Strategies for Building Mobile Apps Using ArcGIS API for JavaScript

Kelly Hutchins, Lloyd Heberlie
Welcome

• Introductions
• Expectations
• Agenda
Why are we here?
Mobile resources in the API

Kelly Hutchins
Mobile resources in the API

ArcGIS API for JavaScript

- Tutorials
- Concepts
- API Reference

- Mobile-specific application
- Build your first application
- Use an ArcGIS.com webmap
- Directions widget
- Geocoder widget
- Labeling features client-side
- Write a class
- Create a re-usable widget
- Sharing maps with secure layers

ArcGIS API for JavaScript

- Tutorials
- Concepts
- API Reference

- Mobile
  - Getting Started Mobile Tutorial
  - Viewport Settings
  - Feature Detection and Browser Sniffing
  - Device Orientation
  - Geolocation
  - Frameworks
  - References

ArcGIS API for JavaScript

- Tutorials
- Concepts
- API Reference

- Mobile
  - Attribute editing – mobile
  - Compass
  - Mobile ArcGIS.com
  - Shake gesture recognition
  - Soil details – mobile
  - Tilt gesture

Esri UC 2014 | Technical Workshop | Strategies for Building Mobile Apps Using ArcGIS API for JavaScript
ArcGIS for Developers website

Kelly Hutchins
Designing for mobile
Lloyd Heberlie
Touch aware map
HTML 5 input types
Geocoder, HomeButton

esri/dijit/Geocoder

esri/dijit/HomeButton
Popup Mobile

esri/dijit/PopupMobile
Basemap Toggle

**View live sample**

Download as a zip file

Explore in the sandbox

**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

Kelly Hutchins
Setup a developer machine

- web server
- Source Control

Code quality and verification
Code formatting and management

```html
<style>
  html, body {width: 100%; height: 100%;}
  label {font-size: 1.5em; color: #ffffff;}
  input {width: 80%;}
  input:invalid {color: #ff0000;}
  form {white-space: pre-wrap;}
</style>
```
Automation and continuous integration
Code validation

- JSLint
- JSHint
CSS preprocessors

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

Lloyd Heberlie
Debugging and testing for mobile

Kelly Hutchins
Emulators / Simulators

- Opera Mobile Emulator
- iOS Simulator *
- Windows Phone Emulator *
- Android Emulator

Strategies for Building Mobile Apps Using ArcGIS API for JavaScript
Physical devices
Browser based testing
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

![Remote Debugging Screenshot](image.jpg)

Chrome remote debugging allows developers to debug code running on a remote device directly from their local browser. This is particularly useful for mobile app development, as it enables the developer to inspect devices without the need for physical access or deployment on the device. The screenshot shows the Chrome DevTools interface with multiple devices listed, including a Nexus 5 and a Chrome browser.

---

<table>
<thead>
<tr>
<th>Device</th>
<th>OS</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nexus 5</td>
<td>Android</td>
<td>5.1.1</td>
</tr>
<tr>
<td>Chrome</td>
<td>(33.0.1750)</td>
<td>32.0.1700</td>
</tr>
</tbody>
</table>

---

This feature is part of the Chrome browser's developer tools and is accessible via the Inspect option in the Chrome DevTools menu. It provides a comprehensive view of the device's state, including network activity, DOM elements, and performance metrics.
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
Optimizing for mobile

Lloyd Heberlie
HTML manifest

CACHE MANIFEST
# This manifest was generated by grunt-manifest HTML5 Cache Manifest Generator
# Time: Fri Jul 11 2014 11:00:00 GMT-0500 (CDT)
# application-boilerplate - v0.0.1 - 2014-07-11
# Copyright (c) 2014 Licensed

CACHE:
#%20application-boilerplate,%20version:%200.0.1
#%20ArcGIS%20API%20for%20JavaScript%20files
# Contents from the HTTP Archive file.
http://js.arcgis.com/3.10compact/js/dojo/dijit/themes/claro/claro.css
http://js.arcgis.com/3.10compact/js/esri/css/esri.css
http://msial.esri.com/esrijs/samples/mobile_citizenrequest/styles/application-styles.min.css
http://msial.esri.com/esrijs/samples/mobile_citizenrequest/js/application-library.min.js
http://js.arcgis.com/3.10compact/init.js
http://js.arcgis.com/3.10compact/js/esri/nls/jsapi_en-us.js
http://js.arcgis.com/3.10compact/js/esri/tasks/QueryTask.js
http://js.arcgis.com/3.10compact/js/esri/tasks/RelationshipQuery.js
http://js.arcgis.com/3.10compact/js/esri/tasks/StatisticDefinition.js
http://js.arcgis.com/3.10compact/js/esri/dojo/dijit/themes/claro/claro.css
http://msial.esri.com/esrijs/samples/mobile_citizenrequest/styles/images/ajax-loader.gif
HTML manifest on device
File compression and caching headers
Dealing with high density resolution
iOS network simulation
Android network speed emulation

<table>
<thead>
<tr>
<th>command</th>
<th>Option</th>
<th>&lt;speed&gt; kilobits/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>emulator</td>
<td>-netspeed</td>
<td>&lt;named value&gt;</td>
</tr>
<tr>
<td>emulator</td>
<td>-netspeed</td>
<td>&lt;num&gt; (both)</td>
</tr>
<tr>
<td>emulator</td>
<td>-netspeed</td>
<td>&lt;up&gt;:&lt;down&gt;</td>
</tr>
<tr>
<td>emulator</td>
<td>-netspeed</td>
<td>gsm</td>
</tr>
<tr>
<td>emulator</td>
<td>-netspeed</td>
<td>14.4 80</td>
</tr>
<tr>
<td>emulator</td>
<td>-netspeed</td>
<td>40.0</td>
</tr>
</tbody>
</table>
Other network simulation options

• Fiddler (Windows)
• Charles Network Debugging proxy
• Network Link Conditioner (Mac OS)
• Dummynet
Additional resources

- Esri resources
- Esri resource proxy
Thank you...

• Please fill out the session survey:

First Offering ID (Tuesday): 1255
Second Offering ID (Thursday): 1366

Online – www.esri.com/ucsessionsurveys
Paper – pick up and put in drop box