esri

Visualizing Temporal Data with the ArcGIS API for JavaScript

Yann Cabon [@yanncabon] | Richie Carmichael [@kiwiRichie]

https://git.io/fjrzO (printer friendly, pdf)

SEE
WHA
OTHE
CAN'T

Agenda

- Time, Filters and Effects
  - Status of Time Support
  - Configuring Temporal Content
  - Time-based Queries
  - Filters and Effects
  - TimeSlider widget
- Visualizing time using Arcade
- Visualizing time using Visual Variables

About Time!

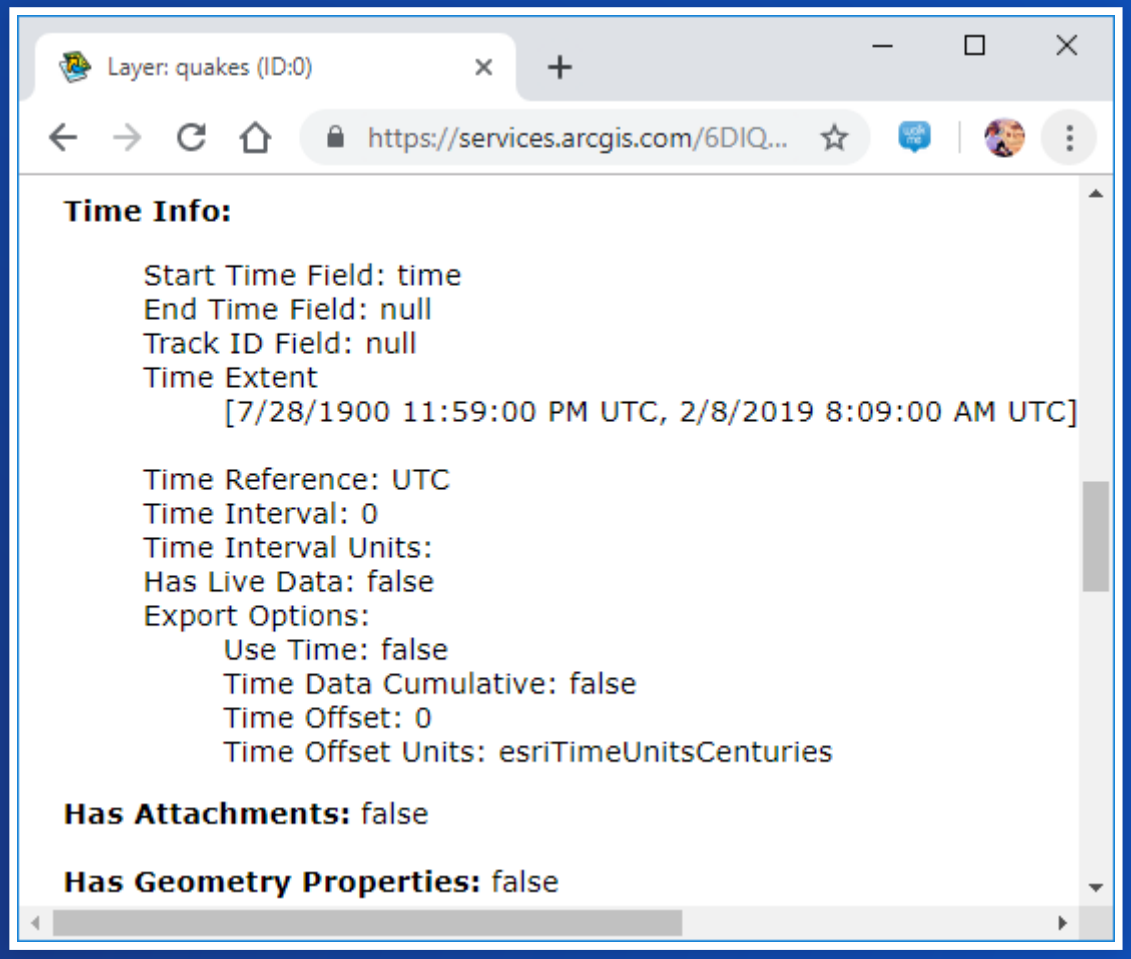
- Time Support @ 4.12
  - Query.timeExtent
  - FeatureLayer, ImageryLayer, MapImageLayer, CSVLayer and GeoJSONLayer
  - MapView.timeExtent & SceneView.timeExtent
  - TimeSlider widget
- Time Support @ 3.29
  - Time offsets, temporal renderer

Publishing Temporal Content

- ArcGIS PRO
  - Set the time properties on data
  - Share as web layer

TimeInfo

Time-aware feature and imagery services require a **TimeInfo** definition.



Defining TimeInfo in Client-Side Layers

```
// Create a layer from a CSV file.
const layer = new CSVLayer({
  url: "ridgecrest-july-2019.csv",
  // Specific geographic fields.
  longitudeField: "longitude",
  latitudeField: "latitude",
  // Specify temporal field.
  timeInfo: {
    startField: "time",
    interval: {
      value: 1,
      unit: "hours"
    }
  }
});
```

Time-based Queries

```
var query = new Query({
  timeExtent: new TimeExtent({
    start: new Date(2000, 0, 1),
    end: new Date(2007, 0, 1)
  })
});
```

```
// Search all features!
var extent   = await featureLayer.queryExtent(query);
var count    = await featureLayer.queryFeatureCount(query)
var features = await featureLayer.queryFeatures(query)
var ids      = await featureLayer.queryObjectIds(query)
```

```
// Only search features downloaded to the browser.
var extent   = await featureLayerView.queryExtent(query);
var count    = await featureLayerView.queryFeatureCount(query)
var features = await featureLayerView.queryFeatures(query)
var ids      = await featureLayerView.queryObjectIds(query)
```

Filters and Effects

- Filter - *Client-side spatial/aspatial/temporal filtering.*

```
// Only show earthquakes that occured between 2000 and 2006.
featureLayerView.filter = new FeatureFilter({
  timeExtent: new TimeExtent({
    start: new Date(2000, 0, 1),
    end: new Date(2007, 0, 1)
  })
});
```

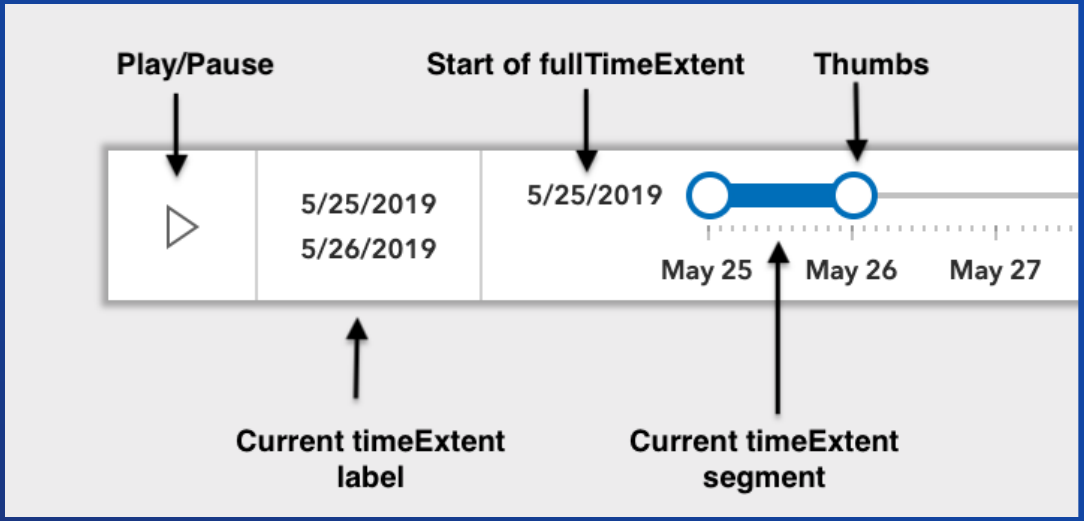
- Effects - *Visual effects applied to included/excluded features.*

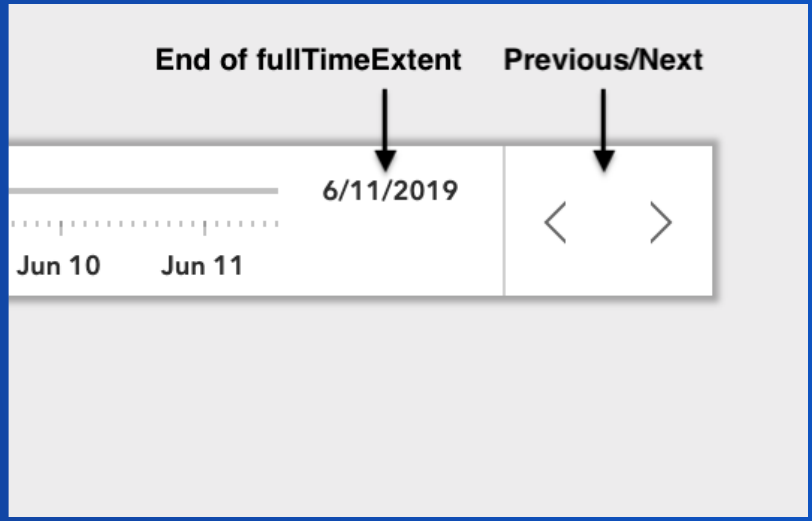
```
// Show quakes less than 7 magnitude as faint shadows.
featureLayerView.effect = new FeatureEffect({
  filter: {
    where: "magnitude >= 7"
  }
  excludedEffect: "grayscale(100%) opacity(0.5)"
});
```

Demo

- Time, Filters and Effects - The Trifecta

TimeSlider Components



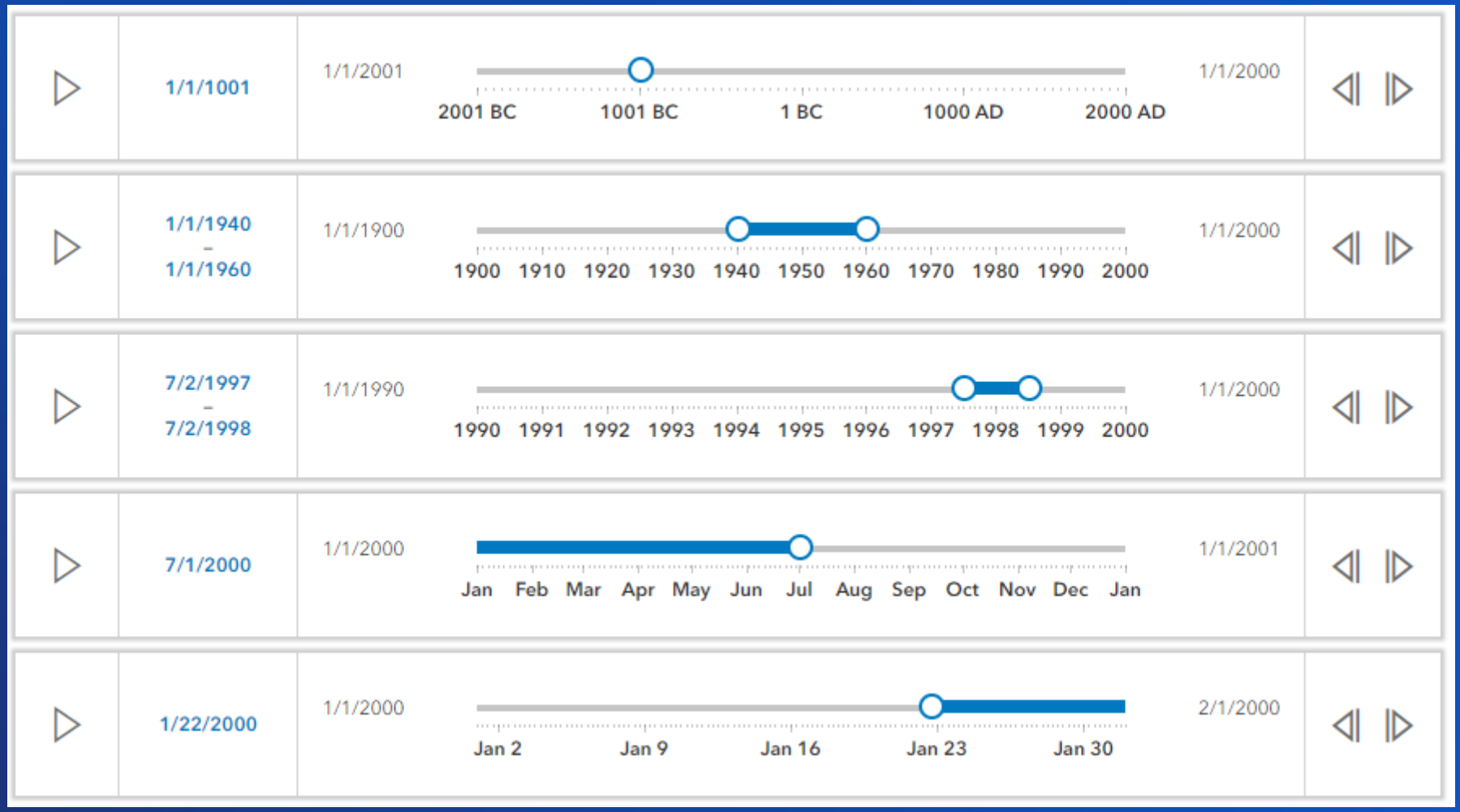


TimeSlider Properties

```
var timeSlider = new TimeSlider({
  container: "timeSliderDiv",       // HTML node
  fullTimeExtent: { start, end }, // Temporal width
  // optional
  mode: "time-window",              // Handle configuration
  playRate: 1000,                   // Play speed
  stops: {                          // Handle stops/steps
    interval: { value: 1, unit: "years" }
  },
  loop: true,                       // Allow looping
  timeVisible: false,               // Show times
  values: [ first, second ],        // Handle positions
  view: view                        // The MapView/SceneView
});
```

TimeSlider Modes

instant | time-window | cumulative-from-start | cumulative-from-end



TimeSlider Stops

```
const timeSlider = new TimeSlider({
  stops: {
    count: 10 // Create ten evenly spaced stops.
  }
});
```

```
const timeSlider = new TimeSlider({
  stops: {
    interval: { // Create stops spaced one year apart.
      value: 1,
      unit: "years"
    }
  }
});
```

```
const timeSlider = new TimeSlider({
  stops: {
    dates: [ // Explicitly create stops at these dates.
      new Date(2000, 0, 1), new Date(2001, 3, 8),
      new Date(2003, 12, 8), new Date(2004, 2, 19)
    ]
  }
});
```

Filter

```
// Only show earthquakes with a magnitude of 7 or greater.
featureLayerView.filter = new FeatureFilter({
  where: "magnitude >= 7"
});
```

```
// Only show buildings within 10 miles of the mouse cursor.
mapView.on("pointer-move", function(e) {
  buildingLayerView.filter = {
    geometry: mapView.toMap({e.x, e.y})
    distance: 10,
    units: "miles"
  }
});
```

API Reference | filter by attribute sample | filter by geometry sample

Effects

```
// Show quakes less than 7 magnitude as faint shadows.
featureLayerView.effect = new FeatureEffect({
  filter: {
    where: "magnitude >= 7"
  }
  excludedEffect: "grayscale(100%) opacity(0.5)"
});
```

```
// brightness(0.4);
// contrast(200%);
// grayscale(50%);
// hue-rotate(90deg);
// invert(75%);
// opacity(25%);
// saturate(30%);
// sepia(60%);
```

CSS reference | CSS filter parser | API Reference | Plenary Demo

Tips for client-side queries

- Wait for layer to be intialized and data downloaded

```
view.whenLayerView(featureLayerQuake).then(function(lv){
  // Layer initalized.
  lv.watch("updating", function(value){
      if (!value){
          // Data finished downloading.
      }
  });
});
```

- Geometry may be generalized
- Restrict queries to data within visual extent

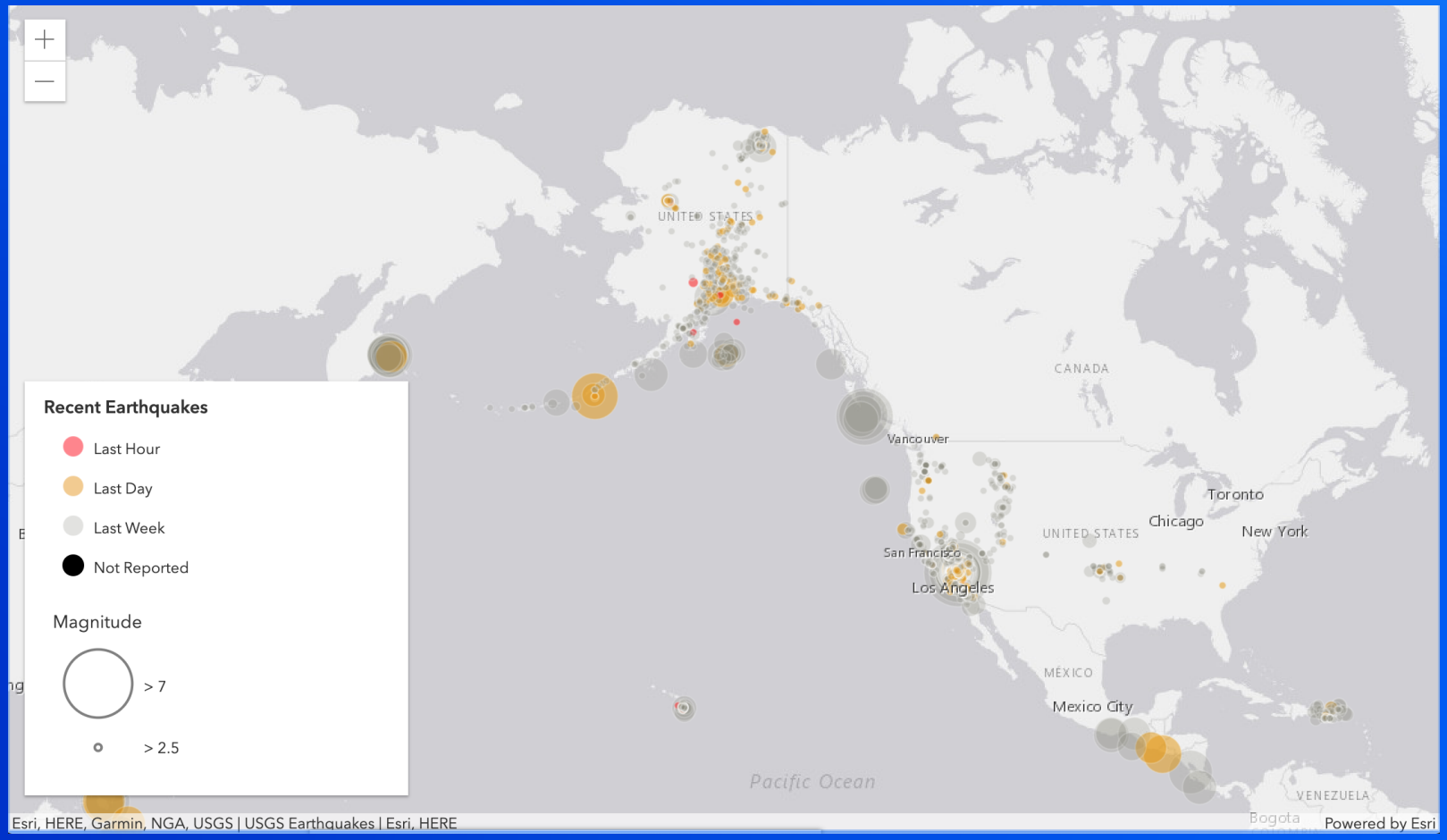
What's coming with Time?

- Time Support @ 4.13
  - "opt-out" for layers
    (i.e. ignore View.timeExtent)
  - Time offset
    (e.g. for analyzing yearly or seasonal trends)
  - Responsive TimeSlider widget

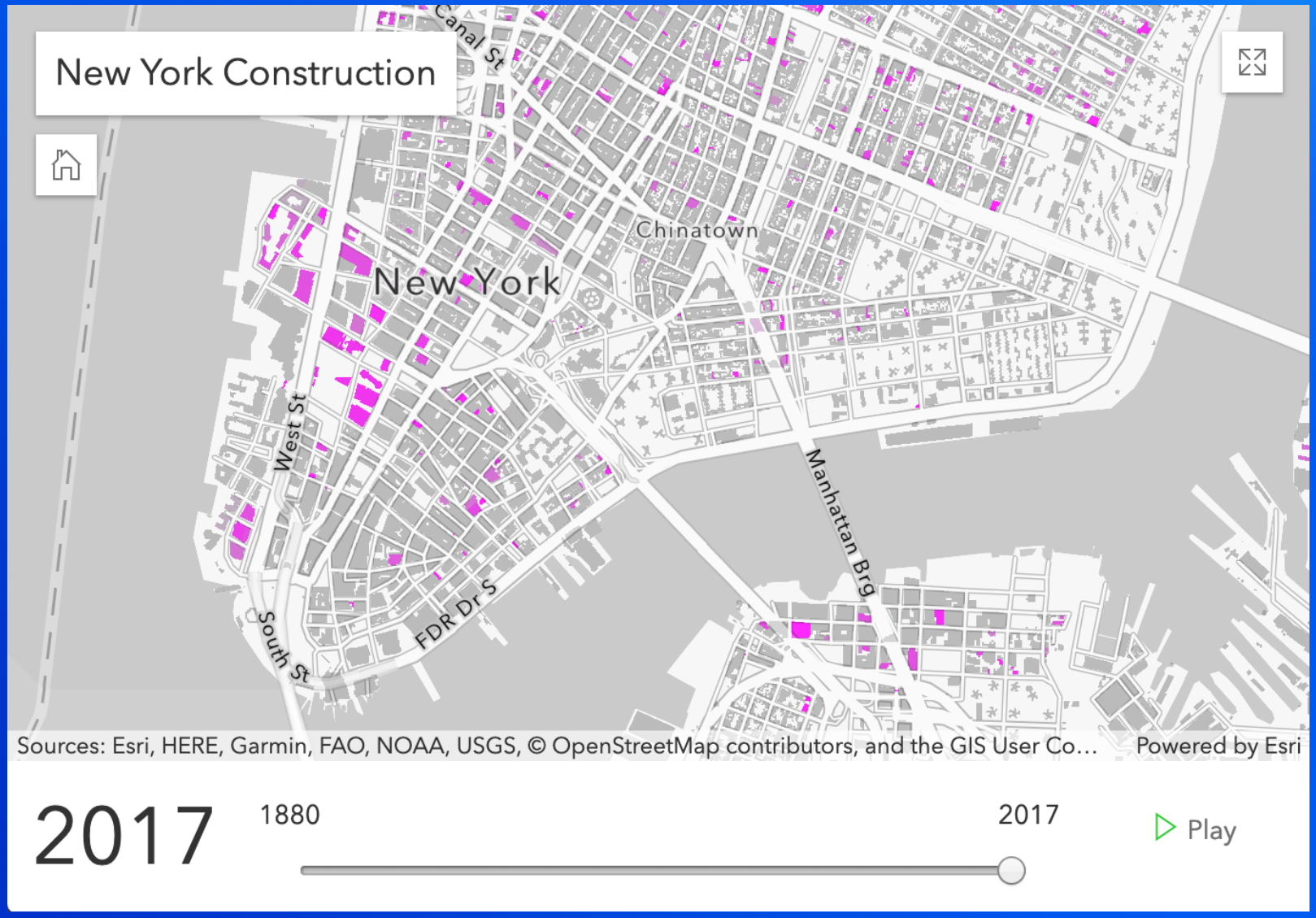
Wait, there's more.

- A century of Earthquakes
- May/June Quakes
- Hurricanes and Storms
- Imagery with Time
- MapServer with Time
- Quakes in 3d

Visualizing time using Arcade
Color earthquakes by age



Visualizing time using Visual Variables




esri
THE
SCIENCE
OF
WHERE