting into editing constraints




```
queryParameters =  
queryParameters().apply {  
    filter { it.price > 200 }  
}  
viewModelScope.launch {
```



Editor: snapping & tooltips

Jose Bañuelos

FormTemplate & Smart Forms

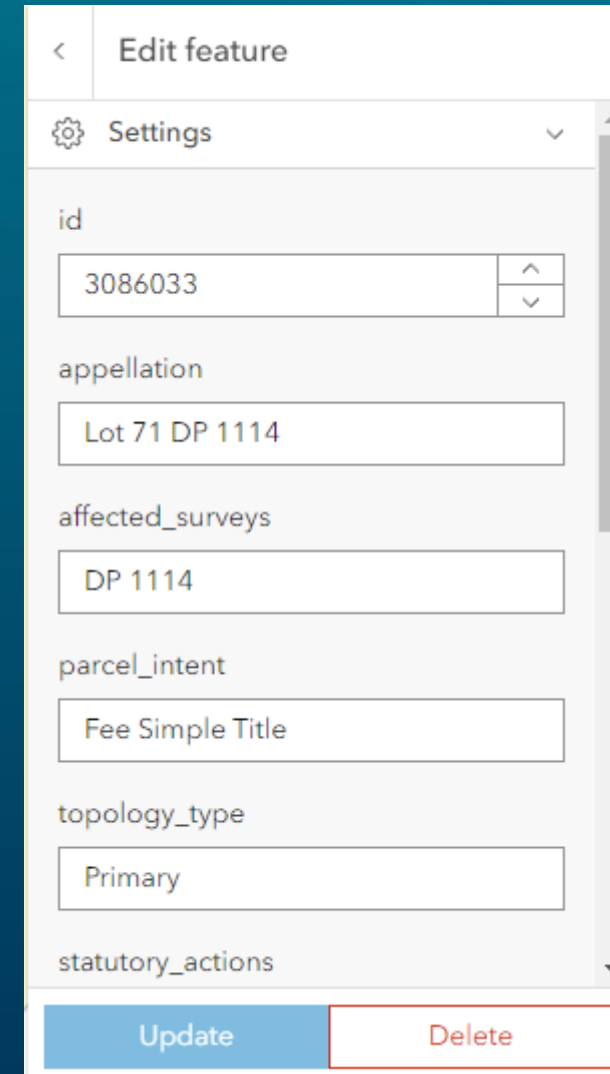
Jose Bañuelos

```
view.goTo({  
  center: [-126, 49]  
})  
.catch(function(error) {  
  if (error.name !== "AbortError") {  
    console.error(error);  
  }  
});
```

```
queryParams =  
queryParams().apply {  
  where{ouse = "price > 200"  
}  
viewModelScope.launch {
```

Working through a FormTemplate

- By default, FeatureForm and Editor will display all fields in a layer whenever a feature is passed in for editing.
- Sometimes it doesn't make sense to edit all the fields



The screenshot shows a web-based form titled "Edit feature". At the top, there is a back arrow and the title. Below the title is a "Settings" section with a gear icon and a dropdown arrow. The form contains several input fields, each with a label above it: "id" (a text box with "3086033" and a small up/down arrow button), "appellation" (a text box with "Lot 71 DP 1114"), "affected_surveys" (a text box with "DP 1114"), "parcel_intent" (a text box with "Fee Simple Title"), "topology_type" (a text box with "Primary"), and "statutory_actions" (a text box that is currently empty). At the bottom of the form, there are two buttons: "Update" (a blue button) and "Delete" (a red button).

Field Name	Value
id	3086033
appellation	Lot 71 DP 1114
affected_surveys	DP 1114
parcel_intent	Fee Simple Title
topology_type	Primary
statutory_actions	

Update Delete

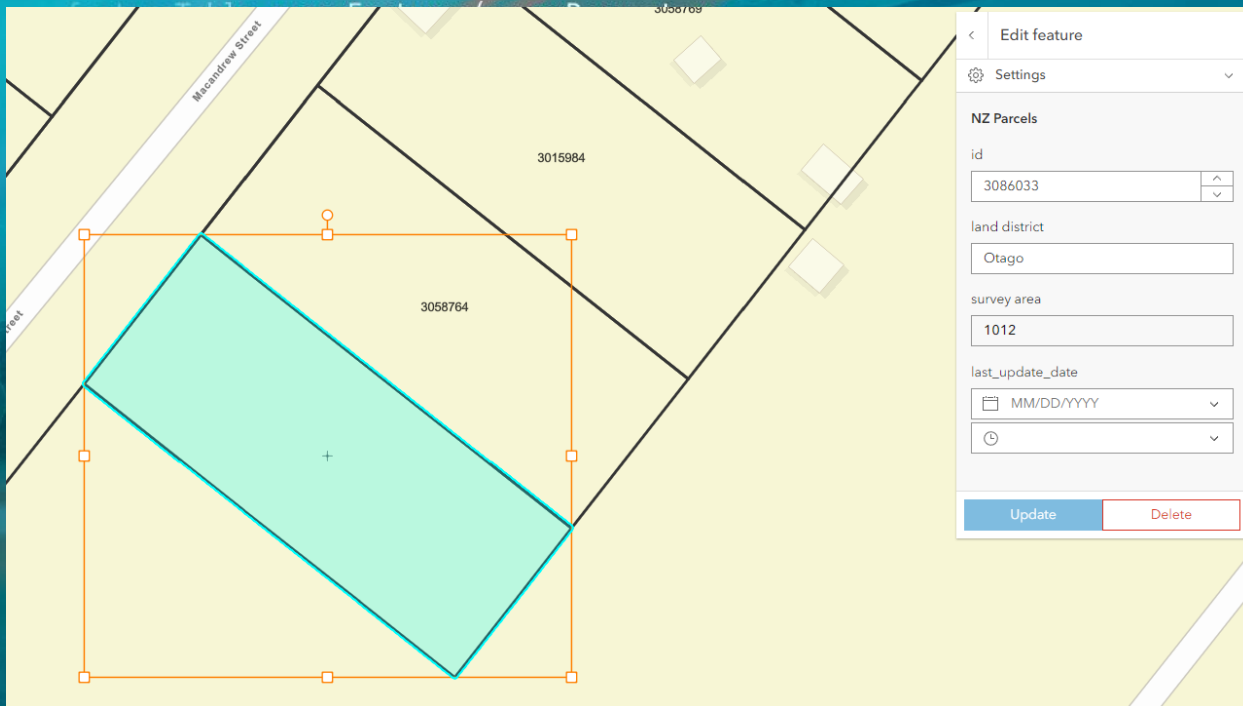
Working through a FormTemplate

- By default, FeatureForm and Editor will display all fields in a layer whenever a feature is passed in for editing.
- Sometimes it doesn't make sense to edit all the fields
- A FormTemplate provides control of what fields will be displayed on the form, and what fields are editable to the user.
- Leads to a cleaner display
- Supports related records
- Supports arcade expressions

The screenshot shows a mobile application interface for editing a feature. At the top, there is a back arrow and the title 'Edit feature'. Below this is a 'Settings' section with a gear icon and a dropdown arrow. The main section is titled 'NZ Parcels' and contains several form fields: 'id' with a text input containing '3086033' and a dropdown arrow; 'land district' with a text input containing 'Otago'; 'survey area' with a text input containing '1012' and a dropdown arrow; and 'last_update_date' with a date picker icon and a dropdown arrow showing 'MM/DD/YYYY'. At the bottom, there are two buttons: 'Update' (blue) and 'Delete' (red).

Edit feature	
Settings	
NZ Parcels	
id	3086033
land district	Otago
survey area	1012
last_update_date	MM/DD/YYYY
Update	
Delete	

```
queryParameters =  
queryParameters().apply {  
    filter { it.price > 200 }  
}  
viewModelScope.launch {
```



FormTemplate

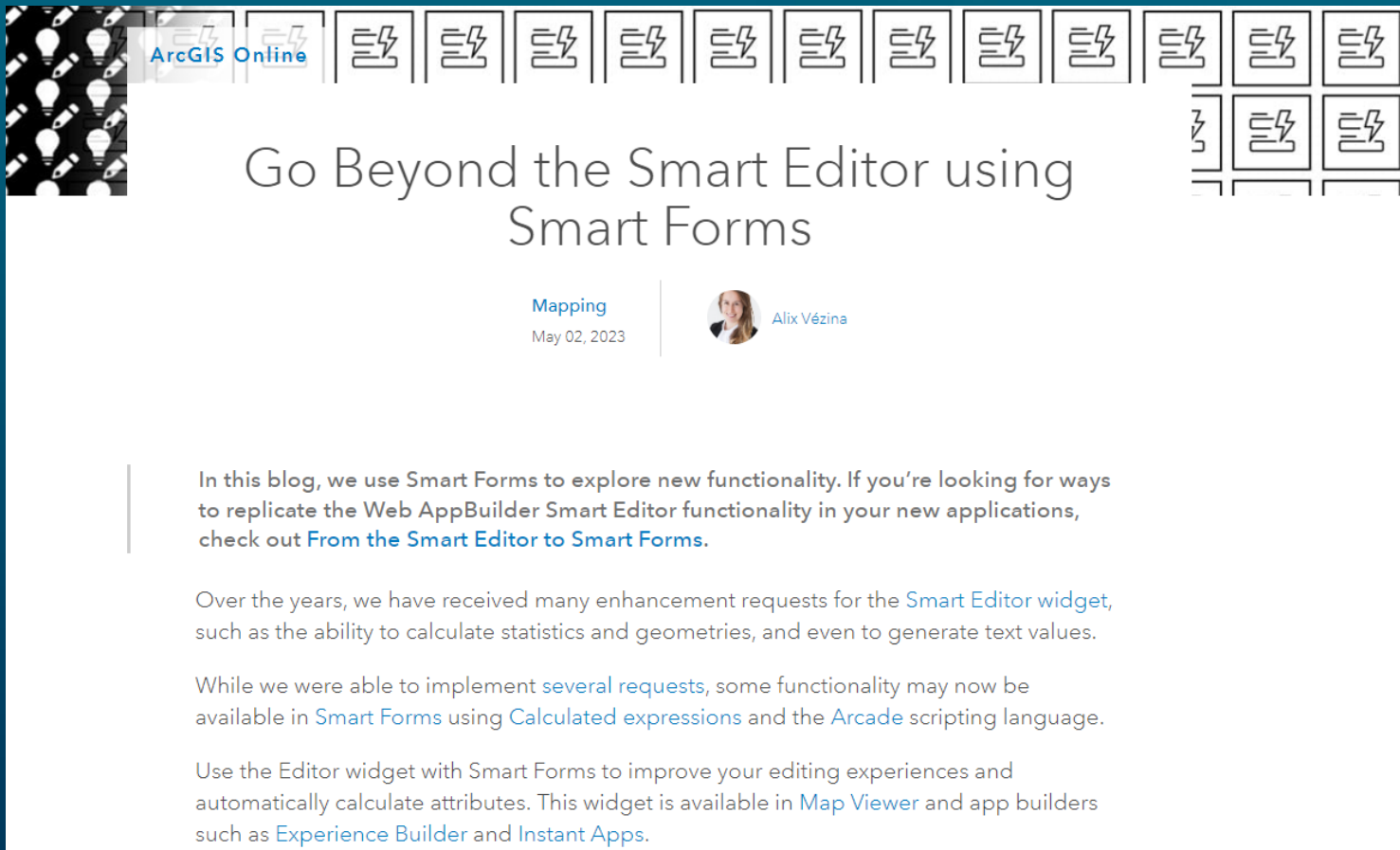
Jose Bañuelos

Smart Forms

- Central location for authoring the form template
- Provides a simple user interface for setting up your form templates
- Everything supported in the previous FeatureTemplate section is supported
- Can be reused by any application that uses the JS SDK, so that the form for each layer only needs to be setup once.
 - MapViewer
 - SceneViewer
 - Experience Builder
 - Instant Apps
- The form will also be respected in the next ArcGIS Maps SDK for Native release, unifying the Web and Mobile form setup.
 - Android, iOS, .NET

Smart Forms

- Looking for examples and tutorials
- [Go Beyond the Smart Editor using Smart Forms](#) – Alix Vézina



```
queryParams =  
queryParams().apply {  
    filter { it.price > 200 }  
}  
viewModelScope.launch {
```

Configure form

NZParcels

id

land_district

survey_area

last_update_date

Permits
Preview count: 1

Owners
Preview count: 3

Form builder

Form elements

Group

Fields
Fields store values within the layer

Add all (7)

Find fields

appellation
(appellation) abc

affected_surveys
(affected_surveys) abc

parcel_intent
(parcel_intent) abc

topology_type
(topology_type) abc

statutory_actions
(statutory_actions) abc

titles
(titles) abc

calc_area
(calc_area) 223

Related records

Form settings

Cancel OK

Configure a form through Smart Forms

Jose Bañuelos

Custom Editing: Editing with `applyEdits()`

Jose Bañuelos

```
view.goTo({
  center: [-126, 49]
})
.catch(function(error) {
  if (error.name !== "AbortError") {
    console.error(error);
  }
});
```

```
queryParams =
  QueryParameters().apply {
    where{ouse = "price > 200"
  }
viewModelScope.launch {
```

Editing with applyEdits() programmatically

- ApplyEdits is the rest call that is sent to the server with the edits from the client
- Available on FeatureLayer, SubtypeGroupLayer (through the sublayers), and SceneLayer
- Ex:
 - Add a feature with FeatureLayer
 - Note: No unique-identifier is needed for a create operation
 - Service takes care of it

```
// Add a new feature to FeatureLayer
const point = {
  type: "point",
  longitude: -49.97,
  latitude: 41.73
};

const attributes = {
  rotation: 34,
  description: "Adding feature programmatically",
  eventtype: 0
}

const pointGraphic = new Graphic({
  geometry: point,
  attributes: attributes
});

featureLayer.applyEdits({
  addFeatures: [pointGraphic]
});
```


Editing with applyEdits() programmatically

- Add, Update, Delete features simultaneously
- Ex:
 - Add & Delete features

```
// Add a feature and delete a couple
const point = {
  type: "point",
  longitude: -65.23,
  latitude: 22.50
};

const attributes = {
  rotation: 12,
  description: "Add & Delete",
  eventtype: 22
}

const featuresToDelete = [
  { objectId: 121 },
  { objectId: 25 }
];

const pointGraphic = new Graphic({
  geometry: point,
  attributes: attributes
});

featureLayer.applyEdits({
  addFeatures: [pointGraphic],
  deleteFeatures: featuresToDelete
});
```

Editing with applyEdits(): returnServiceEditsOption

- Edits sent to a server can trigger edits on other layers in the feature service
 - Attribute rules
 - Deleting Composite relationships
- These are common Enterprise workflows with feature services.
- Default applyEdits() does not refresh the other layers present on the map with edits. Only the layer you called FeatureLayer.applyEdits();
- We need every layer on the map with edits to refresh
- returnServiceEditsOption: "original-and-current-features"

```
const options = {
  returnServiceEditsOption: "original-and-current-features"
};

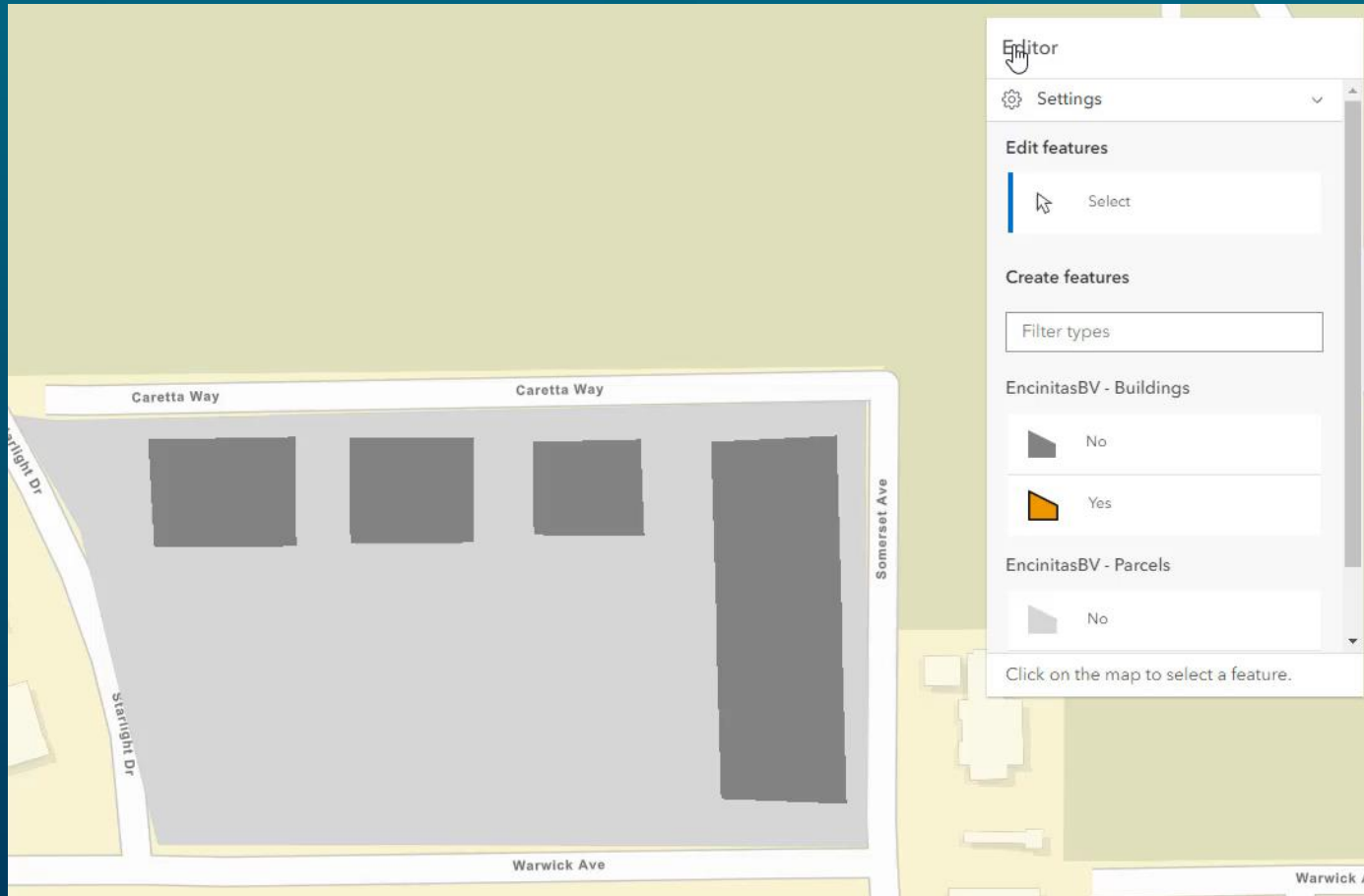
const parcelFeature = new Graphic({
  geometry: {
    type: "polygon",
    rings: [
      [6249371.47783947, 1958364.8923093975],
      [6254021.161453471, 1958486.0567652285],
      [6253869.706047714, 1954457.3407413065],
      [6249159.440205798, 1953215.4055611491],
      [6249371.47783947, 1958364.8923093975]
    ]
  },
  attributes: { Parcel_ID: 9999 }
});

const edits = {
  addFeatures: [parcelFeature]
};

parcelsFL.applyEdits(edits, options);
```

Editing with applyEdits(): returnServiceEditsOption

- returnServiceEditsOption: "original-and-current-features"



Custom Editing: FeatureTable

Jonathan Uihlein

```
view.goTo({  
  center: [-126, 49]  
})  
.catch(function(error) {  
  if (error.name !== "AbortError") {  
    console.error(error);  
  }  
});
```

```
queryParams =  
queryParams().apply {  
  where{ouse = "price > 200"  
}  
viewModelScope.launch {
```



```
queryParameters =  
queryParameters().apply {  
    filter { it.price > 200 }  
}  
viewModelScope.launch {  
  
    featureTable.queryFeatures(queryParameters  
    ).onFailure {  
        showError("Feature query failed")  
    }.onSuccess { featureQueryResult ->  
        val featureResult =  
        featureQueryResults.firstOrNull()  
    }  
}
```

FeatureTable & FeatureForm

Jonathan Uihlein

Custom Editing: Supporting branch- versioning

Jose Bañuelos

```
view.goTo({
  center: [-126, 49]
})
.catch(function(error) {
  if (error.name !== "AbortError") {
    console.error(error);
  }
});
```

```
queryParams =
  QueryParameters().apply {
    where{ouse = "price > 200"
  }
viewModelScope.launch {
```



```
queryParameters =  
queryParameters().apply {  
    filter { it.price > 200 }  
}  
viewModelScope.launch {  
  
    featureTable.queryFeatures(queryParameters  
    ).onFailure {  
        showError("Feature query failed")  
    }.onSuccess { featureQueryResult ->  
        val featureResult =  
        featureQueryResults.firstOrNull()  
    }  
}
```

Branch- versioning editing workflows

Jose Bañuelos

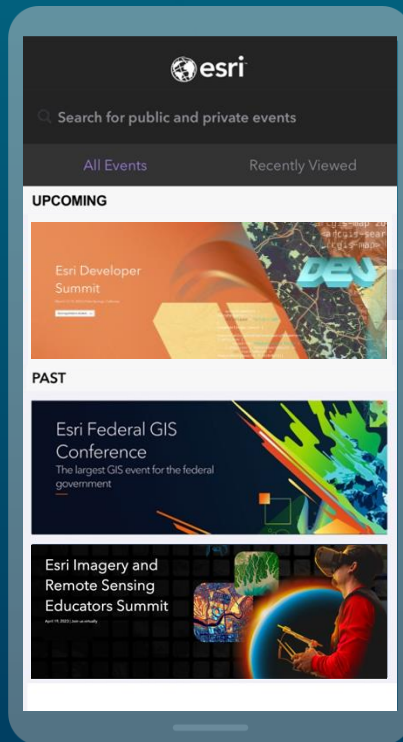
Slides & Demos available:



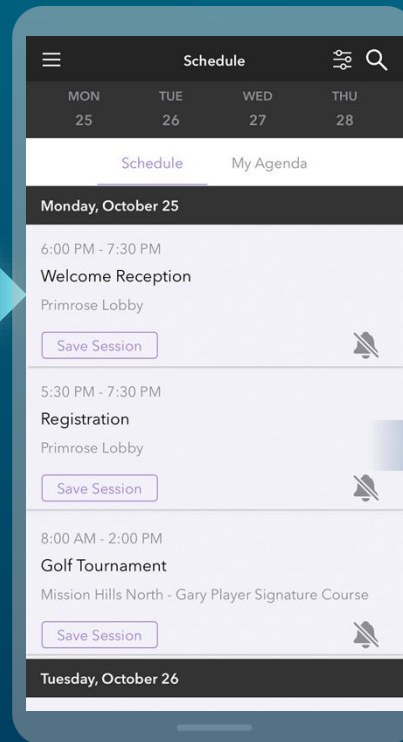
[https://github.com/banuelosj/
DevSummit-presentation/tree/
main/2024/web-editing](https://github.com/banuelosj/DevSummit-presentation/tree/main/2024/web-editing)

Please Share Your Feedback in the App

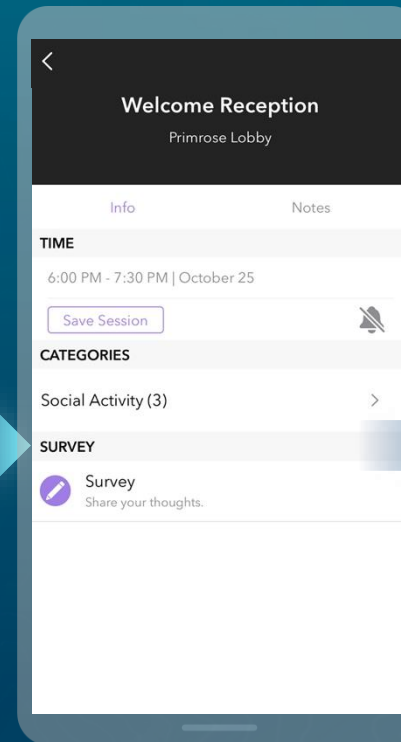
Download the Esri Events app and find your event



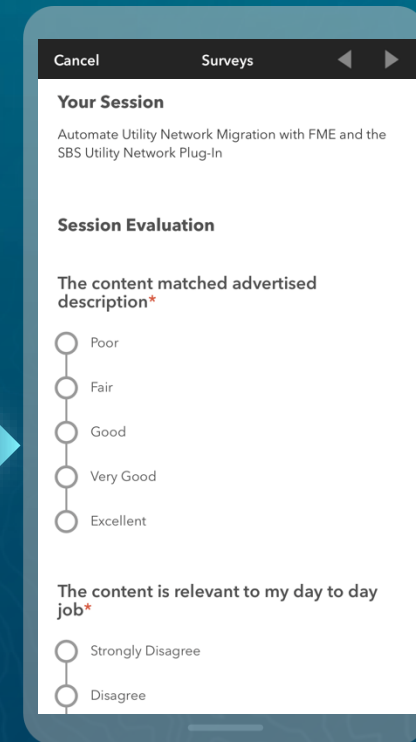
Select the session you attended



Scroll down to "Survey"



Log in to access the survey



The information shared in this presentation is
CONFIDENTIAL and MAY NOT BE REDISTRIBUTED.

For details on confidentiality see the Confidentiality clause in your
Esri partnering agreement or the Esri Distributor Agreement.

Connect With Us On Social

And Join the Conversation Using #EsriDevSummit2024

 twitter.com/EsriDevs #EsriDevSummit2024

 twitter.com/EsriDevEvents

 youtube.com/c/EsriDevelopers

 links.esri.com/DevVideos

 github.com/Esri

 github.com/EsriDevSummit

 links.esri.com/EsriDevCommunity

```
view.goTo({  
  center: [-126, 49]  
})  
.catch(function(error) {  
  if (error.name !== "AbortError") {  
    console.error(error);  
  }  
});  
  
<arcgis-map zoom="4" center="118,34" />
```



```
// show the compass and pass the  
mapRotation state data  
Compass(rotation = mapRotation) {  
  // reset the ComposableMap viewpoint  
rotation to point north using the  
mapViewModel  
  mapViewModel.setViewpointRotation(0.0)  
}
```



esri®

**THE
SCIENCE
OF
WHERE®**

```
const layerList = new LayerList({  
  view: view  
});  
  
// Add widget to the top right corner  
of the view  
view.ui.add(layerList, "top-right")
```

```
<arcgis-map zoom="4" center="-118,34">
```