ArcGIS API for JavaScript
Building Mobile Web Apps

Andy Gup, @agup
Lloyd Heberlie, @lheberlie
Welcome

- Introductions
- Expectations
- Agenda
- Resources
Why are we here?
Resources

- https://github.com/lheberlie/mobile-webapps-js
- https://github.com/lheberlie/mobile-webapps-js/blob/develop/Resources.md
Mobile resources in the API
Lloyd Heberlie
Designing for mobile

Lloyd Heberlie
Touch aware map
HTML 5 input types

type="email"

Email
Number

Telephone

Range

List input

Done

Date

Jul 10, 2014

DateTime

Set date

Cancel Clear Set

Color picker

Email
Geocoder, LocateButton

- esri/dijit/Geocoder
- esri/dijit/LocateButton
Popup Mobile

esri/dijit/PopupMobile
ArcGIS API for JavaScript

Basemap Gallery

Description

This sample shows how you can use buttons to switch between different layers in a map. All the layers in this map happen to be ArcGIS TiledMapServiceLayers from ArcGIS Online. The BasemapGallery widget, introduced at version 2.1 of the ArcGIS JavaScript API simplifies the process of toggling the display of ArcGIS Online basemaps. In this sample a menu item is created for each basemap in the gallery. Since showArcGISBasemaps is set to true, the gallery contains basemaps from ArcGIS Online.

basemapGallery = new esri.dijit.BasemapGallery({
  showArcGISBasemaps: true,
  map: map
});

View live sample
Download as a zip file
Explore in the sandbox
Basemap Toggle

Description

Toggle between basemaps using the BasemapToggle widget. When creating the widget define a basemap that users will be able to switch to from the current basemap. Note: At version 3.7 the widget does not work with maps created from web maps using createMap.
Custom components

Custom basemap switcher
Designing for mobile
Lloyd Heberlie
Productivity and code management

Lloyd Heberlie
Setup a developer machine
Code formatting and management
Automation and continuous integration
Code validation

- JSLint
- JSHint
CSS preprocessors

- Sass
- Less
- Stylus
- Compass
- SASS
development
Productivity and code management

Lloyd Heberlie
Debugging and testing for mobile

Lloyd Heberlie
Emulators / Simulators

The Power of Location
Build applications for web, mobile and desktop with Esri’s cloud services, developer APIs, ready-to-use content and self-hosted solutions.
Physical devices
Remote debugging

- Safari Web Inspector Remote
- Google Chrome remote debugging
- Adobe Edge Inspect
  - Synchronized screen capture, device details
  - Node.js (weinre local)
- Web Inspector Remote (weinre)
Enable Safari remote web inspection
Safari remote debugging
Safari remote debugging
Chrome remote debugging
Chrome remote debugging
Adobe Edge Inspect
Adobe Edge Inspect
Adobe Edge Inspect

device_model = iPhone
device_res = 1136x640
orientation = portrait
os_name = iOS
os_version = 7.0.4
pixel_density = 326 ppi

device_model = LGE Nexus 5
device_res = 1080x1776
orientation = portrait
os_name = Android
os_version = 4.4.2
pixel_density = 480 dpi

device_model = iPad
device_res = 1024x768
orientation = landscape
os_name = iOS
os_version = 7.0.4
pixel_density = 132 ppi
Debugging mobile
Lloyd Heberlie
Many UI frameworks

Bootstrap
Angular.js
jQuery
Ionic
...
...?
Frameworks on github.com/esri

github.com/Esri/bootstrap-map-js

github.com/Esri/angular-esri-map
DEMO
<!-- Step 1. Add CSS for the mapping components -->

<link rel="stylesheet" type="text/css" href="//js.arcgis.com/3.15/esri/css/esri.css">

<link rel="stylesheet" type="text/css" href="http://esri.github.io/bootstrap-map-js/src/css/bootstrapmap.css">

<style type="text/css">
    #mapDiv {
        min-height: 100px;
        max-height: 1000px;
    }
</style>
Set up the div

<!-- Step 2. Add HTML to define the layout of the page and the map -->
<div class="container">
    <div id="mapDiv"></div>
</div>
Load BootstrapMap

<!-- Step 3. Add JS to load the responsive map -->
<script type="text/javascript">
    var package_path = "//esri.github.com/bootstrap-map-js/src/js";
    var dojoConfig = {
        packages: [{
            name: "application",
            location: package_path
        }]
    };
</script>
<script src="//js.arcgis.com/3.15compact"></script>
<script>
    require(["esri/map", "application/bootstrapmap", "dojo/domReady!"],
        function(Map, BootstrapMap) {
            // Get a reference to the ArcGIS Map class
            var map = BootstrapMap.create("mapDiv",{
                basemap:"national-geographic",
                center:[-122.45,37.77],
                zoom:12
            });
        });
</script>
Load Bootstrap + jQuery

<!-- jQuery (for Bootstrap's JavaScript plugins) -->
<script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include all plugins or individual files as needed -->
<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>
Mobile Popups
### Popups

<table>
<thead>
<tr>
<th></th>
<th>Tablets</th>
<th>Smartphones</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>InfoWindow</strong></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>MobilePopup</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
InfoWindow

Text may be hard to read
Not touch friendly
Not centered on rotate
MobilePopup

Readable
Touch friendly
Rotate friendly

http://developers.arcgis.com/javascript/samples/mobile_arcgis/
Custom Popups

Modal popup example

http://andygup.github.io/modal-popup-js/
Mobile Performance
Smartphone vs Your Laptop

App performance not the same
Internet speeds fluctuate/limited
Power limitations
Reducing UI Jank (Jerkiness)

Interruptions in frame production (fps) and latency

Checkerboarding
Long pauses
Scrolling latency
Delayed animation start
Common causes of Jank

Slow internet response

Data processing overload

Excessive garbage collection
HTTP response times

Minimize, concatenate, optimize
Make sure server uses gzip compression!
Use Query MODE_ONDEMAND
Limit the Extent and use layer dependencies
Generalize features
Using Web Workers

Move expensive tasks off main thread!

Parsing large number of features
json.parse() blocks
Serialization/deserialization costs
Timers
Main UI thread

```javascript
var worker = new Worker("libs/MyWorker.js");
worker.postMessage(
    {
        // TODO
    }
);
worker.onmessage = function(result){
    // TODO
};
worker.onerror = function(err){
    console.log("Worker error: " + err.message);
};
```

Background thread

```javascript
// Receive data from main thread
onmessage = function(msg) {
    // TODO
}
// Send data back to main thread
postMessage( data );
```
Background thread

Main UI thread

CPU

SERIALIZE / DE-SERIALIZE
Using Web Workers

Example:

Parsing GeoJSON

https://github.com/andygup/earthquake-heatmap-layer
Example: 7x Performance boost!

1MB of data with web workers

parseFeaturesInWorkerThread: 2.478ms
parseTestTimer: 139.338ms

Without web workers

parseTestTimer: 1028.427ms
Garbage collection

De-reference any variables no longer in use

```javascript
var feature = { feature: { ... }};
feature = null;
```
Managing memory in loops

Re-use objects where possible

Versus `new Object();`

Reset arrays: `myArray.length = 0;

Versus this: `myArray = []`;
Geolocation

Advanced Geolocation Session – 10a – 11a Friday

Outdoor geolocation – turn off WiFi
Not very accurate – 3 – 10 meters
Hybrid – PhoneGap/Cordova

Wednesday 10:30am, Mesquite B

github.com/Esri/quickstart-map-phonegap
Resources

• https://github.com/lheberlie/mobile-webapps-js
• https://github.com/lheberlie/mobile-webapps-js/blob/develop/Resources.md
Andy Gup
agup@esri.com
@agup

Lloyd Heberlie
lheberlie@esri.com