ArcGIS Maps SDK for JavaScript: Animating Your Data

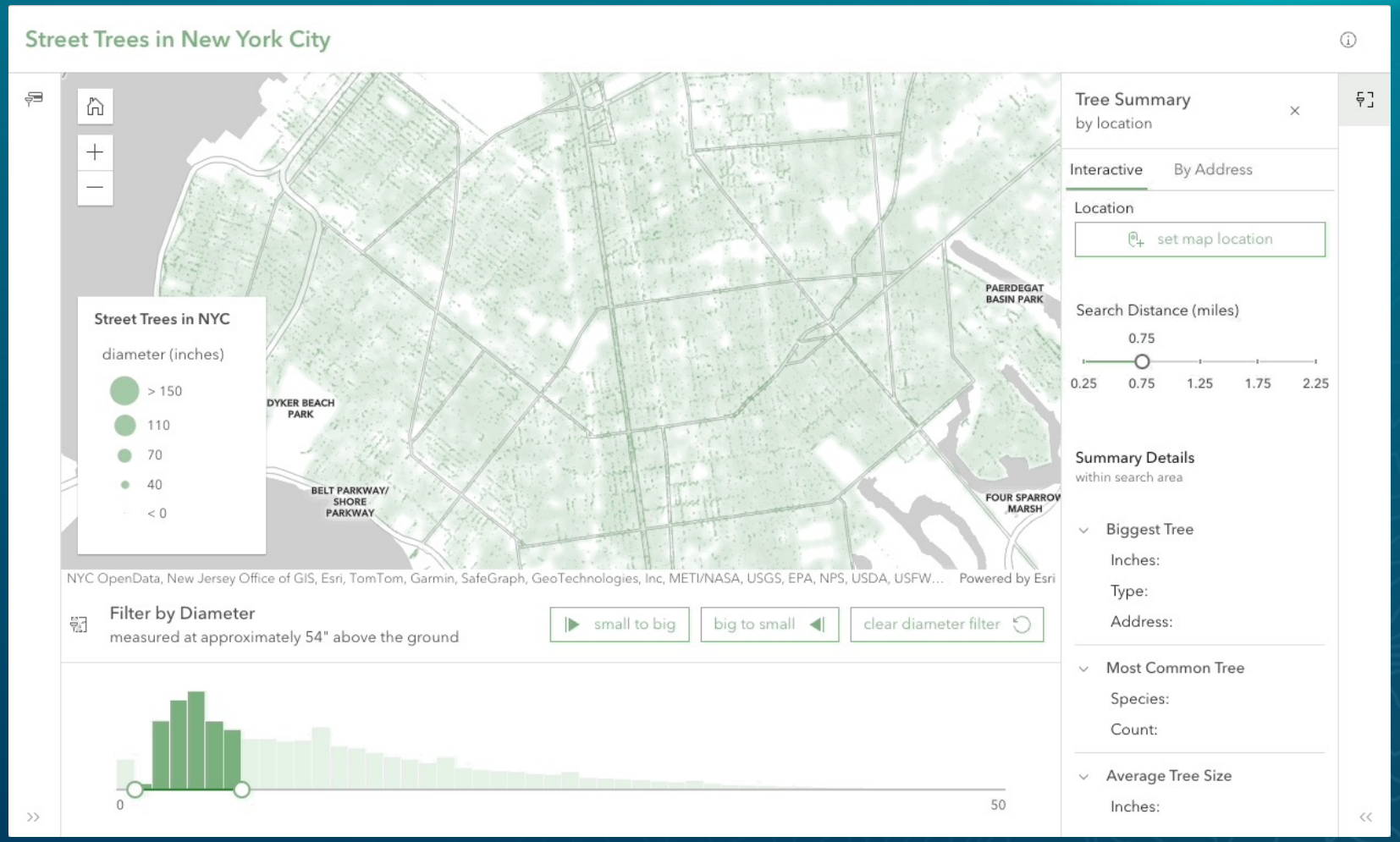
Anne Fitz and Undral Batsukh

2024 ESRI DEVELOPER SUMMIT

Animations

Why use them?

- **Interactive Exploration**
  - This can include zooming, filtering, or highlighting specific data points of interest

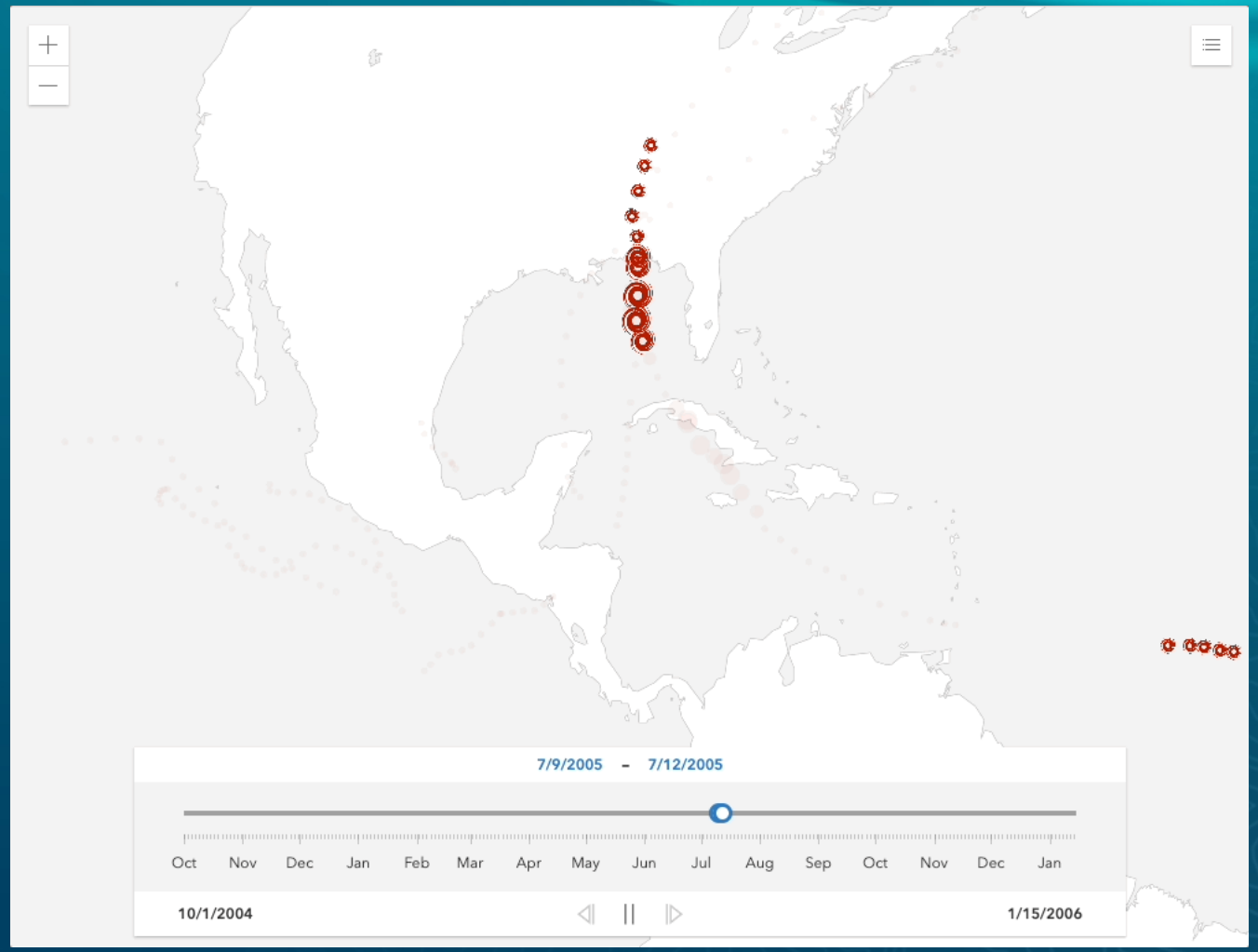


Animations

Why use them?

Temporal Insights

- Visualize change over time. This can help in identifying seasonal patterns, trends, or anomalies

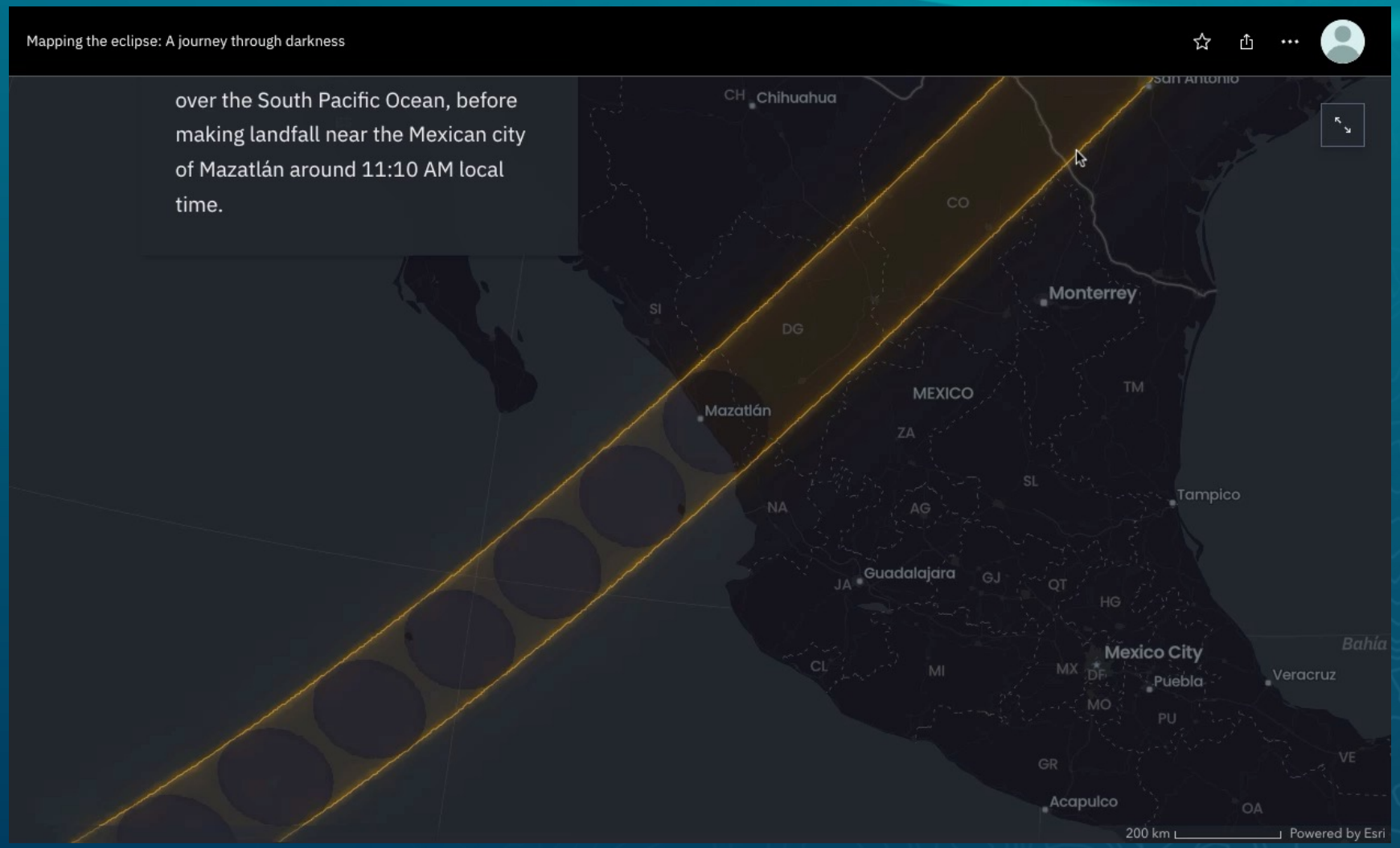


Animations

Why use them?

Storytelling

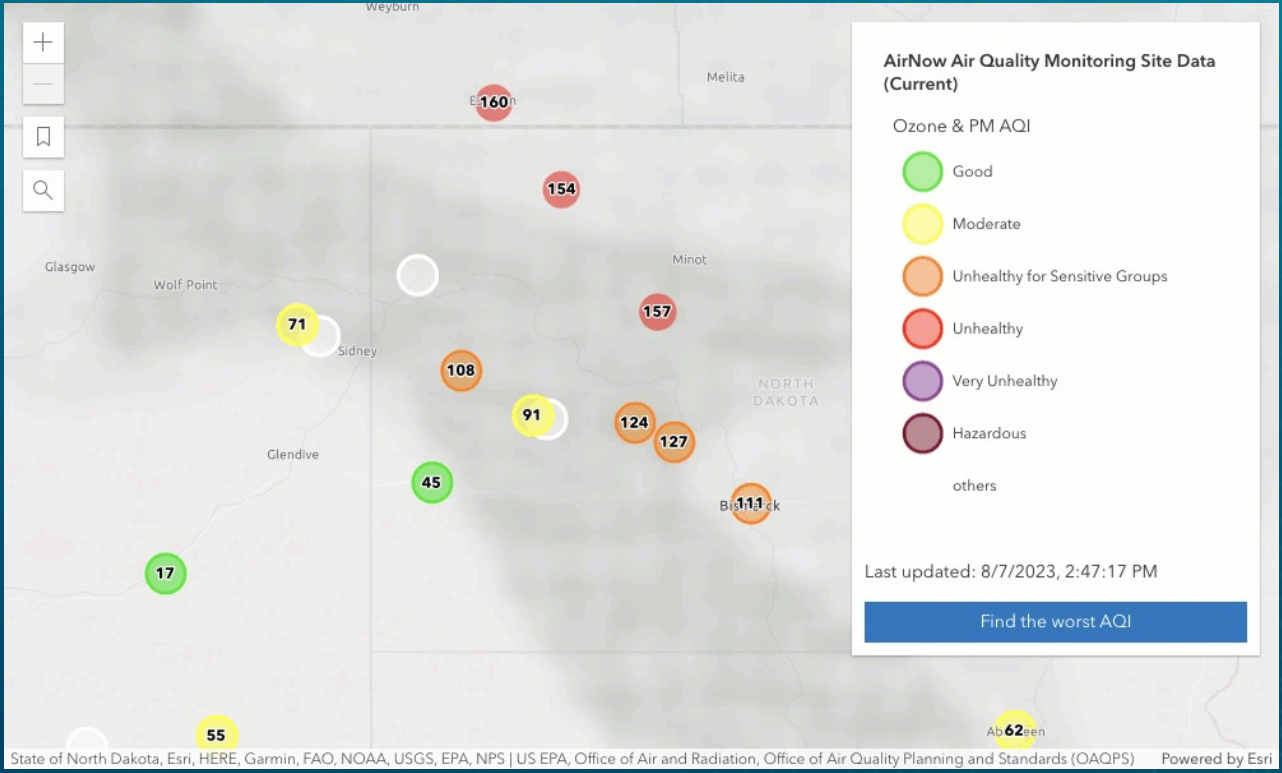
- Guide viewers through a sequence of visualizations.



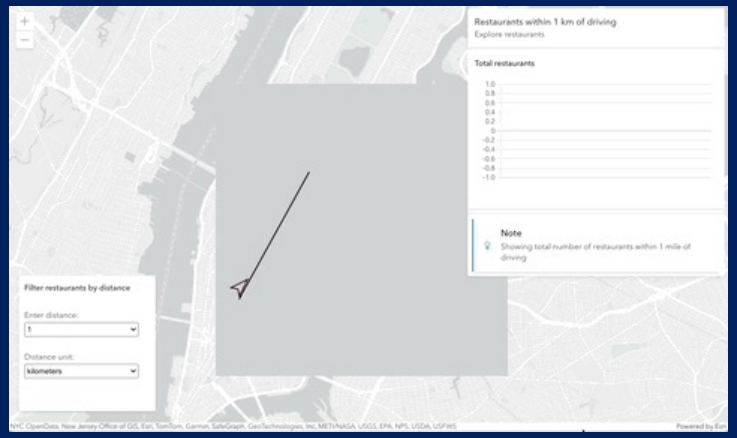
Animations

Why use them?

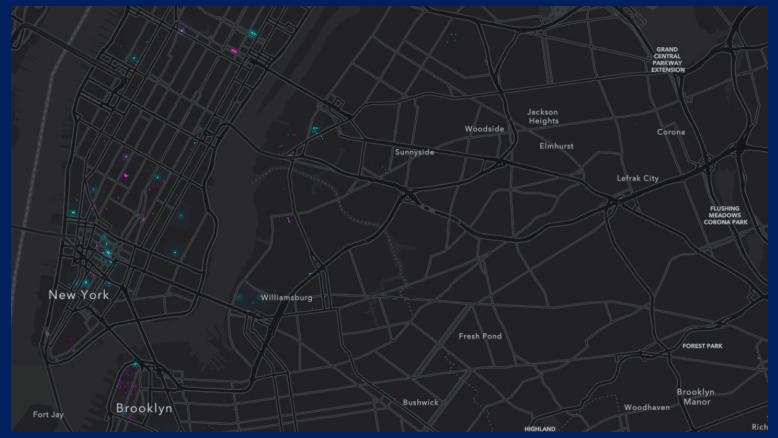
- **Communication**
  - Animations can help quickly communicate critical information to users



Data filter animation

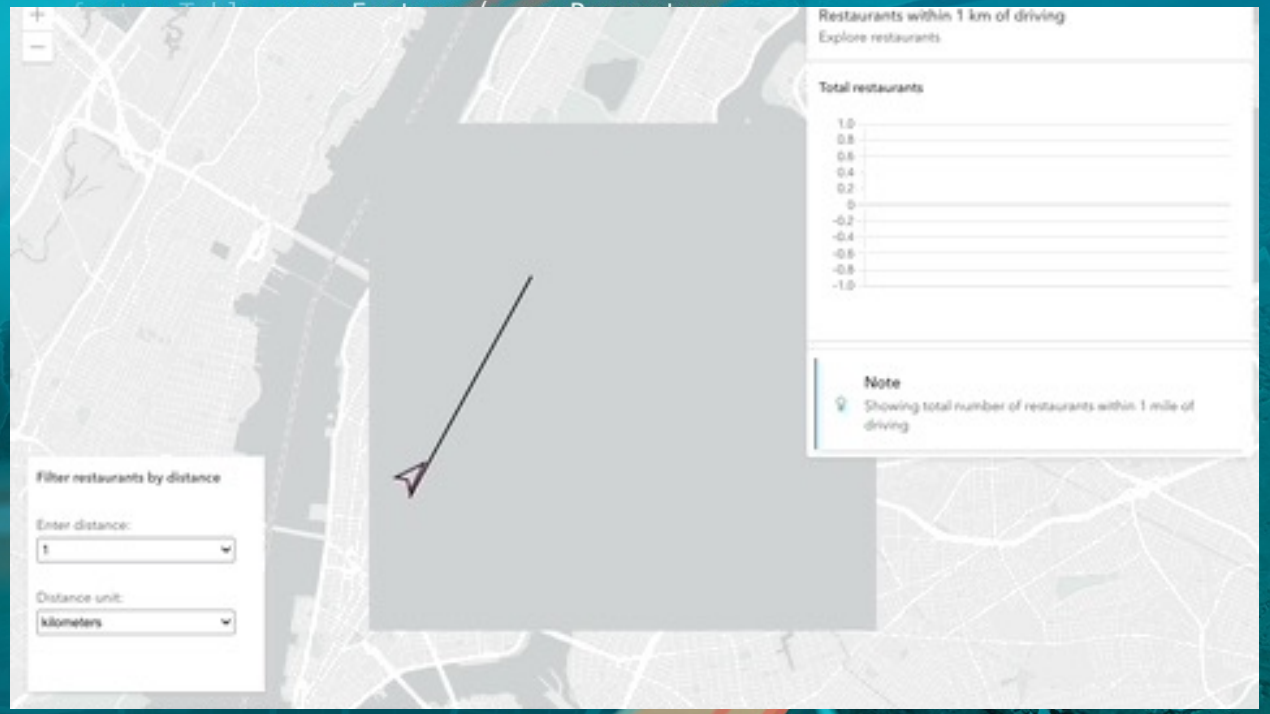


Visual variable animation



Data variable animation

Types of Animation



Data filter animation

Undral Batsukh

Feature based layers

- Attributes and geometries of features can be accessed
- Data filters can be applied
- Uniform API

Server-side layers

- Consume services
- Fetch or stream their features on demand
  - FeatureLayer
  - OGCFeatureLayer
  - StreamLayer
  - SceneLayer

Client-side layers

- Fetch all of their features at once on load
- No feature fetching once loaded unless requested
  - Featurelayer with feature collection
  - CSVLayer
  - GeoJSONLayer
  - WFSLayer

Data filter animation

- Filter data by attributes, geometry or time extent
  - Attributes filters - layer.definitionExpression, layerView.filter.where or layerView.featureEffect
  - Geometry filters - layerView.filter.where or layerView.featureEffect
  - Temporal filters – requires service timeInfo
    - Layer.timeExtent, layerView.filter.timeExtent, layerView.featureEffect.timeExtent
    - Use TimeSlider to animate temporal data
    - For date fields use definitionExpression, layerView.filter.where or layerView.featureEffect.filter.where
  - Filters can be combined

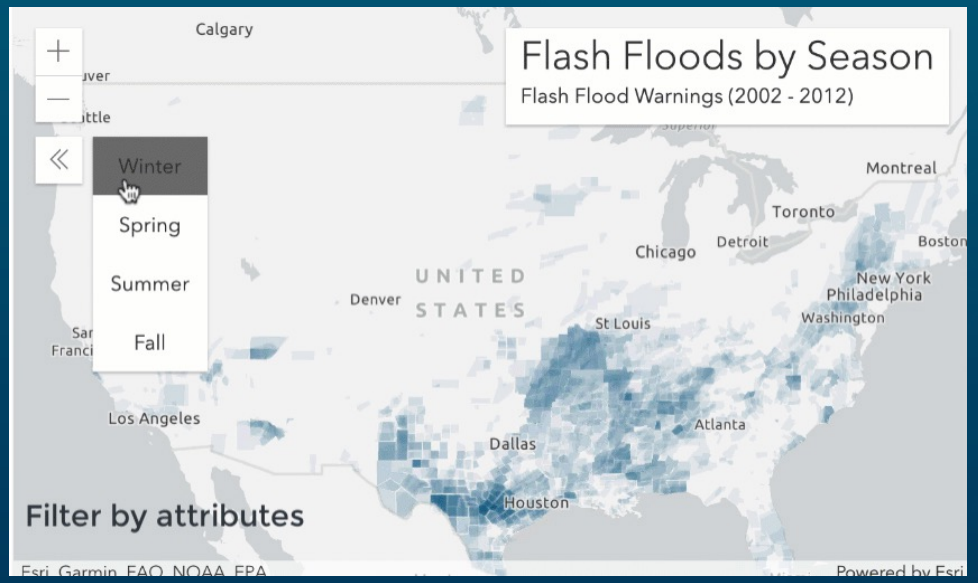
- Renderer is fixed

Filter data by attributes

- Set layer's definitionExpression
  - All features in the layer is used

```
// Set definition expression in constructor to only
// display earthquakes magnitude 5.0 and greater
const layer = new CSVLayer({
  url: "/feed/v1.0/summary/2.5_week.csv",
  definitionExpression: "mag >= 5"
});
map.add(featureLayer);
```

- (Feature|OGCFeature|CSV|GeoJSON|Stream|WFS)LayerView
  - Set layerview's filter.where or featureEffect.filter.where parameter
  - Features available for drawing are used



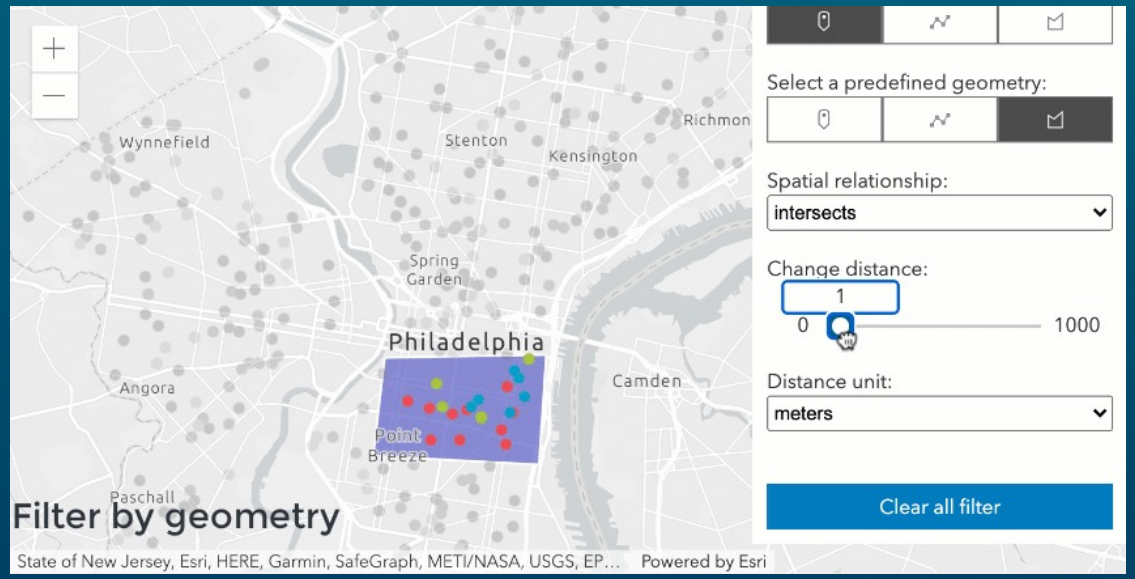

```
// flash flood warnings layer
const layer = new FeatureLayer({
  portalItem: {
    id: "f9e348953b3848ec8b69964d5bceae02"
  },
  outFields: ["SEASON"]
});

view.whenLayerView(layer).then(function (layerView) {
  floodLayerView = layerView;
});

function filterBySeason(event) {
  const selectedSeason = event.target.getAttribute("data-season");
  floodLayerView.filter = {
    where: "Season = '" + selectedSeason + "'"
  };
}
```

Demo 1
Bevery Hills Trees

Live app
Source code

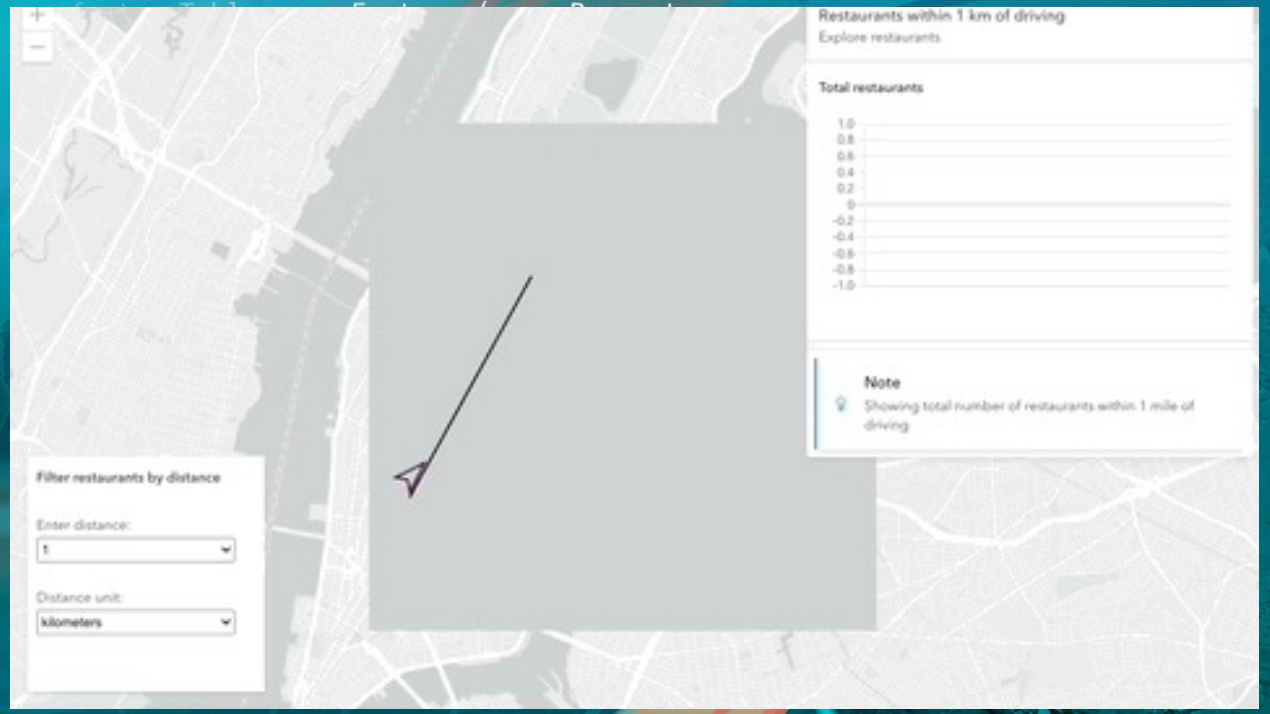
Filter data by geometry

- (Feature|OGCFeature|CSV|GeoJSON|Stream)LayerView
  - Set LayerView's filter.geometry, filter.distance, filter.unit or
  - Set LayerView's featureEffect.filter.geomery, effect.filter.distance, effect.filter.unit




```
view.whenLayerView(layer).then(function (layerView) {
  featureLayerView = layerView;
});
```

```
// set the geometry filter on the visible FeatureLayerView
function updateFilter() {
  featureFilter = {
    // autocasts to FeatureFilter
    geometry: filterGeometry,
    spatialRelationship: "intersects",
    distance: distance,
    units: unit
  };
  // set effect on excluded features
  // make them gray and transparent
  if (featureLayerView) {
    featureLayerView.effect = {
      filter: featureFilter,
      excludedEffect: "grayscale(100%) opacity(30%)"
    };
  }
}
```

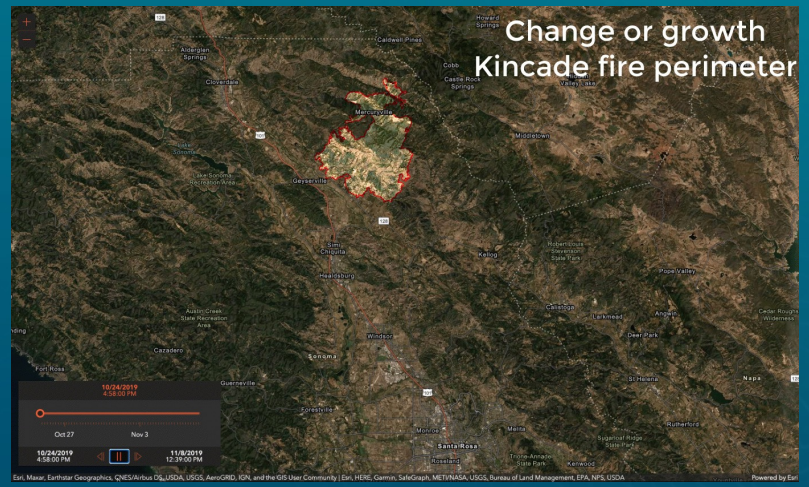


Demo 2
New York restaurants

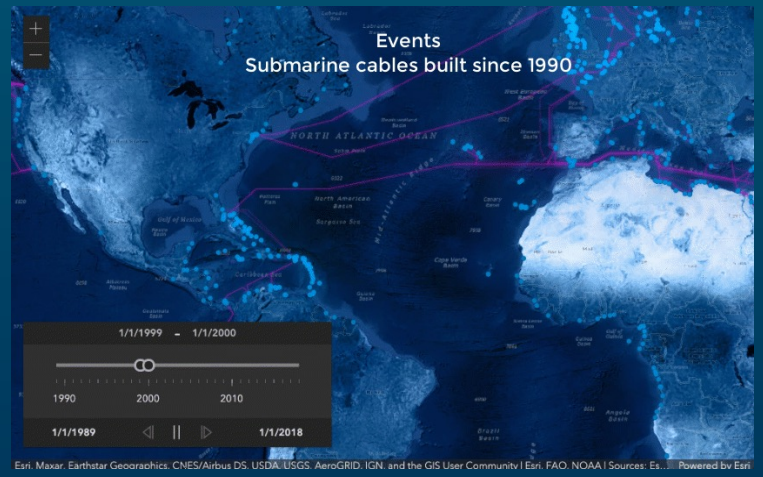
[Live app]
[Source code]

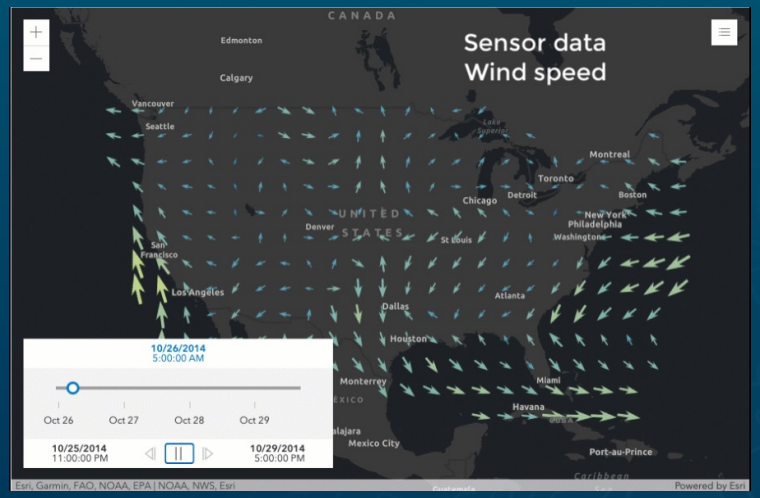
Animating temporal data

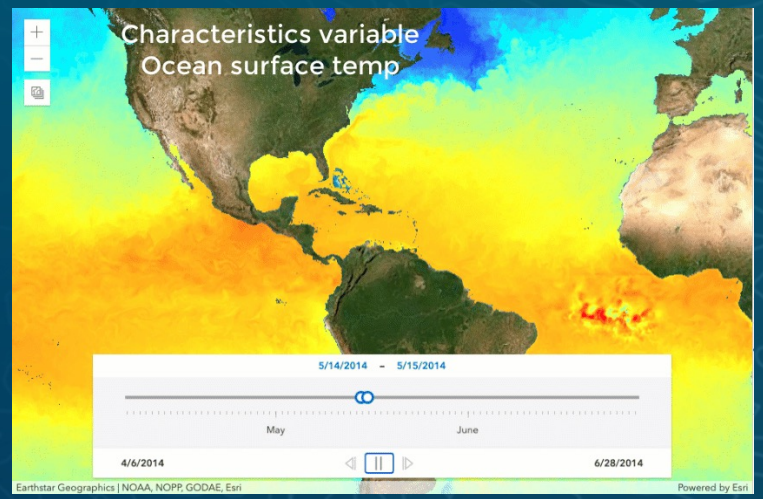
- Detect change or pattern over time
- Location/geometry changes over time



- Stationary objects







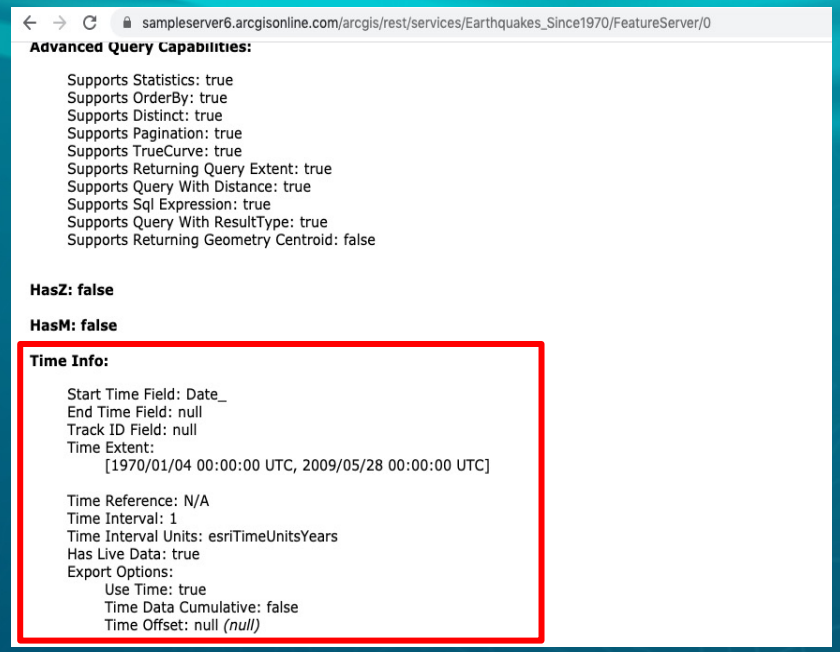
Animating temporal data with TimeSlider

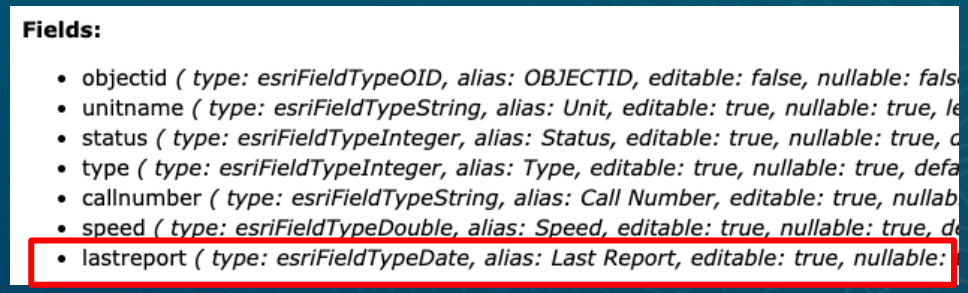
Makes temporal animation simple

1. Animate time enabled services or layer
   - Service (map service, feature service, image service) must have timeInfo
   - Client-side layers with timeInfo defined
   - Set TimeSlider's view property

2. Animate data using date field
   - Layer must have a date field
   - Watch TimeSlider.timeExtent property

     Filter data by date using layer's definitionExpression, layerView.effect or layerView.filter





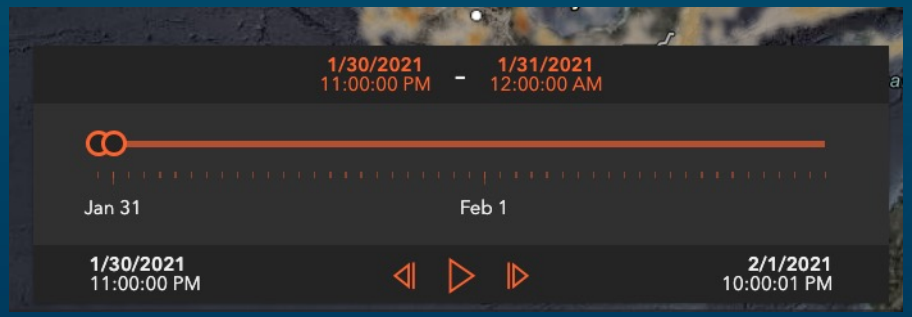
Animating time enabled services

- Map, feature and image services can be published with time info
- Set TimeSlider.view property

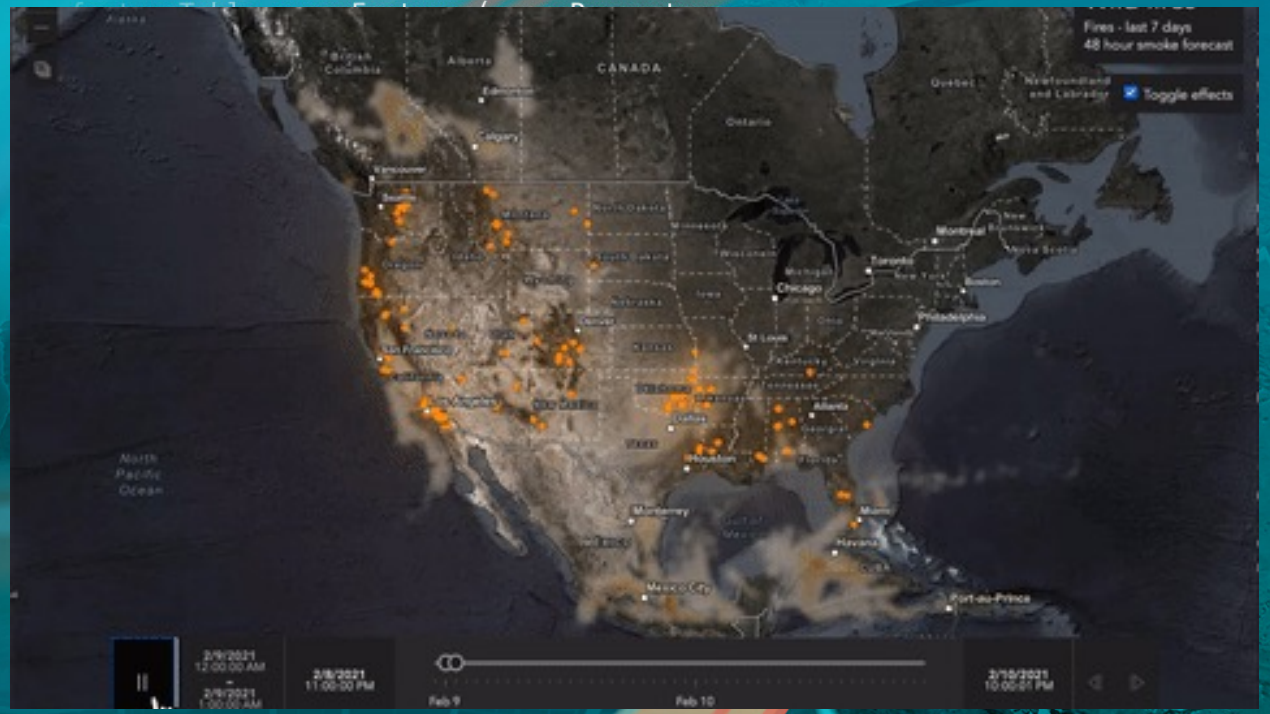
Time Info:

Start Time Field: todate
End Time Field: null
Track ID Field: null
Time Extent
[1/31/2021 7:00:00 AM UTC, 2/2/2021 6:00:01 AM UTC]

Time Reference: UTC
Time Interval: 1
Time Interval Units: esriTimeUnitsHours
Has Live Data: false
Export Options:
Use Time: false
Time Data Cumulative: false
Time Offset: 0
Time Offset Units: esriTimeUnitsCenturies




```
const smokeLayer = new FeatureLayer({
  url: "https://services9.arcgis.com/RHVPKKiFTONKtxq3/arcgis/rest/services
    /NDGD_SmokeForecast_v1/FeatureServer/0"
});

// set up time slider
view.whenLayerView(smokeLayer).then(function (lv) {
  const { fullTimeExtent, interval } = smokeLayer.timeInfo;

  const timeSlider = new TimeSlider({
    container: "timeSlider",
    mode: "time-window",
    view: view,
    timeVisible: true,
    fullTimeExtent,
    playRate: 100,
    loop: true,
    stops: {
      interval
    }
  });

  view.ui.add(timeSlider, "manual");
});
```



Demo 3
Time enabled smoke data

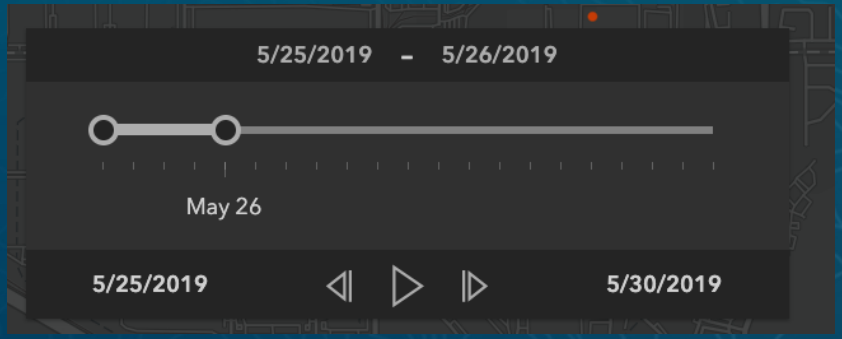
[Live app]
[Source code]

Animating time enabled client-side layers

- Set timeInfo property at the time of initialization for CSVLayer, GeoJSONLayer and client side FeatureLayer
- Data must have a date field
- Set TimeSlider.view property

```
const layer = new GeoJSONLayer({
  url:
    "https://bsvensson.github.io/various-tests/geojson
     /usgs-earthquakes-06182019.geojson",
  copyright: "USGS Earthquakes",
  title: "USGS Earthquakes",
  // set the CSVLayer's timeInfo based on the date field
  timeInfo: {
    startField: "time", // name of the date field
    interval: {
      // set time interval to one day
      unit: "days",
      value: 1
    }
  },
  renderer: {...},
  popupTemplate: {...}
});
```

```
const timeSlider = new TimeSlider({
  container: "timeSlider",
  playRate: 50,
  fullTimeExtent:{
    start: new Date(2019, 4, 25),
    end: new Date(2019, 4, 30)
  },
  stops: {
    interval: layer.timeInfo.interval
  },
  view: view
});
view.ui.add(timeSlider, "bottom-left");
```



Demo 4
Enable timeInfo on CSVLayer

[Live app]
[Source code]

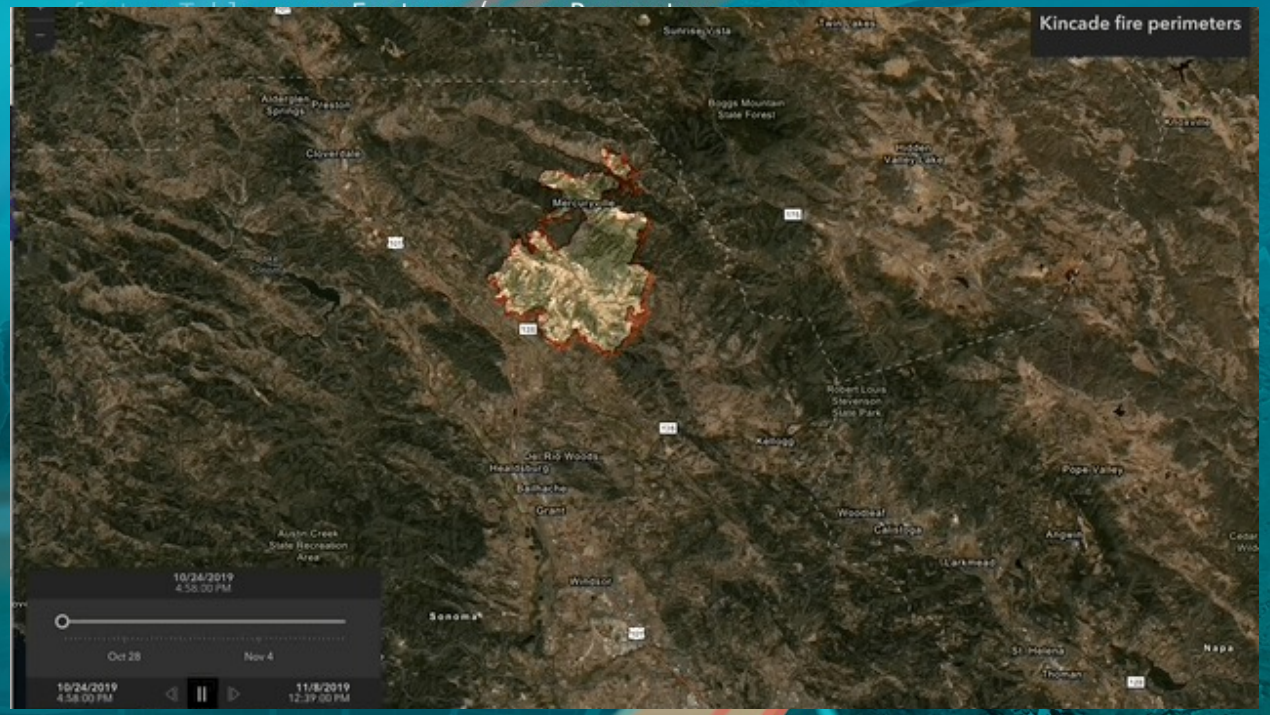
Animating data using date field

- Initialize TimeSlider without view
  - Set TimeSlider properties as needed
  - Watch TimeSlider.timeExtent property and filter data

```
const featureLayer = new FeatureLayer({
  url: "https://services3.arcgis.com/T4QMspbfLg3qTGWY
    /arcgis/rest/services/Historic_GeoMAC_Perimeters_2019
    /FeatureServer/0",
  definitionExpression: "incidentname = 'KINCADE'",
  outFields: ["*"],
  renderer: {...},
  effect: "brightness(600%)",
  blendMode: "overlay",
});
```

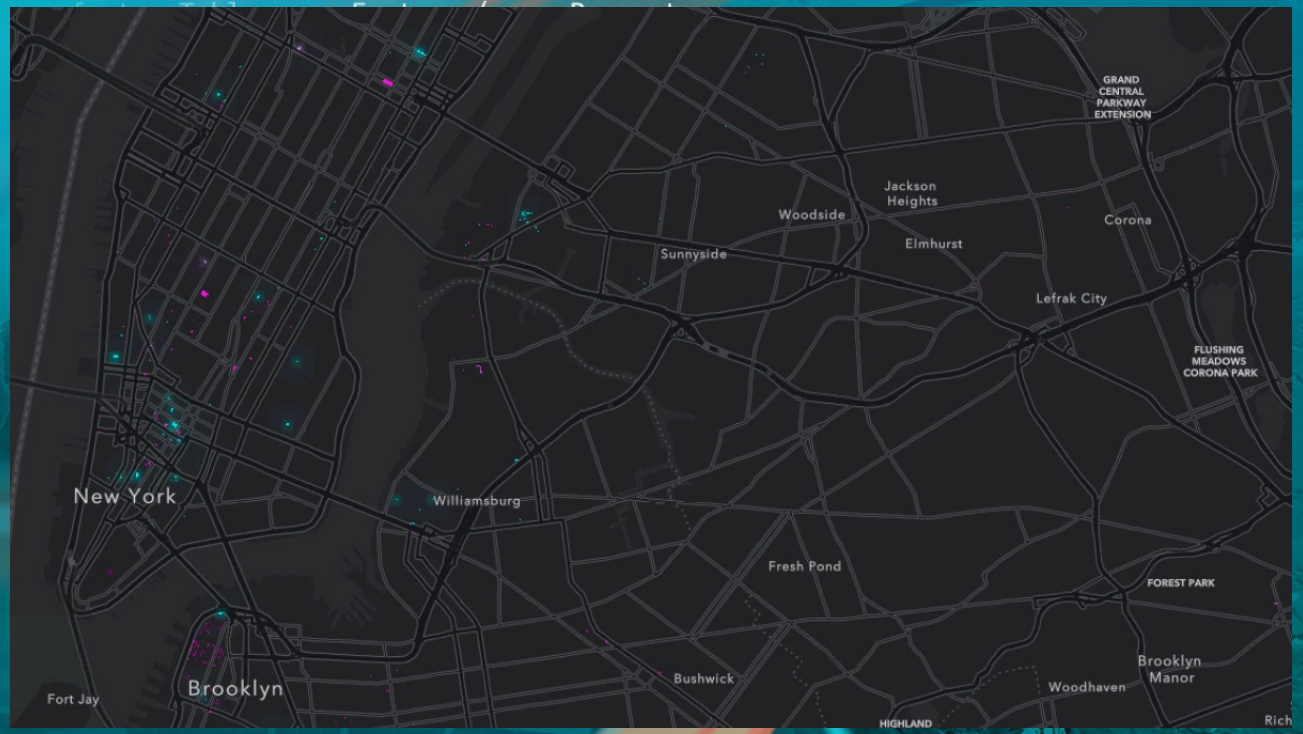
```
// add a new time slider without view settings
const timeSlider = new TimeSlider({
  container: "timeSlider",
  mode: "instant",
  timeVisible: true,
  playRate: 200,
  stops: {dates},
  fullTimeExtent: new TimeExtent({
    start: dates[0],
    end: dates[dates.length-1]
  })
});
```

```
timeSlider.watch("timeExtent", () => {
  const start = timeSlider.timeExtent.start.toISOString().replace("T"," ").substring(0, 19);
  const where = `perimeterdatetime = TIMESTAMP '${start}'`;
  layerView.filter = new FeatureFilter({
    where
  });
});
```



Demo 5
Kincade fire perimeters

Live app
Source code



Visual variable animation

Anne Fitz

Visual variable animation

- Fixed geometries/locations
  - Administrative boundaries (Countries/states/counties)
- **Update renderer** on each animation frame or slider change
  - Fixed data variable (usually a date, but could be another variable)
  - Change visual variable stops (color values, size, etc)
- Single field/column of interest in table

Data structure: Date field (or number representing year)

Footprints (Features: 1082433, Selected: 0)

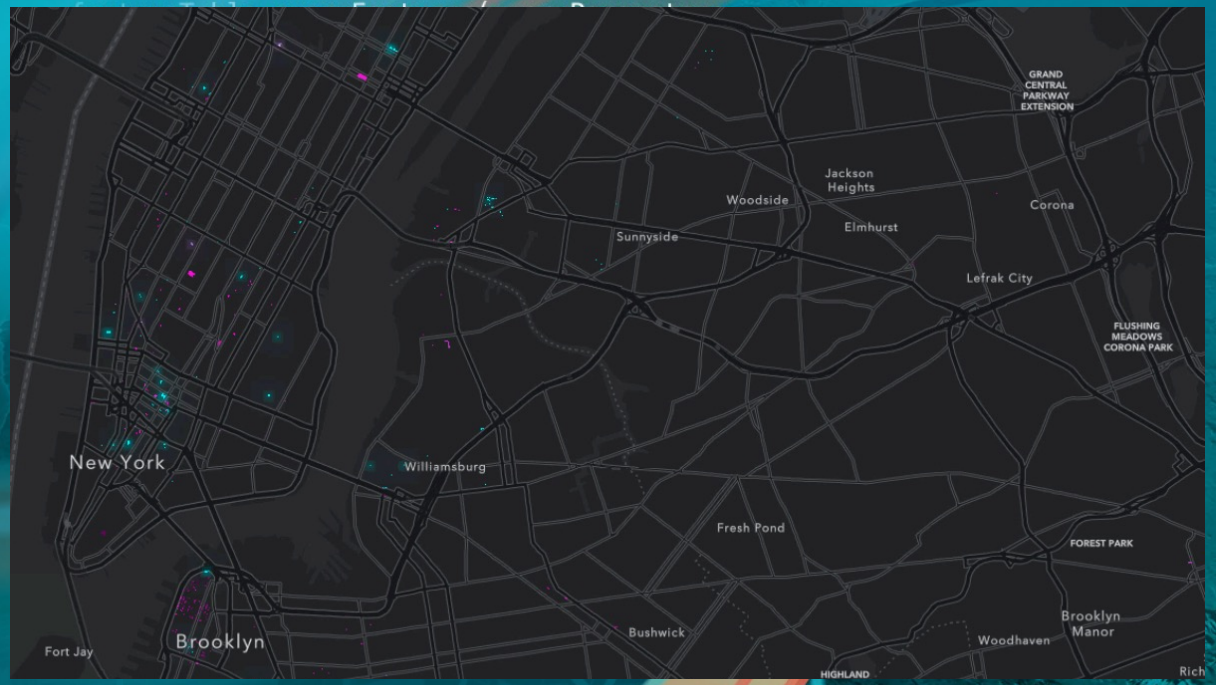
NAME	BIN	BBL	CNSTRCT_YR	LSTMODDATE	LSTSTATYPE	DOITT_ID	HEIGHTROOF	FTRCODE
				PM				
	4,465,036	4163500300	1,938	2/13/2009, 4:00 PM	Constructed	539,338	18.82	2,100
	4,529,251	4159720050	1,920	2/13/2009, 4:00 PM	Constructed	980,512	11.86	5,110
	4,307,100	4163500002	1,929	9/18/2013, 5:00 PM	Constructed	137,895	22.31	2,100
	4,306,730	4163020016	1,920	2/13/2009, 4:00 PM	Constructed	730,180	35.62	2,100
	4,304,594	4162180003	1,930	2/13/2009, 4:00 PM	Constructed	408,322	37.11	2,100
	4,531,671	4156210015	2,006	6/8/2008, 5:00 PM	Constructed	998,124	34.62	2,100
	4,305,555	4162540064	1,935	2/13/2009, 4:00 PM	Constructed	599,241	22.40	2,100
	4,306,123	4162740064	1,930	2/13/2009, 4:00 PM	Constructed	546,445	30.61	2,100
	4,448,991	4161230094	1,930	2/13/2009, 4:00	Constructed	8,479	11.29	2,100

Same data attribute…new visual variable stops…

```javascript
function animate(startValue) {
  var animating = true;
  var value = startValue;

  var frame = function (timestamp) {
    if (!animating) {
      return;
    }

    value += 0.5;
    if (value > 2017) {
      value = 1880;
    }

    setYear(value);

    // Update at 30fps
    setTimeout(function () {
      requestAnimationFrame(frame);
    }, 1000 / 30);
  };

  frame();

  return {
    remove: function () {
      animating = false;
    }
  };
}
```

```javascript
function setYear(value) {
  sliderValue.innerHTML = Math.floor(value);
  slider.viewModel.setValue(0, value);
  layer.renderer = createRenderer(value);
}
```

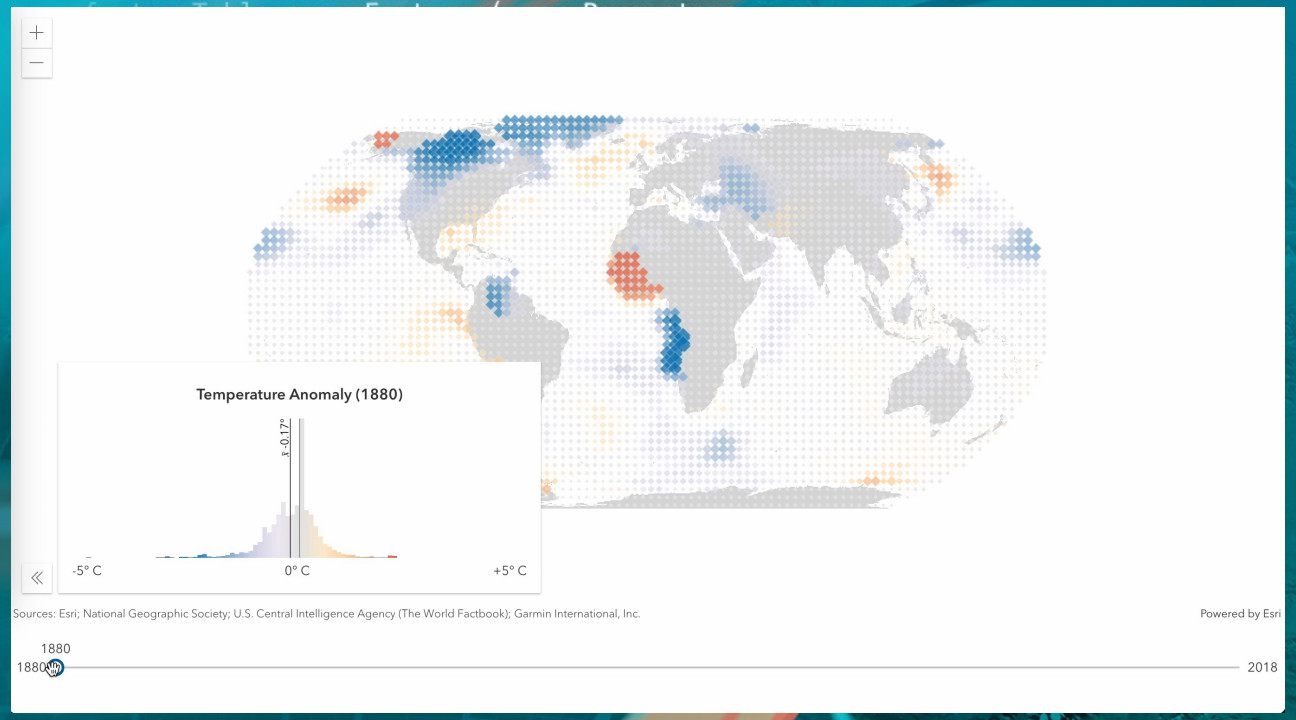
```javascript
{
  type: "color",
  field: "CNSTRCT_YR",
  legendOptions: {
    title: "Built:"
  },
  stops: [
    {
      value: year,
      color: "#0ff",
      label: "in " + Math.floor(year)
    },
    {
      value: year - 10,
      color: "#f0f",
      label: "in " + (Math.floor(year) - 20)
    },
    {
      value: year - 50,
      color: "#404",
      label: "before " + (Math.floor(year) - 50)
    }
  ]
}
```



Demo 6
New York Construction

Live app

Source code



Data variable animation

Anne Fitz

Data variable animation

- Fixed geometries/locations
- **Update renderer** on each animation frame or slider change
  - Changing data variable
    - Column name represents a year, but could be another dimension (like depth)
    - Data value could represent anything (temperature, population, votes, etc)
  - Fixed visual variable stops (color values, size, etc.) – IMPORTANT
- Requires one field per attribute per time interval

Preparing the data

Global_Temperatures_1880_to_2018_not_pivoted | Overview | Data | Visualization | Usage | Settings

Table | Fields

Editing is disabled but you have privileges to edit this layer.

Double-click a value in the table to change it.

Data updated: Mar 8, 2023, 12:00 PM

Global_Temperatures_1880_to_2018_not_pivoted_points (Features: 603816, Selected: 0)

YEAR	VALUE
1,907	-0.3941666
1,908	-0.6575
1,909	-0.4216667
1,910	-0.5866666
1,911	-0.7125
1,912	[illegible]

600k point features
2 fields: year & temperature

4344 point features
140 fields: one field per year

Global Temperatures 1880 to 2018 | Overview | Data | Visualization

Table | Fields

Data updated: Jun 13, 2019, 1:01 PM

GlobalTemp_AnnualMean (Features: 4344, Selected: 0)

1907	1908	1909	1910	1911	1912	1913	1914	1915
-0.0533333253115	-0.1424999913821	-0.3774999907861	-0.3574999892152	-0.7833333114782	-0.5266666534977	0.03916666532556	-0.0800000006953	0.233333325634
-0.394166659719	-0.6574999863902	-0.4216666559999	-0.5866666585206	-0.7124999935428	-0.2908333237282	-0.3341666537647	-0.1699999962002	-0.219166661147
-1.4238888389534	-1.3124999586078	-0.0808333359244	-0.1047222106200	0.00416666972968	-0.7627777649193	-0.8597221954001	-0.2822222223298	-0.205833331164
-0.0563888891289	-0.3786111001649	-0.2541666599507	-0.1616666619148	-0.2452777682079	-0.7524999810589	-0.6813888736069	-0.7274999904135	-0.333611101222
-0.9958333012958	-0.7016666376342	-0.4558333454964	-2.1449999958276	-1.1674999945486	-0.7666666454946	0.34083331966151	0.63833331316709	-0.524166652311
-0.3388888835906	-0.4899999863571	-0.5644444392787	-0.5155555473433	-0.8055555373430	0.07555555697116	-0.0122222242256	-0.0166666646384	0.136666663818
-0.0679166656142	-0.5999999865889	-0.5370833243553	-0.6945833202141	-0.5154166519641	-0.4395833266898	-0.5299999877655	-0.3758333244671	-0.472916654543
-0.3124999929374	-0.4762499881908	-0.6608333125089	-0.7545833153029	-0.4479166573534	-0.1949999959518	-0.3537499923259	-0.1937499997826	-0.179999993725
-1.1270832996815	-1.6599999675527	-0.0675000017508	-0.2408333361769	-0.2208333324330	-1.1570832890768	-0.9616666398942	0.09250000274429	-0.127916662798
0.09749999735504	-0.2658333286332	-0.1258333292789	-0.1129166651517	-0.0895833338145	-0.8391666443397	-0.7545833168551	-0.7945833193759	-0.499583316346

Update data attribute on animation frame

```
function updateRenderer(value) {
  renderer = layer.renderer.clone();
  const sizeVariable = renderer.visualVariables[0];
  const colorVariable = renderer.visualVariables[1];
  sizeVariable.valueExpression = getSizeValueExpression(value);
  colorVariable.valueExpression = getColorValueExpression(value);
  renderer.visualVariables = [ sizeVariable, colorVariable ];
  layer.renderer = renderer;
}
```

Visual variable stops stay the same!
Only the Arcade expression updates

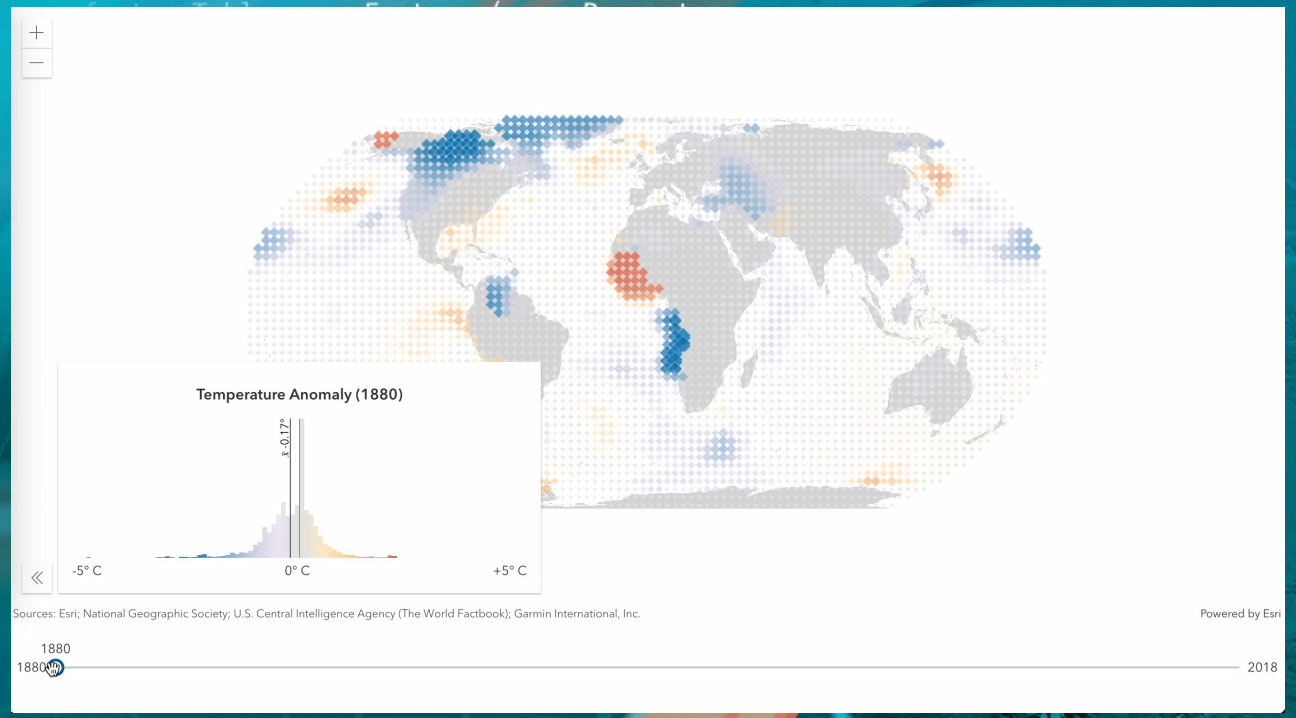
Update data attribute on animation frame

```
function updateRenderer(value) {
  renderer = layer.renderer.clone();
  const sizeVariable = renderer.visualVariables[0];
  const colorVariable = renderer.visualVariables[1];
  sizeVariable.valueExpression = getSizeValueExpression(value);
  colorVariable.valueExpression = getColorValueExpression(value);
  renderer.visualVariables = [ sizeVariable, colorVariable ];
  layer.renderer = renderer;
}
```

```
function getSizeValueExpression(value) {
  return `
    var AbsTEMP = Abs($feature.F${value});
    var vs = $view.scale;
    var TempSize = when(
      AbsTEMP > 5, 35,
      AbsTEMP > 4, 30,
      AbsTEMP > 2.5, 25,
      AbsTEMP > 1, 20,
      AbsTEMP > 0.5, 15,
      AbsTEMP > 0.01, 12,
      AbsTEMP < 0.01, 10,
      8
    );
    when(
      vs >=37000000, TempSize,
      vs >=18500000, 2 + TempSize,
      vs >=9300000, 4 + TempSize,
      vs >=4700000, 6 + TempSize,
      vs >=2000000, 8 + TempSize,
      10 + TempSize
    );
  `;
}
```

Update data variable used in the Arcade expression

Live demo



Demo 7
Global Temperature Anomalies

Live app

Source code

Symbol Animations

Anne Fitz

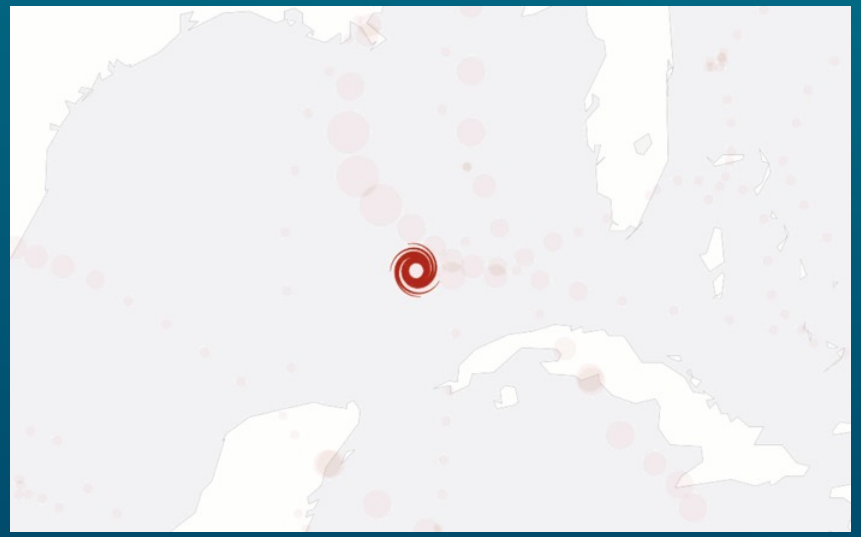
```
view.goTo({
  center: [-126, 49]
})
.catch(function(error) {
  if (error.name != "AbortError") {
    console.error(error);
  }
});
```

```
queryParameters =
QueryParameters().apply {
  whereClause = "price > 200"
}
viewModelScope.launch {
```

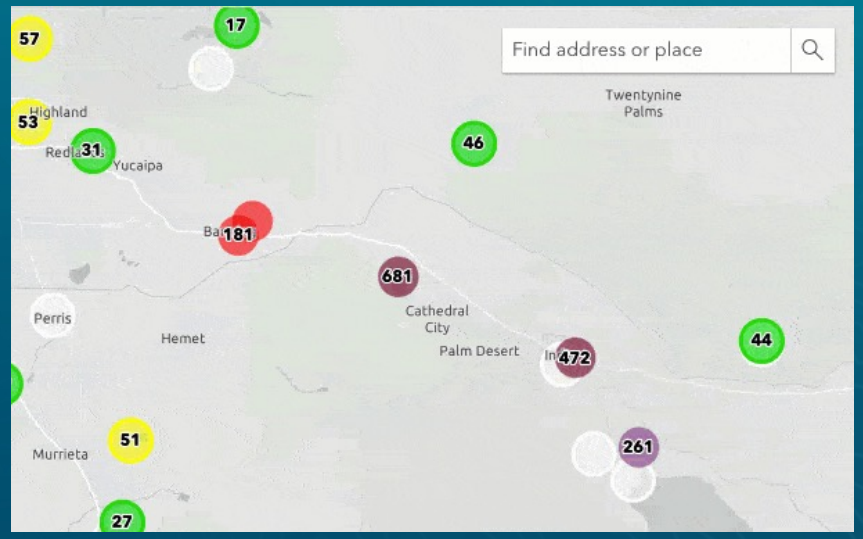
Animated Symbols

Take advantage of GIFs!

PictureMarkerSymbol



CIMSymbol



- Multi-layer symbols supported
- Adjust the properties of animation (duration, repeat type, etc)
- Can be **data driven**!

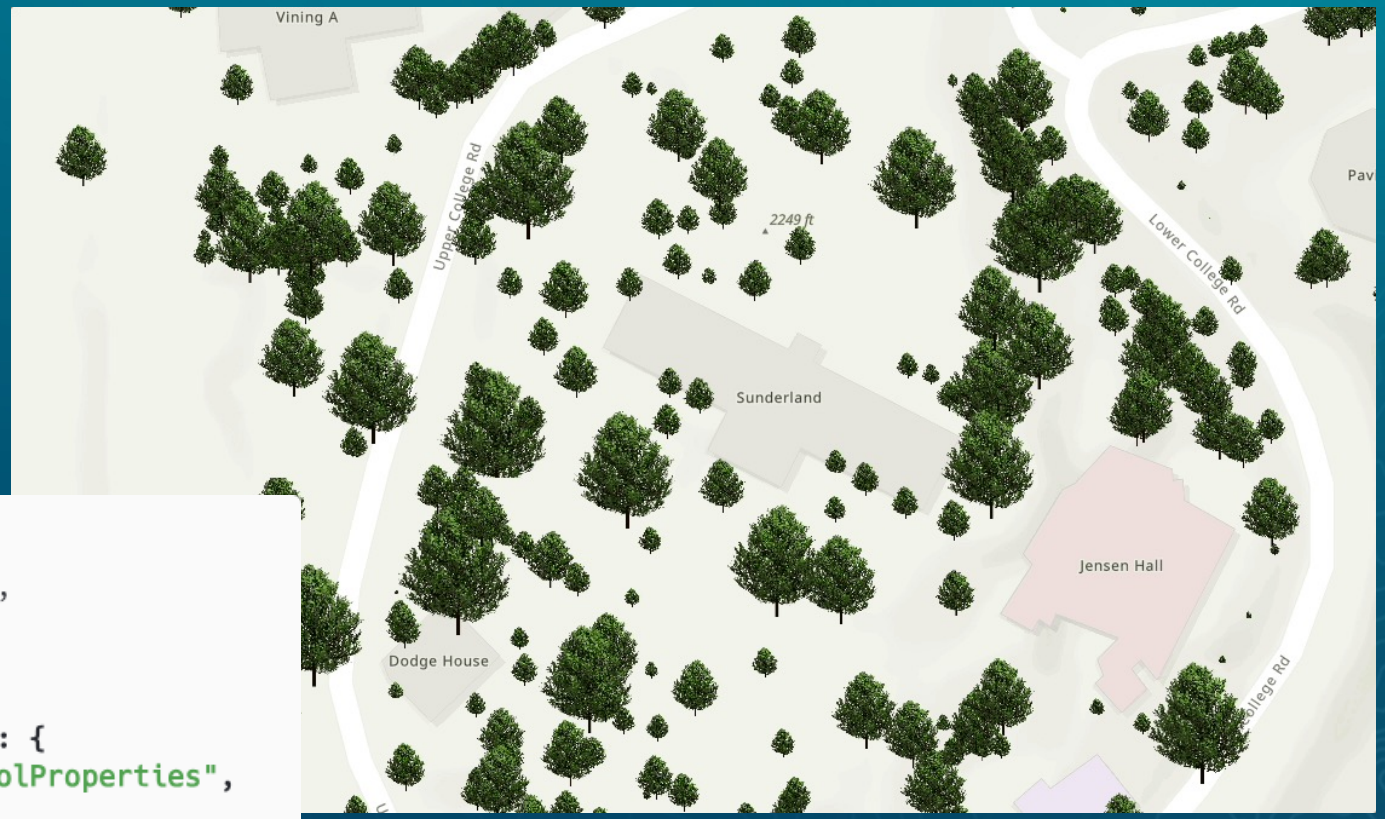
CIM Animated Symbols

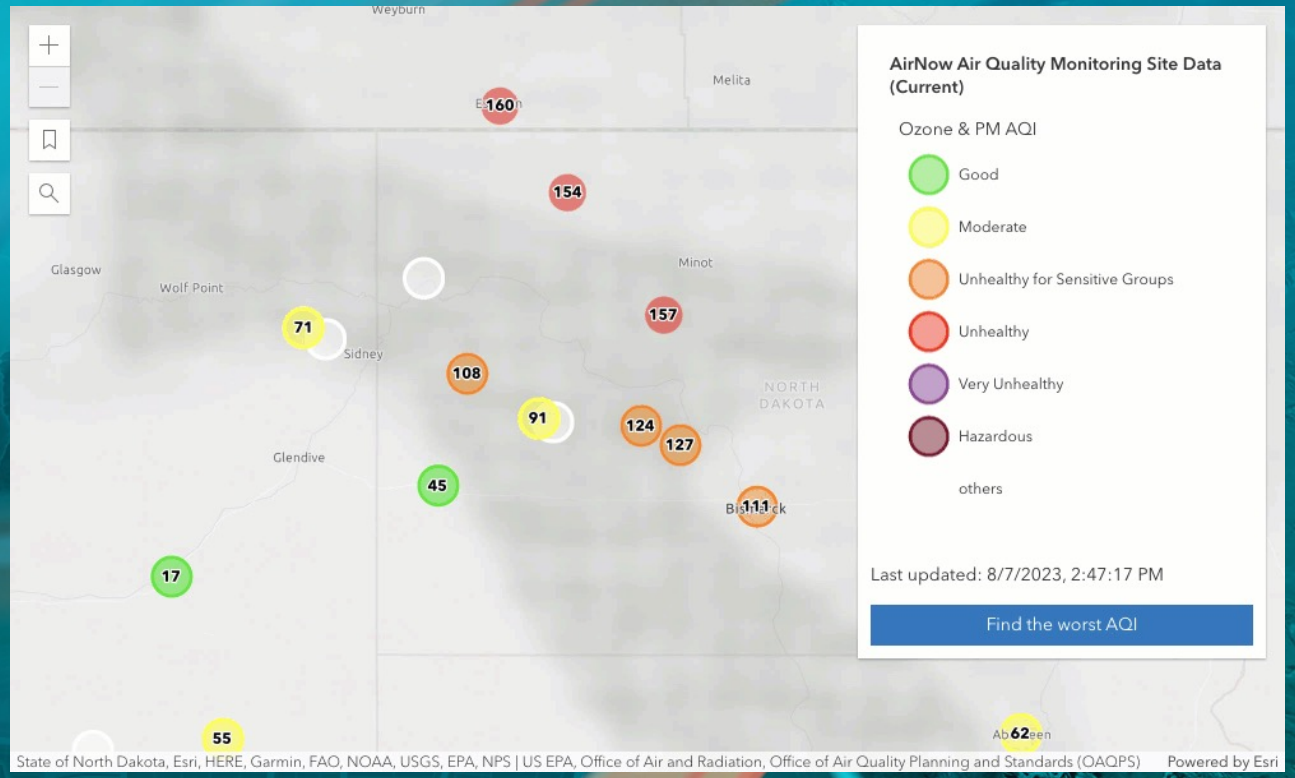
Animated symbol properties

- **CIMAnimationSymbolProperties**
  - playAnimation
  - reverseAnimation
  - randomizeStartTime
  - duration
  - repeatType

Can be data driven using primitive overrides!

```
{
  type: "CIMPictureMarker",
  enable: true,
  size: 10,
  url: "url to GIF",
  animatedSymbolProperties: {
    type: "CIMAnimatedSymbolProperties",
    playAnimation: true,
    randomizeStartTime: true,
    repeatType: "Loop"
  }
}
```





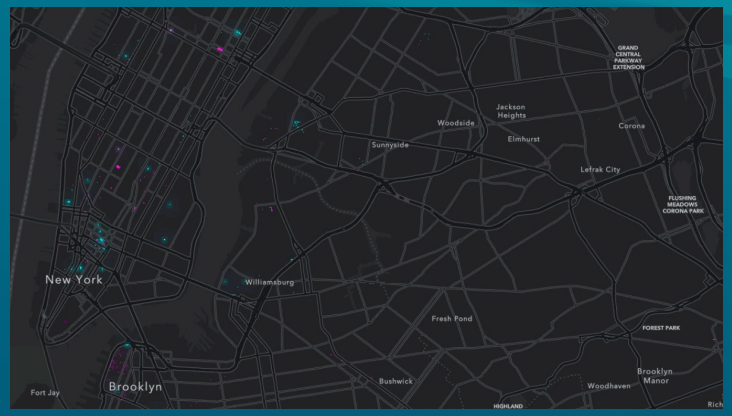
Demo 8
Air Quality

Live app

Source code

Conclusion

How do I know which technique to use?

	Data filter animation	Visual variable animation	Data variable animation
			
Animate locations? Changing geometry?	✓	✗	✗
A single event in time and location?	✓	✗	✗
One data value?	✓	✓	✗
Changing data values in same location?	✗	✗	✓

Useful presentations

- ArcGIS Maps SDK for JavaScript: Client-side Queries & Analysis
- Unveiling the Magic: The Secrets Behind 10 Amazing Web Apps
- ArcGIS Maps SDK for JavaScript: Visualizing Large Datasets
- ArcGIS Maps SDK for JavaScript: Data Visualization

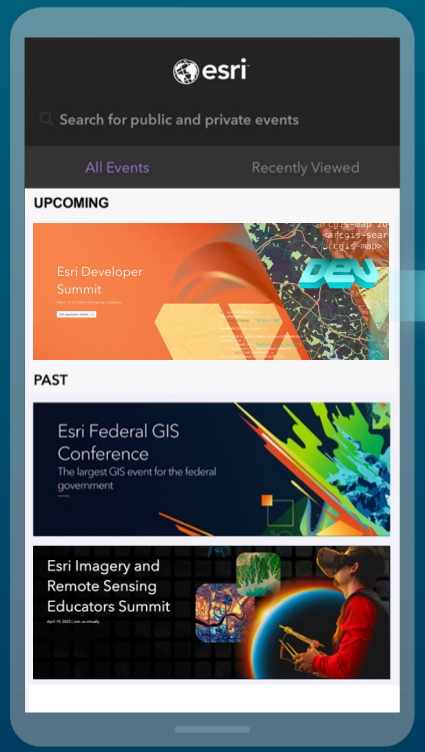
Demo links

Undral’s demos

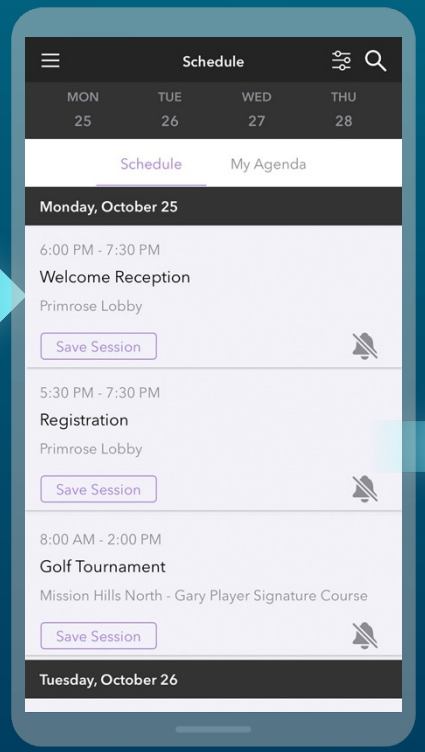
Anne’s demos

Please Share Your Feedback in the App

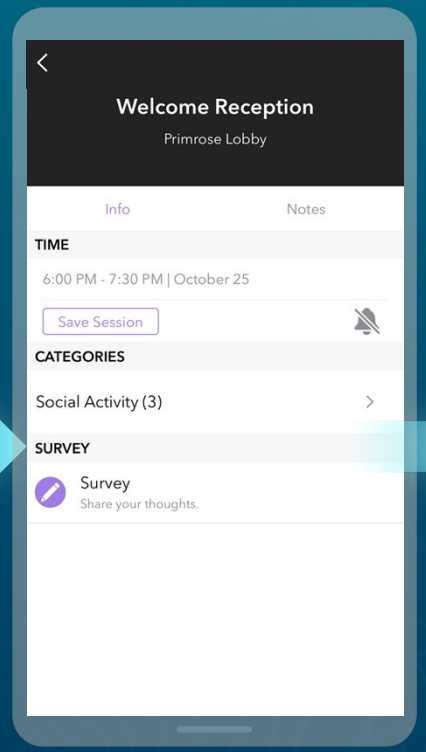
Download the Esri Events app and find your event



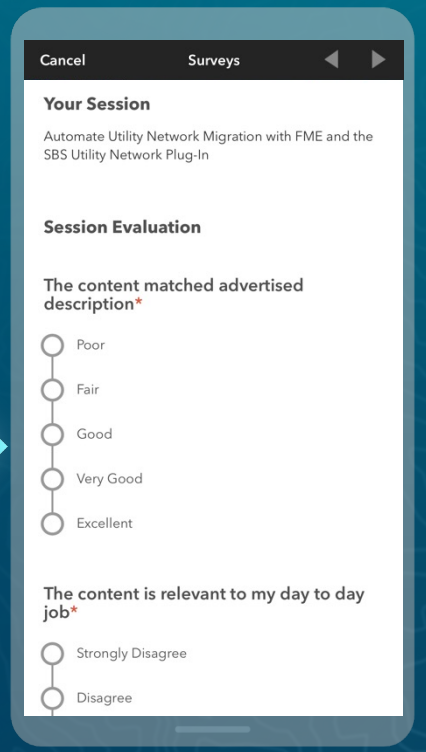
Select the session you attended



Scroll down to "Survey"



Log in to access the survey



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

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

```
// 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)
}
```

```
const layerList = new LayerList({
  view: view
});

// Add widget to the top right cor
of the view
view.ui.add(layerList, "top-right"

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

esri®

THE SCIENCE OF WHERE®

Copyright © 2024 Esri. All rights reserved.