

DevSummit DC

February 11, 2015 | Washington, DC



ArcGIS API for JavaScript: Building Web Mobile Apps

Andy Gup
&
Ben Ramseth

Agenda

Why responsive design?

Web AppBuilder

Bootstrap-map-js

GPS and offline

Web Optimizer

Andy Gup

agup@esri.com

@agup

Ben Ramseth

bramseth@esri.com

@esriMapNinja

Why Responsive Design?

Default browser behavior



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

Responsive design components

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

Responsive Web Frameworks

- Bootstrap 3
- Foundation 3
- Skeleton
- YAML 4
- jQuery Mobile Responsive Grid

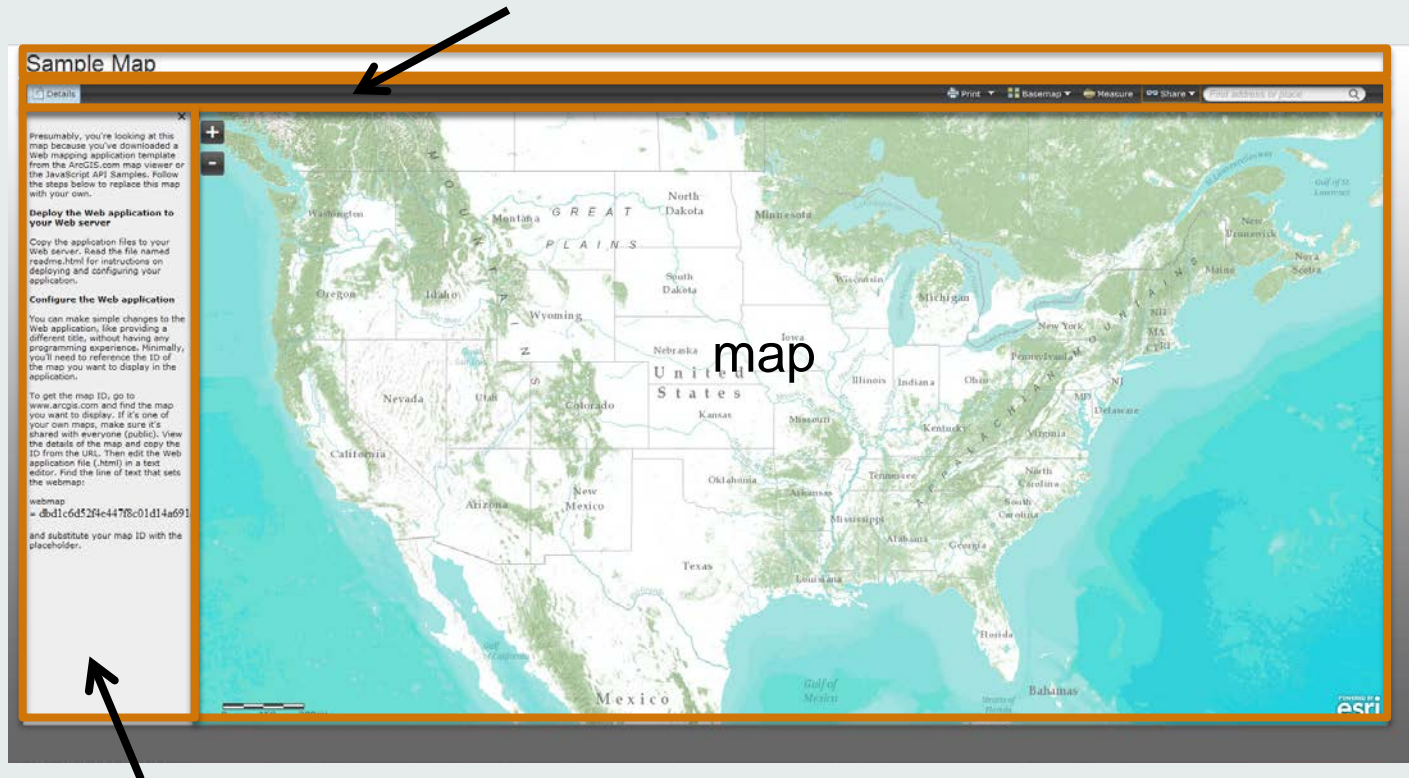
Media Queries

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

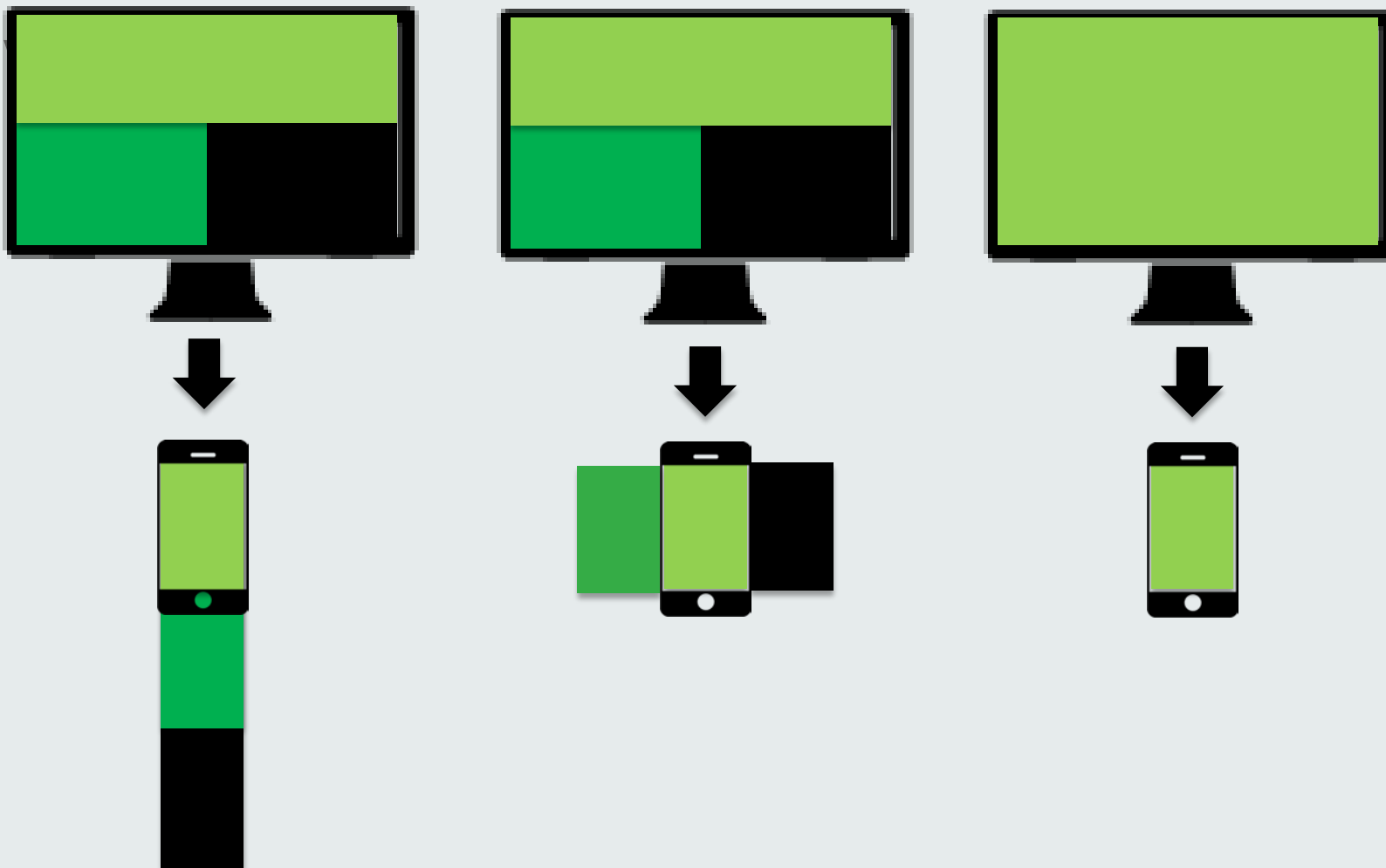
```
@media only screen and (max-device-width:480px) {  
  
    /* Custom css styles */  
    body {  
        font-size: 0.5em;  
    }  
    #titleArea{  
        display: none;  
    }  
}
```

Typical “full-view” Mapping App

3 Row – 2 Column



Responsive Grid Layouts



Large: 3 Rows - 3 Columns



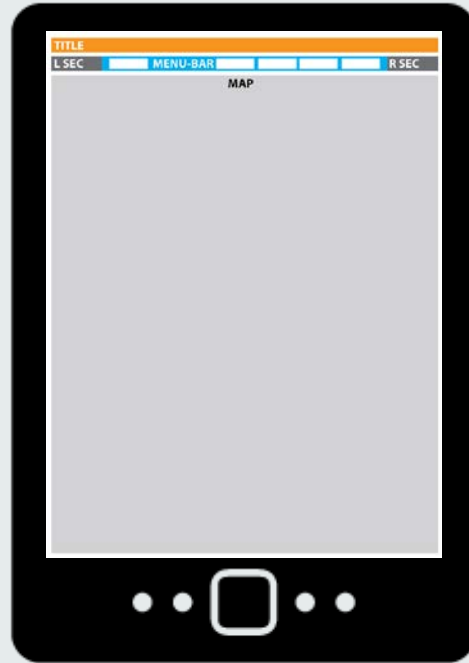
$\geq 1280\text{px}$

Medium: 2 Columns



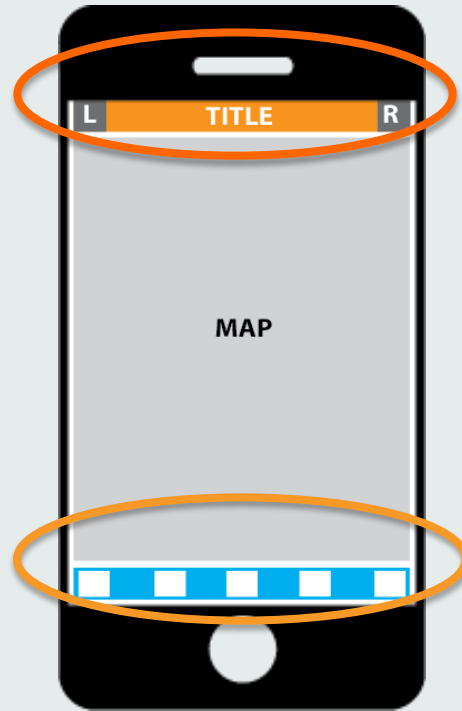
1024 - 1280px

Small: Single Column



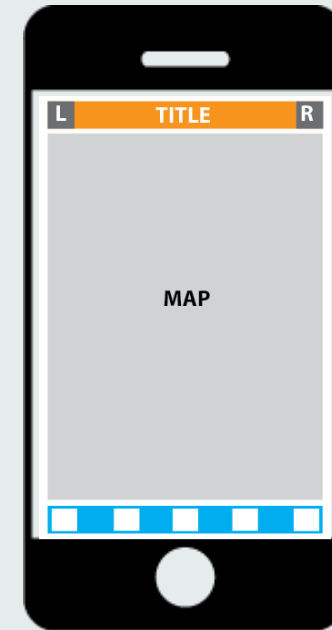
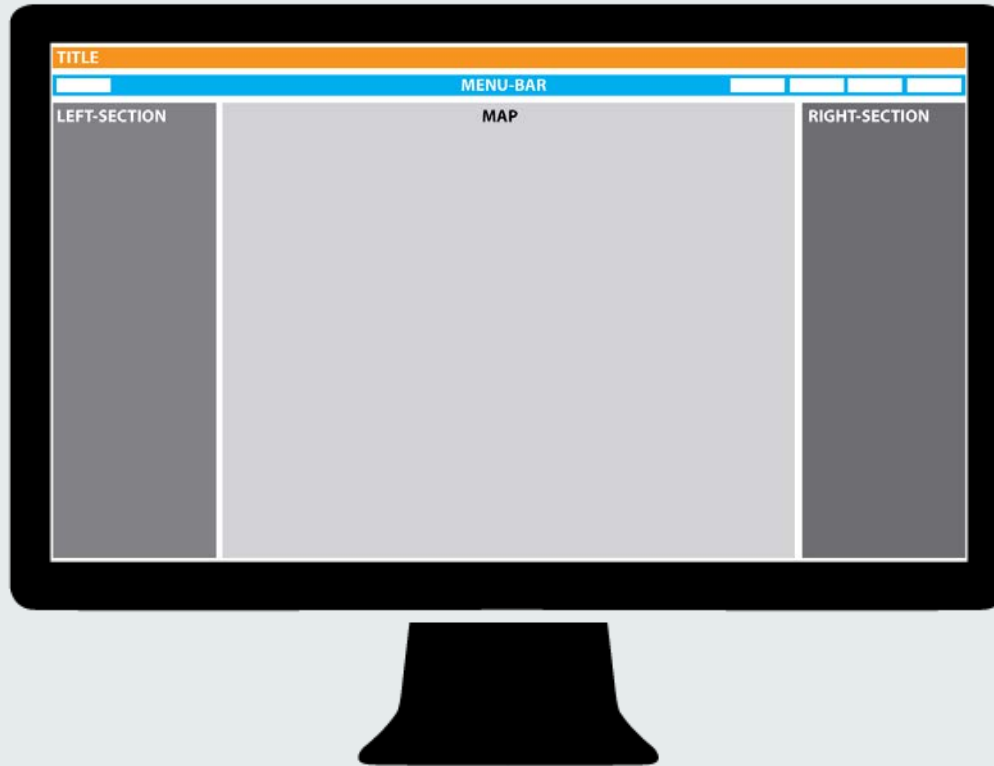
768 - 1024px

Extra Small: 1 Column, Minimized



$\leq 768\text{px}$

Responsive HTML5 Application



Esri's open source responsive libs

[github.com/esri/...](https://github.com/esri/)

[response-map-js](#)

[bootstrap-map-js](#)

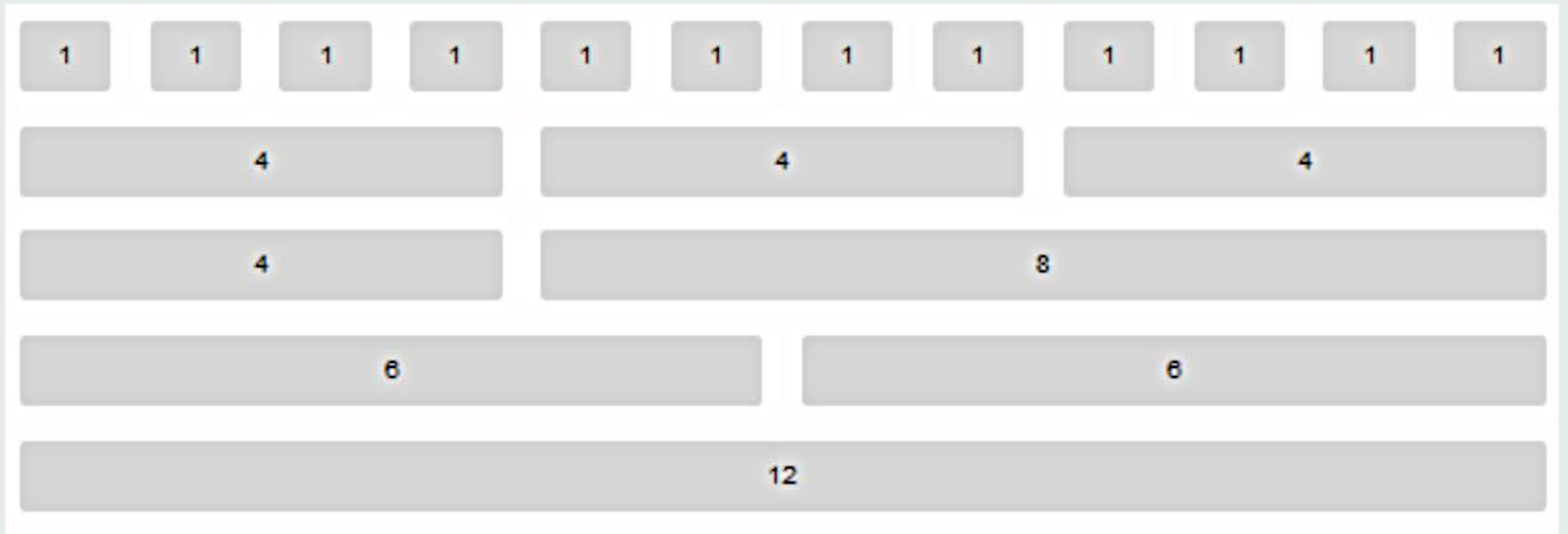
[dojo-bootstrap-ui-for-maps-js](#)

Bootstrap 101

The Fluid Grid System

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

The Fluid Grid System



The Fluid Grid System

| | Extra small devices Phones (<768px) | Small devices Tablets (≥768px) | Medium devices Desktops (≥992px) | Large devices Desktops (≥1200px) |
|------------------------|---|--|--|--|
| Grid behavior | Horizontal at all times | Collapsed to start, horizontal above breakpoints | | |
| Container width | None (auto) | 750px | 970px | 1170px |
| Class prefix | <code>.col-xs-</code> | <code>.col-sm-</code> | <code>.col-md-</code> | <code>.col-lg-</code> |
| # of columns | 12 | | | |
| Column width | Auto | 60px | 78px | 95px |

The Fluid Grid System

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

Define Column



The diagram illustrates the relationship between device size, number of columns, and column definition in a fluid grid system. It features a code snippet at the top: `<div class="col-xs-12 col-sm-8">`. Below the code, three blue arrows point upwards to the class names. The arrow from 'Define Column' points to 'col-xs-12'. The arrow from 'Device Size' points to 'col-sm-8'. The arrow from 'Number of Columns' points to '12'.

Device Size

**Number of
Columns**

The Fluid Grid System

```
<div class="row">  
  <div class="col-xs-12 col-sm-12 col-lg-12">  
    <h5>Top 12</h5>  
  </div>  
</div>  
<div class="row">  
  <div class="col-xs-12 col-sm-8 col-lg-9">  
    <!-- Bootstrap-map-js -->  
    <div id="mapDiv1"></div>  
  </div>  
  <div class="col-xs-12 col-sm-4 col-lg-3">  
    <h5>Right 3</h5>  
  </div>  
</div>
```

Bootstrap breakpoints

```
/* Extra small devices (phones, less than 768px)
   No media query since this is the default in Bootstrap
*/
```

```
/* Small devices (tablets, 768px and up) */
@media (min-width: @screen-sm-min) { ... }
```

```
/* Medium devices (desktops, 992px and up) */
@media (min-width: @screen-md-min) { ... }
```

```
/* Large devices (large desktops, 1200px and up)*/
@media (min-width: @screen-lg-min) { ... }
```


Bootstrap-map-js

github.com/Esri/bootstrap-map-js

Bootstrap-map-js

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

A screenshot of the Bootstrap-map-js website header. The background is a light gray map of San Francisco with the Golden Gate Bridge visible. The text 'Golden Gate' is in the top right corner. The main title 'Bootstrap-map-js' is in a large, bold, dark gray font. Below it, a subtitle reads 'Examples of how to build beautiful web mapping apps that run on any device.' At the bottom, there are two buttons: a blue one labeled 'Dojo + jQuery' and a green one labeled 'Pure Dojo'.

Bootstrap-map-js

Examples of how to build beautiful web mapping apps that run on any device.

Dojo + jQuery

Pure Dojo

ArcGIS Ready

Build web mapping and GIS applications that use the [ArcGIS JavaScript API](#) and [Bootstrap](#). Just add a few lines of CSS and JS to your app and you are good to go! For more mapping examples visit [quickstart-map-js](#).

Responsive

Use the [Bootstrap 3](#) grid system to design a responsive layout that works on mobile, tablet and desktops. Make your maps and pop-ups automatically resize and reposition for different devices.

CSS Components

Use the [Bootstrap 3](#) css styles and components with jQuery or Dojo to build interactive navigation bars, menus, pop-up windows and other user interface controls for your mapping apps.

Demo Bootstrap-map-js

Web AppBuilder

Create a suite of custom apps

Extensible framework for developers



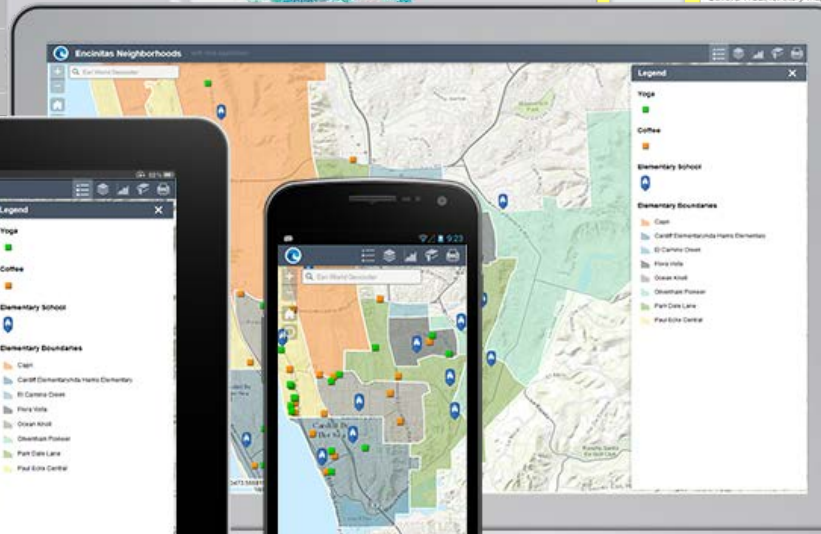
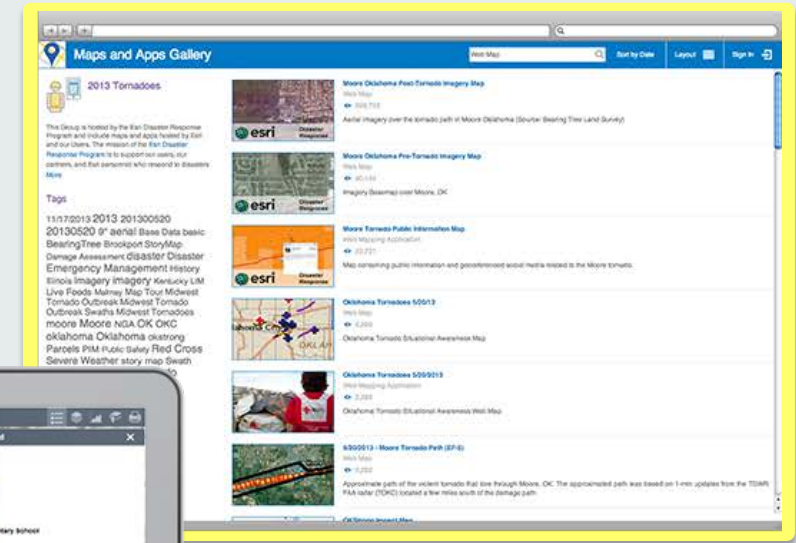
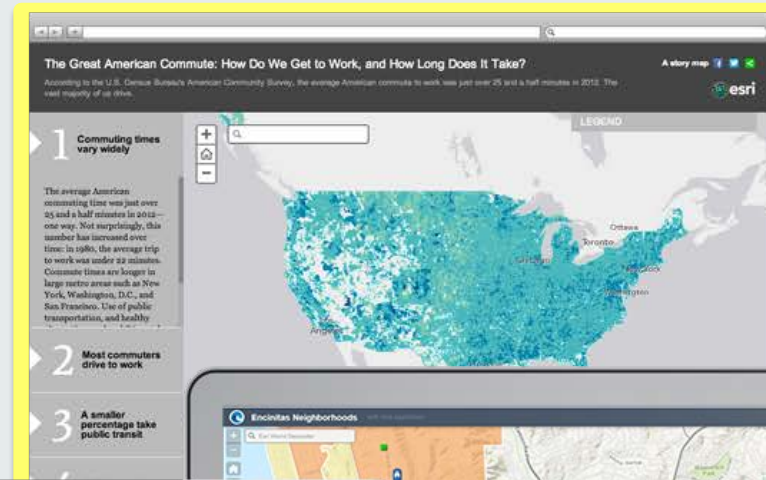
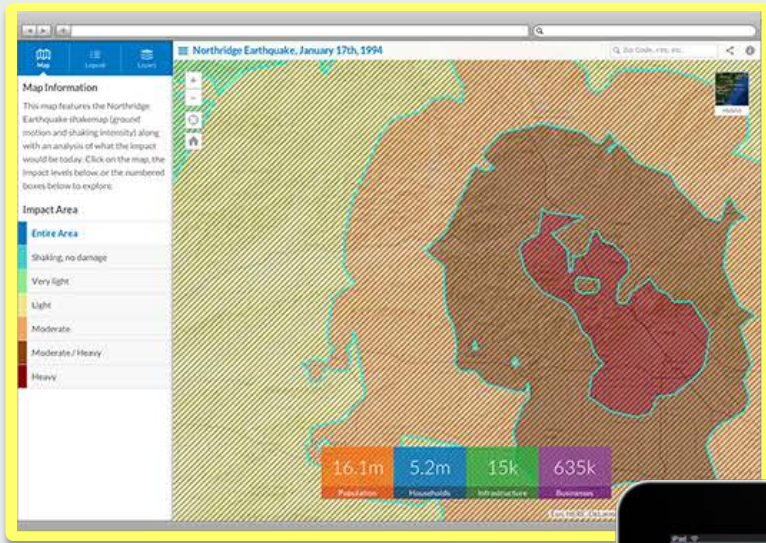
What

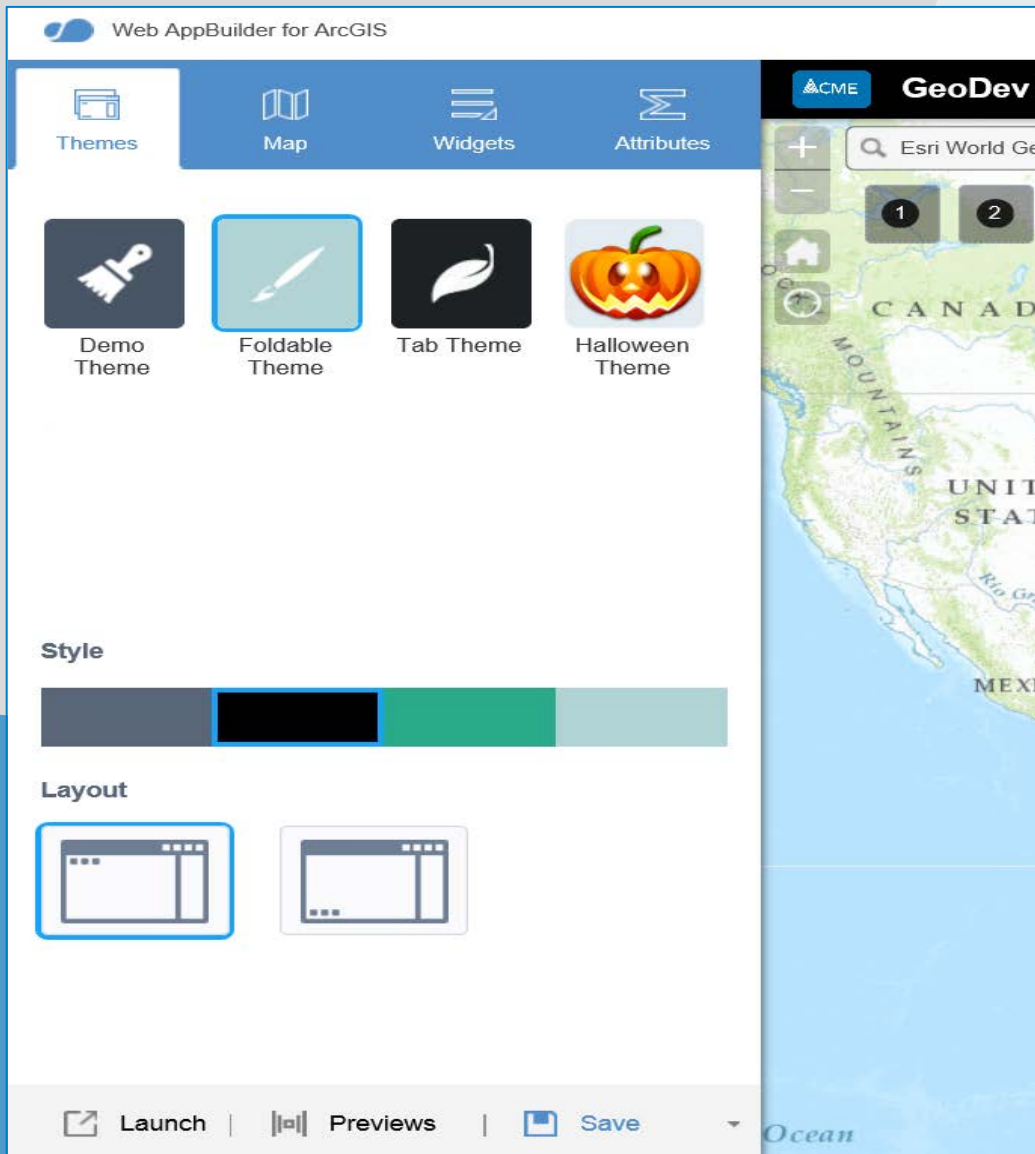
is the Web AppBuilder?

Empower non-developers

Bring “builder” experience from Flex/Silverlight to JavaScript

Multiple Devices





Demo Web AppBuilder

Working with GeoLocation

GeoLocation Startup - Waiting for map to load

```
var mapLoaded = false;
```

```
navigator.geolocation.watchPosition(  
    locationSuccess,  
    locationError,  
    {setHighAccuracy:true}  
);
```

```
map.on("load",function(evt){  
    mapLoaded = true;  
});
```

```
function locationSuccess(evt){  
    if(mapLoaded){ . . .}  
}
```

GeoLocation Best Practices

Start GeoLocation as early as possible

When viewing non-map pages (MVC):

- Don't draw points on map

Cache points in localStorage

- Use last point when starting map

GeoLocation Best Practices

When returning to map page or browser restart

- Turn GPS back on
- Recenter map
- Write cache points to map

Offline JavaScript

Why disconnected JavaScript?

Mobile web editing in areas of intermittent or no internet

Ability to reload or restart map in areas of intermittent or no internet

Lightweight cross-browser functionality

Need a full featured, robust offline solution?

ArcGIS Runtimes for iOS, Android, Qt and .NET!

Includes integrated support for offline editing and synchronization.

Also fully supports related tables, sub-types, domains and much more.

Offline-editing-js Library

Experimental



<https://github.com/Esri/offline-editor-js>

Offline-editor-js

Examples of how to build lightweight offline mapping applications for the web.

Editing

Build web mapping and geospatial applications that use the [ArcGIS JavaScript API](#) and allow you to temporarily store adds, updates and deletes of geographic features while the device is offline.

Tiles

Stores tile images from tiled map services locally. Once tiles are stored your mapping app can seamlessly transition between online and offline modes.

TPK

Add .tpk files (binary tile packages) directly to your app whether it is online or offline. Use TPKs as the stand-alone mapping source file or alongside tiled map services.

Demo Offline-editor-js

ArcGIS Web Optimizer

Use the ArcGIS Web Optimizer

- Works with the ArcGIS API for JavaScript
- Create smaller, optimized builds
- For hybrid, can host locally

<https://jso.arcgis.com/>

Andy Gup

agup@esri.com

@agup

Ben Ramseth

bramseth@esri.com

@esriMapNinja



Understanding our world.



