ArcGIS API for JavaScript: What's New

Yann Cabon, Julie Powell, Bjorn Svensson
Simplicity, Productivity & Consistency

Redesigned widgets for building compelling apps
Streamlined architecture & programming patterns
Improved SDK developer experience
Consistency to improve productivity
Redesigned widgets for building compelling apps
PLATFORM INTEGRATION

Manage and display groups, users, content
Web Maps

What is a web map?

Web maps are 2D and 3D graphics that combine information from various data sources into a single, interactive platform. ArcGIS for Developers provides developers with the tools to create web maps using ArcGIS API for JavaScript. Web maps can be used for a wide range of applications, from urban planning to environmental monitoring.

The web map specification

The specification for ArcGIS web maps is based on the OpenGIS GeoServer specification and can be easily transformed and extended. It includes specifications for layers, filters, and data types.

Web Scenes

What are web scenes?

Web scenes are 3D graphics that combine information from various data sources into a single, interactive platform. ArcGIS for Developers provides developers with the tools to create web scenes using ArcGIS API for JavaScript. Web scenes can be used for a wide range of applications, from urban planning to environmental monitoring.

The web scene specification

The specification for ArcGIS web scenes is based on the OpenGIS GeoServer specification and can be easily transformed and extended. It includes specifications for layers, filters, and data types.

Arcade

Scripting and expressions through Arcade

Arcade is a portable, powerful, and simple expression language written for use with the ArcGIS API for JavaScript. It is available for free and can be used to create custom expressions for web maps and web scenes. Arcade includes built-in functions for common tasks such as distances, areas, and angles, and can be easily extended with custom functions.
```javascript
var scene = new WebScene({
  portalItem: // outscouts as new PortalItem()
  id: "d10b2f212a4845780bea52241083ba7e17"
});
```

This map shows the international airports found in the U.S.

by julie.powell

Last Modified: May 11, 2016

Description

Add an in-depth description of the item.

Layers

FAA_5010_Airports

Access and Use Constraints

Owner

julie.powell
```javascript
require([  
  "esri/Map",  
  "esri/views/SceneView",  
  "esri/layers/Layer",  
  "do/for/domReady!"  
], function(  
  Map, SceneView, Layer  
  ){  
  var map = new Map({  
    basemap: "gray"  
  });  
  
  var view = new SceneView({  
    map: map,  
    container: "viewDiv",  
    zoom: 7,  
    center: [-87, 34]  
  });  
  
  Layer.FromPortalItem({  
    portalItem:  
      // autocast as esri/portal/PortalItem  
      id: "c8c479e3eb0d4e3e250f22be79f9a45c"  
  }).then(addLayer)  
  .otherwise(rejection);  
  
  // Adds the layer to the map once it loads  
  function addLayer(lyr) {  
    map.add(lyr);  
  }  
  
  function rejection(err) {  
    console.log("Layer failed to load: ", err);  
  }  
});
```
WEB MAP SUPPORT

- Approx 85% can be read by 4.4 API
- 4.4 is a big improvement over 4.3
- To Do: Labeling, Heatmap, Time, WFS, ...
MULTIVARIATE THEMATIC MAPPING

Type
Visual variables
Continuous size, color, opacity, rotation
MULTIVARIATE THEMATIC MAPPING

Visual variables
Continuous size & color on thematic or realistic representations
THEMATIC MAPPING IN 3D

3d Object SceneLayer
Continuous Color
High to low, above and below
THEMATIC MAPPING IN 3D

With textures
Without textures
REALISTIC SCENES

Realistic buildings (3d Object Scene Layers)
Street furniture (Point Scene Layers)
ELEVATION

- Esri world elevation service
- Custom elevation services
- Display and access elevation data
FUSING IMAGERY AND ELEVATION

Integrated Mesh Scene Layer
VECTOR TILES 2D and 3D
VECTOR TILES
Localized labeling with RTE support
VECTOR TILES

Printing support added in 4.4 and 3.21
POINT CLOUDS

130 ft
Improved developer experience
Dynamic Workspaces
watchUtils,
dom,
domConstruct,
on

})
esriConfig.request.proxyUrl = "/sproxy/";
esriConfig.request.corsEnabledServers.push(
  "http://gibs.earthdata.nasa.gov");

layer = new WMSTLayer(
  url: "https://gibs.earthdata.nasa.gov/wmts/epsg4326/best/",
  copyright: "<a target='_top' href='https://earthdata.nasa.gov'>Earthdata</a> by <a target='_top' href='https://www.nasa.gov'>NASA</a>",
  activeLayer: {
    id: "SRTM_Color_Index",
  }
);

map = new Map(
  layers: [layer]
);
view = new MapView(
  container: "viewDiv",
  map: map
);
view.then(function() {
  view.extent = layer.fullExtent;
  var layerList = new LayerList(
    view: view
  );
  view.ui.add(layerList, "bottom-left");
});
</script>
</head>
FEATURELAYER

Drawing enhancements for large datasets
Arcade expressions
BUILD INTERACTIVE APPS

Local geometry engine
Map interaction
Access elevations from elevation layer
BUILD INTERACTIVE APPS

Callout Lines
Highlight
Popup
BUILD INTERACTIVE APPS

Local filtering
Highlight
Callout lines
Popup
Thematic 3D styles
BUILD A GREAT USER EXPERIENCE

Redesigned API widgets
New pop-up design
Flexible UI placement
Responsive design
Widget development pattern
```javascript
var view = new MapView({
  ...
  ui: {
    components: ["zoom", "compass", "attribution"]
  }
});
view.then(function() {
  view.ui.add(legend, "bottom-right");
  view.ui.add(searchWidget, "top-right");
  view.ui.add(homeWidget, "top-left");
  view.ui.add(locateWidget, "top-left");
});
```
WIDGETS

Use OOB
Style with CSS
Create a custom view
Create a custom widget

Customize!
POPUPS

- Responsive
- Docking
- Custom actions
Styling

Default stylesheets

The following section describes the default stylesheet options available:

**main.css**
Includes styles for everything in the ArcGIS API for JavaScript application.

```html
<link rel="stylesheet" href="https://example.com/main.css">
```

**view.css**
If your application just needs the CSS for views, use this.

```html
<link rel="stylesheet" href="https://example.com/view.css">
```

Themes
Widget development

Widgets are reusable user-interface components and are key to providing a rich user experience. The ArcGIS for JavaScript API provides a set of ready-to-use widgets. Beginning with version 4.2, it also provides a foundation for you to create custom widgets.

This guide topic discusses the basic fundamentals of widget development. It does so by discussing specific areas that you should focus on when transitioning to this new framework. The foundation for creating custom widgets remains consistent, regardless of the widget’s intended functionality. The Additional information section has extra resources to help get you started.

Please note that this framework is not intended to be a direct replacement for all Dijits. One such example would be when working with dgrid. Here, you would still need to use Dijit.

This topic discusses:

- Development requirements
- Widget life cycle
- TypeScript decorators
- Widget implementation
- Completed code
EXTENSIBILITY

Custom layers for new visualizations or connect to 3rd party services.

Presentation and video from Devsummit
EXTENSIBILITY

Access to the low-level WebGL engine externalRenderers Presentation & video from DevSummit
ARCGIS API 3.X FOR JAVASCRIPT
3.X & 4.X RELEASE STRATEGY

**Q2 2016**

3.17
- Smart mapping: temporal
- Feature table - attachments
- Web map & OGC: styling & pop-ups

**Q3 2016**

3.18
- Vector tiles revamp
- Feature table - related records
- Web map & OGC: WCS, custom params

**Q4 2016**

3.19
- Arcade
- Vector tiles enhancements
- Web map updates

**Q1 2017**

3.20
- Non-earth measurement
- Vector tile improvements
- Hindi

---

**4.0 production release**

- API groundwork & SDK
- Unified 2D/3D API
- UI/UX
- Realistic & thematic visualization
- Portal integration
- Initial ArcGIS layers
- Tasks
- Identity management

**4.1**

- Dynamic Layers
- CSV Layer
- Print Task
- Integrated Mesh
- Web Scene writing

**4.2**

- LayerList widget
- Print Widget
- Input manager
- 3D Smart mapping (points)
- Vector tiles 2D/3D
- Widget development pattern
- Point Cloud Layer
- Elevation API
- Arcade

**4.3**

- “on-demand” Feature Layer (2D)
- Feature Layer ApplyEdits
- Vector tile improvements
- GeoRSS Layer
- Widget standardization
- Basemap Gallery widget
- Expand widget
- WorkflowManager tasks

*These are only highlights; this is not an exhaustive list*
ARCGIS API 3.X FOR JAVASCRIPT

What to expect...

Releases planned through summer 2018
Tech support available until July 2021
Will be hosted on esri’s CDN for the foreseeable future
MORE RESOURCES
<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Latest activity</th>
<th>Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context Menu on iOS</td>
<td>franc44</td>
<td>June 6, 2017 4:51:06 PM</td>
<td>1</td>
</tr>
<tr>
<td>Creating turn by turn navigation widget for ESRI web app builder</td>
<td>RP_Cole</td>
<td>June 6, 2017 4:22:07 PM</td>
<td>3</td>
</tr>
<tr>
<td>Dijox chart in panel</td>
<td>MicheleJean</td>
<td>June 6, 2017 2:16:37 PM</td>
<td>537</td>
</tr>
<tr>
<td>Performance problem when draw graphics around 400,000 points</td>
<td>mattswhazard</td>
<td>June 6, 2017 1:55:26 PM</td>
<td>88</td>
</tr>
<tr>
<td>DISPLAY QUERY RESULTS IN A GRID/TABLE</td>
<td><a href="mailto:laura_ruskol@adlands.esri_unitedlands">laura_ruskol@adlands.esri_unitedlands</a></td>
<td>June 6, 2017 12:53:17 PM</td>
<td>282</td>
</tr>
<tr>
<td>How to create an attachment with the current map screenshot?</td>
<td>asarjaer</td>
<td>June 6, 2017 9:04:44 AM</td>
<td>31</td>
</tr>
<tr>
<td>Get number of related records</td>
<td>Schienm_1</td>
<td>June 6, 2017 1:55:14 AM</td>
<td>33</td>
</tr>
<tr>
<td>How to utilize queryTask.executeForExtent to zoom to the extent of features corresponding to a definitionExpression?</td>
<td>JTTodd</td>
<td>June 5, 2017 8:29:51 AM</td>
<td>36</td>
</tr>
<tr>
<td>How to Get TileLayer in 4.39f's work in 3.20</td>
<td>trax2003</td>
<td>June 5, 2017 7:39:36 PM</td>
<td>52</td>
</tr>
</tbody>
</table>
Creating a predominance visualization with Arcade

by Kristan Dianes on May 20, 2017

At the March 2017 Esri Developer Summit, Dave Bayer and I gave a presentation on how to use Arcade expressions in web apps built on the ArcGIS platform. In that presentation I demonstrated a succinct way to create a predominance visualization using Arcade.

Visualizing predominance involves coloring a layer’s features based on which attribute among a set of competing numeric attributes wins or beats the others in total count. Common applications of this include visualizing election results, survey results, and demographic majorities.

Arcade is a good solution for predominance visualizations because it allows you to avoid creating new fields in a service for storing the predominant category and the margin of victory. With Arcade, you write the expression and it will return the values at runtime, allowing you to drive the color of the visualization based on the predominant category.

In our Dev Summit presentation I shared this webmap, which depicts the predominant educational attainment achieved by people in Mexico on the municipal level.
A collection of useful resources for developers using the ArcGIS API for JavaScript.

jsapi-resources

A collection of resources for developers using the ArcGIS API for JavaScript.

ArcGIS API for JavaScript 3.x

- Bower - Create custom builds of the API
- TypeScript - Class and interface definitions

ArcGIS API for JavaScript 4.x

- Bower Build - Create custom builds of the API