



# Designing Responsive Web Mobile Mapping Applications

Andy Gup, @agup

Esri Developer Summit  
Middle East & Africa

19-21 November 2013  
Park Hyatt Dubai

# Speed Bump

By Dave Coverly



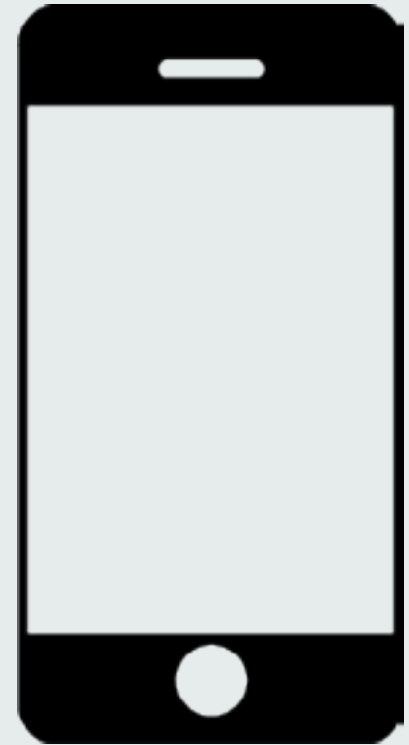
Reponsive Libraries

And

User Interface Frameworks

# Web Mobile Design

Key Considerations



# Mobile Devices

Are different...

- § Physical device
- § Screen size
- § Button size

# User Interactions

Are different...

- § Touch
- § Orientation
- § Keyboard
- § Voice

# User Expectations

Are different...

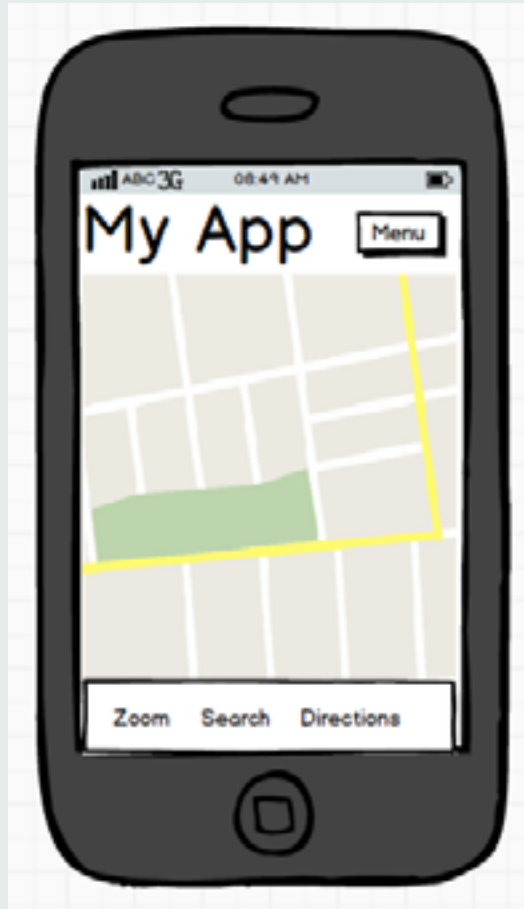
- § App to work like an app
- § Websites to work like an app

# Design Patterns

**Need to be different!**



# Think mobile first!



320px

# Responsive Design



# Responsive Design

§ Single web app that works *well* across a variety of devices/screen sizes

§ Re-use content and software

§ Considers

§ Device limitations

§ User's behavior



# mediaqueri.es



# Responsive Design Components

1. Fluid Grid System
2. Media Queries
3. HTML5, CSS & JavaScript

# Fluid Grid System

- § Layout adapts to different screen sizes
- § Based on percentages
- § 12 column / 960px

# Fluid Grid System



# Media Queries

- § Detect device screen size and orientation
- § Apply CSS at specific break points
- § Typical: 480px, 768px, 1024px, 1280px

```
@media only screen and (max-device-width:480px) {  
    /* Custom css styles */  
    body {  
        font-size: 0.5em;  
    }  
    #titleArea{  
        display: none;  
    }  
}
```



# HTML5 & CSS3

## HTML5

- § Meta tags
- § Input types (text, dates...)

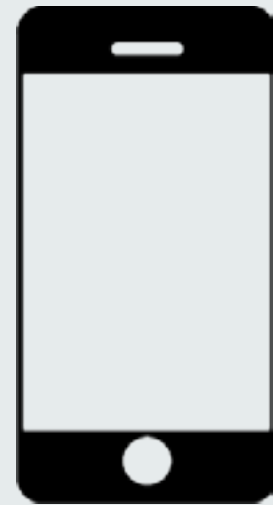
## CSS3

- § Selectors, transitions

## JavaScript

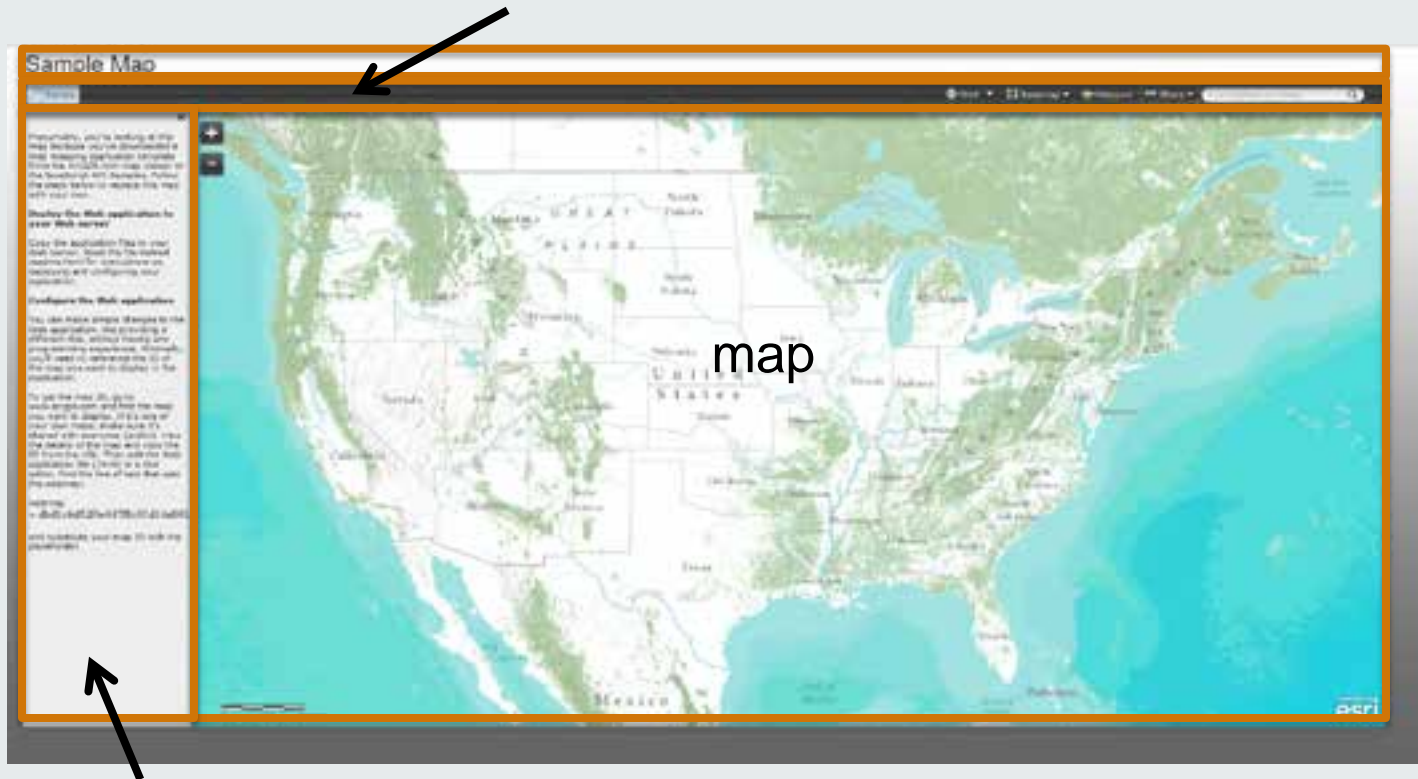
- § Touch events
- § Geolocation, localStorage, websockets, appcache....

# How does responsive work?



# Typical “full-view” Mapping App

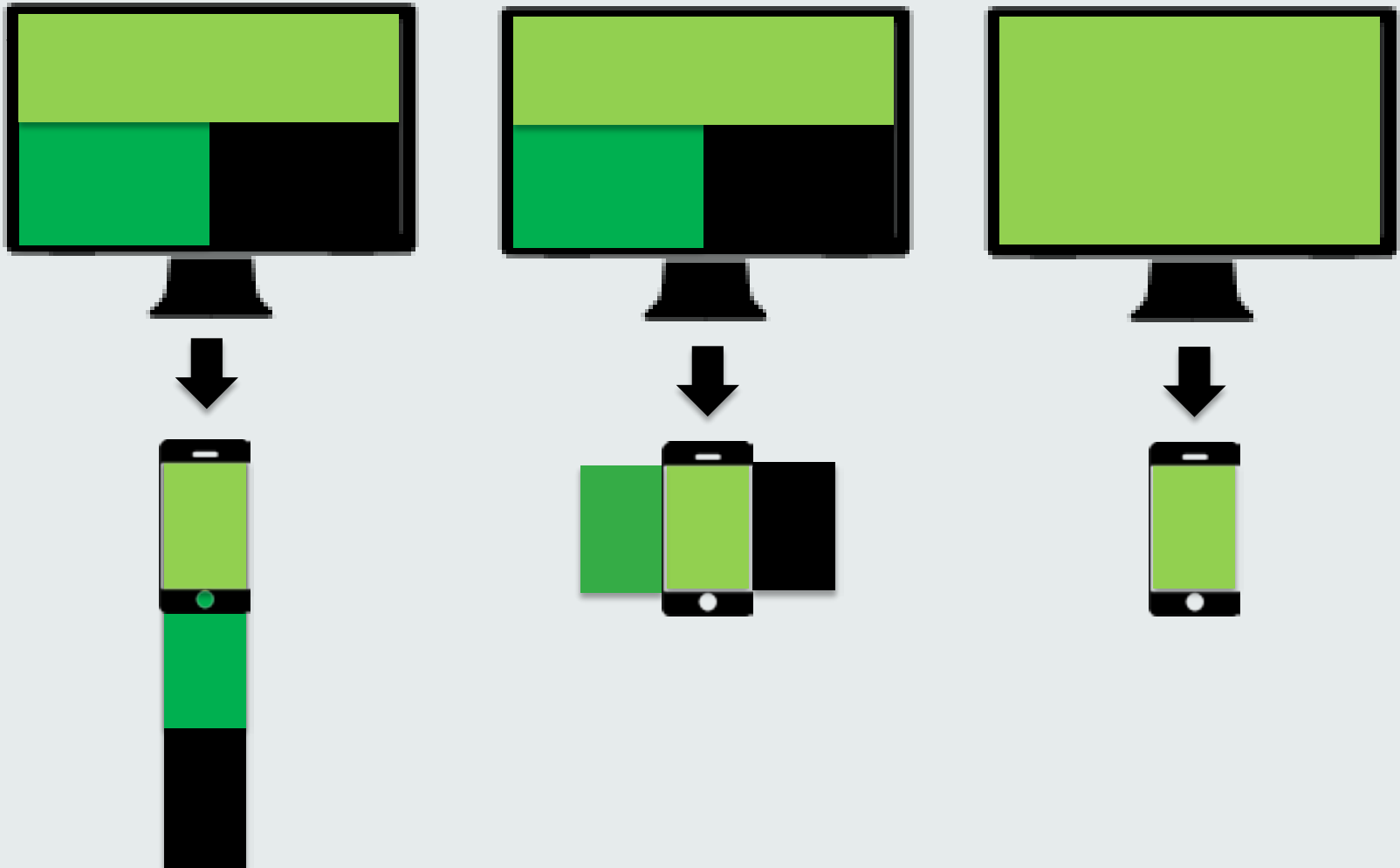
## 3 Row – 2 Column



# Default Behavior



# Responsive Grid Layouts



# Large: 3 Rows - 3 Columns



**$\geq 1280\text{px}$**

# Medium: 2 Columns



**1024 - 1280px**

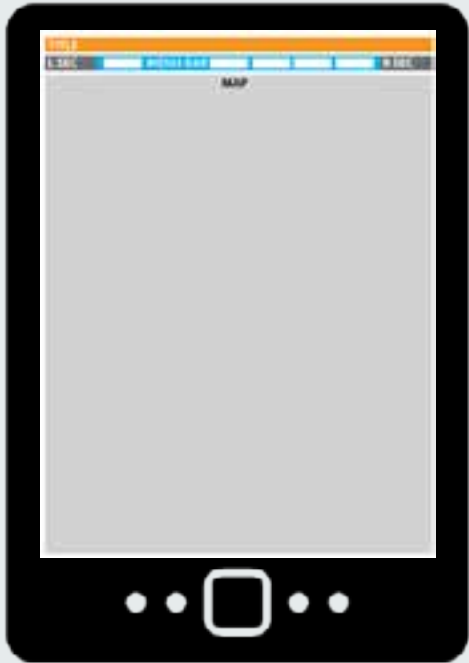
# Medium: 2 Columns



**1024 - 1280px**

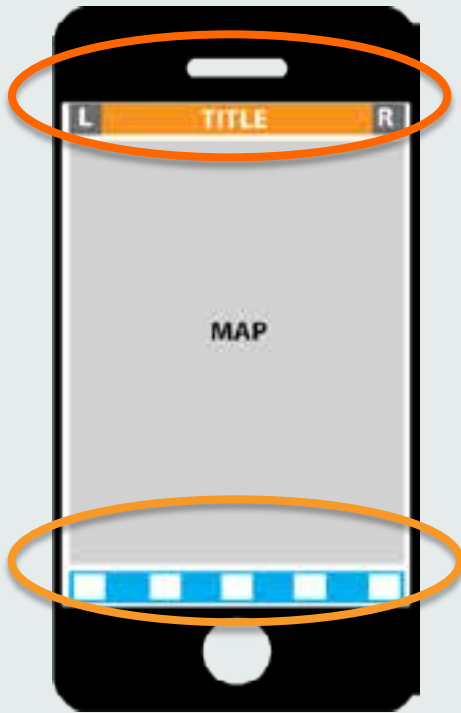


# Small: Single Column



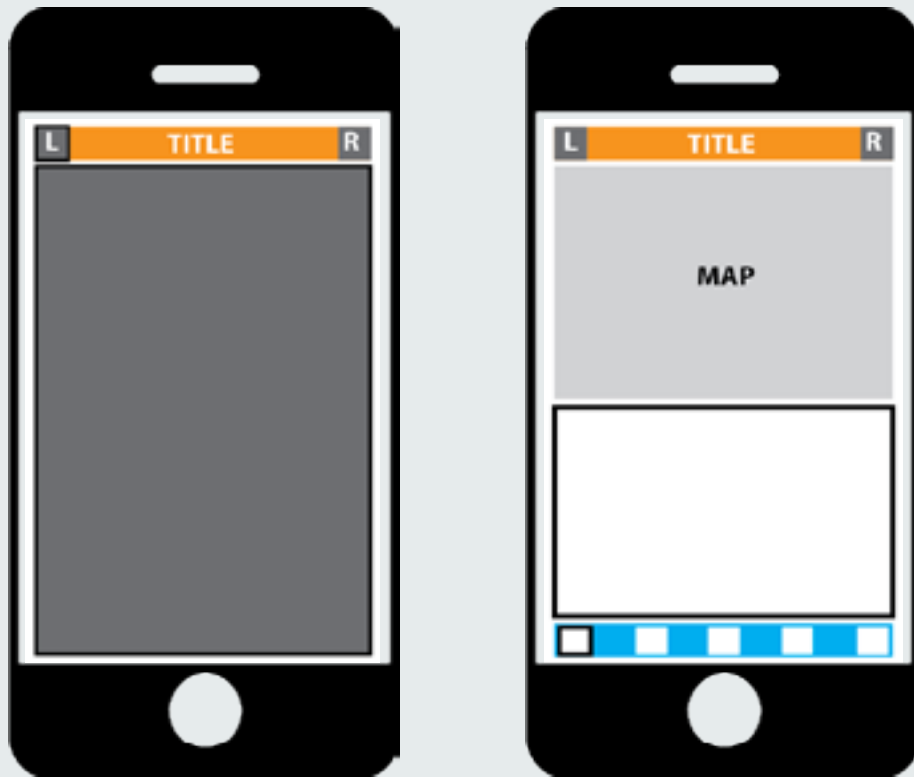
**768 - 1024px**

# Extra Small: 1 Column, Minimized



**=< 768px**

# Extra Small: Windowing



**$\leq 768\text{px}$**

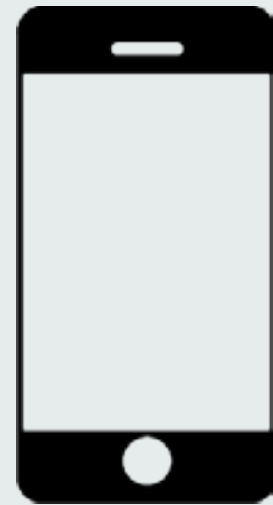
# Responsive HTML5 Application



[github.com/esri/responsive-map-js](https://github.com/esri/responsive-map-js)

[ami.responsivedesign.is](http://ami.responsivedesign.is)

# Responsive Frameworks



# Responsive Web Frameworks

§ Bootstrap 3

§ Foundation 3

§ Skeleton

§ YAML 4

**[www.getbootstrap.com](http://www.getbootstrap.com)**



# Bootstrap-map-js

- § Bootstrap ver 3 framework
- § Responsive map
- § Resize and re-center
- § Pop-ups, widgets
- § Touch
- § CSS Styles

# Step 1: Get bootstrap-map-js

[github.com/Esri/bootstrap-map-js](https://github.com/Esri/bootstrap-map-js)

# Step 2: Create a page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap 101 Template</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <!-- Bootstrap -->
    <link href="../../assets/css/bootstrap.min.css" rel="stylesheet"
media="screen">
  </head>
  <body>

    <h1>Hello, world!</h1>

    <!-- jQuery (for Bootstrap's JavaScript plugins) -->
    <script src="../../assets/js/jquery.js"></script>
    <!-- Include all plugins or individual files as needed -->
    <script src="../../assets/js/bootstrap.min.js"></script>
  </body>
</html>
```

# Step 2: Create a page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap 101 Template</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <!-- Bootstrap -->
    <link href="../../assets/css/bootstrap.min.css" rel="stylesheet"
media="screen">
  </head>
  <body>

    <h1>Hello, world!</h1>

    <!-- jQuery (for Bootstrap's JavaScript plugins) -->
    <script src="../../assets/js/jquery.js"></script>
    <!-- Include all plugins or individual files as needed -->
    <script src="../../assets/js/bootstrap.min.js"></script>
  </body>
</html>
```

# Step 3: Style map div

```
<!-- ArcGIS css -->
<link rel="stylesheet" type="text/css"
href="http://js.arcgis.com/3.7/js/esri/css/esri.css">
```

```
<!-- Bootstrap-map-js css -->
```

```
<link rel="stylesheet" type="text/css" href="../../src/css/bootstrapmap.css">
```

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

# Step 3: Style map div

```
<!-- ArcGIS css -->
<link rel="stylesheet" type="text/css"
href="http://js.arcgis.com/3.7/js/esri/css/esri.css">

<!-- Bootstrap-map-js css -->
<link rel="stylesheet" type="text/css" href="../../src/css/bootstrapmap.css">

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

# Step 4: Add a responsive map

```
<div class="container">
  <div id="mapDiv"></div>
</div>
...
<script src="http://js.arcgis.com/3.7compact"></script>
<script>
  <!-- Load Bootstrap Map - responsive map -->
  require(["esri/map",
    ".../src/js/bootstrapmap.js",
    "dojo/domReady!"],
    function(Map, BootstrapMap) {
      <!-- Get a reference to the ArcGIS Map -->
      var map = BootstrapMap.create("mapDiv",{
        basemap:"national-geographic",
        center:[-122.45,37.77],
        zoom:12
      });
    });
</script>
```

# Step 4: Add a responsive map

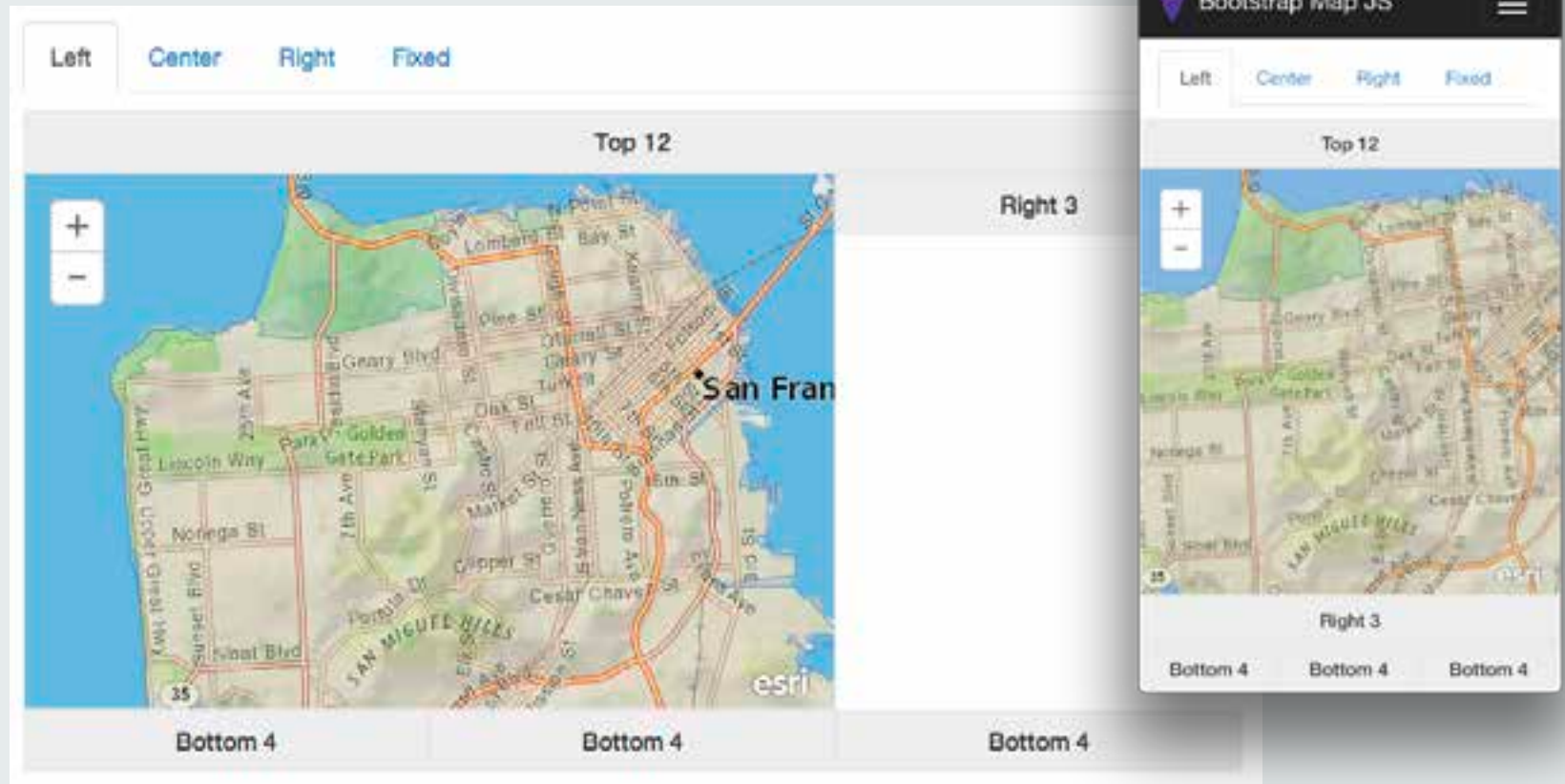
```
<div class="container">
  <div id="mapDiv"></div>
</div>
...
<script src="http://js.arcgis.com/3.7compact"></script>
<script>
  <!-- Load Bootstrap Map - responsive map -->
  require(["esri/map",
    ".../src/js/bootstrapmap.js",
    "dojo/domReady!"],
    function(Map, BootstrapMap) {
      <!-- Get a reference to the ArcGIS Map -->
      var map = BootstrapMap.create("mapDiv",{
        basemap:"national-geographic",
        center:[-122.45,37.77],
        zoom:12
      });
    });
</script>
```



**[github.com/esri/bootstrap-map-js](https://github.com/esri/bootstrap-map-js)**

**A quick look inside...**

# Bootstrap Fluid Grid CSS



# Bootstrap Fluid Grid CSS

```
<div class="col-xs-12 col-sm-8">
```

**Define Column**



```
graph BT; A[Define Column] --> B[col-xs-12]; C[Device Size] --> B; D[Number of Columns] --> B;
```

The diagram illustrates the components of Bootstrap CSS classes. Three blue arrows point from the labels 'Define Column', 'Device Size', and 'Number of Columns' to the corresponding parts of the class string 'col-xs-12 col-sm-8'. 'Define Column' points to 'col', 'Device Size' points to 'xs', and 'Number of Columns' points to '12'.

**Device Size**

**Number of  
Columns**

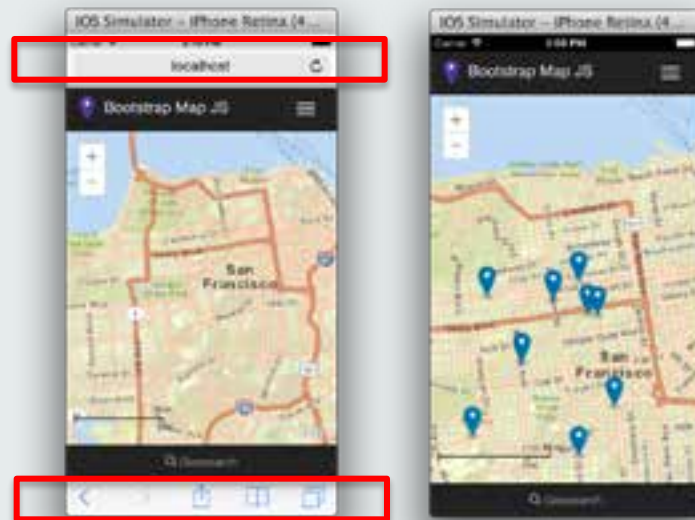


# Dynamic visibility

```
<div class="page-header" hidden-xs>
  <div class="row">
    <div class="col-xs-9"><h2>Windows</h2>
      <p class="lead">Show modal & responsive pop-ups.</p>
    </div>
    <div class="col-xs-3">
      <button . . .>Show Window</button>
    </div>
  </div>
</div>
```

# Viewport behavior

```
// Optimize for mobile - prevent scaling on pinch
<meta name="viewport" content="width=device-width,
  initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
// Safari iOS - apps only
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style"
content="black">
// Chrome for Android - NEW!
<meta name="mobile-web-app-capable" content="yes">
```



# Map: Scroll without pan

```
// Set touch behavior
createMap: function() {
    var options = {smartNavigation:false};
    var map = new Map(mapDiv,options);
    map.on('load', this.setTouchBehavior);
    return map;
},

setTouchBehavior: function() {
    this.disableScrollWheelZoom();
},
```



# Map: Auto-resizing

```
// Responsive resize
var resizeWin function() {
    var w = window.innerHeight;
    if (w != this._w) {
        this._w = w;
        var b = document.body.clientHeight;
        var mh = this._mapDiv.clientHeight;
        var ms = this._calcSpace(this._map);
        var mh1 = mh - ms;
        var room = w - b;
        var mh2 = room + mh1;
        style.set(this._mapDivId, {
            "height": mh2+"px"});
    }
}

on(window, "resize", lang.hitch(this, resizeWin));
```

# Map: Auto-recentering

```
// Auto-center map
var recenter = function(extent, width, height) {
    this._map.__resizeCenter =
        this._map.extent.getCenter();
    var timer = function() {
        this._map.centerAt(this._map.__resizeCenter);
    }

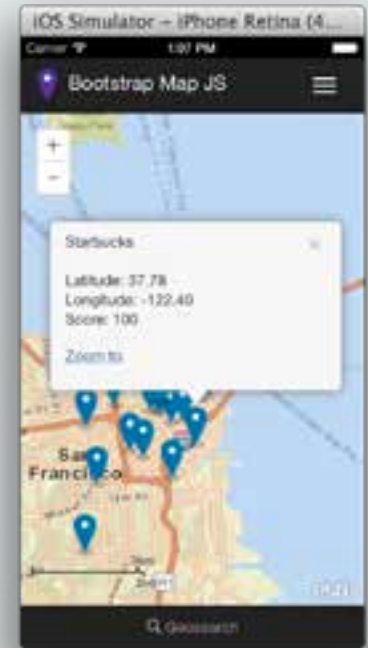
    setTimeout(lang.hitch(this, timer), this._delay);
}

on(window, "resize", lang.hitch(this, recenter));
```

# Popups: Smart touch

```
// Smart-center popup
var updatePopup = function(obj) {
    var infoW = obj._map.infoWindow;
    updateTitle(infoW);
    obj._repositionInfoWin(infoW.features[0]);
}

on(this._map.graphics, "click", lang.hitch(this,
    function(){
        updatePopup(this);
    }));
```



Tip: Listen for onTouchStart and onClick

# Using Media Queries

```
/* Extra small devices (phones, up to 480px) */  
@media (max-width: 480px) {  
    .map { height: 100px; }  
}
```

```
/* Small devices (tablets, 768px and up) */  
@media (min-width: 768px) {  
    .map { height: 200px; }  
}
```

```
/* Medium devices (desktops, 992px and up) */  
@media (min-width: 992px) {  
    .map { height: 300px; }  
}
```

```
/* Large devices (large desktops, 1200px and up) */  
@media (min-width: 1200px) {  
    .map { height: 400px; }  
}
```

# Other Tips and Tricks

§ Watch for `.container` CSS

§ Remove all `:hover` states

§ Listen for *onTouchStart* and *click* to avoid 300ms delay

**[github.com/esri/bootstrap-map-js](https://github.com/esri/bootstrap-map-js)**

# Final Notes

- § Write code once
- § Write is “responsively”
- § Make your maps responsive
- § Know your device
- § Know your user

Don't re-invent the wheel!



# JavaScript User Interface (UX) Frameworks





dōjō

embedjs

# What is a JavaScript UX framework?

Mimic native look-and-feel

Mimic native development patterns

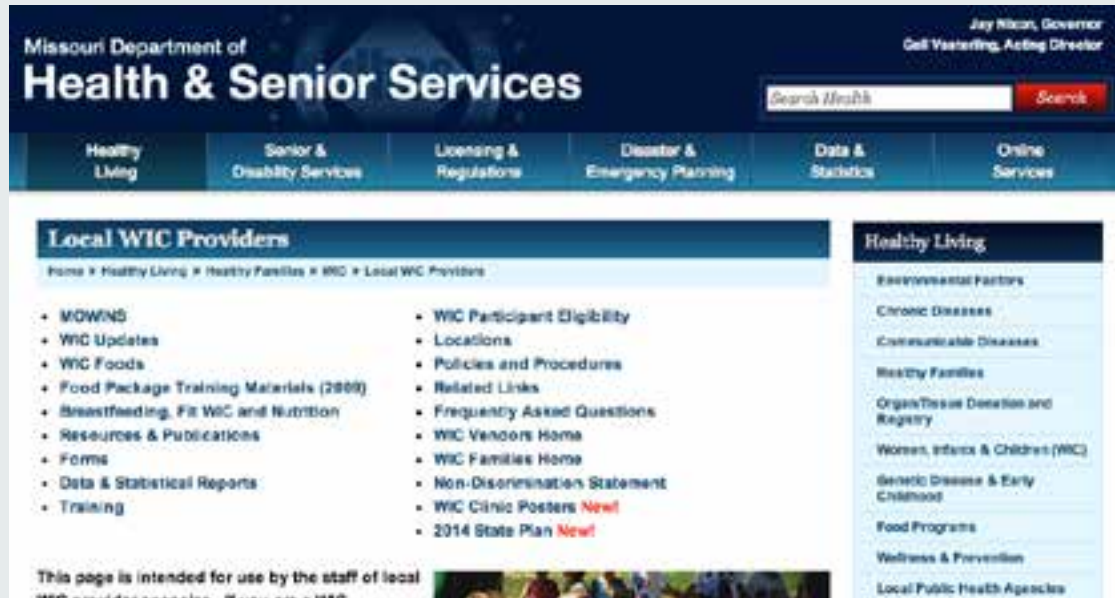
Cross-platform

Can be used with PhoneGap and Titanium

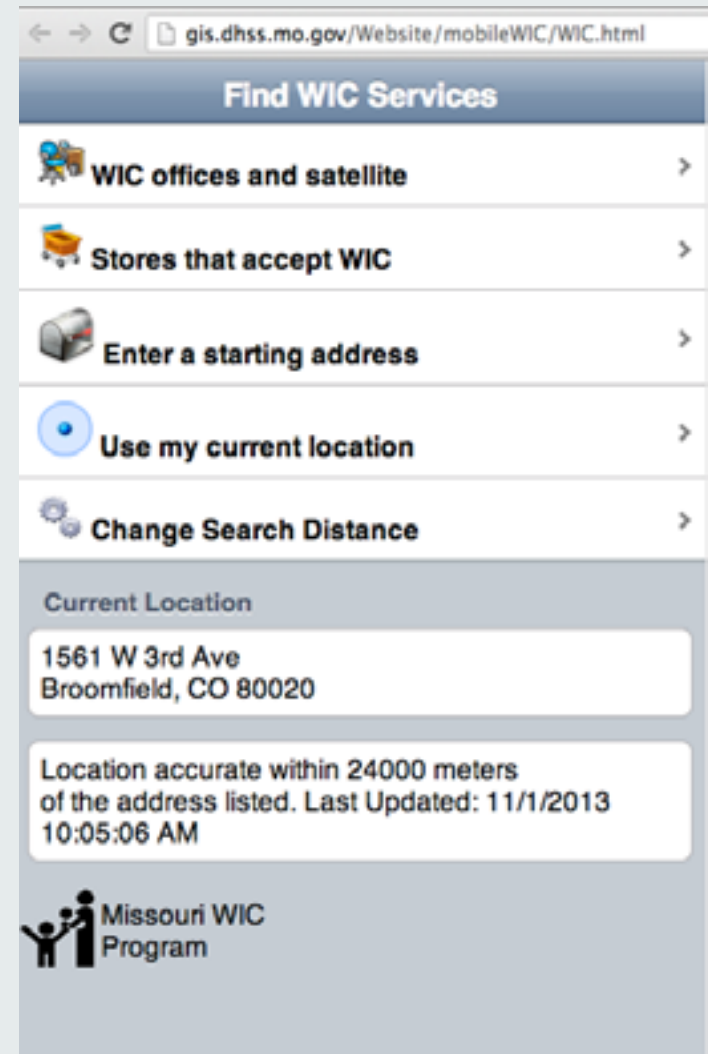
# Why use JavaScript UX framework?



# Mimic Native



Desktop

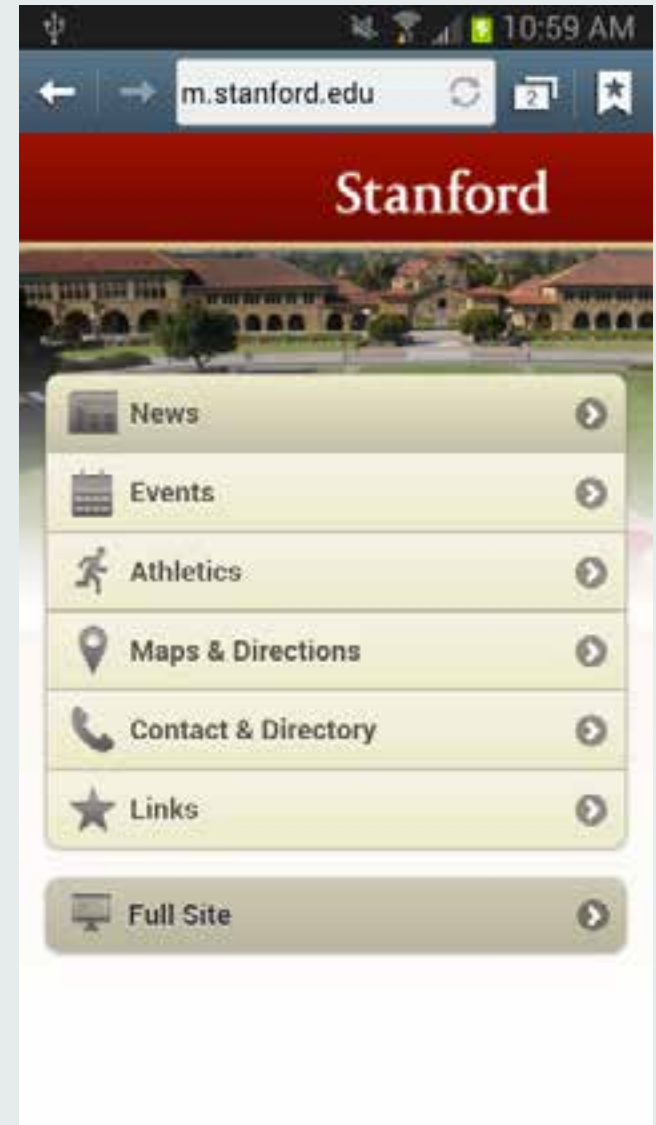


Phone

# Mimic Native



Desktop

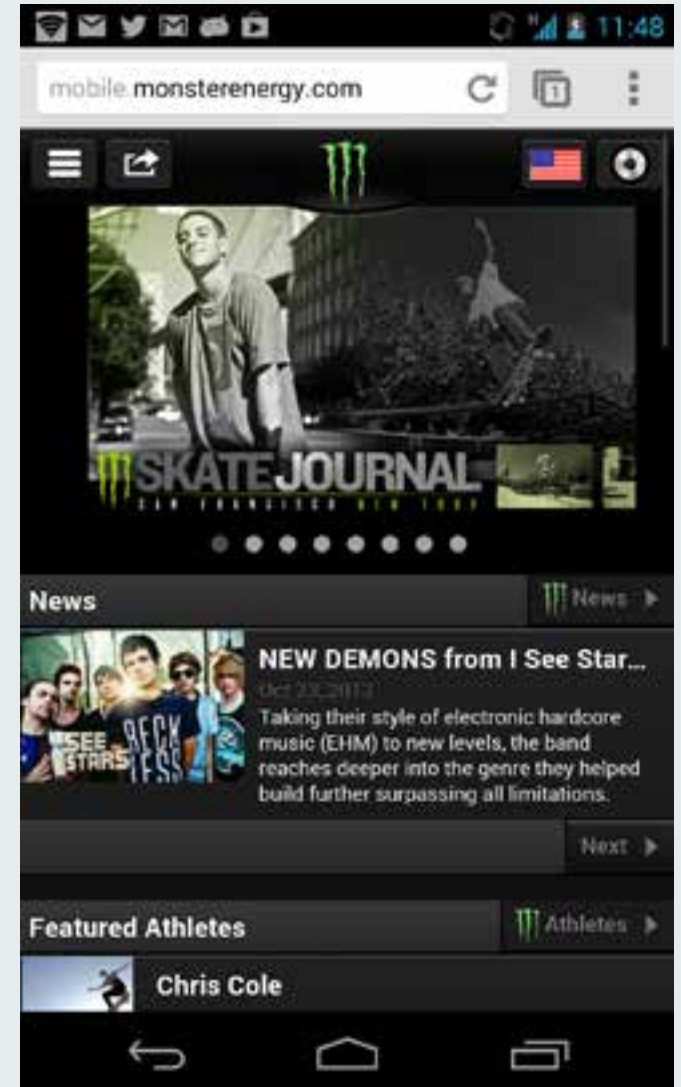


Phone

# Mimic Native



Desktop



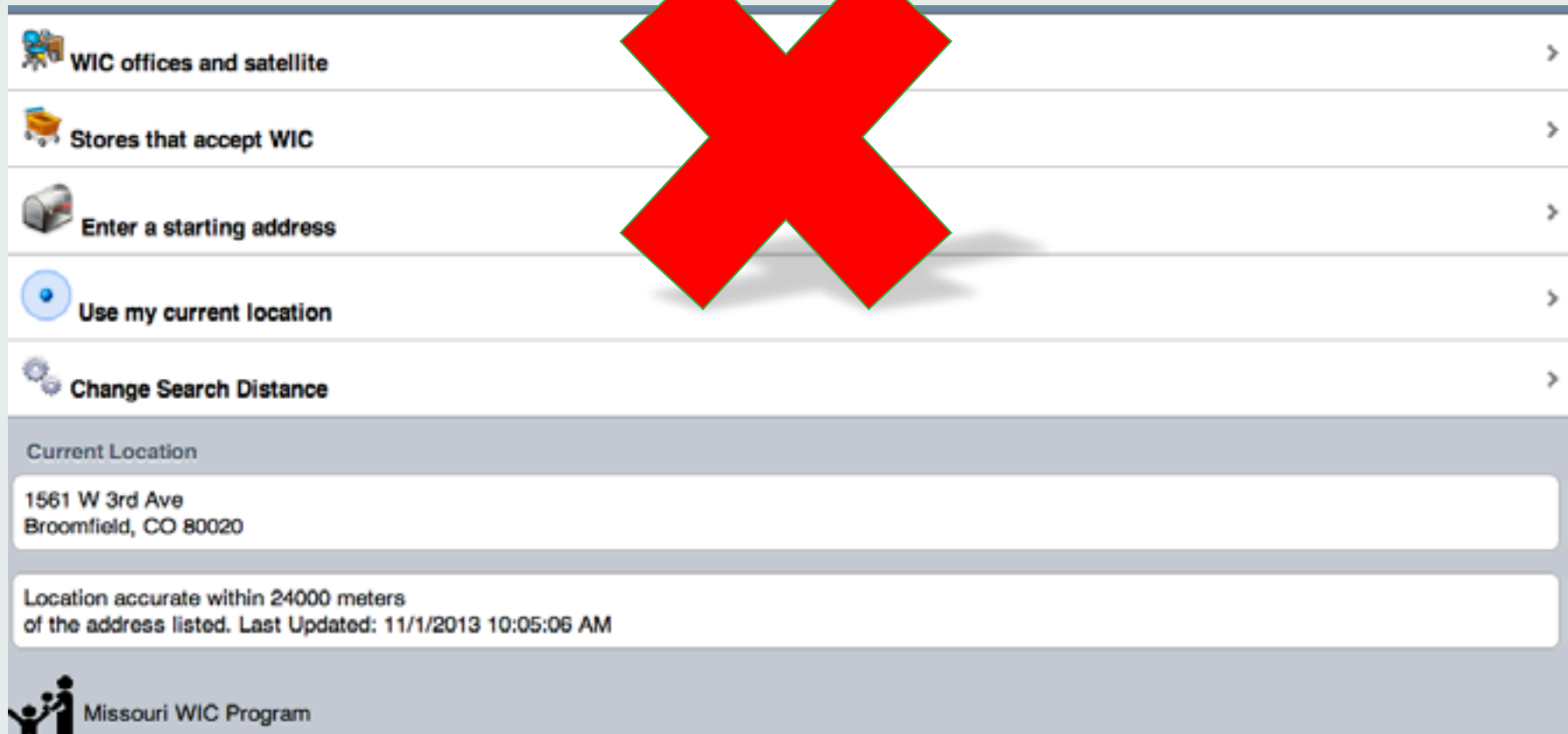
Phone

# Mobile device detection

Server-side detection (Fast)

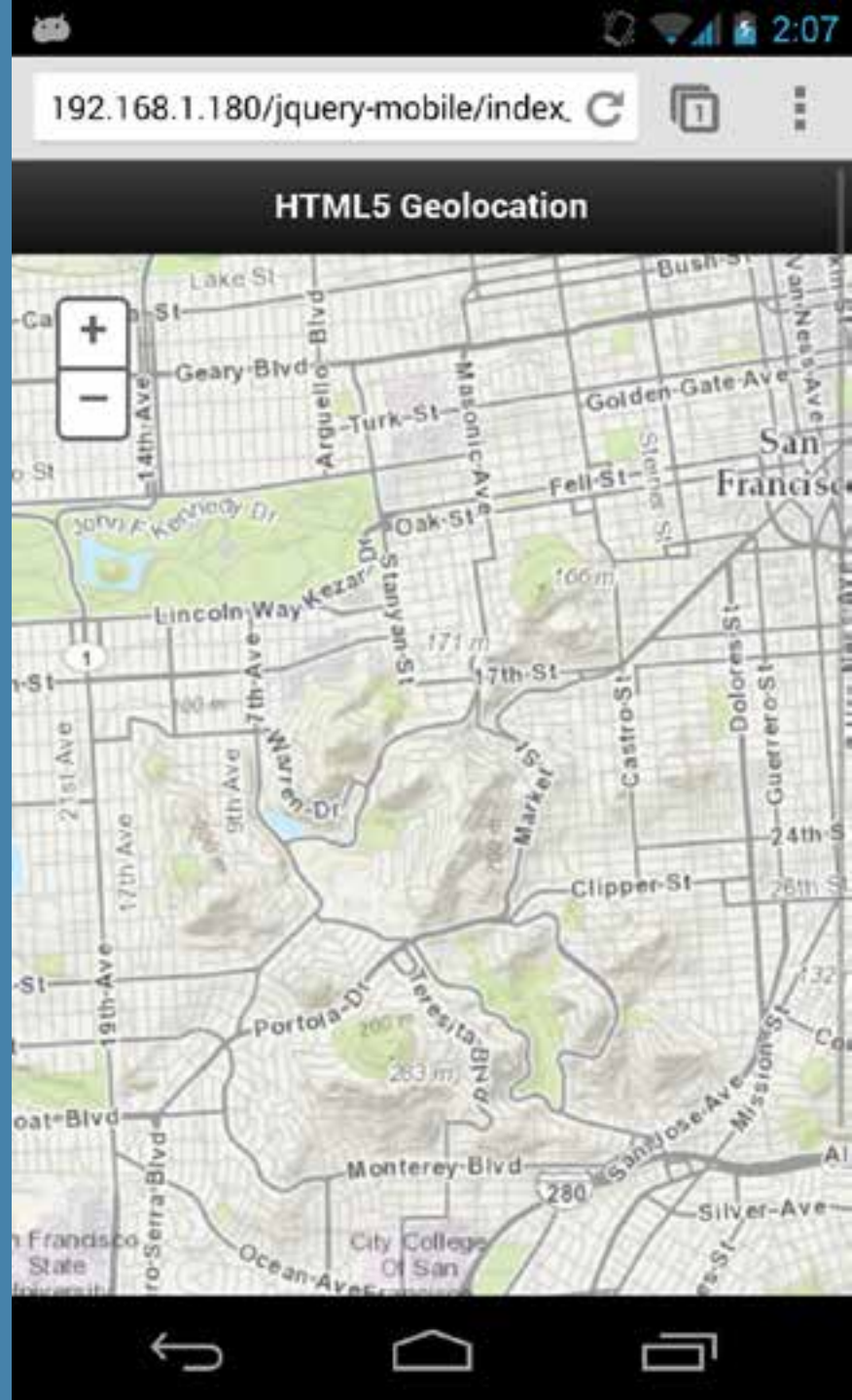
Client-side device detection  
(Slower)

# Desktop view of a mobile web page





# Single Page Map



# Single Page View

Page structure

Apply CSS, roles and themes

# Single Page Map - HTML

```
<div data-role="page" id="home">
```

```
//Header
```

```
<div data-theme="a" data-role="header"  
  data-position="fixed">
```

```
<h3>HTML5 Geolocation</h3>
```

```
</div>
```

```
//Map
```

```
<div data-role="content">
```

```
<div id="mapDiv"></div>
```

```
</div>
```

```
</div>
```

# Single Page Map - HTML

```
<div data-role="page" id="home">
```

```
//Header
```

```
<div data-theme="a" data-role="header"  
  data-position="fixed">
```

```
  <h3>HTML5 Geolocation</h3>
```

```
</div>
```

```
//Map
```

```
<div data-role="content">
```

```
  <div id="mapDiv"></div>
```

```
</div>
```

```
</div>
```

# Single Page Map - HTML

```
<div data-role="page" id="home">  
  //Header  
  <div data-theme="a" data-role="header"  
    data-position="fixed">  
    <h3>HTML5 Geolocation</h3>  
  </div>  
  
  //Map  
  <div data-role="content">  
    <div id="mapDiv"></div>  
  </div>  
</div>
```

# Single Page Map - CSS

```
html,body, div[data-role ="page"] {  
    height: 100%;  
    width: 100%;  
    margin: 0px;  
    padding: 0px;  
}  
.ui-header{  
    margin: 0px !important;  
    padding: 0px !important;  
    float: left;  
}  
.ui-content{  
    height: 100%;  
    width: 100%;  
    margin: 0px;  
    padding: 0px;  
}  
#mapDiv {  
    position: absolute;  
    background-color: #EEEEDD;  
    height: 100%;  
    width: 100%;  
    padding: 0px;  
    z-index: 0;  
    left: 0px;  
}
```

# Single page app – **App** Life Cycle

1. Load CSS
2. Load jQuery
3. Load jQuery Mobile
4. Load HTML
5. Load ArcGIS API for JavaScript

# Single page app – **Map** Life Cycle

1. jQuery “**pageinit**” event
2. ArcGIS JS API - **dom/domReady!**
3. Instantiate the Map
4. On map “**load**” event
5. Add additional layers or start GPS



# Single Page Map

## Code layout

```
html>

http-equiv="Content-Type" content="text/html; charset=
name="viewport" content="width=device-width, initial
e>Basic jQuery Mobile Map</title>

data-role="page" id="home">

div data-theme="a" data-role="header" data-position="
  <h3>HTML5 Geolocation</h3>
/div>

div data-role="content">
  <div id="mapDiv"></div>
/div>

>

Load the library and CSS references for ArcGIS API fo
rel="stylesheet" type="text/css" href="http://js.arcg
pt type="text/javascript" src="http://js.arcgis.com/3
rel="stylesheet" href="http://code.jquery.com/mobile
pt type="text/javascript" src="//ajax.googleapis.com/
pt type="text/javascript" src="http://code.jquery.com

style type="text/css">
html,body,div[data-role="page"] {
  height: 100%;
  width: 100%;
  margin: 0px;
  padding: 0px;
}

ui-header{
  margin: 0px !important;
  padding: 0px !important;
  float: left;
}

ui-content{
  height: 100%;
  width: 100%;
  margin: 0px;
  padding: 0px;
}
```

# Use Case - Recenter the map



# Use jquery-mobile-map-js library!

```
var helper =  
    new jQueryHelper( map );
```

<http://github.com/esri/jquery-mobile-map-js>

# Auto-recenter after orientation change

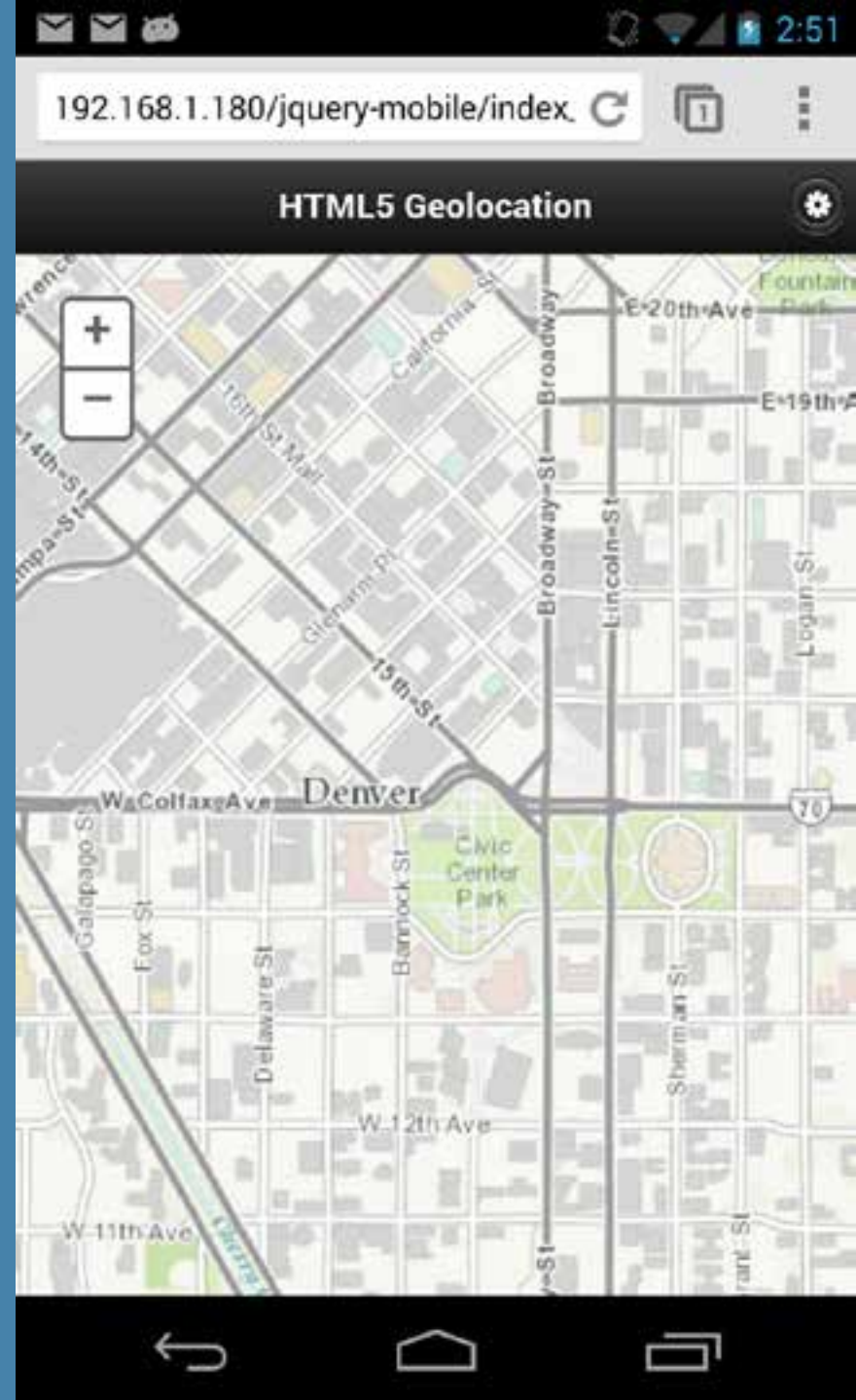
```
//Listen for map load event  
map.on("load",init);
```

```
function init(){  
    try{  
        helper = new jQueryHelper(map);  
        helper.setCenterPt(x,y,4326);  
    }  
    catch(err) {  
        console.log("jQueryHelper " + err.message);  
    }  
}
```

```
//Some browsers don't show full height after onLoad  
map.reposition();  
map.resize();
```

```
}
```

# Multiple Page App

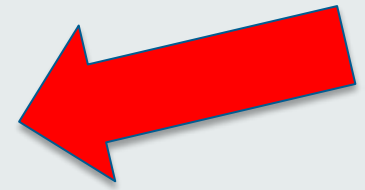
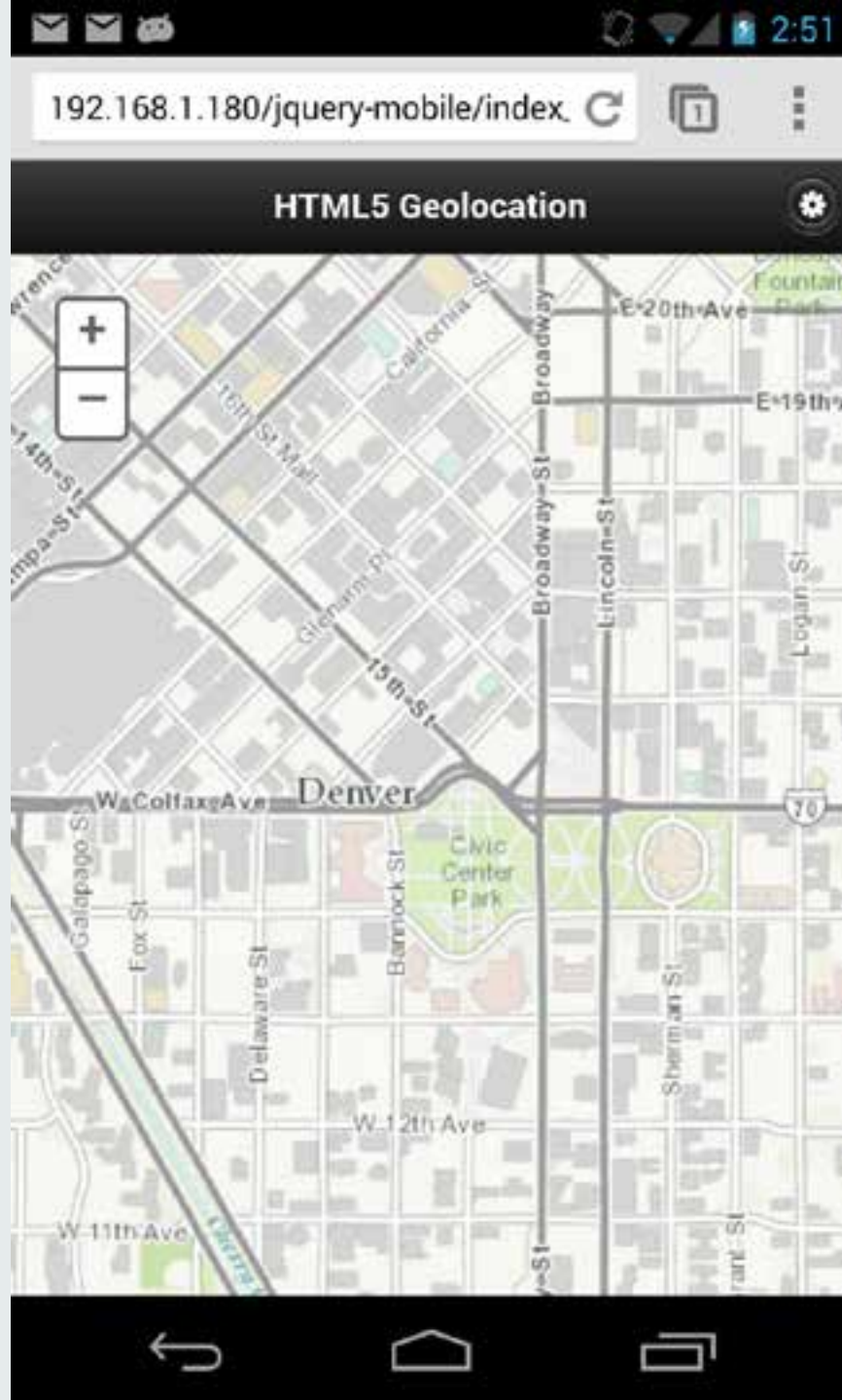


# Multi Page View

Set up additional pages

Set up page navigation

Additional CSS







# Multiple Page app – Page 2 HTML

```
<div data-role="page" id="settings" data-add-back-btn="true">
  //Our Header
  <div data-theme="a" data-role="header"
    data-position="fixed">
    <h1>Settings</h1>
  </div>

  //Page Content
  <div data-role="content">
    <div class="settings" style="margin-top: 15px;">
      <div>Geolocation:</div>
      <select name="slider" id="slider-geo-on-off"
        data-role="slider" data-theme="a">
        <option value="off">Off</option>
        <option value="on" selected>On</option>
      </select>
    </div>
  </div>
</div>
```

# Multiple Page app – Page 2 HTML

```
<div data-role="page" id="settings" data-add-back-btn="true">
  //Our Header
  <div data-theme="a" data-role="header"
    data-position="fixed">
    <h1>Settings</h1>
  </div>

  //Page Content
  <div data-role="content">
    <div class="settings" style="margin-top: 15px;">
      <div>Geolocation:</div>
      <select name="slider" id="slider-geo-on-off"
        data-role="slider" data-theme="a">
        <option value="off">Off</option>
        <option value="on" selected>On</option>
      </select>
    </div>
  </div>
</div>
```





# Multiple Page app – Page 2 CSS

```
<style type="text/css">
  /* Class for aligning buttons */
  .settings{
    margin-left: auto;
    margin-right: auto;
    text-align: center;
    width: 100%;
  }
</style>
```

# Multiple page app – Map Life Cycle

Listen for “helper-map-loaded” event

```
//Custom event from jQueryHelper
$(document).on("helper-map-
loaded",function(evt){
    map = helper.map;

    if(geoEnabled == true) {
        startGeolocation();
    }
});
```

# Page Transitions & Orientation Change

Page and orientation life cycle

Map destruction & resizing

# Page Transitions

**Page 1**



**Page 2**



**~250 ms**



**HTML/CSS**

**jQuery**

**ArcGIS API  
for JavaScript**

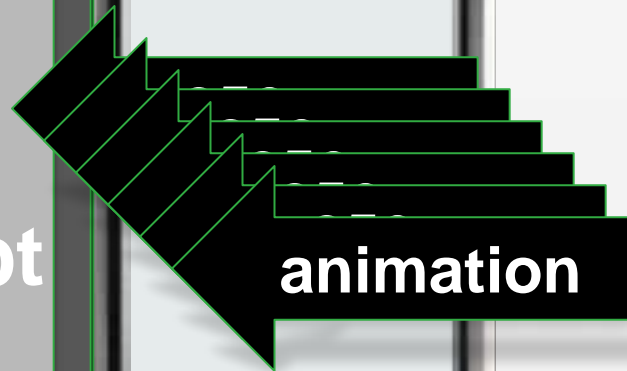
**animation**

**< Back**

**Settings**

**Geolocation:**

**On**



# User-driven Page **Transition** Events

pagebeforeshow

pageshow

pagechangefailed

# Page Transitions & Orientation Change

**Page 1**

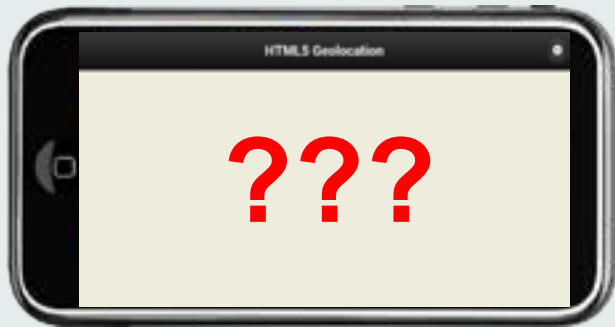
**Page 2**

**Page 2**

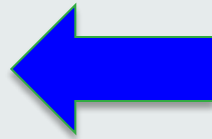


# Page Transitions & Orientation Change

**Page 1**



**Page 2**





✕ Elements Resources Network Sources Timeline Profiles Audits Console

✖ XMLHttpRequest cannot load <http://services.arcgisonline.com/ArcGIS/rest/info?f=json>. Origin <http://services.arcgisonline.com/ArcGIS/rest/info?f=json> is not allowed by Access-Control-Allow-Origin.  
▶ Resource interpreted as Script but transferred with MIME type text/plain: "http://services.arcgisonline.com/ArcGIS/rest/info?f=json&callback=dojo.io.script.io&dojoToScript1.isonopCallback".

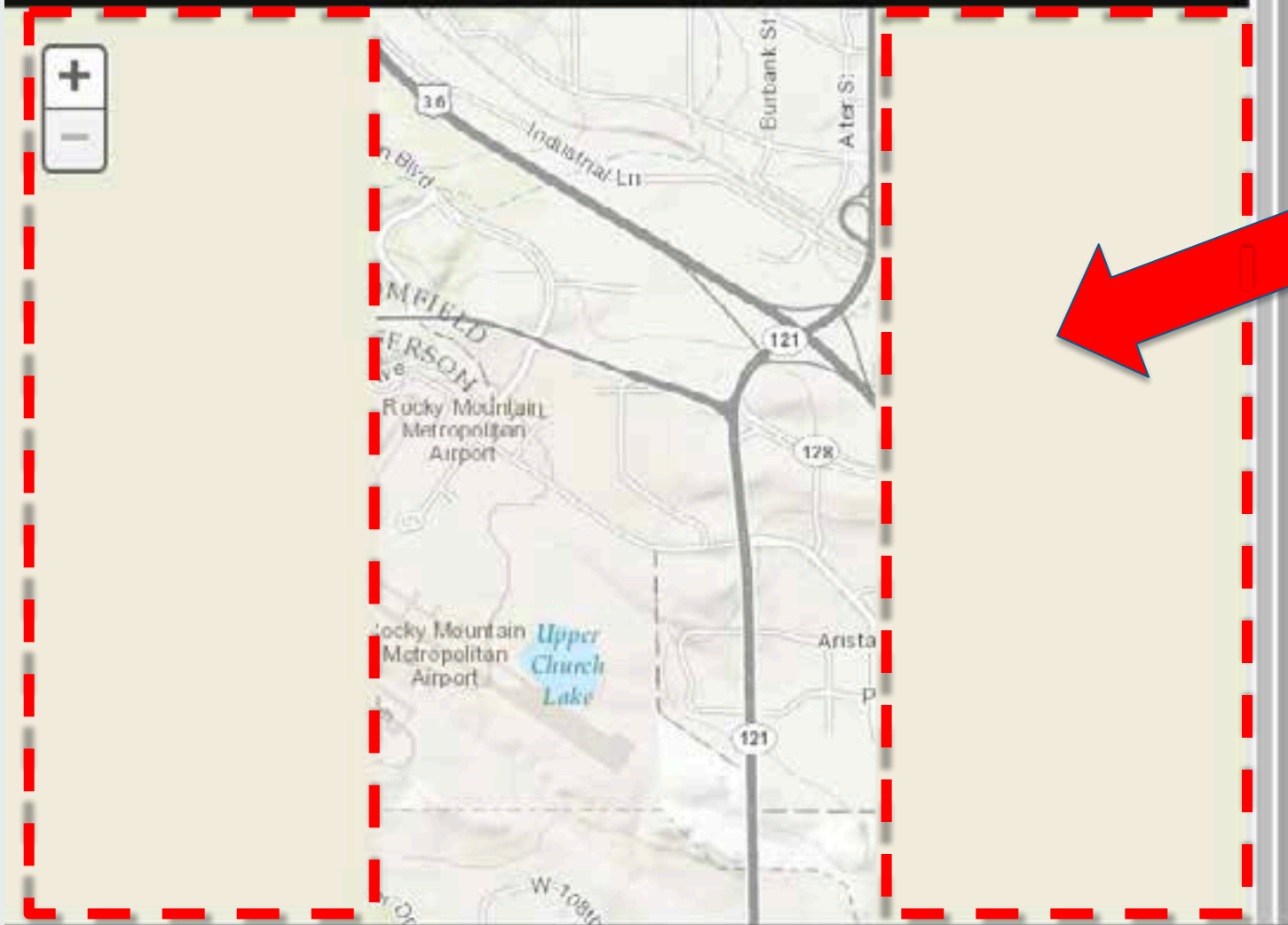
🕒 home showing

✖ ▶ Error: Invalid value for <image> attribute x="NaN"

✖ ▶ Error: Invalid value for <image> attribute y="NaN"

>

# HTML5 Geolocation



Elements Resources Network Sources Timeline Profiles Audits Console

XMLHttpRequest cannot load <http://services.arcgisonline.com/ArcGIS/rest/info?f=json>. Origin <http://services.arcgisonline.com/ArcGIS/rest/info?f=json> is not allowed by CORS policy. Resource interpreted as Script but transferred with MIME type text/plain: "http://services.arcgisonline.com/ArcGIS/rest/info?f=json&callback=dojo.io.script.io.\_loadScript1\_isonopCallback".

home showing

Error: Invalid value for <image> attribute x="NaN"

Error: Invalid value for <image> attribute y="NaN"

# Use jquery-mobile-map-js library!

```
var helper =  
    new jQueryHelper(map);
```

# Auto-recenter after orientation change

```
//Listen for map load event  
map.on("load",init);
```

```
function init(){
```

```
  try{
```

```
    helper = new jQueryHelper(map);  
    helper.setCenterPt(x,y,4326);
```

```
  }
```

```
  catch(err) {
```

```
    console.log("jQueryHelper " + err.message);
```

```
  }
```

```
//Some browsers don't show full height after onLoad
```

```
map.reposition();
```

```
map.resize();
```

```
}
```



# Use jquery-mobile-map-js library!

The following won't work for all use cases:

```
map.reposition( );
```

```
map.resize( );
```

# Working with GPS



# Working with GPS

```
// Wait until map has loaded before
// starting geolocation
map.on("load", startGeolocation);

function startGeolocation(){
    navigator.geolocation.getCurrentPosition(
        locationSuccess,
        locationError,
        {setHighAccuracy:true}
    );
}
```

# GPS Best Practices

When viewing non-map pages

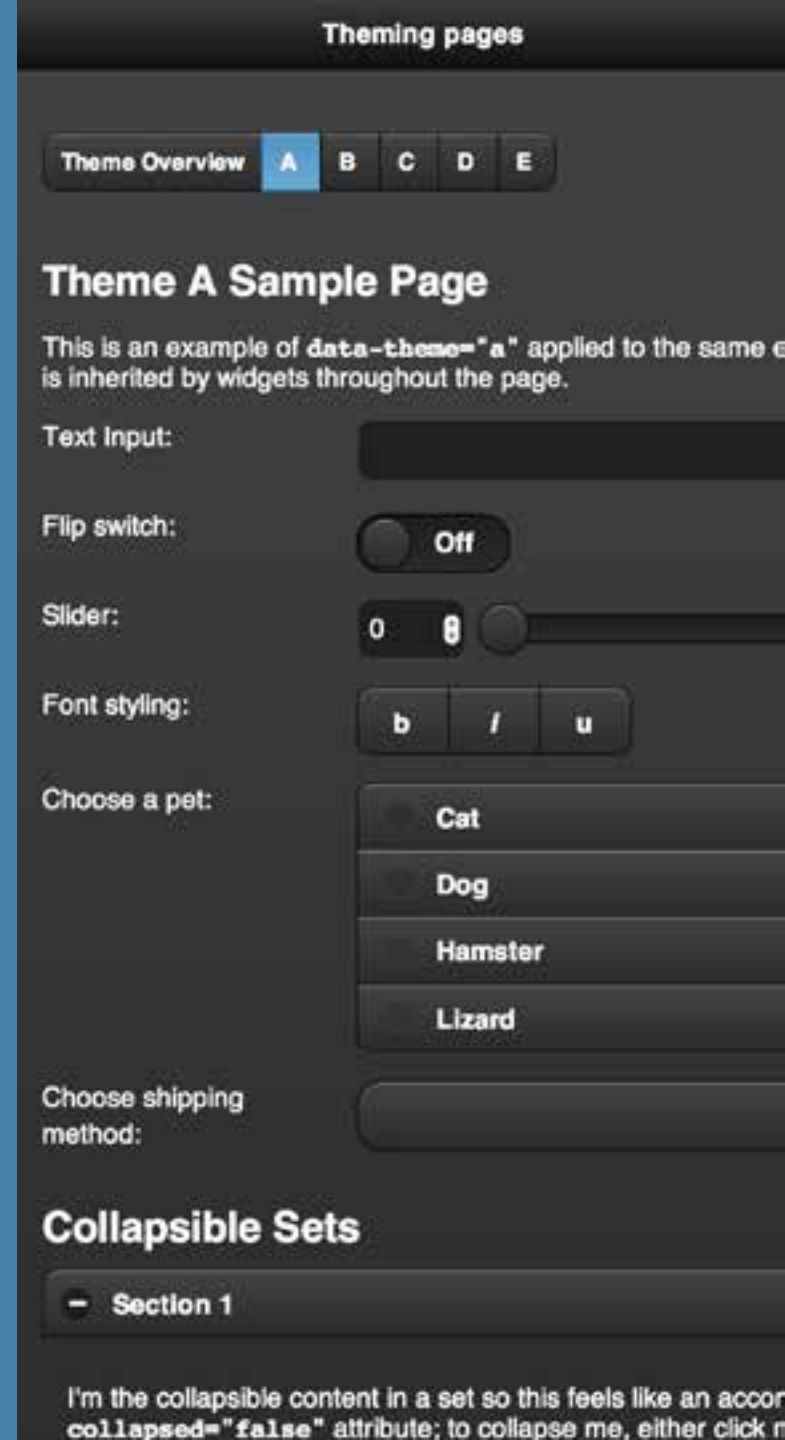
- Turn off GPS
- Or, do not write points to map
- Or, cache points in localStorage

# GPS Best Practices

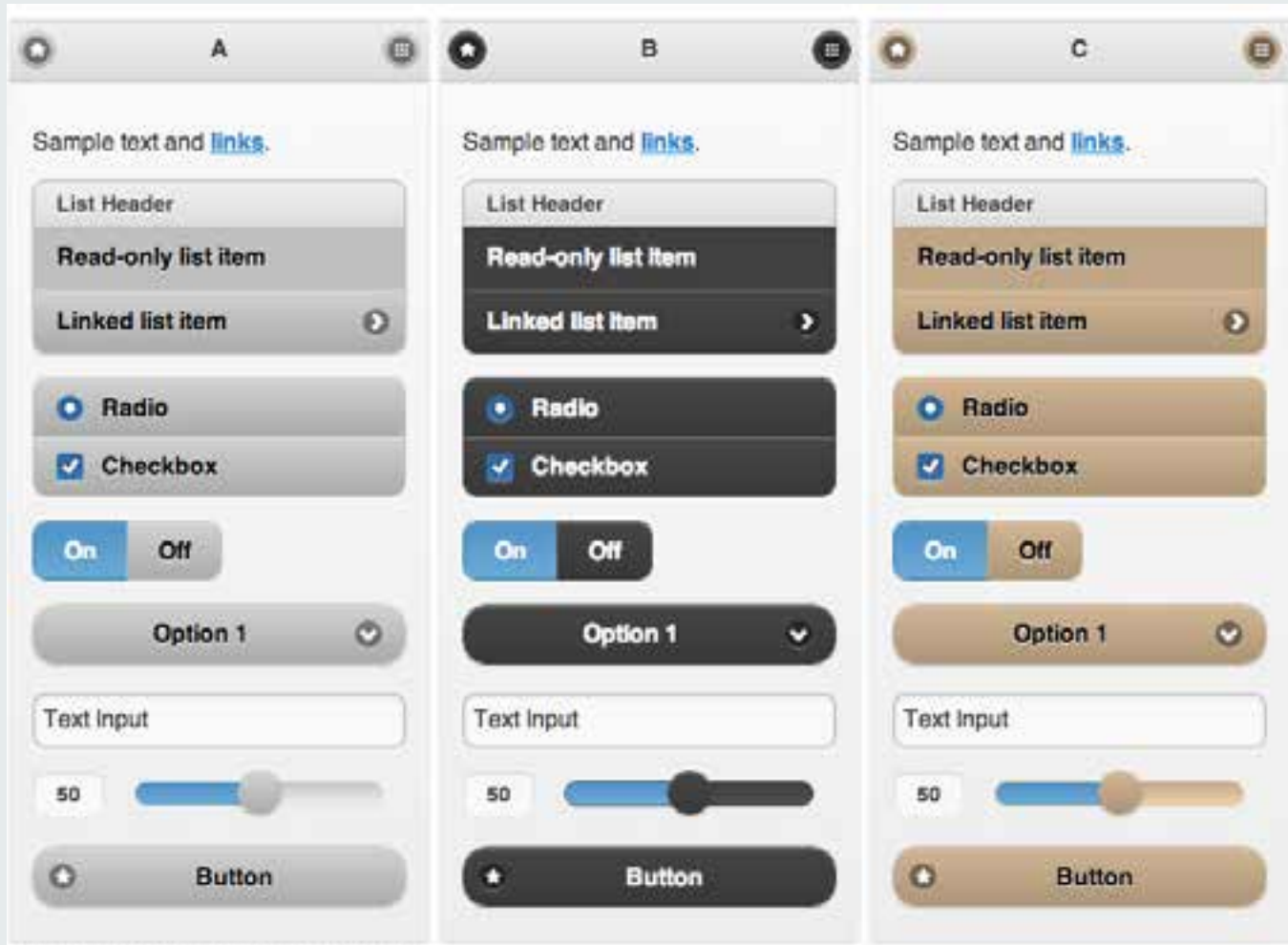
When returning to map page

- “helper-map-loaded” event
- Turn GPS back on
- Recenter map
- Write cache points to map

# Themes



# Themes and Styling



# JavaScript UX Frameworks wrap up

Mimic native behavior

Gives more control over life cycle

Can you used on Google Play and  
App Store



# Questions?

@agup

[Github.com/andygup](https://github.com/andygup)



Understanding our world.