



# ArcGIS API for JavaScript: Advanced Topics

Andy Gup

@agup

Esri Developer Summit  
Washington, DC

# My goals?

Discuss important coding patterns

Help your long-term coding skills

Faster, better, more maintainable code

Save you a ton of time!



# Your skill sets?

Moderate knowledge of JavaScript

Moderate usage of ArcGIS JavaScript API

Basic familiarity with dojo & AMD

Moderate usage of 3<sup>rd</sup> party libraries



# Who am I?

Andy Gup

Developer Evangelist

[agup@esri.com](mailto:agup@esri.com)

[www.andygup.net](http://www.andygup.net)

[github.com/agup](https://github.com/agup)

@agup

# **We will cover:**

Object-oriented JavaScript

Unit Testing

Custom widgets

Performance

Working with 3<sup>rd</sup> party JS libraries

Offline JavaScript

# Object Oriented JavaScript

AMD

require, define

modules

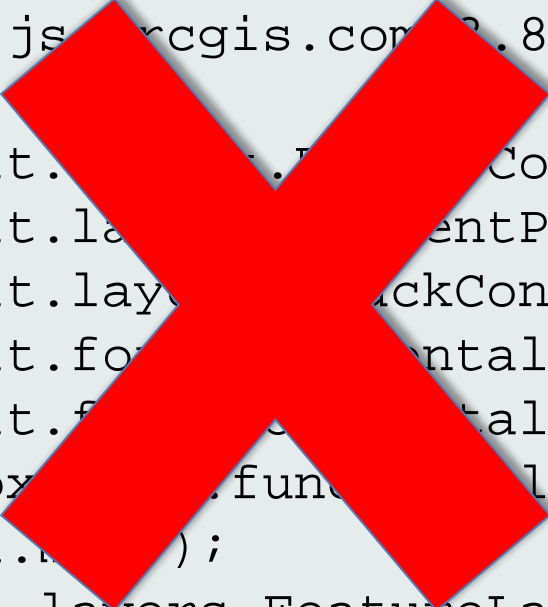
declare Classes

Closure

Scope

# Legacy dojo loader – bye bye

```
<script>var dojoConfig = { parseOnLoad: true };</script>  
<script src="http://js.arcgis.com/3.8/"></script>  
<script>  
  dojo.require("dijit.layout.BorderContainer");  
  dojo.require("dijit.layout.ContentPane");  
  dojo.require("dijit.layout.StackContainer");  
  dojo.require("dijit.form.HorizontalSlider");  
  dojo.require("dijit.form.HorizontalRuleLabels");  
  dojo.require("dojox.lang.functional.fold");  
  dojo.require("esri.layers.FeatureLayer");  
  dojo.require("esri.layers.FeatureLayer");
```



# AMD

Asynchronous Module Definition API

Global functions for loading code

*Normally* asynchronous & concurrent

Decreases page load times

Also used by RequireJS, curl, bdLoad, etc



# What about Dojo AMD??

AMD community standards compliant

Modules work with other AMD loaders/libraries!

# require – load modules

```
require(  
    // (optional; object) configuration object  
    configuration,  
  
    // (optional; array of strings) module identifiers to  
    // load before calling callback  
    dependencies,  
  
    // (optional; function) function to call when  
    // dependencies are loaded  
    callback  
)
```

# require – load modules

```
require([
```

```
  "esri/map",
```

Module

```
  "esri/arcgis/utils",
```

```
  "dojo/domReady!"
```

Alias

```
], function(Map, arcgisUtils){
```

```
  var map = new Map("mapDiv" ...);
```

```
});
```

# define – load modules (no config)

```
define(  
    // (optional; string) explicit module identifier  
    moduleId,  
  
    // (optional; array of strings) list of module  
    // identifiers to load before calling factory  
    dependencies,  
  
    // (function or value) the value of the module, or a  
    // function that returns value of the module  
    factory  
)
```

# define

Module creation is lazy and asynchronous

Modules are cached in the module namespace

# define

```
// extras/gps.js
```

```
define(["esri/geometry"], function(geometry) {
```

```
return {
```

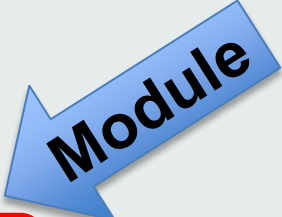
```
  gpsToPoint: function(geolocation) {
```

```
    var point = new geometry.Point(...);
```

```
  }
```

```
}
```

```
})
```



Module



Alias



# Custom module pattern – using dojo CDN

```
<script><!-- Declare before loader executes -->
  var locationPath = location.pathname.replace(/\[/[^\]]+$/, "");
  var dojoConfig = {
    async: true,
    paths: {
      extras: locationPath + "/extras"
    }
  }
</script>
<script src="http://js.arcgis.com/3.8/"></script>
<script>
require(["esri/map", "extras/gps"], function(map, gps) {
  . . .
  var point = gps.gpsToPoint(...);
}
</script>
```

# Incorrect path


✘ "NetworkError: 404 Not Found - <http://js.arcgis.com/3.8/js/dojo/extras/test2.js>"

✘ ▶ Error: scriptError

The resource from this URL is not text: <http://js.arcgis.com/3.8>

src: dojoLoader

info: [ "<http://js.arcgis.com/3.8/js/dojo/extras/test2.js>", error ]



```
/  
index.html  
js/  
  lib/  
    dojo/  
    dijit/  
    dojox/  
my/  
util/
```



# DOM ready

```
//Works with both define and require
define(["esri/geometry", "dojo/domReady!"],
    function(geometry) {

        //Wait until DOM is ready
        function init(){
            // TO-DO
        }() // Run immediately
    }
)
```

# declare - named Classes

```
// You must define "dojo/_base/declare" first
// Allows for inheritance
declare("namespace.ClassName",null, {
  _privateVar : null,
  publicString : "",
  init: function(){
    //TO-DO
  },
  changeSymbol : function(){ . . . }
})
```

# declare - anonymous Classes

```
// Scoped anonymous class (no namespace)
// Allows for inheritance
var ClassName = declare(null, {
  _privateVar : null,
  publicString : "",
  init: function(){
    //TO-DO
  },
  changeSymbol : function(){ . . . }
})
```

# Why care about require, define, declare?

Scalable

Avoids repeated code (smaller download)

More stability

Reduced global variable leakage

Reduced debugging time

You “can” use it with 3<sup>rd</sup> party JS libraries!

# require, define, declare

```
var locationPath = location.pathname.r
var dojoConfig = {
  paths: {
    extras: locationPath + "/extr
  }
}
</script>
<script src="http://js.arcgis.com/3.8/"></
<script>
  "use strict"
  var map; //Global

  require(["esri/map", "extras/test2", "do
    map = new Map("map", {
      basemap: "topo",
      center: [-122.45, 37.75], // lo
      zoom: 13,
      sliderStyle: "small"
    });

    var myTest = new extras.MapTools()

    map.on("load", function(evt){

      myTest.setExtent(map);
      console.log("Test 2 get map id

    })

  });
```

# Unit Testing

Validates methods, properties and callbacks

Isolates the smallest piece of testable code

Jasmine, Mocha and others

<http://pivotal.github.io/jasmine/>

# Why unit tests?

Absolutely helps you write better code

Helps you quickly test code after changes

# Downsides to unit tests?

Requires learning the testing library.

They are “only” as good as the test case!!



# Jasmine – one example

[Open source project](#)

Behavior driven

Independent of 3<sup>rd</sup> party JS frameworks

Does not require a DOM (fast)

Can be used [stand-alone](#) (locally)

# Jasmine – app life cycle

Load CSS

Load Map

Load Layer(s)

Final Layer event loads Jasmine

e.g. “load” or “layer-add-result”

Jasmine loads unit test spec

# Jasmine – example

```
var map;

require(["esri/map",
  "esri/layers/ArcGISTiledMapServiceLayer", "dojo/domReady!"],
function(Map, Tiled) {
  map = new Map("map");
  var tiled = new Tiled("http://. . ./Terrain/MapServer");
  map.on("load",runJasmine)
  map.addLayer(tiled);

  function runJasmine(){
    . . .
  }
}
);
```

# Jasmine – example spec

```
"use strict"
describe("My map tests", function(){
  it("validate map", function(){
    expect(map.id).toEqual("map");
    expect(map).toEqual(jasmine.any(Object));
  })
  it("load feature layer", function(){
    var done = function(evt){
      expect(evt.layer.type).toEqual("Feature Layer")
    }
    map.on("layer-add-result", done);
  }
})
}
```

# Jasmine test example



# Widgets

Application framework

Singular purpose

Reusable/shareable

Examples:

[Attachment Editor Widget](#)

[Directions Widget](#)

[Geocoding Widget](#)

# Custom Widgets

## [Lifecycle of dijits](#)

dojo/dom.byId

dijit/registry

registry.byId()

dijit/\_WidgetBase

dijit/\_TemplatedMixin

# Dijit Basics

Created with `dojo.declare`

Inherit from `dijit._WidgetBase`

Use `dijit._TemplatedMixin`



# Dijit Lifecycle (minimum req's)

Override *at least* the following:

`constructor`

`startup`

`destroy`

# Custom widget example



# Client Performance

How long does user have to wait??

...on their machine

...on their internet connection

# Client Performance

Generalizing features

Custom scale dependencies

Reducing number of layers

Reducing attributes and properties

# Client Performance

Querying/Retrieving features

Parsing

Displaying

Analyze CPU performance

Use timers to check elapsed times

# Performance - Parsing Features

How many features per response?

How complex are the features?

How efficient is your parser algorithm?

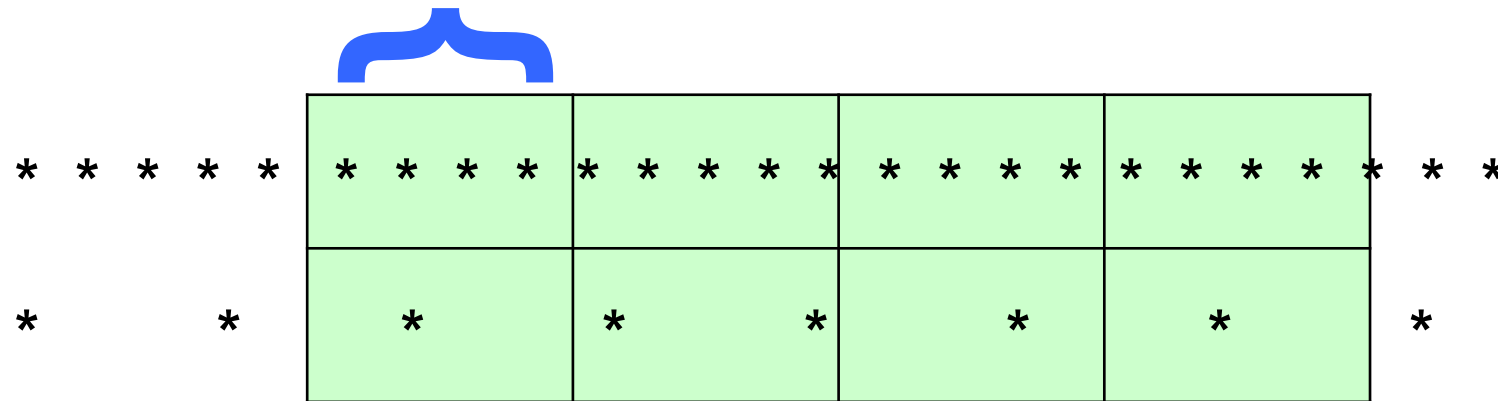
# maxAllowableOffset (FeatureLayer)



Display more than one vertex per pixel

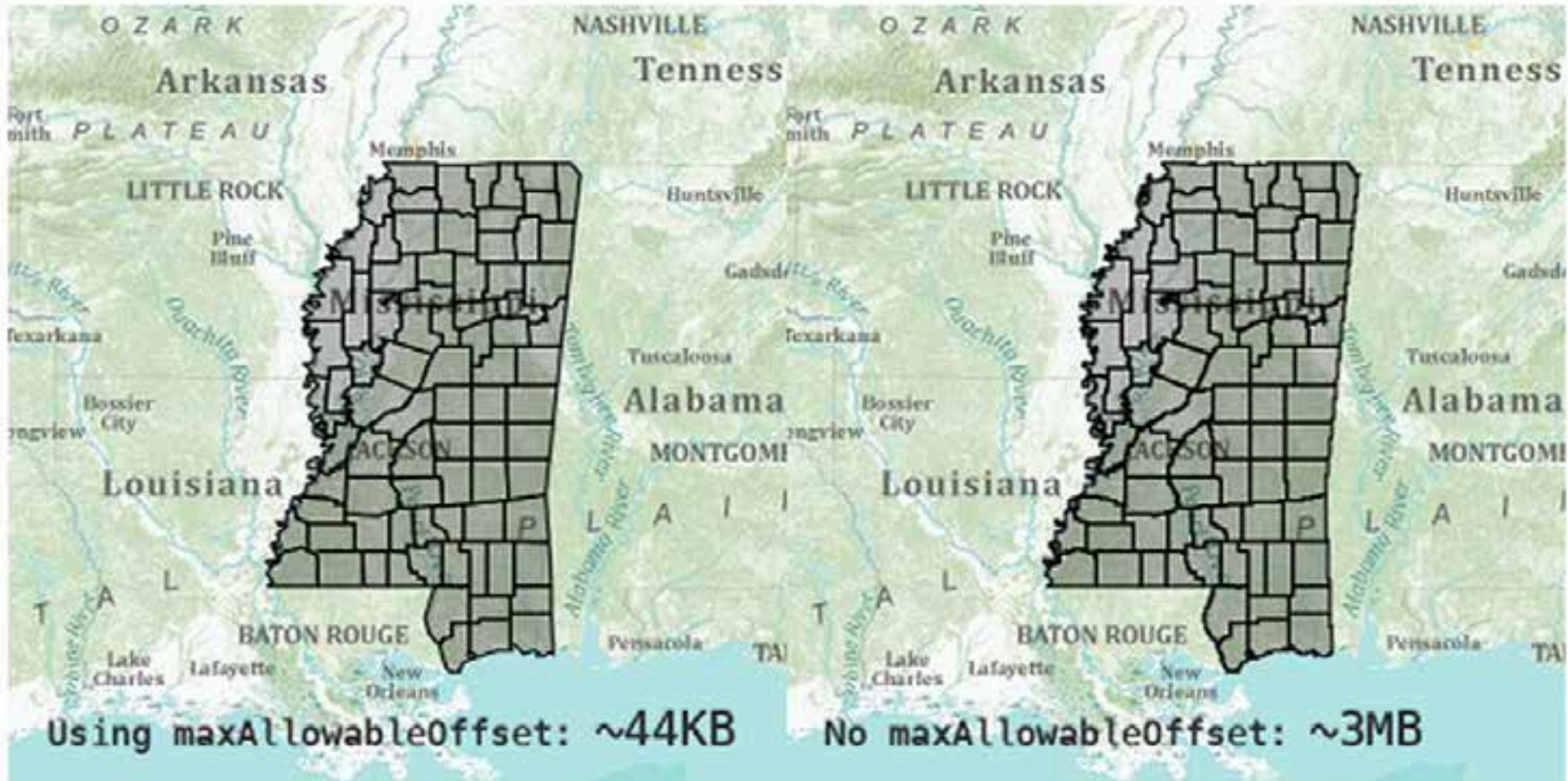
JS API v2.7+ with ONDEMAND mode

Reduces graphics size by up to 95%!



~5 vertex/Pixel  
~1 vertex/Pixel

# maxAllowableOffset (FeatureLayer)





# FeatureLayer.minScale & maxScale

```
// Limit zoom levels
var fl = new esri.layers.FeatureLayer(url, options);
    on("load", function() {
        fl.minScale = minScale; //min visible. Number
        fl.maxScale = maxScale; //max visible. Number
    });
```

Performance demos

Polygon

Performance



Performance Playg

et:

single level. Zooming ha



# Performance Profiling

Elements Network Sources Timeline Profiles Resources Audits Console

Profiles

Select profiling type

- Collect JavaScript CPU Profile  
CPU profiles show where the execution time is spent in your page's JavaScript functions.
- Take Heap Snapshot  
Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.
- Record Heap Allocations  
Record JavaScript object allocations over time. Use this profile type to isolate memory leaks.

Start

# Working with 3<sup>rd</sup> Party JavaScript

jQuery, Bootstrap, Angular.js, node.js etc etc

Problem areas:

- Getting map to load
- Rotating map
- Resizing map
- Loading maps in secondary views (or tabs)

# Working with 3<sup>rd</sup> Party JavaScript

[github.com/esri](https://github.com/esri)

[Twitter bootstrap](#)

[Dojo bootstrap](#)

[jQuery Mobile](#)

[PhoneGap Quickstart](#)

[Sencha Touch](#)

[angular.js \(coming soon\)](#)

# Working with 3<sup>rd</sup> party JavaScript libraries

The screenshot shows the top navigation bar of the Bootstrap Map JS website. It includes a purple location pin icon, the text 'Bootstrap Map JS', and navigation links for 'Get Started', 'Maps', 'Grid', and 'Web Map'. The main content area features a large heading 'Bootstrap Map JS', a sub-heading 'Build responsive mapping apps with a powerful web', and a blue 'Get Started »' button. Below this is a light blue note box stating 'NOTE: Supports Bootstrap ver 3.0.0.'. At the bottom, there are two columns: 'ArcGIS Ready' with the text 'Build powerful mapping applications with the' and 'Responsive' with the text 'Use the new Bootstrap 3 grid s'.

Bootstrap Map JS

Get Started Maps Grid Web Map

# Bootstrap Map JS

Build responsive mapping apps with a powerful web

Get Started »

NOTE: Supports Bootstrap ver 3.0.0.

## ArcGIS Ready

Build powerful mapping applications with the

## Responsive

Use the new [Bootstrap 3](#) grid s

# Offline JavaScript

Partially or fully disconnected workflows

Reuses existing JavaScript skills

Cross-platform

Offline data collection

Search & Rescue, inspections, delivery, etc

# Offline JavaScript

Offline Editing

Offline Tile Management

Resyncing Edits

[github.com/esri](https://github.com/esri) repository coming soon!!



# Offline JavaScript – storing data in browser

IndexedDB (~50 MB)

Web Storage (= $\leq$  5 MB)

SQLite (PhoneGap)

SD Card (PhoneGap)

FileWriter/FileReader (?)

# Offline JavaScript

## Map: Sample Webmap

Sample Webmap to test Offline Tile library. You can use your

### Min Zoom Level

### Current Zoom Level

### Max Zoom Level

Level	Tile Count	Size Mb (approx.)
13	2	0.04 Mb
14	4	0.09 Mb
15	6	0.13 Mb
16	12	0.26 Mb
17	35	0.77 Mb
18	104	2.28 Mb
19	375	8.2 Mb
<b>Total</b>	<b>551</b>	<b>12.05 Mb</b>

[Prepare for Offline](#)[Delete All Tiles](#)[Go Offline](#)[Usage: 2.29 Mb \(67 tiles\)](#)[Show Stored Tiles](#)

1. Navigate to your area of

2. Click 'Prepare for Offline'

# Recap

Object-oriented JavaScript

Unit Testing

Custom widgets

Performance

Working with 3<sup>rd</sup> party JS libraries

Offline JavaScript

# Questions??

Andy Gup

Developer Evangelist

[agup@esri.com](mailto:agup@esri.com)

[github.com/agup](https://github.com/agup)

@agup



Understanding our world.