Building Mobile Apps with the ArcGIS API for JavaScript

Andy Gup, Lloyd Heberlie, Thomas Other
Agenda

Capabilities
Managing app life-cycle
Working with locally hosted builds
Working from JS frameworks
Debugging
Working with 3D on Mobile
Expectations
Capabilities
Lloyd Heberlie
BasemapToggle
LayerList action
Locate

Demo
Popups
Calcite Maps - Settings
Calcite Maps - Settings
Calcite Maps - Basemaps
Calcite Maps - Basemaps
HTML5 input types

type="email"

Email

Number

Telephone

Range

List input

DateTime

Time

Color picker

Set date

Cancel Clear Set
Mobile media queries

```html
<!DOCTYPE html>
<html data-browser="Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Mobile Safari/537.36">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no">
  <title> - 4.4</title>
</head>
<body>
<script>
// ------------------------------------------
// js, sass - user-agent mixin
// ------------------------------------------

document.documentElement.setAttribute("data-browser", navigator.userAgent);
</script>
</body>
</html>
```
Mobile media queries

- macOS
- Android
- iPhone
- iPad
Mobile media queries

Constructors

new MapView(properties)

Parameter:

properties Object

Example:

// Typical usage
var view = new MapView();
// ID of DOM element containing the view container: "viewDiv",
// Map/WebMap object
map: new Map();
Mobile media queries

```html
<!DOCTYPE html>
<html data-browser="Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Mobile Safari/537.36">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no">
  <title></title>
</head>
<body>
  <script>
    // Table viewport queries need a different CSS selector than desktop viewports.
    var mqDesktop = window.matchMedia("(min-width: 1024px) and (max-width: 2560px)");
    var mqTablet = window.matchMedia("(min-width: 768px) and (max-width: 1023px)");
    var mqPhone = window.matchMedia("(max-width: 767px)");
    if (mqDesktop.matches){
      console.log("Desktop");
    }
    if (mqTablet.matches){
      console.log("Tablet");
    }
    if (mqPhone.matches){
      console.log("Phone");
    }
  </script>
</body>
</html>
```
Capabilities
Demos
Managing initial load times
Initial loading in a mobile browser

- Often misunderstood
- Many things happening
- Vulnerable to timing issues
- Consider user’s experience
Initial loading in a mobile browser

CSS
3rd party libraries
Frameworks
Framework plugins
ArcGIS JS API
GIS queries
GET map tiles
Initial loading in a mobile browser

- CSS
- 3rd party libraries
- Frameworks
- Framework plugins
- ArcGIS JS API
- GIS queries
- GET map tiles
Initial loading in a mobile browser

Wait for framework initialization

- In Angular: implements OnInit
- In Cordova: ‘deviceready’ event
- In ArcGIS: dojo/domReady!
Consider synchronous patterns

Framework init

Map ready

dojo/ready!

Feature service loaded
Lazy load modules and libraries

Map ready

Wait for user input

Load widget
Lazy load demo

```javascript
function lazyLoadPointCloudLayer()
{
    require(["esri/layers/PointCloudLayer"], function(PointCloudLayer)
    {
        var plc = new PointCloudLayer();
    });
}
```
Reserve async for after app load

Wait for framework initialization
- In Angular: implements OnInit
- In Cordova: ‘deviceready’ event
- In ArcGIS: dojo/domReady!
MapView ready?

The View has a Map
View container size is greater than zero
View has a spatial reference

API Reference MapView.ready

Demo
Loadable Patterns

Any resource that handles async data access
Examples: feature layers, tasks, web maps

not-loaded

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>not-loaded</td>
<td>The object’s resources have not loaded.</td>
</tr>
<tr>
<td>loading</td>
<td>The object’s resources are currently loading.</td>
</tr>
<tr>
<td>loaded</td>
<td>The object’s resources have loaded without errors.</td>
</tr>
<tr>
<td>failed</td>
<td>The object’s resources failed to load. See loadError for more details.</td>
</tr>
</tbody>
</table>
API reference
WebMap.loadStatus
load() example

```javascript
var webmap = new WebMap(
    portalItem: { // autocasts as new PortalItem()
        id: "f2e9b762544945f390ca4ac3671cfa72"
    }
);

var view = new MapView(
    //map: webmap,
    container: "viewDiv"
);

webmap.load().then(function(r){
    view.map = webmap;
    console.log(r);
    console.log("Load Status: " + webmap.loadStatus)
}).otherwise(function(err){
    console.log("Problem loading webmap: " + error);
});
```
Splash Screens

Distract end user from long load times
Gives you control over when main app visible
Wait for MapView.then()

https://github.com/andygup/splash-screen-js
Using local builds

Lloyd Heberlie
Why?

More modules – more requests
Optimized build for app
**Web application vs. Hybrid**
Reduce http service calls for Hybrid apps
How to create custom builds?

ArcGIS API for JavaScript Web Optimizer – 3x

Bower

npm
Using local builds

Demo
More information

- Optimizing Your JavaScript App for Performance
  - Thursday, March 8 - 10:30 - 11:30 am
  - Primrose A
Working with Cordova and Ionic Native Mobile
Wait for “deviceready” event

https://github.com/Esri/quickstart-map-phonegap
Ionic and loading ArcGIS JavaScript modules

Many helper libraries!
- [angular-esri-loader](#) (Angular 4+, wraps esri-loader)
- [esri-loader](#) (platform agnostic)
- [Ionic2-esri-map](#) (Sample app)
- [ArcGIS TypeScript definitions](#)
Working with Frameworks
Mobile browser-only
Other frameworks

- Plain old Bootstrap
- calcite-maps
- ember-esri-loader
- esri-loader-react
- esri-system-js
Physical devices

http://blog.adtile.me/2014/01/08/adtile-device-lab/
Remote debugging

- Safari Web Inspector Remote
- Google Chrome remote debugging
- http://livereload.com/
- Browser-sync
Enable iOS remote web inspection

To use the Web Inspector, connect to Safari on your computer using a cable and access your iPhone from the Develop menu. You can enable the Develop menu in Safari’s Advanced Preferences on your computer.
Enable Android remote web inspection
Content Security Policy (CSP)

```html
<meta http-equiv="Content-Security-Policy" content="
    default-src
    ws://localhost:35729 'self'
    gap:
    https://dopotoolkit.org
    http://js.arcgis.com
    https://js.arcgis.com
    https://jsdev.arcgis.com
    https://static.arcgis.com
    https://services.arcgisonline.com
    https://server.arcgisonline.com;
    font-src
    'self'
    http://js.arcgis.com
    https://js.arcgis.com
    data;
    style-src 'self'
    https://js.arcgis.com
    'unsafe-inline';
    img-src
    'self'
    data;
    script-src
    'self'
    http://js.arcgis.com
    https://js.arcgis.com
    'unsafe-inline'">
```
Debugging Mobile
Demo
Working with 3D on Mobile
Topics

- GPU Hardware
- GPU Architecture
- 3D on Mobile Devices
- Performance
- JavaScript API
- Example
- Progressive Web Apps
<table>
<thead>
<tr>
<th>GPU Hardware</th>
<th>Apple A10</th>
<th>Intel Kaby Lake</th>
<th>nVidia Pascal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TBDR Architecture</td>
<td>TB/EZ IMR Architecture</td>
<td>TB/EZ IMR Architecture</td>
</tr>
<tr>
<td></td>
<td>650 MHz Clock</td>
<td>1115 MHz Clock</td>
<td>1465 MHz Clock</td>
</tr>
<tr>
<td></td>
<td>12 Clusters</td>
<td>48 Clusters</td>
<td>256 Clusters</td>
</tr>
<tr>
<td></td>
<td>&gt; 500 GFLOPS</td>
<td>&gt; 730 GFLOPS</td>
<td>&gt; 750 GFLOPS</td>
</tr>
</tbody>
</table>

Source: [http://kyokojap.myweb.hinet.net/gpu_gflops/](http://kyokojap.myweb.hinet.net/gpu_gflops/)
GPU Architecture

Immediate Mode Renderer

Tile Based Delayed Renderer

Source: https://www.imgtec.com/blog/a-look-at-the-powervr-graphics-architecture-tile-based-rendering/
# GPU Architecture

## Simple Rendering

<table>
<thead>
<tr>
<th></th>
<th>Immediate Mode Renderer</th>
<th>Tile Based Delayed Renderer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture Reads</td>
<td>150 * 4 Bytes</td>
<td>150 * 4 Bytes</td>
</tr>
<tr>
<td>Depth Reads</td>
<td>200 * 4 bytes</td>
<td>0 bytes</td>
</tr>
<tr>
<td>Depth Writes</td>
<td>150 * 4 bytes</td>
<td>0 bytes</td>
</tr>
<tr>
<td>Color Writes</td>
<td>150 * 4 bytes</td>
<td>0 bytes</td>
</tr>
<tr>
<td>Total Bandwidth</td>
<td>2600 bytes</td>
<td>600 bytes</td>
</tr>
</tbody>
</table>

ArcGIS JavaScript API

3D on Mobile Devices

- Works on mobile
  - Apple iPhone 8
  - Apple iPad Pro 2
  - Samsung Galaxy S8
  - Samsung Galaxy Tab S3
- It's as simple as opening a webpage
ArcGIS JavaScript API

Performance

- Performance can vary across
  - Hardware
  - Operating System
  - Browser Vendor

- Poor performance can be avoided by
  - keeping the number of layers limited (10 - 30)
  - using appropriate symbols for every visualization
  - using advantageous viewpoints
  - throttling resource consumption
  - turning off features
```javascript
require(['esri/Map', 'esri/layers/FeatureLayer', 'esri/layers/SceneLayer', 'esri/views/SceneView'], function(Map, FeatureLayer, SceneLayer, SceneView)
{
    var view = new SceneView({
        map: new Map({
            basemap: "satellite",
            ground: "world-elevation",
            layers: [
                new FeatureLayer({ url: "//services.arcgis.com/..." }),
                new SceneLayer({ url: "//services.arcgis.com/..." })
            ]
        }),
        container: "viewDiv"
    });
});
```

- `featureLayer.renderer.symbol`
- `view.environment.lighting.directShadowsEnabled`
- `view.qualityProfile`
- `view.environment.atmosphere.quality`
// index.html
if ('serviceWorker' in navigator) {
    navigator.serviceWorker.register('./service-worker.js', { scope: './' }).then(function() {
        if (navigator.serviceWorker.controller) {
            // Service Worker is active
        }
        else {
            // Service Worker is installed but inactive
        }
    })
    .catch(function(error) {
        // An error occurred during Service Worker creation
    });
} else {
    // The current browser doesn't support service workers
}
// service-worker.js

// Called during app installation
self.addEventListener("install", function(event) {
    event.waitUntil(
        caches.open("cache_v1")
            .then(cache => {
                return cache.add("./index.html"); // Add index to cache
            })
    );
});

// Called during app start
self.addEventListener("activate", function(event) {});

// Called on every network request when installed and active
self.addEventListener("fetch", function(event) {
    event.respondWith(
        caches.open("cache_v1").then(function(cache) {
            return cache.match(event.request).then(function(response) {
                if (response) {
                    return response; // Serving response from cache
                } else {
                    return fetch(event.request.clone()).then(function(response) {
                        return response; // Serving response from network
                    });
                }
            });
        })
    );
});
Progressive Web Apps

Web App Manifest

// index.html
<link rel="manifest" href="./manifest.json">

// manifest.json
{
  "short_name": "Hiking trails",
  "name": "Hiking trails Swiss National Park",
  "icons": [{
    "src": "src/img/android-icon-36x36.png",
    "sizes": "36x36",
    "type": "image/png"
  }],
  "start_url": "index.html",
  "display": "standalone",
  "background_color": "#2d2b07",
  "theme_color": "#b5e2c1"
}
Lloyd Heberlie – lheberli@esri.com
Andy Gup – agup@esri.com
Thomas Other – TOther@esri.com