esri Canada

THE SCIENCE OF WHERE™

JavaScript API Technical Updates

Jackie Roberts, Esri Inc.

@webdevjackie

Tech Trek 2020

Agenda

- Saving webmaps
- Widget updates
  - Editor Widget - 3D feature layers
  - Swipe Widget
  - Feature Table (beta)
- Custom visualization using Arcade
- Updates to arcgis-cli
- jsapi-next
- Significant API updates as of v4.15

JSAPI v4.15 as of 4/20/20

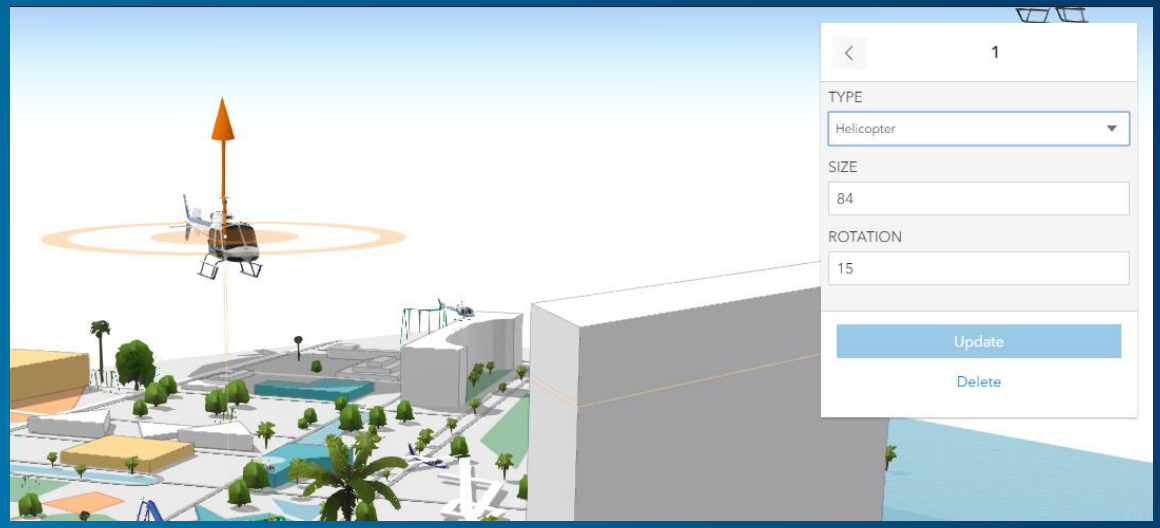
Save or update webmap via JSAPI

```
/**************************************************************
 * Loading the portal will trigger authentication and once the
 * returned Promise is resolved, the WebMap will be saved as
 * a new PortalItem using the given title.
 **************************************************************/
portal.load().then(function() {
  map.saveAs({
    title: "Empty WebMap",
    portal: portal
  });
});
```

Editor Widget – Editing 3D Features

```
let editor = new Editor({
  view: view
  // Pass in any other additional property as needed
});
```

```
// Add widget to top-right of the view
view.ui.add(editor, "top-right");
```



Swipe Widget

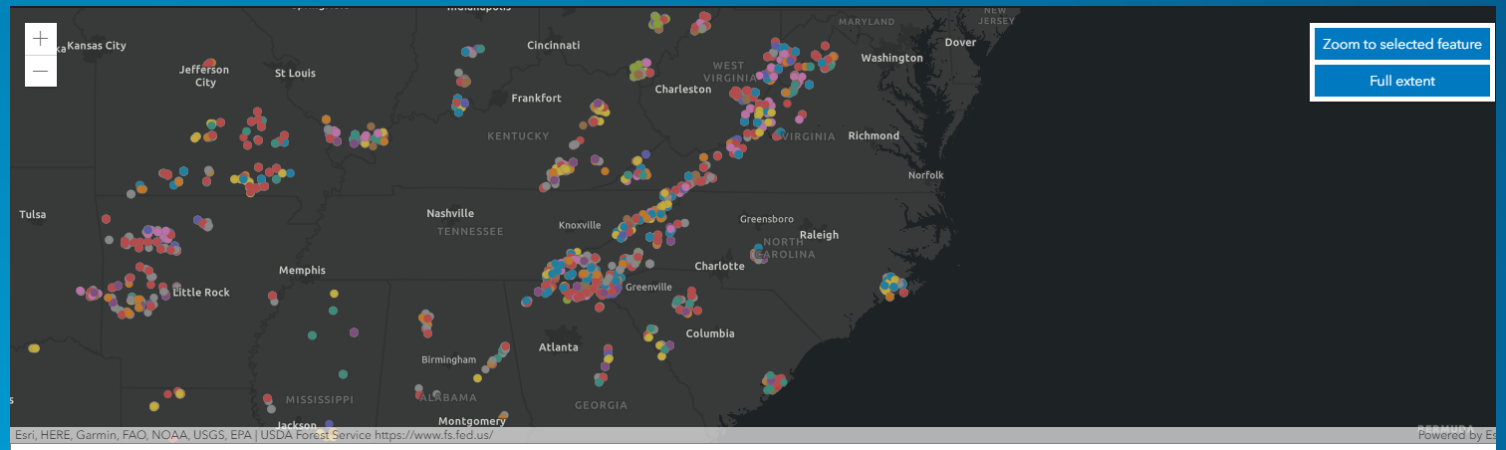
```
// create a new Swipe widget
const swipe = new Swipe({
  leadingLayers: [infrared],
  trailingLayers: [nearInfrared],
  position: 35, // set position of widget to 35%
  view: view
});
// add the widget to the view
view.ui.add(swipe);
```

Swipe Widget – Infinite Scrolling

```
// create a swipe widget for each layer
swipes = layers.map(function(layer) {
  return new Swipe({
    view: view,
    disabled: false,
    position: 100,
    direction: "vertical",
    trailingLayers: [layer],
    visibleElements: {
      handle: false,
      divider: true
    }
  });
});
```

```
// To achieve this infinite scroll effect we need to swap the layers:
//   The layer starts at the bottom, the divider goes up.
//   Then the next layer starts to show up, so we put back the divider at the bottom and swap the layers.
if (position < 0 && swipe.trailingLayers.length) {
  swipe.leadingLayers.addMany(swipe.trailingLayers);
  swipe.trailingLayers.removeAll();
} else if (position >= 0 && swipe.leadingLayers.length) {
  swipe.trailingLayers.addMany(swipe.leadingLayers);
  swipe.leadingLayers.removeAll();
}
```

Feature Table Widget - beta



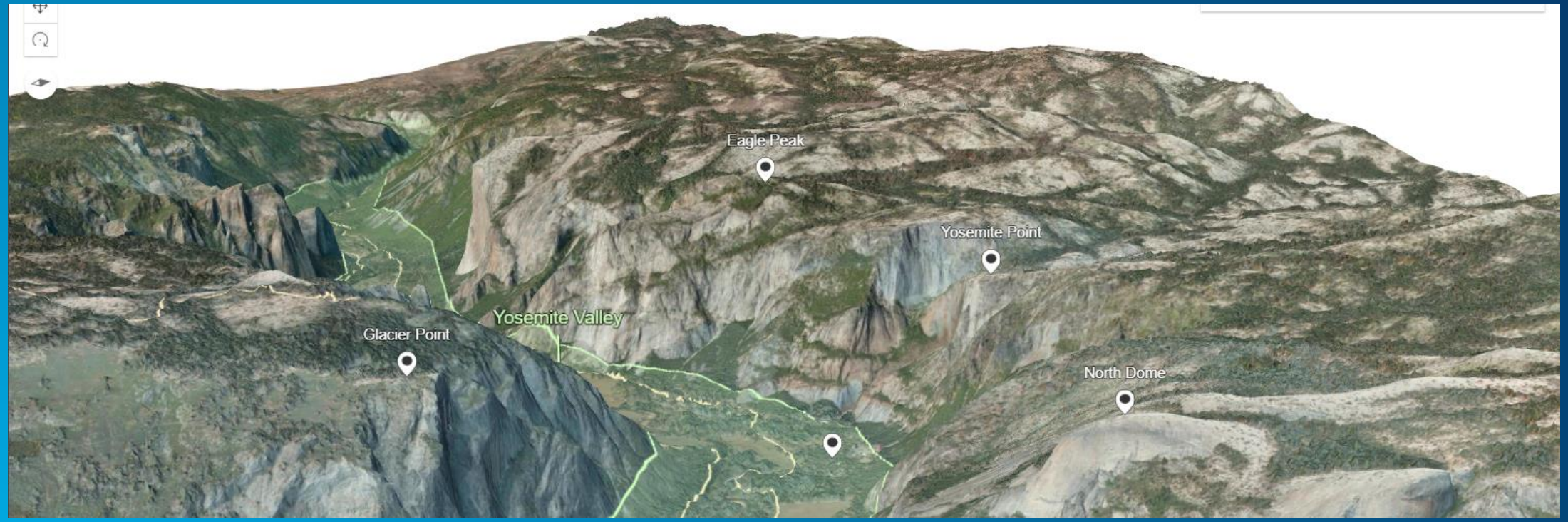
USFS Recreational areas (Features: 49,037, Selected: 0)

Recreation area name	Forest name	Open status	Season begins	Recreation description
Timberline Trail: Pigs Ear Trailhead	Allegheny National Forest	closed	Memorial Day	This trailhead parking lot is located off of Forest Road 125 via State Route 66 just northeast of the main State Route 66 trailhead. Facilities inclu
Timberline Trail: Pigs Ear Trailhead	Allegheny National Forest	closed	Memorial Day	This trailhead parking lot is located off of Forest Road 125 via State Route 66 just northeast of the main State Route 66 trailhead. Facilities inclu
Tracy Ridge Recreation Area	Allegheny National Forest	closed	Apr. 1	This campground sits on a ridge top three miles east of the Allegheny Reservoir just minutes south of the PA/NY state border. This 45-site cam The
Tracy Ridge Recreation Area	Allegheny National Forest	closed	Apr. 1	This campground sits on a ridge top three miles east of the Allegheny Reservoir just minutes south of the PA/NY state border. This 45-site cam The
Tracy Ridge Recreation Area	Allegheny National Forest	closed	Apr. 1	This campground sits on a ridge top three miles east of the Allegheny Reservoir just minutes south of the PA/NY state border. This 45-site cam The

Known limitations:

- Not supported in Internet Explorer 11.
- Editing is currently not supported.
- Viewing related records is currently not supported.
- Viewing attachments is currently not supported, although if a feature contains attachments, the total count per feature will display.
- Selecting features from a map and having it reflected in the table is currently not implemented but will be in an upcoming release.
- Dark themed CSS is currently not supported.
- Customizing date strings via the FeatureTable API is currently not supported.

IntegratedMeshLayer



Custom visualizations using Arcade

You can use template literals (ES6 or later) to write multi-line Arcade expressions.

```
renderer.valueExpression = `
  var republican = $feature.MP06025a_B;
  var democrat = $feature.MP06024a_B;
  var independent = $feature.MP06026a_B;

  var parties = [ republican, democrat, independent ];
  var total = Sum(parties);
  var max = Max(parties);
  return (max / total) * 100;
`;
```

```
<script type="text/plain" id="adult-population">
  var republican = $feature.MP06025a_B;
  var democrat = $feature.MP06024a_B;
  var independent = $feature.MP06026a_B;

  var parties = [ republican, democrat, independent ];
  var total = Sum(parties);
  var max = Max(parties);
  return (max / total) * 100;
</script>
```

Updates to arcgis-cli

Choose a UI framework

Use the `-t` flag to create a template application with the specified UI framework

React

Use `-t react` to create a template application using React.

```
arcgis create my-react-application -t react
```

Vue

Use `-t vue` to create a template application using Vue as the UI framework.

```
arcgis create my-vue-application -t vue
```

ArcGIS API for JavaScript - next

CDN

Add references to the CDN and you are ready to get started.

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

npm

Install using `npm install --save arcgis-js-api@next`.

esri-loader

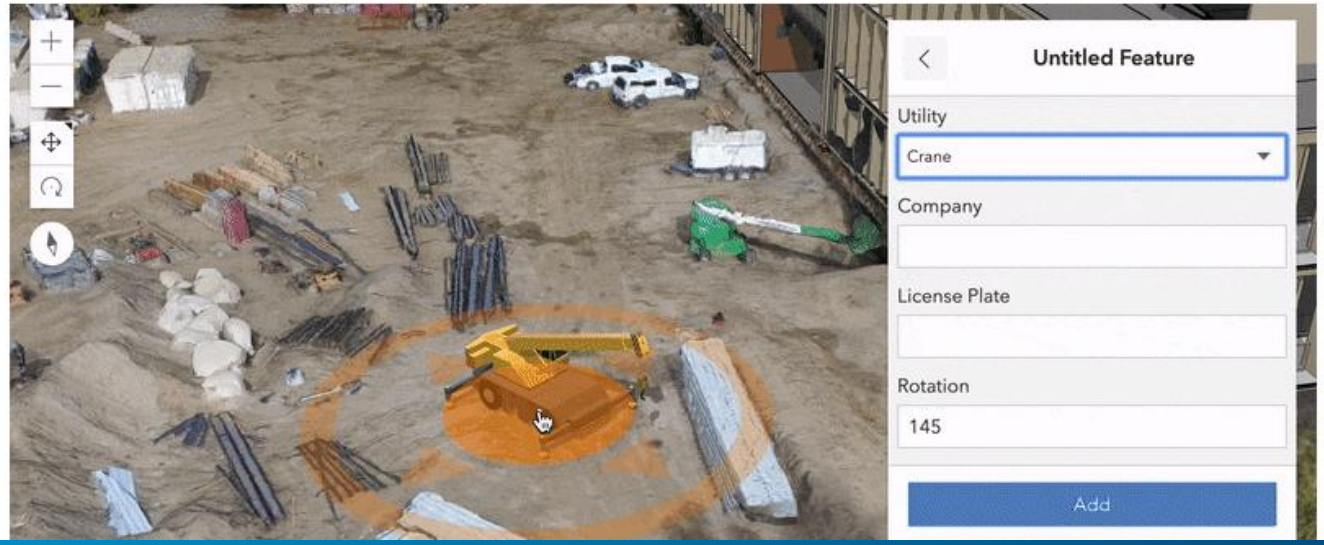
Or if you're using esri-loader to help load the library, use options.version and specify `next` as the value. You can also lazy-load js.arcgis.com/next/esri/themes/light/main.css by adding the `css: true` option, or you can use any of the other methods supported by esri-loader instead.

```
const options = {
  version: 'next',
  css: true
};
const [Map] = await loadModules(['esri/map'], options);
```

Latest release notes

- Tutorials
- Core Concepts
- Data Visualization
- Building your UI
- Working with ArcGIS Online and Enterprise
- Developer Tooling
- Migrating from 3.x
- Reference

While the Sketch tool already supports features on the ground, the new z-aware editing tools also allow you to modify geometries above- and underground. Laser lines projected onto the terrain and objects in the scene help you align geometries in the 3D space.



esriConfig as a global variable

```
<script>
  var esriConfig = {
    portalUrl: "https://myHostName.esri.com/arcgis"
  };
</script>
```

Return Native Promises

- API returns native `Promise` by default
- Dojo promise to be removed ***completely*** by 4.16 (~July 2020)
- Native promises are nearly identical to Dojo deferred/promise, except:
  - No `otherwise()` method. Use `catch()` instead.
  - No `cancel()` method. User `AbortController.abort()` instead
  - No `always()` method. Use the following code instead:

```
.catch(function(error){
  /* do something with the error */
}).then(function() {
  /* this function is always executed */
});
```

Return Native Promises

- Can opt out at v4.15

```
<script>
  var dojoConfig = {
    has: {
      "esri-native-promise": false
    }
  };
</script>
```

References

- What's New in ArcGIS API for JavaScript (April 2020) blog post
- Latest release notes
- JavaScript API sample code
- Arcgis-cli developer tooling guide
- Esri-loader guide
- https://github.com/webdevjackie/esrica-tech-trek
- ArcGIS API for JavaScript: Using TypeScript
- ArcGIS API for JavaScript: Programming Patterns and Fundamentals
- jsapi-next
- CodePens – Esri Canada Tech Trek 2020