



Esri International User Conference | San Diego, CA
Technical Workshops |

ArcGIS API for JavaScript – Advanced Topics

Jeremy Bartley, Kelly Hutchins, and Derek Swingley

Agenda

- **Map Enhancements**
- **InfoWindow**
- **ArcGIS.com Webmap integration**
- **FeatureLayer**
- **Mobile Development**
- **HTML5**

Enhanced Map Control

Wrap Around



Wrap Around -- Notes

- Show no more than 360 degrees at a time
- Map.extent is always linear
- Normalize geometries when working with the server if geometry is beyond ± 180
 - `esri.geometry.normalizeCentralMeridian(geoms)`
- `esri.tasks` auto normalizes for you
 - `esri.config.defaults.geometryService`



Wrap around notes

- **Map control must be used with Web Mercator and ArcGIS Online WGS 84 tiling schemes as the base layer**
- **ArcGIS Map & Image services work best if they ArcGIS server 10 or greater**
 - **Support WKT requests with dynamic central meridian**
- **Services that don't support WKT work but may get clipped when getting close to the dateline**

Client side geodesic operations

- `esri.geometry.geodesicAreas(gcsPolygons, unit)`
 - [sample](#)
- `esri.geometry.geodesicLengths(gcsPolylines, unit)`
 - [Sample](#)
- `esri.geometry.geodesicDensify(gcsPolylinePolygon, maxSegmentLength)`
 - **Densifies the geometry along the great ellipse**



Snapping

- By default snap to all layers in the map
- Snap to points, endpoint, vertices, and edge
- Snapping Manager controls snapping options



InfoWindow and InfoTemplate

Info Window

- Title + Content
- Place anywhere on map
 - `InfoWindow.show(mapPoint, anchorPlacement)`
- Shown when clicked on a graphicslayer if Graphic has defined infoTemplate



Content of InfoWindow

Format Info Window Content

The [InfoTemplate](#) class is used to define the content and title for an Info Window. If you are using version 2.2 or greater, you can define the content and title using either a string or function. Prior to version 2.2, you can only use a string.

Format content using a string

To format content using a string, you will create a string value that defines the content to display. The string can include HTML markup, attribute placeholders and formatting functions. Let's look at a few examples of ways you can format content using a string:

- **Simple String**

In this example, a simple string is used to display text in an info window.



```
myTemplate.content = "This is a location of interest";
```
- **HTML Markup**

You can also include HTML markup in the string, in this example the info window will contain several paragraphs with some content in bold.

```
myTemplate.content = "This is some content with  
<b>HTML</b> markup!<p>This is a paragraph  
</p><p>Another Paragraph</p>"
```
- **Placeholders**

Use placeholders, created using `$()`, to define the attribute value that should display in the info window. At run-time parameter substitution occurs and the placeholder is replaced with the actual attribute value for the selected graphic. In this example, `OIL_CUMULATIVE` and `YEAR` are valid attribute fields and will be replaced with the value when the info window is displayed.

```
myTemplate.content = "<b>Status</b>: ${STATUS}" +  
    "<br><b>Cumulative Gas</b>: ${CUMUL_GAS} MCF" +  
    "<br><b>Total Acres</b>: ${APPROXACRE}" +  
    "<br><b>Avg. Field Depth</b>: ${AVG_DEPTH} meters";
```



Let's go to the doc...

Customize look of InfoWindow

- Simple InfoWindow can be stylized with CSS



- Default InfoWindow can be stylized by creating a new sprite defining the colors and then using it in your app

Extend InfoWindowBase to create your own InfoWindow

- Need to implement a few methods
 - Hide, show, setContent, setTitle, resize
- Once implemented you can set your new InfoWindow on the map
- API for working (show/hide) with InfoWindow will remain unchanged.



Popups

- Implementation of InfoWindowBase
 - Navigate through selection
 - Zoom and highlight
 - Maximize the info window



PopupMobile

- Inherits from InfoWindowBase
- Designed for small screen size
 - Embedded Maps
 - Mobile



ArcGIS.com Integration

esri.arcgis.utils.createMap()

- **Creates an instance of esri.Map initialized with all layers specified in the webmap**
- **Supports:**
 - All basemap and operational layers
 - Supports features stored in the map
 - Supports webmap defined popups
- **Constructed from arcgis online webmap ID or by value from JSON**

ArcGIS.com webmap popups

- If webmap has defined popups createMap will:
 - Listen for map click events and when clicked query the layer for each defined popup
 - Behavior can be disabled via ignorePopups option
 - Selected geometry is generalized if possible
- You can style the InfoWindow that contains the popup using css
 - `dojo.addClass(map.infoWindow.domNode, "chrome");`



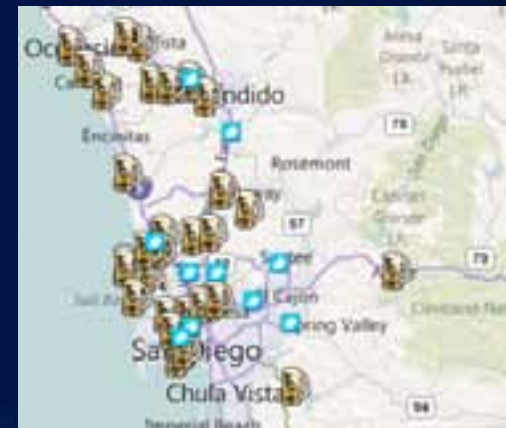
Extending createMap's InfoWindow

- Uses custom InfoWindow extended from InfoWindowBase
- Handles paging, media (one or multiple), and supports full screen view of content
- Includes an actions bar that supports “zoom to”



- You can get the currently selected feature from the InfoWindow. Use it to add your own tool to the actions bar

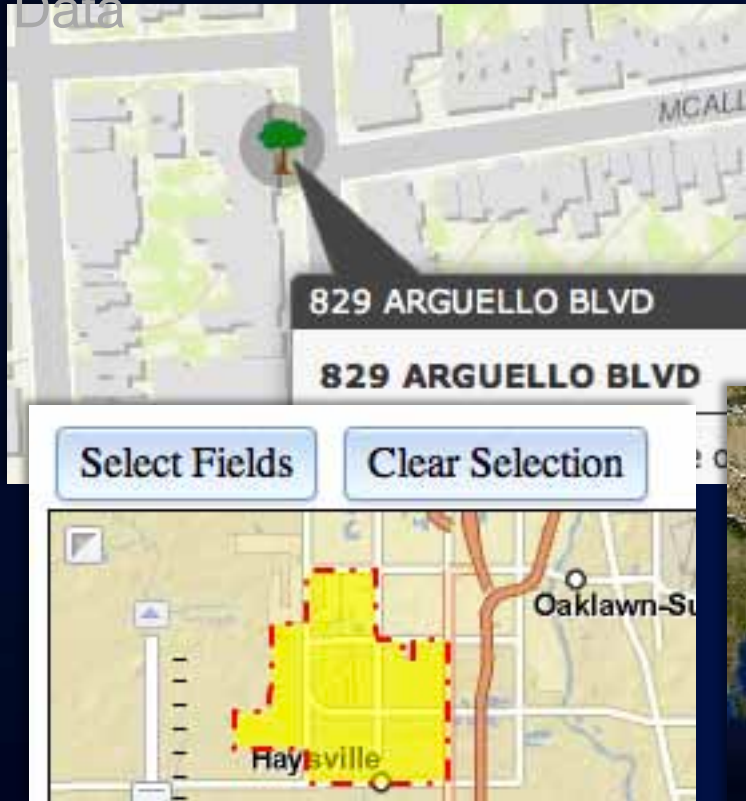
Use ArcGIS.com Web application Templates to get started!



esri.layers.FeatureLayer

FeatureLayer

Client Side
Data



Feature Selections

Web Editing



Pagination

FeatureLayer

Series of Blog Posts on Performance and Best Practices

How many graphics?

Vector Tiling

maxAllowableOffset

Generalize Features

Mobile Development

Mobile support in JSAPI

- Touch enabled Map control
- Touch enabled widgets
- Touch enabled move support

iOS

- **Best Mobile Browser**
- **Supports two finger pinch zoom on map control**
- **Create and edit geometries**
- **No access to photos/videos from the browser**
- **No support for freehand drawing style**
- **Browser supports**
 - **HTML5 Geolocation API**
 - **Web Storage**
 - **Offline Web Applications**
 - **Access to GyroScope**

Blackberry Torch

- Supports touch pan
- Supports create/edit of geometries
- Does have access to the camera roll
- No support for two finger touch
- No support for freehand drawing style

Android

- **2.2 and greater**
 - Also depends on the firmware that the device manufacturers apply to the phone – Sprint EVO
- **SVG is not supported on Android.**
 - JSAPI falls back to canvas
- **Can create geometries, but not edit them**
- **No support for two finger touch**
- **Does have access to the camera roll**

Mobile Widgets

- **Mobile popup**
- **Touch-aware thumbnail gallery**
- **Displays horizontal scrolling view of thumbnail images.**
- **Touch access so users can flick through the gallery to select or view items.**

Mobile Frameworks

- **Native application look and feel**
 - **Animated transitions**
 - **Toolbars**
 - **Buttons**
 - **List views**



jQuery Mobile

- Announced in late 2010
- Currently in alpha 3
- Active user community
- Lots of samples and doc
- Theme Framework



Dojox.mobile

- **Features and behavior similar to jQueryMobile**
- **Good demos**
- **Poor documentation**



Sencha

- ExtJS
- UI components
- iPad support
- Local storage support
- Data integration
- Apple iOS and Android only
 - Blackberry coming soon...



HTML5

HTML5 components in (or soon to be in) the JSAPI



Canvas (pixel control
and android graphics)



CSS3 Transitions

No Proxy

Cross Origin Resource
Sharing

Web Workers

CPU Intensive Operations
(asynch data processing,
client geometric operations)

Simple Marker Symbol	IE6	IE7	IE8	IE9	Chrome 10	Firefox 4
Query Time	1297	1703	1406	1629	688	1800
Draw Time	2375	2844	1125	93	95	84
Clear Time	281	516	157	15	4	29
Pan Time	7577	12218	4751	390	395	370

High performance browsers!

What we are teaching app developers



Geolocation API



Web Storage & Offline Web Apps (app cache)



File API, Drag and Drop API