



ArcGIS API for JavaScript Building 3D Web Apps

Javier Gutierrez, Björn Svensson



ArcGIS API for JavaScript Building 3D Web Apps

Javier Gutierrez, Björn Svensson



<https://developers.arcgis.com/javascript/beta/>

Agenda

- **Introduction**
- **From 3.x to 4.0**
- **Layers**
- **3D Visualization**
- **SceneView**
- **Widgets**
- **What's next**

Introduction

The background features a vibrant blue gradient. On the left side, there are several overlapping geometric shapes: a large purple triangle pointing upwards, a yellow triangle pointing downwards, and a purple triangle pointing to the right. These shapes are layered, creating a sense of depth and movement.

What is the ArcGIS API for JavaScript?

- **Allows JavaScript developers to build applications for web browsers**
- **Provides visual mapping component and widgets**
 - Support for many different layer types (data sources)
- **Analysis and geoprocessing**
- **Integration with ArcGIS platform**
 - security, sign-in, premium services

Basics - ArcGIS API for JavaScript

- **How to get JSAPI**

- **API Library**

- CDN (content delivery network) - Hosted at <http://js.arcgis.com/4.0beta1/>

- Online optimizer

- Download

- <https://developers.arcgis.com/javascript/>

- **Beta**

- <https://developers.arcgis.com/javascript/beta/>

- **Licensing**

- <https://developers.arcgis.com/en/terms/faq/>

3D in ArcGIS API for JavaScript

- **Same ArcGIS API, built with map & layers, for GIS analysis and visualization**
- **Data**
 - 2D tiles, maps, features
 - New 3D data (SceneLayer, ElevationLayer)
- **3D concepts – more realistic**
 - Camera, Light and Shadows
 - Ground surface with Elevation
- **Scene Viewer and WebScenes**

ArcGIS API for JavaScript Beta

[Home](#) [Guide](#) [API Reference](#) [Sample Code](#) [Forum](#)

4.0 Beta 1

Version 4.0 is the next generation ArcGIS API for JavaScript and will support both 2D and 3D. We are planning multiple public betas. The first beta is primarily a developer preview with limited functionality.

[About 4.0 betas](#)

Use the API

Reference the ArcGIS JavaScript API from our CDN and you're ready to get started:

```
<link rel="stylesheet" href="https://js.arcgis.com/4.0beta1/esri/css/esri.css">
<script src="https://js.arcgis.com/4.0beta1/"></script>
```

Your first map



3D in ArcGIS API for JavaScript

- How does it work?
 - WebGL + JavaScript (ECMAScript 5)
- Requirements
 - Good hardware w/ Graphics Card
 - Modern Web Browser
- Limitations
 - Web Mercator only (beta1)
 - Limited number of features



ArcGIS API for JavaScript

From 3.x to 4.0

<https://developers.arcgis.com/javascript/beta/>

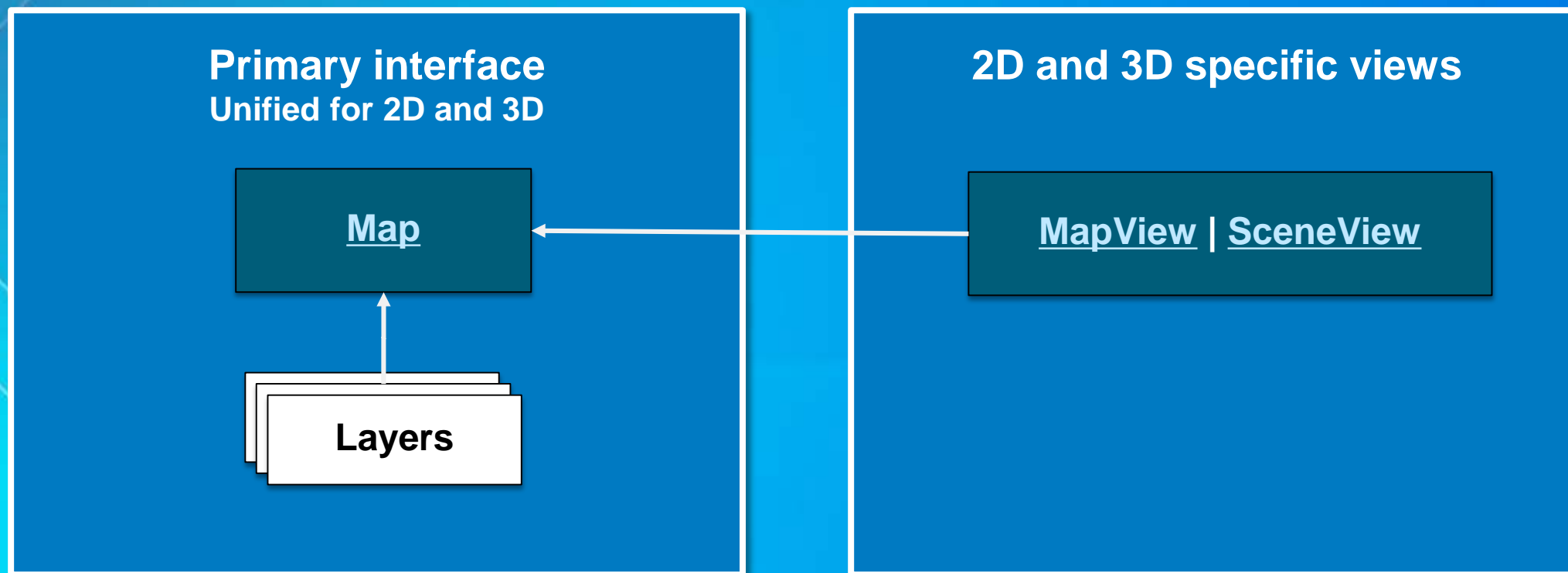
From 3.x to 4.0

1. **Using views**
2. **Working with properties**
3. **Working with promises**
4. **Map and layer specifics**
5. **Module and package updates**
6. **AMD only**
7. **Different system requirements**

<https://developers.arcgis.com/javascript/beta/guide/migrating/>

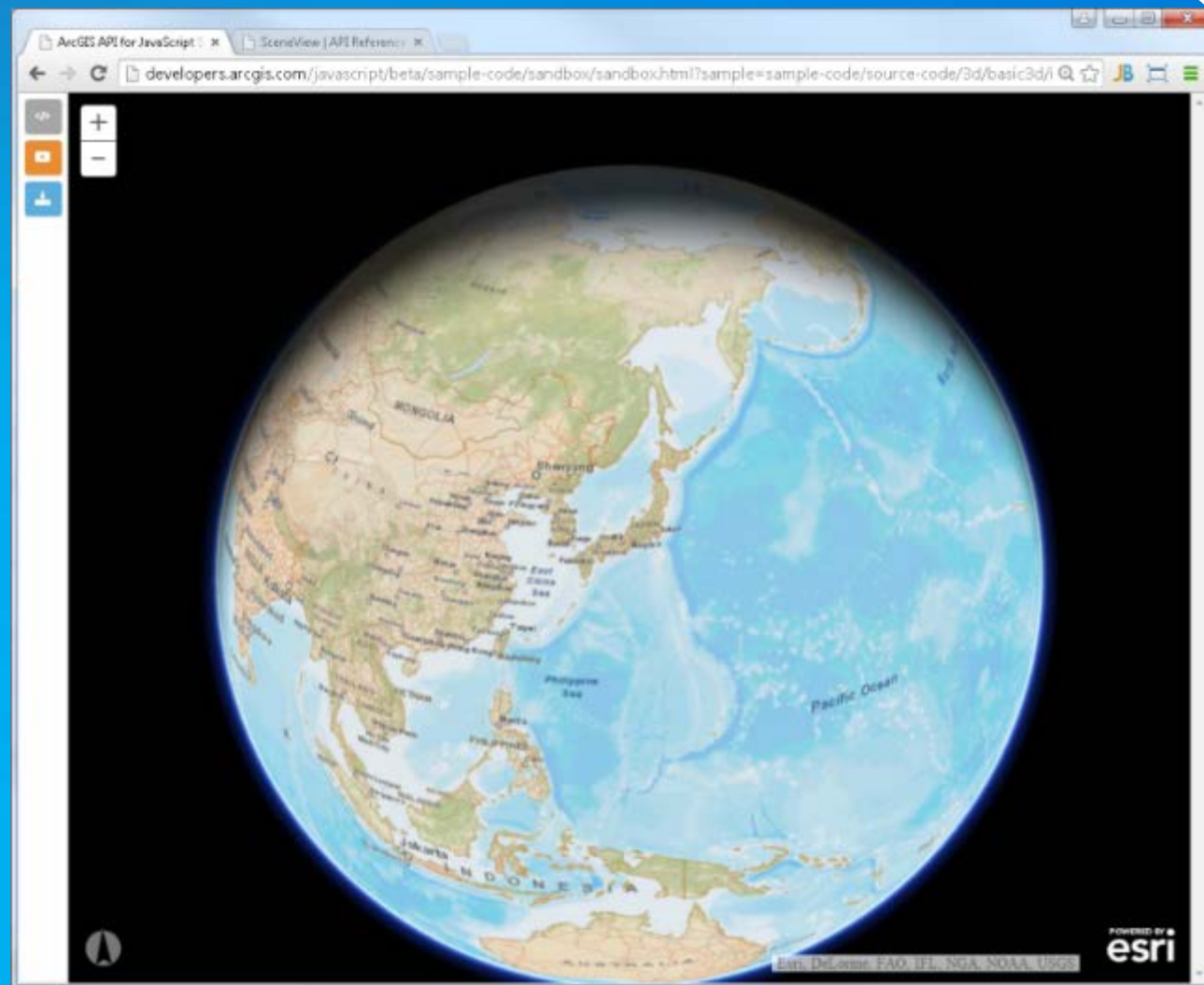
1. Using views

Extending to 3D - one API for both 2D and 3D



Create a basic 3D map

```
<script>
var map, view;
require([
  "esri/Map",
  "esri/views/SceneView",
  "dojo/domReady!"
], function (Map, SceneView) {
  map = new Map({
    basemap: "streets"
  });
  view = new SceneView({
    container: "viewDiv",
    map: map,
    scale: 240000000
  });
});
</script>
```



2. Working with Properties

	Old (3.x)	New (4.0)
Read	<code>map.property</code> or <code>map.getProperty()</code>	<code>map.property</code>
Write	<code>map.property = 10</code> or <code>map.setProperty(10)</code>	<code>map.property = 10</code>
Change event	<code>map.on("extent-change", ...)</code>	<code>view.watch("extent", ...)</code>

- <https://developers.arcgis.com/javascript/beta/guide/working-with-props/>
- <http://odoe.net/blog/arcgis-js-api-4-0beta1-accessors/>
- <http://odoe.net/blog/fun-with-accessors-in-arcgis-js-4beta1/>

3. Working with promises

- A promise is a representation of a future value returned from async task
- Core objects (Map, Layer) are Promises
- Promises are in one of three states:
 - pending
 - resolved
 - rejected

ArcGIS API 3.x

```
map.on("load", function() {  
    // map is initialized  
});
```

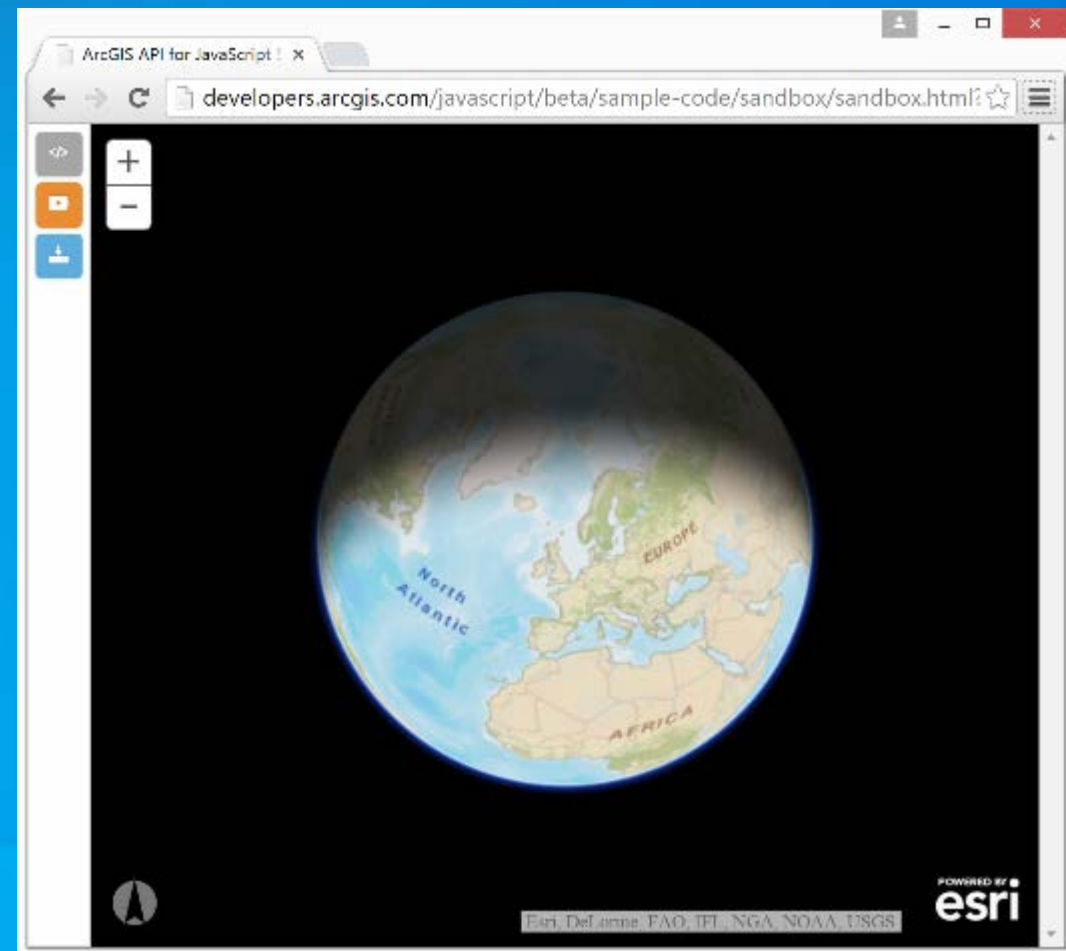
ArcGIS API 4.0

```
map.then(function() {  
    // map is initialized  
});
```

```
view.then(function() {  
    // view is initialized  
});
```

Detect WebGL support with view.then

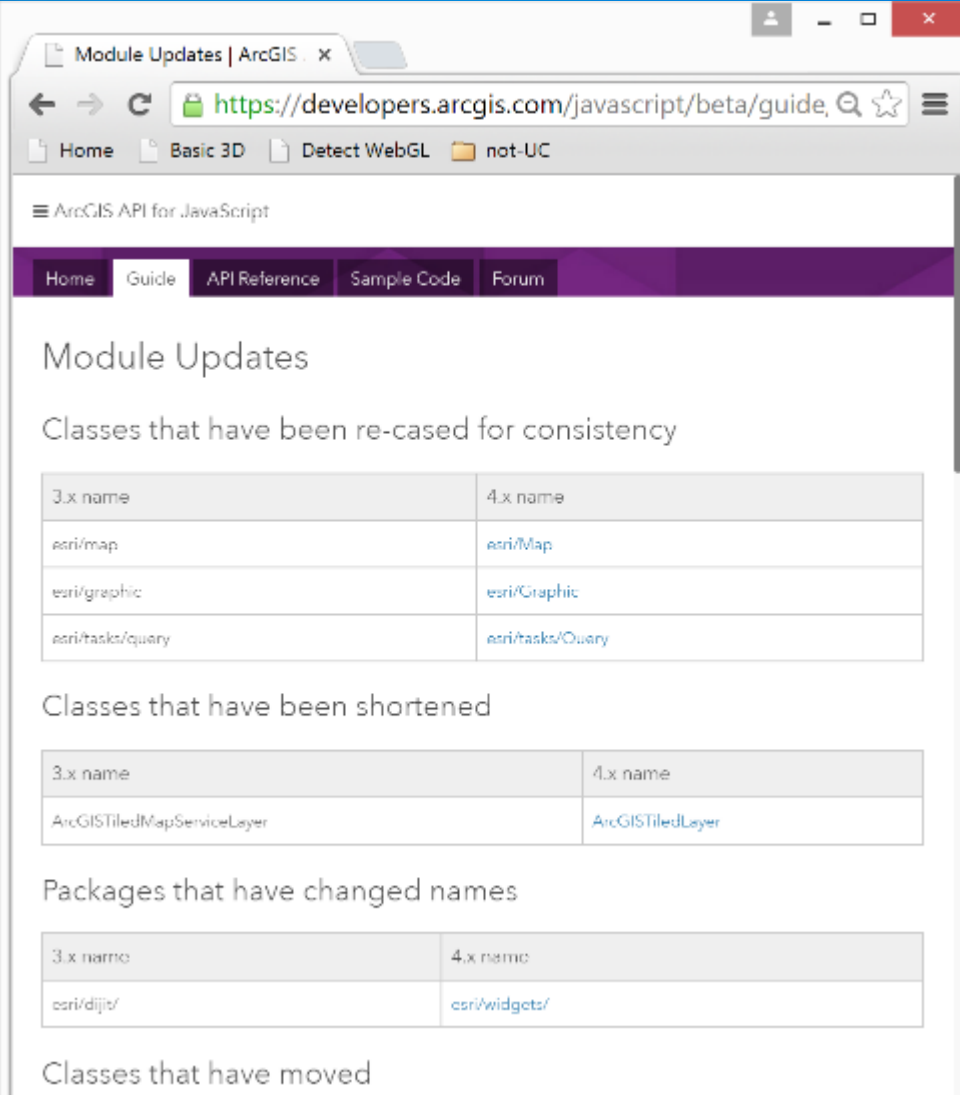
```
view.then(function() {  
    // View successfully loaded,  
    // show viewDiv  
}, function() {  
    // View was rejected,  
    // show webgl unsupported message  
    // and turn off the viewDiv  
});
```



4. Map and layer specific changes

- `map.addLayer(layer)` \Rightarrow `map.add(layer)`
- proper basemap and operational layers separation

5. Module and package updates



The screenshot shows a web browser window displaying the 'Module Updates' page for the ArcGIS API for JavaScript. The page is titled 'Module Updates' and lists several categories of changes:

- Classes that have been re-cased for consistency:**

3.x name	4.x name
esri/map	esri/Map
esri/graphic	esri/Graphic
esri/tasks/query	esri/tasks/Query
- Classes that have been shortened:**

3.x name	4.x name
ArcGISTiledMapServiceLayer	ArcGISTiledLayer
- Packages that have changed names:**

3.x name	4.x name
esri/dijit/	esri/widgets/
- Classes that have moved:**

<https://developers.arcgis.com/javascript/beta/guide/module-updates/>

6. AMD only

7. Different system requirements

- **3D mapping: supported in web browsers that support WebGL**
- **IE11**
- **Chrome**
- **Firefox**
- **Safari**
- **No mobile and iPad support for 3D.**

Layers

The background features a gradient of blue and purple colors. On the left side, there are several overlapping geometric shapes, including a yellow shape that contains a faint outline of a map of the United States. The word "Layers" is written in white, bold, sans-serif font in the upper left quadrant.

Beta 1 – Types of layers

Layer

[ArcGISTiledLayer](#)

[FeatureLayer](#)

[GraphicsLayer](#)

[GroupLayer](#)

[OpenStreetMapLayer](#)

[WebTiledLayer](#)

3D-specific layers

[ArcGISElevationLayer](#)

[SceneLayer](#)

Elevation layers

- Elevation services
 - Tiled image service
 - Format: LERC (Limited Error Raster Compression)
- New layer type: ArcGISElevationLayer

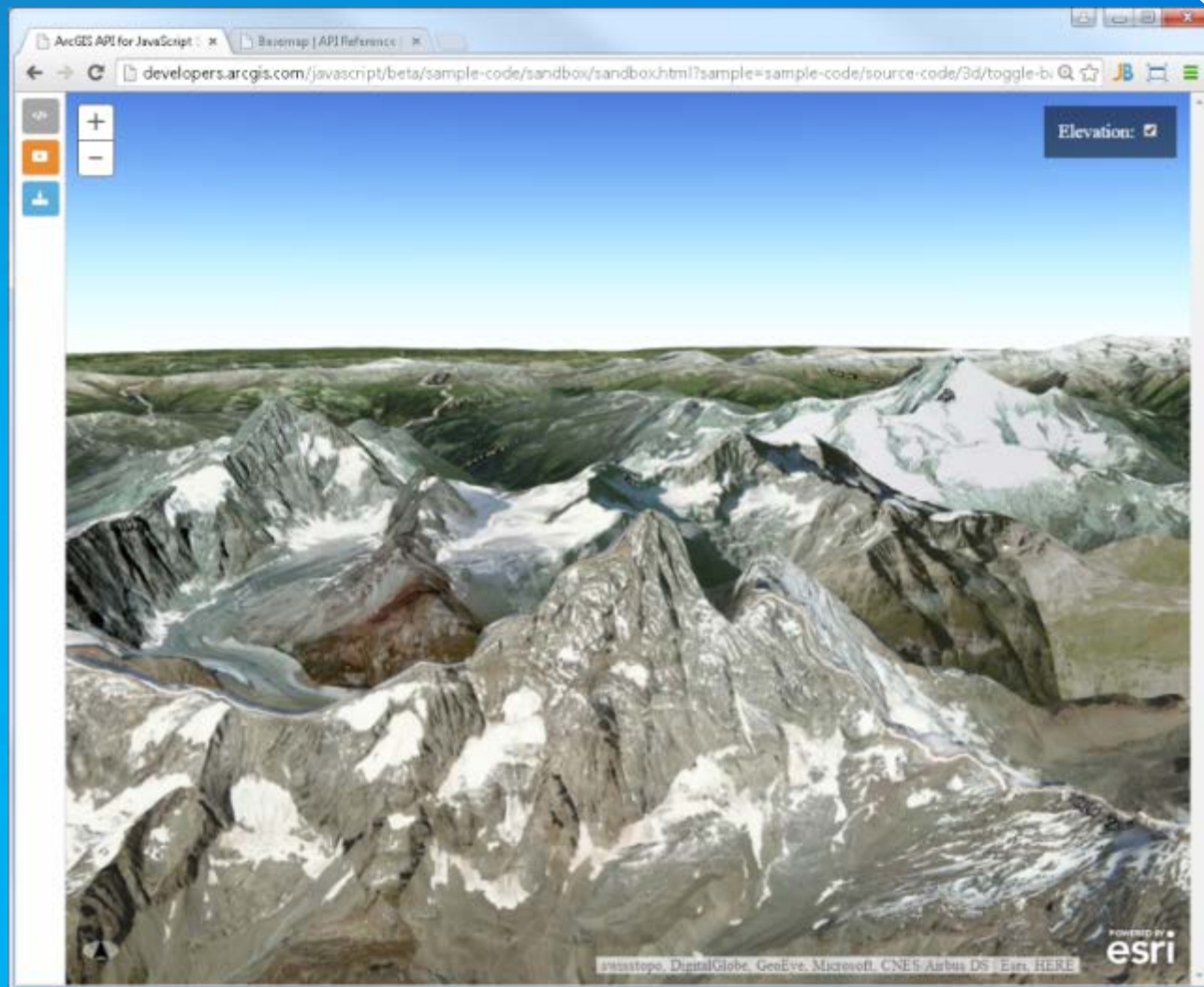
```
var layer = new ArcGISElevationLayer(url);  
map.add(layer);
```

- Multiple elevation layers will overlay just as map layers would
- Planned: elevation data queries

LERC

3D Basemap has default elevation layers

```
// Store the default elevation  
layers  
var elevationLayers =  
map.basemap.elevationLayers.getA  
ll();  
...  
map.basemap.elevationLayers.clea  
r();
```



Scene Layers

- 3D objects served in i3s format
 - Spatially indexed
 - Automatic level of detail
- New layer type: **SceneLayer**

```
var layer = new SceneLayer(url);  
map.add(layer);
```

i3s = Indexed 3D Scene

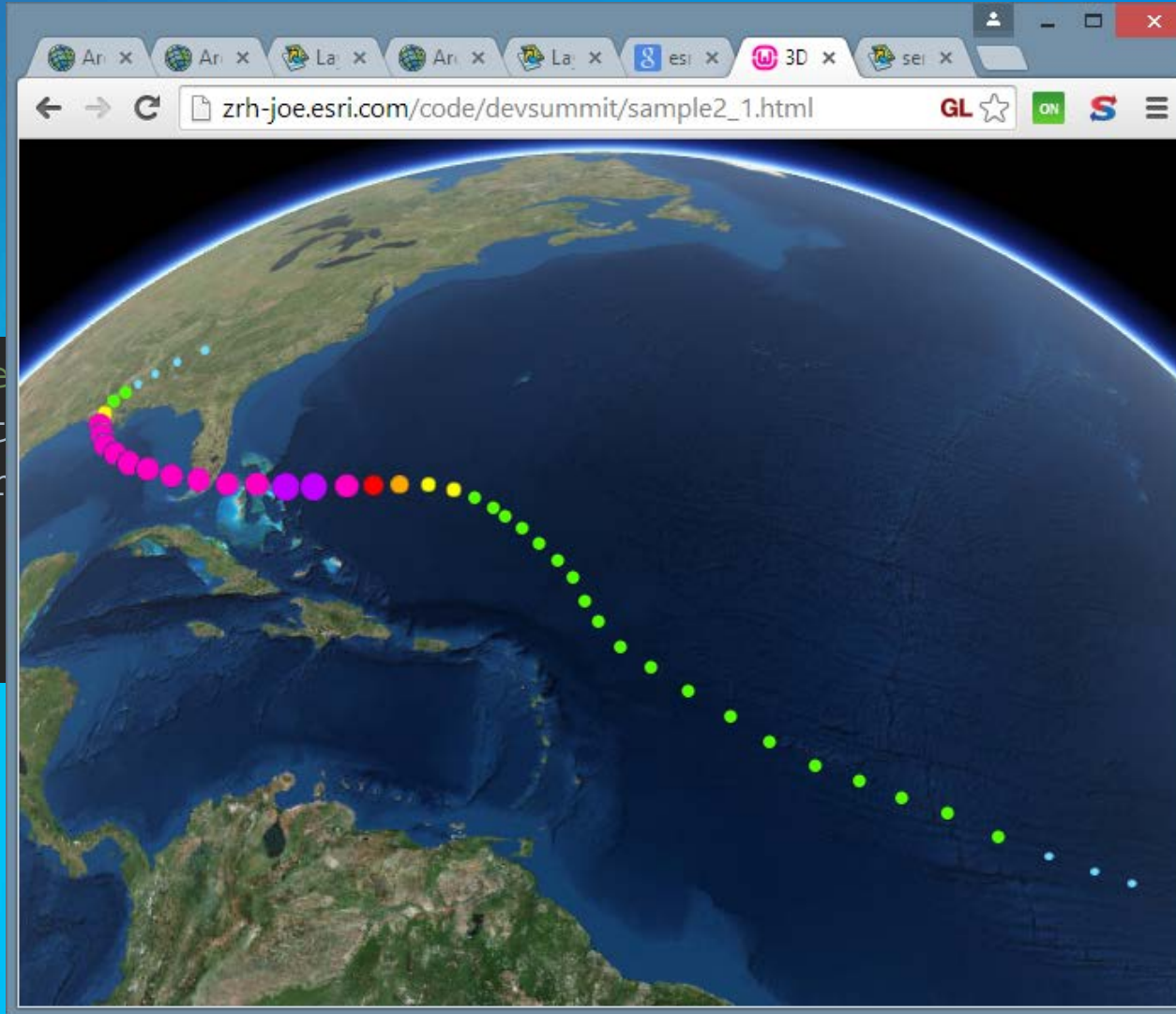
Open specification:

<https://github.com/Esri/i3s-spec>

i3s

Feature layers

```
var url = "http://se  
var layer = new Feat  
mode: FeatureLayer  
});  
map.add(layer);
```



```
2/FeatureServer/0";
```

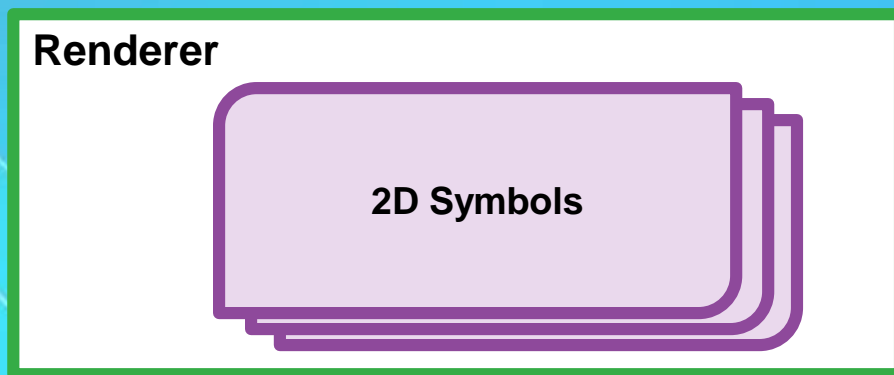
3D Symbology

The background features a gradient from dark purple to bright blue. On the left side, there are several overlapping geometric shapes: a dark purple triangle, a yellow textured shape, and a dark purple rectangle. A large, light blue diagonal shape is also present on the right side of the image.

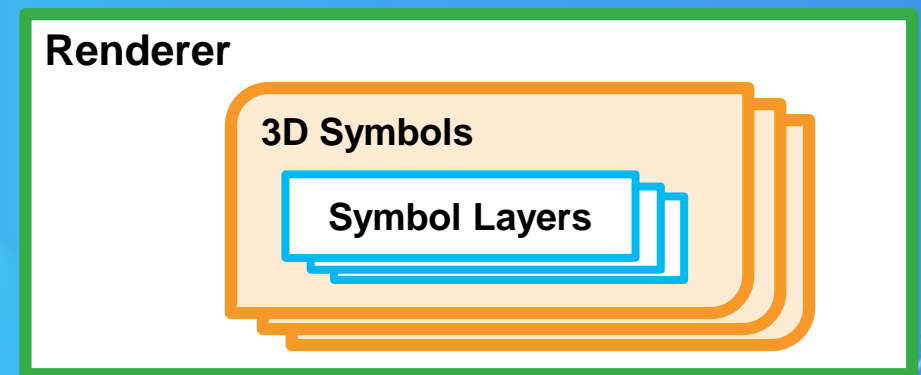
Web3D Symbology

- Modern and simple spec
- Concepts from JavaScript and Pro symbology
- Design with 3D and 2D in mind
 - future proof, flexible and extensible



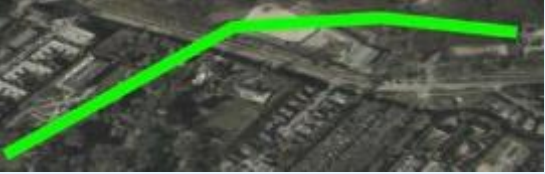


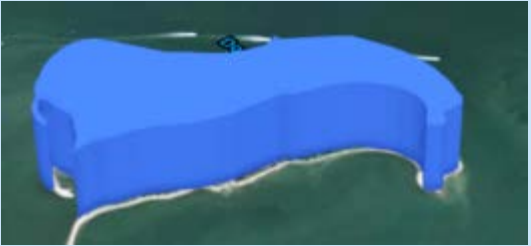
Existing 2D Symbology



New 3D Symbology



Web3D Symbology: flat vs. volumetric

Web3D “flat” symbol layer	Web3D “volumetric” symbol layer
Icon 	Object 
Line 	Path 
Fill 	Extrude 

Web3D Symbology

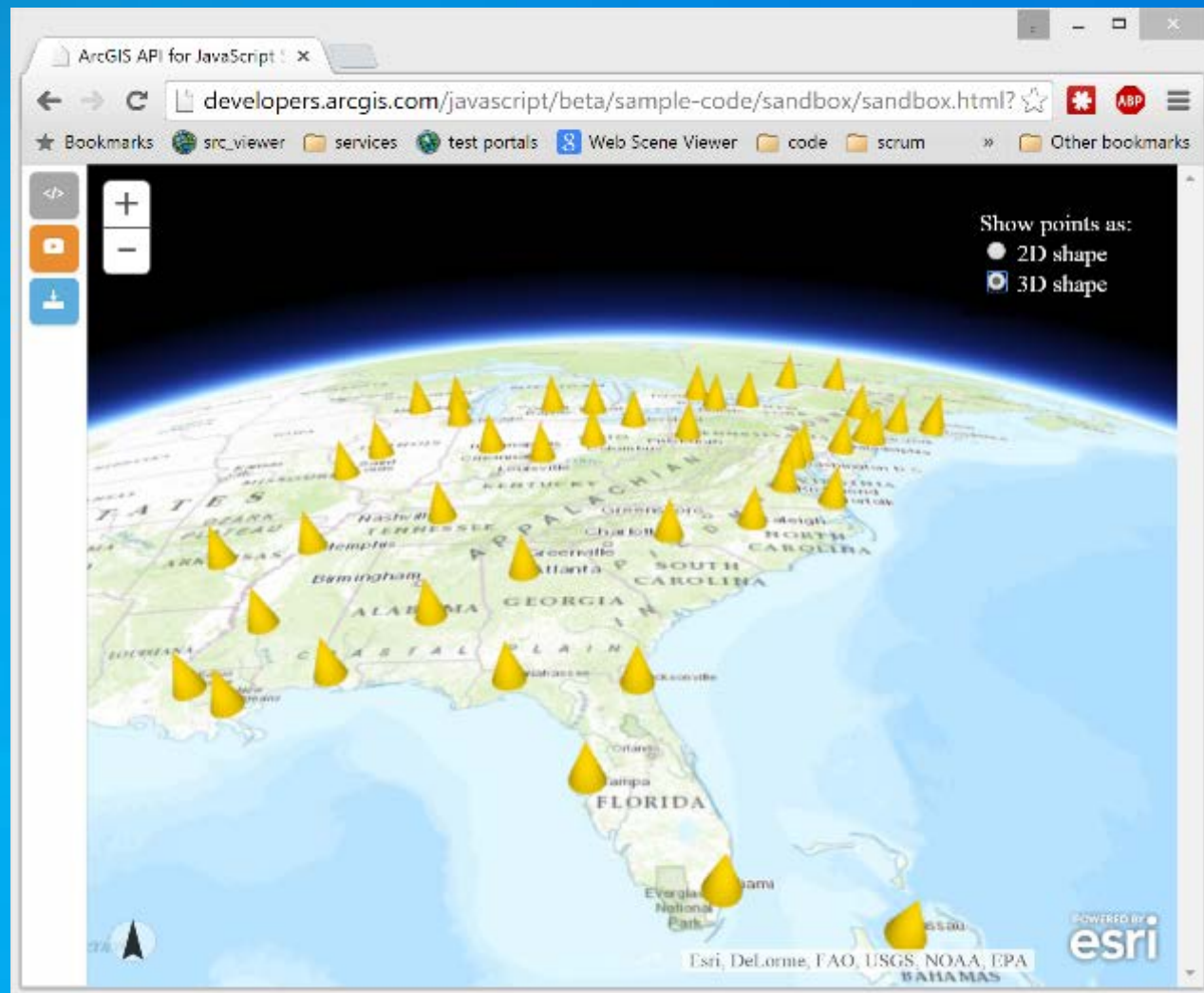
- Symbol – Symbol Layer relationships

	Icon	Object	Line	Path	Fill	Extrude
PointSymbol3D	✓	✓				
LineSymbol3D			✓	✓		
PolygonSymbol3D	✓	✓	✓		✓	✓
MeshSymbol3D					✓	

PointSymbol3D – flat vs. volumetric

```
//Create objectSymbol and add to renderer
var objectSymbol = new PointSymbol3D({
  symbolLayers: [new ObjectSymbol3DLayer({
    width: 70000,
    height: 100000,
    resource: {
      primitive: "cone"
    },
    material: {
      color: "#FFD700"
    }
  })]
});

objectSymbolRenderer = new SimpleRenderer({
  symbol: objectSymbol
});
layer.renderer = objectSymbolRenderer;
```

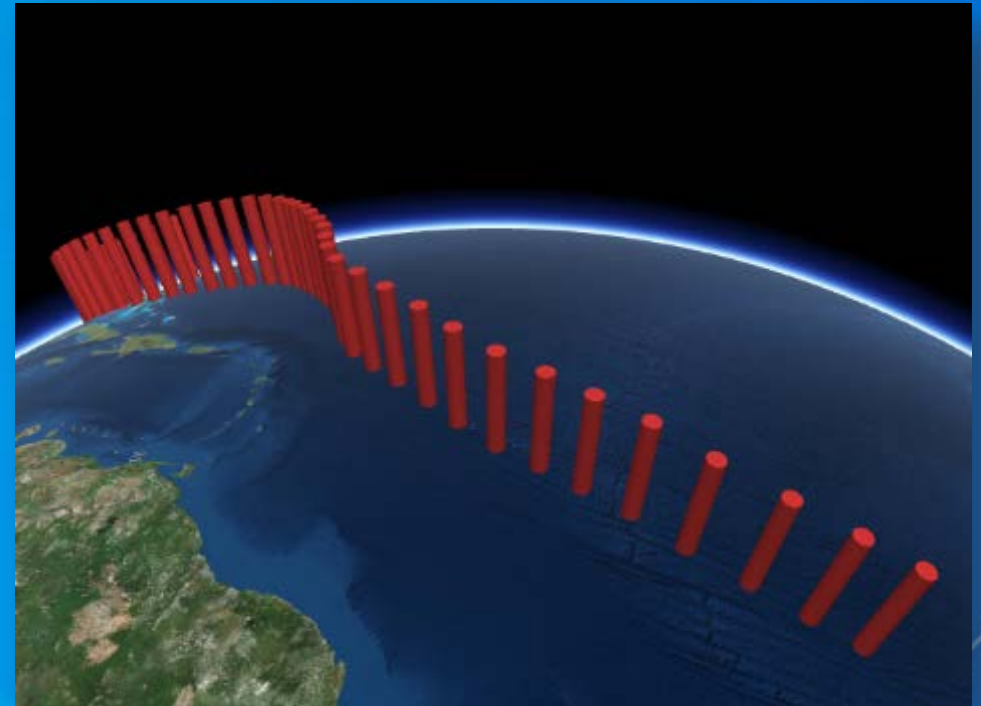


Web3D Symbology

PointSymbol3D + ObjectSymbol3DLayer

```
var symbolLayer = new ObjectSymbol3DLayer({
  resource: {
    primitive: "cylinder"
  },
  material: {
    color: [255, 64, 64]
  },
  width: 80000, // in meters
  height: 500000 // in meters
});
var symbol = new PointSymbol3D(symbolLayer);
var renderer = new SimpleRenderer(symbol);

layer.renderer = renderer;
```



Visual Variables

ArcGIS API 3.12

```
renderer.setSizeInfo({
  field: "WIND_SPEED",
  minSize: 3,
  maxSize: 20,
  minDataValue: 5,
  maxDataValue: 50
});

renderer.setColorInfo({
  field: "M086_07",
  minDataValue: 0,
  maxDataValue: 100,
  colors: [
    new Color([255, 255, 255]),
    new Color([127, 127, 0])
  ]
});
```

ArcGIS API 4.0

```
renderer.set("visualVariables", [
  {
    type: "sizeInfo", // new
    axis: "all", // new
    field: "WIND_SPEED",
    minSize: 3,
    maxSize: 20,
    minDataValue: 5,
    maxDataValue: 50
  }, {
    type: "colorInfo", // new
    field: "M086_07",
    minDataValue: 0,
    maxDataValue: 100,
    colors: [
      new Color([255, 255, 255]),
      new Color([127, 127, 0])
    ]
  }
]);
```

Web3D Symbology

Polygon Extrusion

```
extrudePolygonRenderer = new SimpleRenderer({
  symbol: new PolygonSymbol3D({
    symbolLayers: [
      new ExtrudeSymbol3DLayer({
        size: 1000000,
        material: {
          color: [128, 128, 255]
        }
      })
    ]
  }),
  visualVariables: [{
    type: "sizeInfo",
    field: "POP07_SQMI",
    minSize: 40000,
    ...
```



Elevation alignment

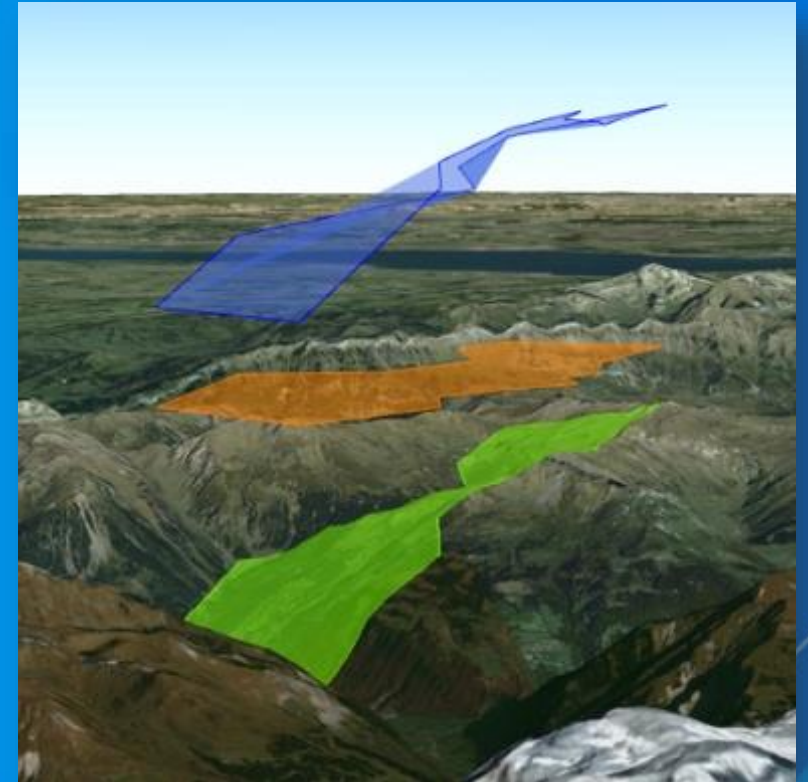
ElevationInfo

- Elevation modes

Elevation mode	Symbol elevation
"onTheGround"	Draped
"absoluteHeight"	Z value (or 0) [+ offset]
"relativeToGround"	Terrain + Z value (or 0) [+ offset]

- Setting the elevation behavior

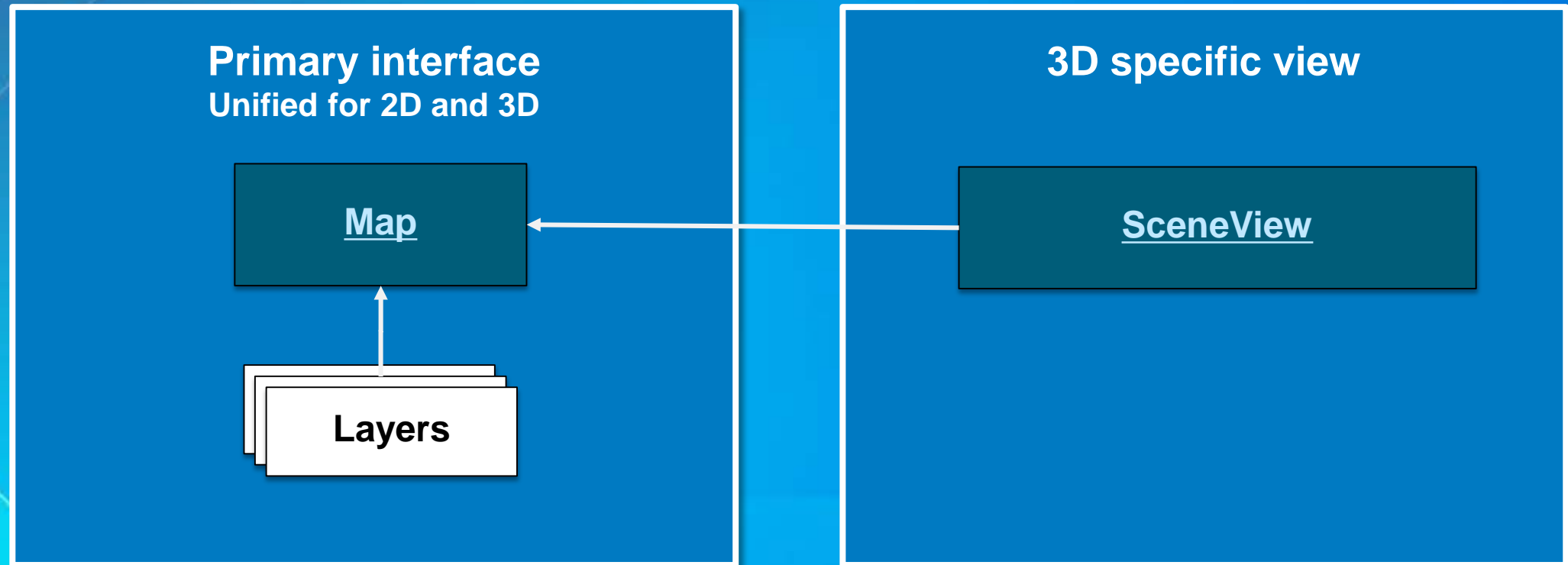
```
layer.elevationInfo = {  
  mode: "relativeToGround",  
  offset: 1000 // meters  
};
```



SceneView

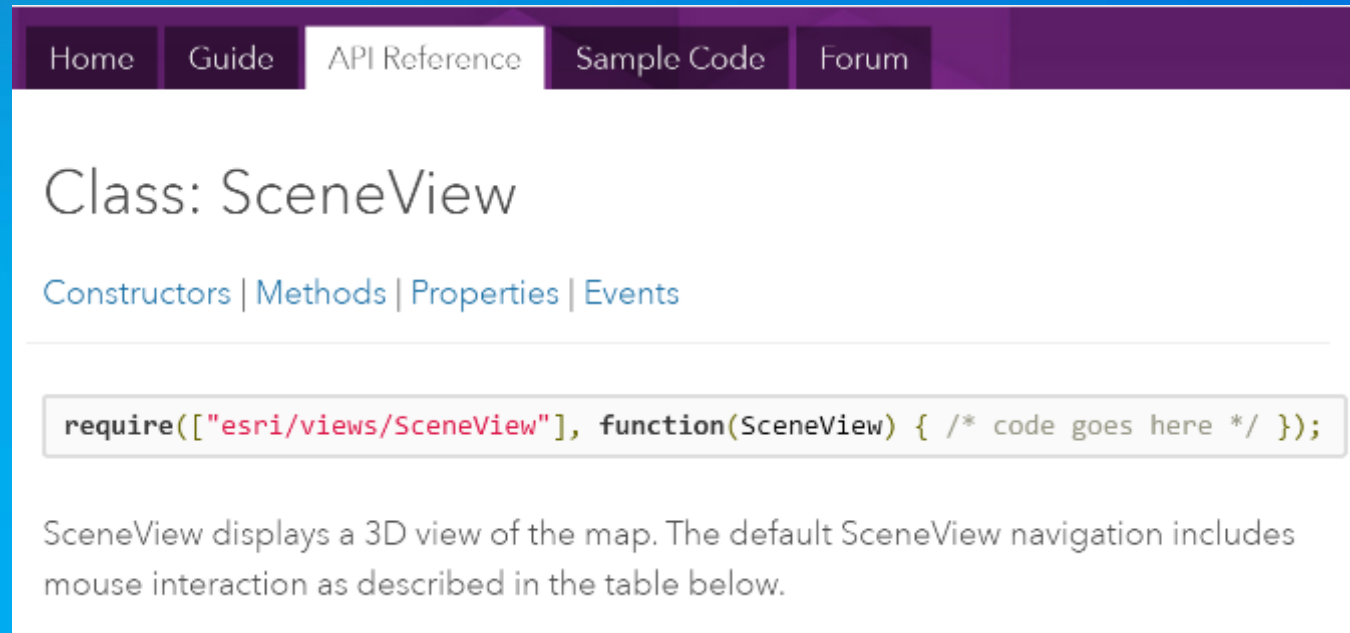
The background features a gradient from dark purple on the left to bright blue on the right. In the bottom-left corner, there are several overlapping geometric shapes: a dark purple triangle, a yellow textured shape, and a dark purple shape. The text 'SceneView' is centered in the upper-left quadrant.

SceneView



SceneView

- **Core component of 3D API**
 - Rendering, Navigation, Ground
- **Specific 3D View properties**
 - Camera
 - Environment
- **Common 2D/3D View properties**
 - Center *
 - Scale / Zoom *
 - Extent *
 - Rotation *



The screenshot shows the Esri API Reference page for the SceneView class. The navigation bar includes Home, Guide, API Reference (selected), Sample Code, and Forum. The page title is "Class: SceneView". Below the title are links for Constructors, Methods, Properties, and Events. A code block shows the require function call: `require(["esri/views/SceneView"], function(SceneView) { /* code goes here */ });`. The text below the code block states: "SceneView displays a 3D view of the map. The default SceneView navigation includes mouse interaction as described in the table below."

* supported in 3D in a “best effort” basis

SceneView

Camera

- **New class**

```
new Camera({
  position: new Point({
    x: -116.54, // longitude
    y: 33.83,   // latitude
    z: 1000,   // altitude in meters
    spatialReference: new SpatialReference(4326)}),
  heading: 30, // in degrees
  tilt: 45     // in degrees
})
```

- **Heading: clockwise, 0 .. 360°**
- **Tilt: 0..180°**
 - 0° straight down
 - 90° horizontal
 - 180° straight up

SceneView

Camera

- **Modifying the camera**

```
// view.camera.heading = 0; → doesn't work  
// need to modify and re-set the camera
```

```
var myCamera = view.camera;  
myCamera.heading = 0;  
view.camera = myCamera;
```

SceneView

animateTo

- Simple view animations with `animateTo`:

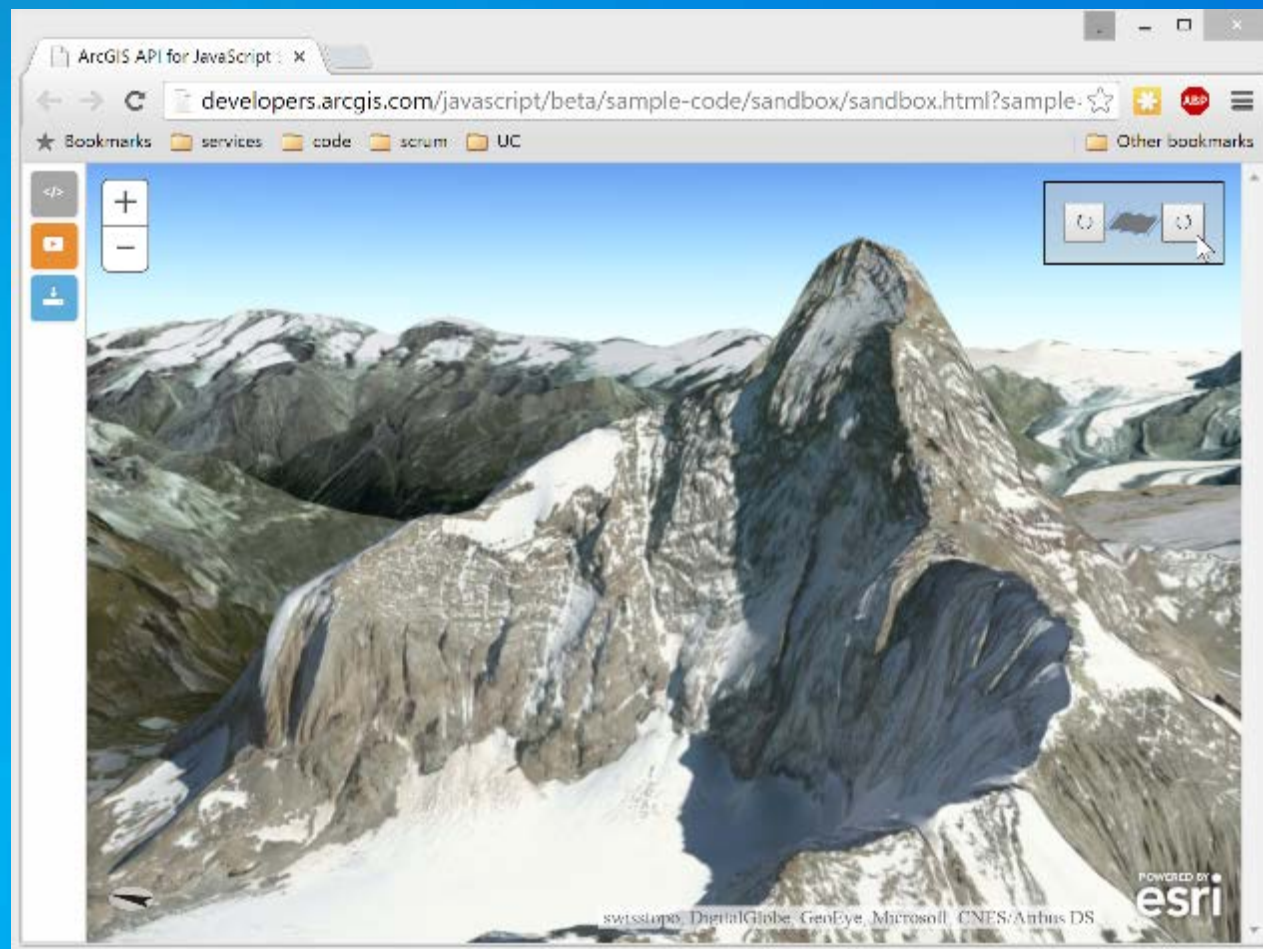
```
view.animateTo(target, options);
```

- Target can be
 - Camera
 - [longitude, latitude] (in WGS84)
 - Any Geometry object, or an array of Geometry objects
 - Graphic, array of Graphic

SceneView

Easy Navigation with animateTo

```
var heading = ...;  
var tilt = ...;  
  
view.animateTo({ heading: heading });  
view.animateTo({ tilt: tilt });
```



SceneView: Sunlight

- **Set time/date**

```
var date = new Date("01 Jan 2015 10:00 GMT-0800");  
view.environment.lighting.date = date;  
  
view.environment.lighting.setTimeForCurrentLocation(hours, minutes);
```

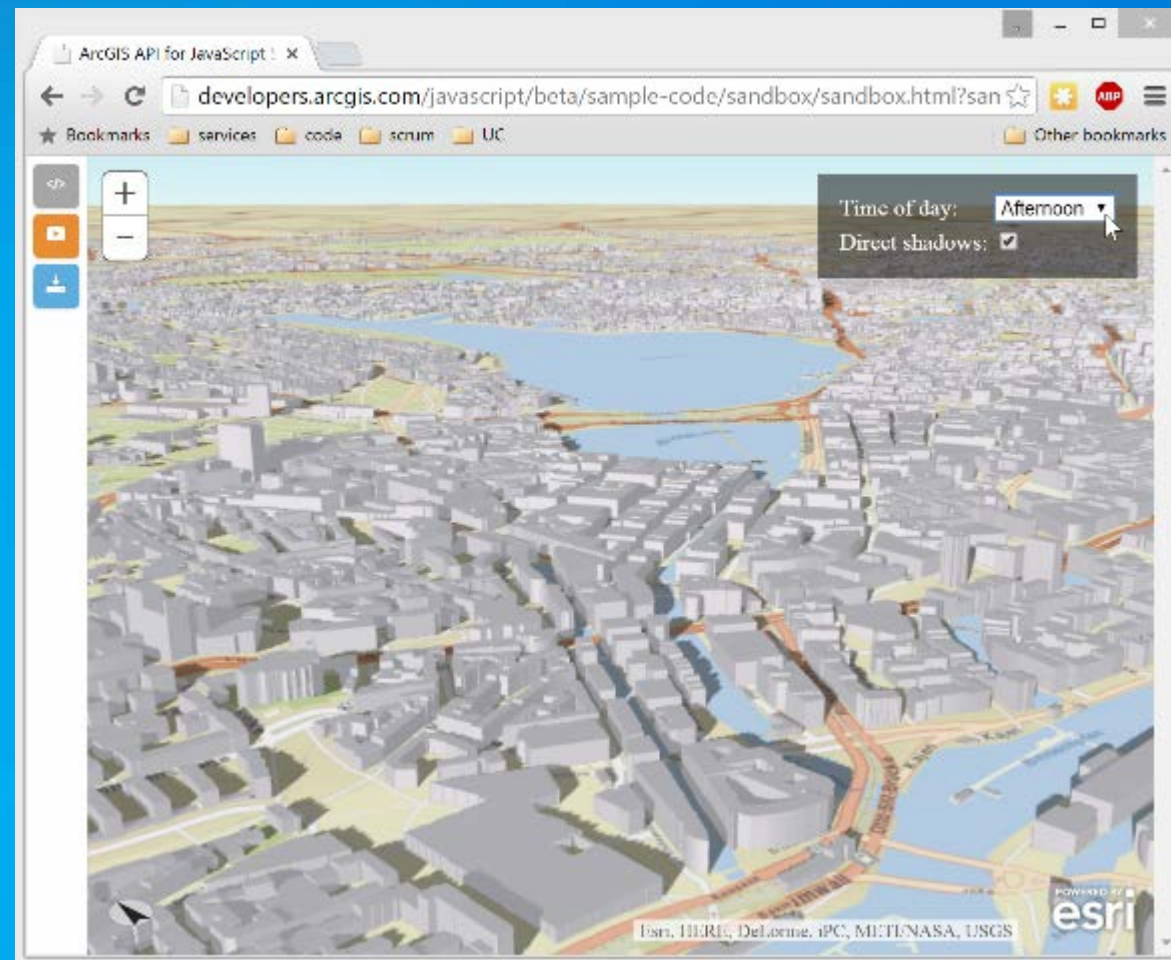
- **Enable shadows**

```
view.environment.lighting.shadows = true;
```

SceneView

Environment: light and shadows

```
//Set the environment in SceneView
view.environment = {
  lighting: {
    directShadows: true,
    date: new Date("Sun Mar 15 2015
                   09:00:00 GMT+0100")
  }
};
```



Widgets

The background features a complex geometric design. On the left, there are overlapping shapes in shades of purple and yellow, some with a grid-like texture. The rest of the background is a gradient of blue, transitioning from a darker shade on the left to a lighter shade on the right.

Beta 1 – Widgets

Map navigation / Controls

[Attribution](#) - [Compass](#) - [Zoom](#)

[BasemapToggle](#)

Other widgets

[Search](#)

[Measurement](#)

[Popup](#) - [PopupTemplate](#)

Wrap-up

- **ArcGIS API for JavaScript 4.0: an API that spans 2D and 3D**
 - Unified where it makes sense
 - Different where necessary
- **Layer types currently supported in 3D**
 - **ArcGISTiledMapServiceLayer**
 - **GraphicsLayer**
 - **FeatureLayer**
 - **ArcGISElevationLayer**
 - **SceneLayer**

The background features a gradient of blue and purple hues. On the left side, there are several overlapping geometric shapes, including a yellow map of the United States and a purple shape. The text "What's next" is centered in the upper half of the image.

What's next

What's next

- **API 4.0**
 - More betas, more functionality
 - 4.0 "final" in 2016
- **Several updates throughout 2015**
 - Support for other layer types
 - New features in 3D symbology
 - Improved performance and visual quality
 - More view options for view configuration

Want to learn more?

- <https://developers.arcgis.com/javascript/beta/>
- **User Conference:**
 - **Advanced workflows for creating 3D Web Scenes in ArcGIS Online**
 - Wednesday, 22 Jul 2015, 3:15pm - 4:30pm
 - Location: Room 14 B
 - **ArcGIS API for JavaScript - Road Ahead**
 - Wednesday, 3:15 – 4:30 pm
 - Room: 10
- **Web Developer Island**

Thank you...

- Please fill out the session survey in your mobile app
- Select “ArcGIS API for JavaScript: Building 3D Web Apps” in the Mobile App
 - Use the Search Feature to quickly find this title
- Click “Technical Workshop Survey”
- Answer a few short questions and enter any comments

Q & A

