Advanced Development with ArcGIS API for JavaScript

Kelly Hutchins (@kellyhutchins)
Derek Swingley (@derekswingley)
Feature Layers – Capabilities

• Client Side Data
• Feature Interactivity / Selections
• Easy Pagination
• Web Editing
Feature Layers – Implementation

- Generalize On the Fly
- Tiled Vectors
- Smart Usage of Browser Cache
- Use Appropriate Precision
Working with third party data
Apps can access Images, CSS, JavaScript (through the Script tag) across different domains.
JSONP

• Script tag inserted into page
• Points to resource that returns json
• Server wraps data in callback function
• Eval to get access to data
CORS

Browser

esri.request

Your code

ArcGIS Server

Remote Server
Cross-Origin Resource Sharing - Working Draft

Method of performing XMLHttpRequests across domains.

<table>
<thead>
<tr>
<th>Show all versions</th>
<th>IE</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Blackberry Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current</td>
<td>10.0</td>
<td>19.0</td>
<td>25.0</td>
<td>6.0</td>
<td>12.1</td>
<td>6.0</td>
<td>5.0-7.0</td>
<td>4.2</td>
<td>7.0</td>
</tr>
<tr>
<td>Near future</td>
<td>20.0</td>
<td>26.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Farther future</td>
<td>21.0</td>
<td>27.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Usage stats:  
  - Support: 68.53%  
  - Partial support: 23.86%  
  - Total: 92.39%
Access external data

- esri.request
- Formats
  - Plain Text
  - XML
  - JSON
  - JSONP
Under the hood

- XMLHttpRequest
- Dojo methods
  - xhrGet, xhrPost, Script get via JSONP
- esri.request
Pseudocode example

```
JSON / XML / plain text:
If JSON / XML / PlainText request
  If target not on same domain && CORS not available
    Rewrite request url to use proxy
  If request size is 2000 characters or less
    Call dojo.xhrGet
  Else
    Call dojo.xhrPost
```
Smart

- CORS and Proxy
- Integrated with the Identity Manager
Using esri.request

```javascript
var request = esri.request({
    // Location of the data
    url: "<String>",
    // Service parameters if required, sent with URL as key/value pairs
    content: {
        parameter1: <value>,
        parameter2: <value>
    },
    // Data format
    handleAs: "<String>"
});
```
Callback functions

```javascript
function requestSucceeded(data) {
    console.log("Data: ", data); // print the data to browser's console
}

function requestFailed(error) {
    console.log("Error: ", error.message);
}

request.then(requestSucceeded, requestFailed);
```
Demo
Using esri.request
Esri.setRequestPreCallback

GET findAddressCanc

<table>
<thead>
<tr>
<th>Params</th>
<th>Headers</th>
<th>Response</th>
<th>JSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>outSR</td>
<td>{&quot;wkid&quot;:102100}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>singleLine</td>
<td>146317</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Deferreds: Making Async Easy

- Handle async requests in response to a value that is not yet available.
Demo
Using Deferreds
Deferred Lists

dParcels = app.qtParcels.execute(app.qParcels);
dBuildings = app.qtBuildings.execute(app.qBuildings);
console.log("deferreds: ", dParcels, dBuildings);
dList = new dojo.DeferredList([[dParcels, dBuildings]]);
dList.then(handleQueryResults);
Custom Info Window

**2800 GEARY BLVD**

**Type** Graffiti Complaint – Private Property

**Date** 08/19/2009

Attachments:
No attachments found

---

**Idaho**

The estimated unemployment rate here was 8.70% in summer 2010, affecting an estimated 67,766 people age 16 and over.

**Total and Unemployed Population**

*Hover cursor over or top column for details.*
Custom Behavior
Object-Oriented JavaScript

- OOJS == Modules
- One class per module
- Avoid single-file, spaghetti code apps
- Re-usable, portable code
- Application maintenance

Tutorials:
- Write a class
- Build a custom widget
Create a Class

- **AMD**: `define()` creates the module and loads dependencies
- **Legacy**: `dojo.provide()` and `dojo.require()`
- `dojo/_base/declare()` or `dojo.declare()`
- [Instagram search example](https://www.instagram.com/search)
AMD

- Asynchronous Module Definition
- Community-driven
  - [https://github.com/amdjs/amdjs-api/wiki/AMD](https://github.com/amdjs/amdjs-api/wiki/AMD)
- Dependency Management
- Portable Code
- Asynchronous (performance, debugging, cross-domain)
- Sitepen resources
AMD and the ArcGIS API for JavaScript

- Converted all modules (100+) to AMD at 3.4
- Created nearly 100 new modules while refactoring
- Say goodbye to esri.*, dojo.* and dojo.require!
- …but 100% compatible with older module loading style
- …and dojo.require supported until Dojo 2.0
- Documentation for our AMD modules to come…
Custom Widgets

- Dijit
- Standard file structure
- JS for custom methods and properties
- HTML template for layout elements
- CSS for style
Dijit Lifecycle

- Set of methods that fire during dijit creation
- Convenient access to the widget as it is created
Custom Dijit – Gauge

- “Dashboard” style display
- Read JSON from an ArcGIS.com webmap
- Provide a way to connect to a map
• Kelly Hutchins (@kellyhutchins)
• Derek Swingley (@derekswingley)

http://esriurl.com/survey

Wednesday, March 27th – Offering ID: 248
Thursday, March 28th – Offering ID: 391