An Introduction to the ArcGIS JavaScript API

October 4, 2017

Alan Hammersmith – Senior GIS Web Application Developer

Mapping the HackVT Datasets Meetup
Background

BS in Geography from University of Utah (2000)

Java Developer with OnStar GIS Team at EDS (2000-2003)

Professional Services Consultant and Instructor at Esri (2003-2011)

Sr. GIS Web Application Developer at Stone Environmental (2011-now)

Outdoor enthusiast that really loves maps!
Overview

ArcGIS JavaScript API Object Model

Using Map Views, Layers and Features

Navigating Esri’s API Reference

Finding Example Code
ArcGIS JavaScript API Object Model (v4.5)

MapView (2D)  SceneView (3D)

Map

layers

GraphicsLayer
FeatureLayer
MapImageLayer
TileLayer
Basemap

Graphic/Feature
- geometry
- attributes
- symbol
- popupTemplate
- visibility
Map

Container for your layer collection
• Referenced by a MapView (2D) or SceneView (3D)
• Can have multiple views referencing same map

Example

MapView/SceneView
• Renders map in DIV element
• Set scale level, rotation, extent, etc.

```javascript
var mapView = new MapView({
  map: myMap,
  container: "viewDiv"
});
```
Basemaps

Bottom layer in your map collection

Image Tile Basemap
• Static images on server (256 x 256px)
• Created by ArcGIS Server or ArcGIS Online

Vector Basemap
• Features drawn on client side
• Can override style, if desired
• Labels remain horizontal when map rotated
• No pixelization at intermediate scales

Can use default basemap by strings

Can create your own (more effort)
Layers

Unique ID
• Used when referencing layer

```javascript
lyrOfInterest = myMap.findLayerById('myLayerID');
lyrOfInterest.visible = myCheckbox.checked;
```

Title
• Used by Legend or LayerList

Opacity
• Makes the layer see-through
• 0-1 where 1 is completely opaque, 0 is invisible

Visibility
• Hide layer from view without removing from map

Events (create/destroy)
• Used to do something after layer is ready/removed

Common types:
- TileLayer
- MapImageLayer
- FeatureLayer
- GraphicsLayer
- GroupLayer
(many others…)
Layers

2 Main Data Source Types for Layers

- **Vector** – made up of discrete coordinates (point, line, polygon…)
- **Raster** – made up of pixels (PNG, TIF, GIF…)

Group Layer

- Multiple layers that you want to relate together (ie - visibility, opacity)
- Use `visibilityMode`
  - `independent`: user can choose which child layer is displayed
  - `inherited`: user can only toggle parent layer to show/hide all
  - `exclusive`: only 1 of the child layers can be visible (*new)

TileLayer

- Created from a cached ArcGIS Server `MapService` (256 x 256 px image tiles)
- Make sure spatial reference matches your basemap
  - `WebMercator`: if using Esri basemap (*recommended)
  - `StatePlane`: if using your own StatePlane basemap

ImageryLayer

- Created from an ArcGIS Server `ImageService` (cached or not)
Layers

MapImageLayer
• Map image rendered on server and returned to client for display
• No feature information included (geometry/attributes)
• Created from MapService
• Sublayers can be turned on/off independently
• If one sublayer is turned off, entire image needs to be re-rendered
• http://maps.vcgi.vermont.gov/arcgis/rest/services/VCGI_services/VCGI_BASEMAP_WM_v2/MapServer

FeatureLayer
• Layer rendered on client (all same geometry type)
• Features returned with geometry, attributes and symbol
• Created from MapService or FeatureService layer
• FeatureService can support editing
• http://maps.vcgi.vermont.gov/arcgis/rest/services/VCGI_services/VCGI_BASEMAP_WM_v2/MapServer/20
GraphicsLayers and Graphics

GraphicsLayer

- Typically top most layer(s) in map
- Features rendered on client (like Feature Layer)
- In-memory layer with no schema (can have different geometry types)
- Can be initialized empty or from JSON data
- Can be used for representing feature selections for MapImageLayers

Graphics properties

- `geometry` - discrete coordinates of point, line or polygon
- `attributes` - data associated with feature (name, type, etc)
- `symbol` - how feature is rendered on client
- `popupTemplate` - HTML display of attributes when clicked
- `visible` - can hide feature without removing it
Querying Layers (esri.tasks and esri.tasks.support)

**IdentifyTask**
- Can return features from **MANY layers** in the MapService
- Executed with IdentifyParameters (click point, layer IDs, map extent…)
- Returns IdentifyResults (list of features at point clicked)

**QueryTask**
- Returns features from a **SINGLE layer** in a MapService
- Uses Query for filtering data spatially or by where clause
- Options for executing for count, list of IDs, related records

**Only used to get data for MapImageLayers**
- FeatureLayers already have all the data in the client
- Use FeatureLayer.queryFeatures() to get local features
- LayerView will give access to features in current extent
A note on Popups

Views contain a default Popup
- Not tied to a specific layer or feature

Can set a PopupTemplate on Layers and Graphics

```javascript
popupTemplate.title = "Stats for {ZIP}";
popupTemplate.content = "{pctImperv}% of land for {ZIP} " +
  "is impervious and has a total population of {pop2010}";
```

**content property can be set to:**
- String: use `{curlyBraces}` to replace with attribute values
- Object array: use for creating charts/graphs or displaying attachments
- Promise or Function: must resolve to string or object array
Finding more information

Navigating Esri’s API Reference

Finding Examples
• Some found in API reference
• https://developers.arcgis.com/javascript/latest/sample-code/index.html

Learning more
• https://developers.arcgis.com/javascript/
Thank you.

For more information / www.stone-env.com

Contact / alan@stone-env.com