



Web App Builder Developer Edition

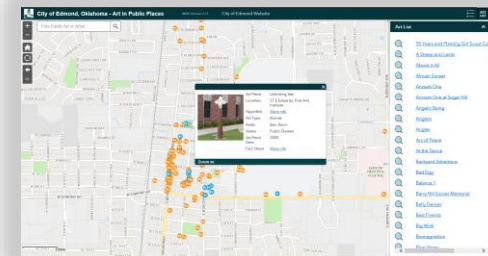
Customizing and Adding Functionality

By

Ian Peebles, GISP

City of Edmond Information Technology

Wednesday, April 3, 2019





Objectives

Explore the Web App Builder Developer Edition, covering the following topics:

- What are the differences between the AGS Online Version and Developer Edition?
- Getting Started
- Web App Builder Popups (with Demo)
- Web App Builder Splash Screen (with Demo)
- Web App Builder Custom Widgets (with Demo)
- Resources to help get started
- Closing Statements and Questions





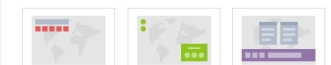
Web App Builder

ArcGIS Online WAB vs. Developer Edition WAB

What are the differences?

ArcGIS Online WAB

Home ▾ Web AppBuilder for ArcGIS



Billboard Theme Box Theme Dart Theme



Foldable Theme Dashboard Theme Launched Theme



Jewelry Box Theme Plateau Theme Tab Theme



Pocket Theme

Style



Easily Created through ArcGIS Online Account

Application Hosted on ArcGIS Online

Applications Created without Coding

Number of Widgets can be added and configured without any Coding

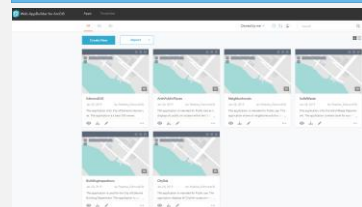
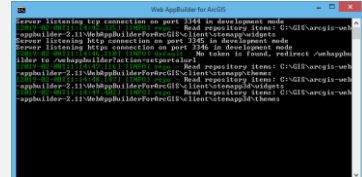
An easy solution for organizations looking to deploy simple applications compatible on numerous devices

Developer Edition WAB

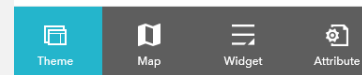
ArcGIS for Developers / Web AppBuilder
Web AppBuilder for ArcGIS (Developer Edition)

Download SDK
Version 2.11 - January 2019

Home Guide API Reference Sample Code Support



Web AppBuilder for ArcGIS EdmondGIS



Downloadable application/SDK that can be hosted on your local machine or server

Application hosted within your organization

Code is opened up to the applications and can be customized

Custom widgets and themes can be created

More control over your applications and can modify the layout, display, and content to match your organization content

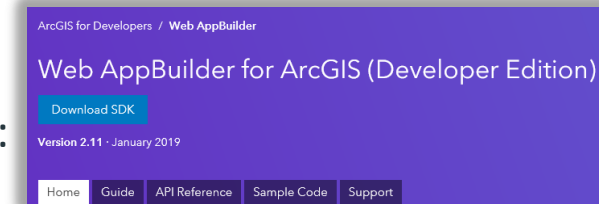




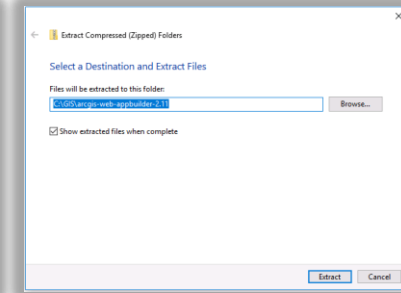
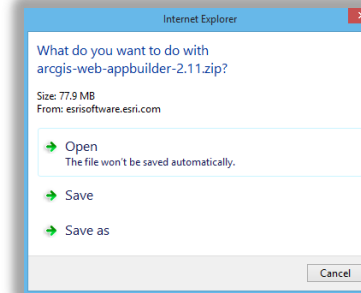
Getting Started

Installing the WAB Developer Edition

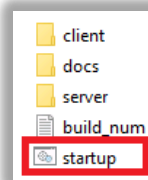
- Download the WAB Developer Edition, from the following link:
<https://developers.arcgis.com/web-appbuilder/>



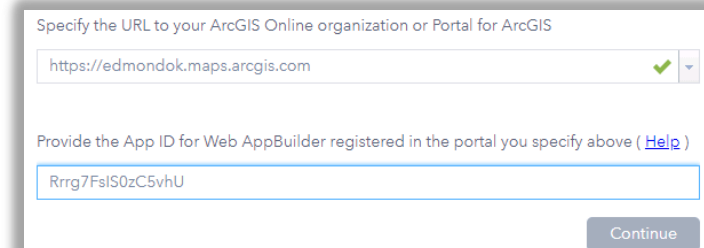
- Save the WAB to a local directory such as C:\GIS
- Extract all files from the .zip folder to your directory



- Following the install, click on the startup.bat file
<C:\GIS\arcgis-web-appbuilder-2.11\WebAppBuilderForArcGIS>



- Enter in the Portal ID and App ID from ArcGIS Online

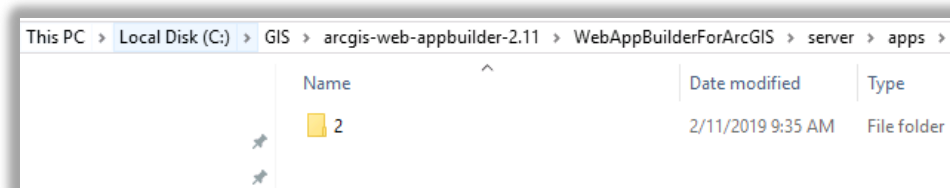
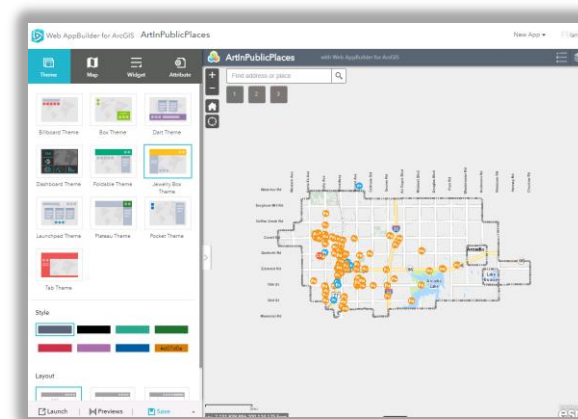
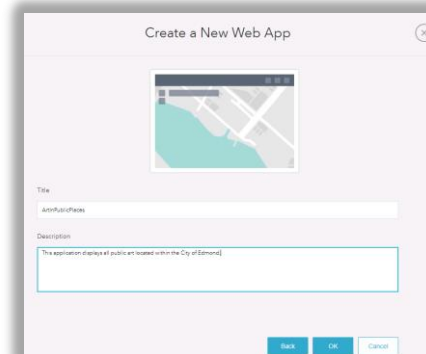
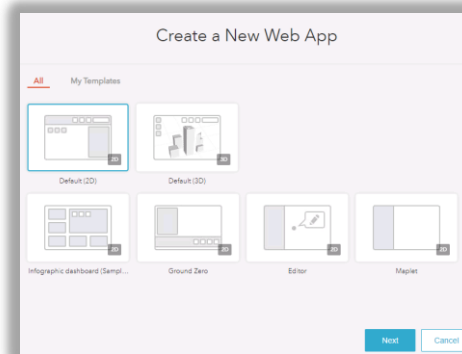




Getting Started

Create an Application

- Create a new application of your choice
- Give the application a title and description
- Choose theme, base map (through AGS Online), then save changes
- Following the save, a new Web App ID will be assigned
- Application directory now created and can be accessed



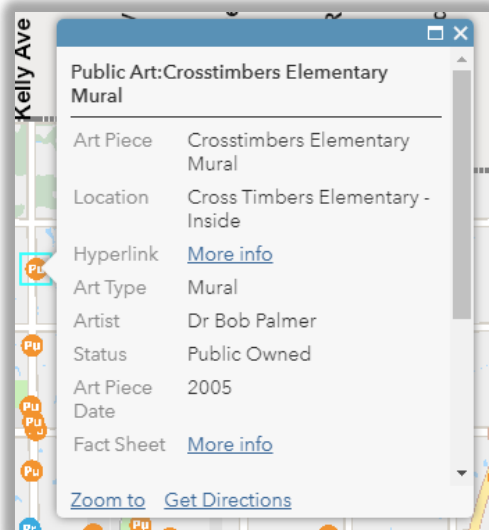


Web App Builder Popups

Popups display attribute information for features within the application

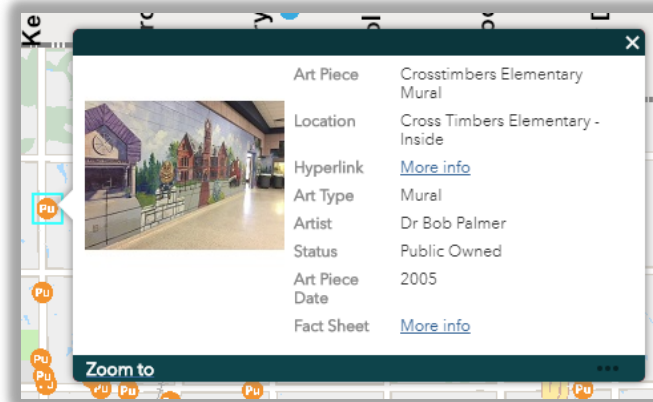
Sample #1

AGS Online Popup (Not Customized)



Sample #2

WAB Dev Edition Popup (Customized)



- I want to improve the popup, but how?
- Introduce simple CSS code to make modifications

```
popup.css
1 .esriPopup, .esriPopupMobile, .esriMobileNavigationBar,
2   z-index: 108
3 }
4
5 .esriRtl .esriPopupWrapper {
6   text-align: right;
7 }
8
9 .esriPopup .esriPopupWrapper {
10  font-family: "Avenir Light";
11 }
12
13 .esriPopup .header, .esriMobilePopupInfoView .header {
14  font-family: "Avenir Heavy";
15 }
16
17 .esriPopup .contentPane {
18  background-color: #grey;
19 }
```

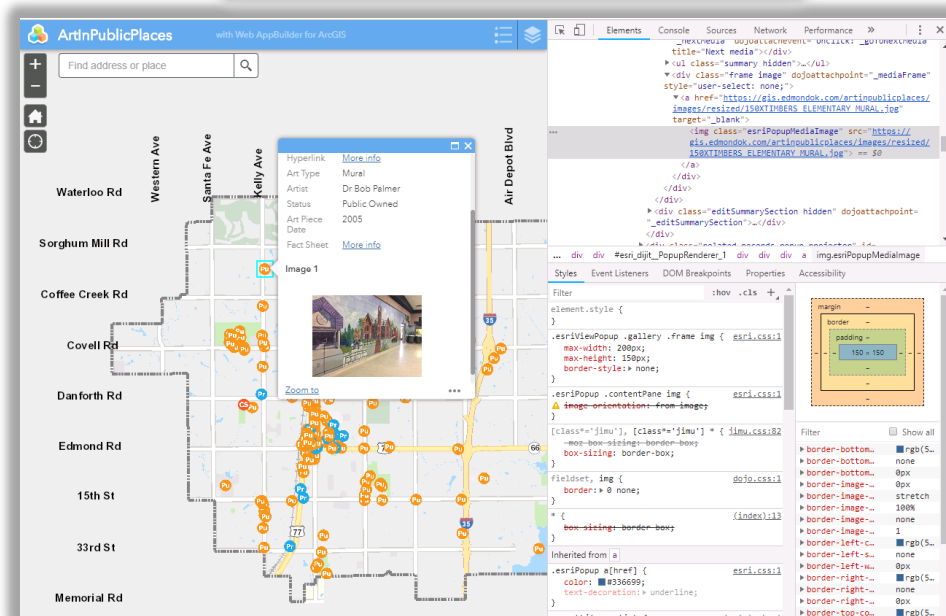




Web App Builder Popups

- Introduce CSS modifications. Go to:
...\\WebAppBuilderForArcGIS\\server\\apps\\8\\jimu.js\\css\\popup.css
- Launch the application - Identify the elements of the popup using debugging tools (CTRL + Shift + I in Google Chrome)
- You can use CSS properties from the select elements to determine the code you need to update the layout of the popup
- Add in the CSS you need in the common.css file

Name	Date modified	Type	Size
fonts	4/2/2019 9:07 AM	File folder	
images	4/2/2019 9:07 AM	File folder	
sprite-icons	4/2/2019 9:07 AM	File folder	
clamp	1/17/2019 1:03 PM	Cascading Style S...	2 KB
ColorPicker	1/17/2019 1:03 PM	Cascading Style S...	2 KB
dojo-override	1/17/2019 1:03 PM	Cascading Style S...	6 KB
jimu	1/17/2019 1:03 PM	Cascading Style S...	161 KB
jimu-ie	1/17/2019 1:03 PM	Cascading Style S...	3 KB
jimu-theme	1/17/2019 1:03 PM	Cascading Style S...	10 KB
popup	4/2/2019 10:01 AM	Cascading Style S...	7 KB



```

107  /* Popup Attribute Table - Includes Attributes and Data */
108  .esriPopup .esriPopupWrapper .attrTable {
109      float:right;
110      margin-right: 5px;
111      margin-top:-5px;
112      width:255px;
113  }

```

```

115  /* Popup Attribute Table List */
116  .esriPopup .esriPopupWrapper .attrName {
117      width: 80px;
118      font-weight: bold;
119  }

```

```

134  /* Preview Image 150x150px */
135  .esriPopup .esriPopupWrapper .gallery .frame img {
136      margin-top: 0px;
137      margin-left: -18px;
138      width: 150px;
139      height: 150px;
140  }

```





Web App Builder Popups

DEMO



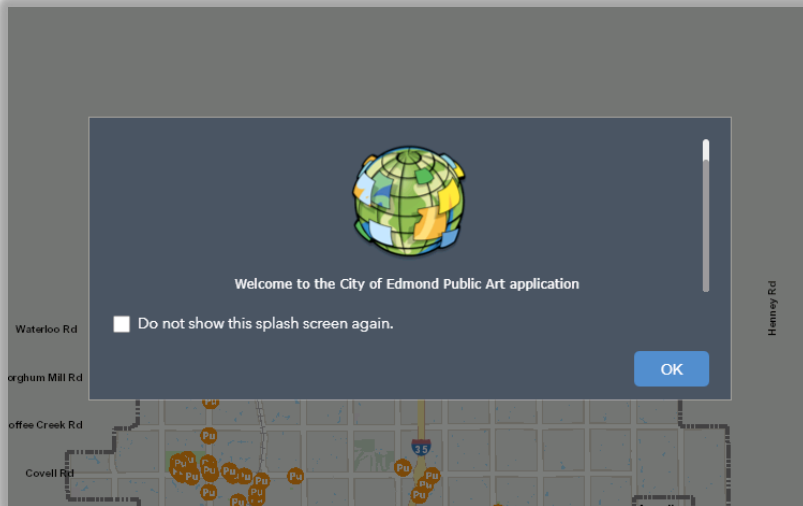


Web App Builder Splash Screen

The splash screen appears when the application launches and contains a logo, description, disclaimer or other information pertinent to the application.

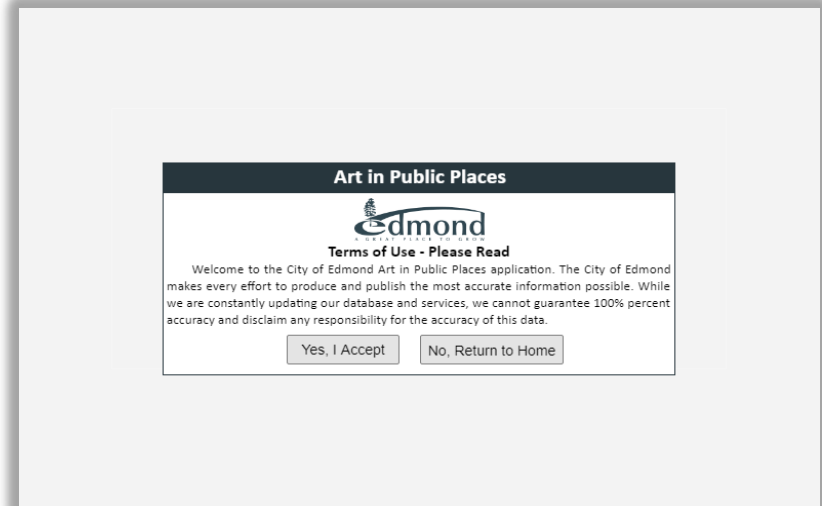
Sample #1

AGS Online Splash (Not Customized)



Sample #2

WAB Dev Edition Splash (Customized)



- How can I improve the splash screen?
- Simply activate the splash screen widget within the app and save as part of app
- Introduce HTML and CSS code to the splash widget

Name	Date modified	Type	Size
css	1/28/2019 8:18 AM	File folder	
images	1/28/2019 8:35 AM	File folder	
nls	1/28/2019 8:18 AM	File folder	
setting	1/28/2019 8:18 AM	File folder	
config	1/17/2019 1:04 PM	JSON File	32 KB
manifest	1/17/2019 1:04 PM	JSON File	1 KB
VersionManager	1/17/2019 1:04 PM	JS File	3 KB
Widget	1/28/2019 8:36 AM	HTML File	4 KB
Widget	1/17/2019 1:04 PM	JS File	18 KB





Web App Builder Splash Screen

- Introduce HTML modifications. Go to:
...\\WebAppBuilderForArcGIS\\server\\apps\\2\\widgets\\Splash\\Widget.html
- Introduce CSS modifications. Go to:
...\\WebAppBuilderForArcGIS\\server\\apps\\2\\widgets\\Splash\\css\\style.css
- As you add in the HTML DIVS, you can define the DIV properties in the style.css
- As you add in the code for the DIVS and define the CSS, you can have the application up in the browser and have the debugging tools on
- Refresh the application as needed to test your customization

```
10 <!-- Splash Screen Header || Ian Added: 2017-05-08 -->
11 <div id="splashScreenHeader">
12   <div id="splashScreenHeaderTextContainer">
13     <center>Neighborhoods</center>
14   </div>
15 </div>
16
17 <!-- Splash Screen Thumbnail Image -->
18 <div id="splashImageFrame">
19   <center><center>Terms of Use - Please Read</center></div>
24
25 <!-- Terms of Use Text -->
26 <div id="termsOfUseTextContainer">
27   <center><div id="termsOfUseText">
28     <p><strong>Welcome to the City of Edmond</strong> Neighborhoods
29     makes every effort to produce and publish the most accurate information p
30     database and services, we cannot guarantee 100% percent accuracy and disc
31     data.
32   </div>
33 </div>
34 <div class="custom-content" data-dojo-attach-point="customContentNode">
```

```
218 #splashContainerFrame{
219   position: absolute;
220   top: 60px;
221   background-color: white;
222   border: 1px solid #26363d;
223   width: 500px;
224   height: 215px;
225 }
226
227 @media only screen and (max-width: 530px) {
228   #splashContainerFrame{
229     position: absolute;
230     top: 60px;
231     background-color: white;
232     border: 1px solid #26363d;
233     width: 300px;
234     height: 215px;
235   }
236 }
237
238 #splashScreenHeader{
239   position: absolute;
240   height: 30px;
241   width: 500px;
242   background-color: #26363d;
243   color: white;
244 }
245
```





Web App Builder Splash Screen

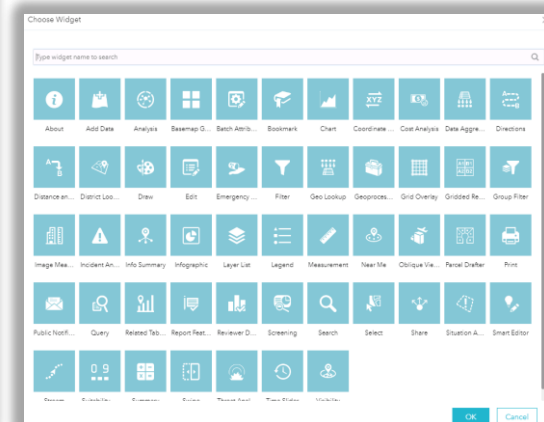
DEMO





A widget allows for adding functionality to an application by working with the map and web services

- Both the AGS Online and Developer Edition versions of the WAB contain a number of ready-to-use widgets.
- Out-of-the-Box widgets are great, but certain functionality may not be available Out-of-the-Box.
- The WAB Developer Edition provides the environment to customize, test, and deploy custom widgets that may not be part of the Out-of-the-Box functionality
- Extend your WAB functionality leveraging the JavaScript API



Zoom to Sc...







Web App Builder Custom Widgets

Getting started, let's build a simple widget (Night Mode):


1. Go to the following location:

...\WebAppBuilderForArcGIS\client\stemapp\widgets\samplewidgets

2. Copy the  Demo folder and rename it to  NightMode .

Widget Folder Structure

Name	Date modified	Type
css	2/11/2019 4:20 PM	File folder
images	2/11/2019 4:21 PM	File folder
nls	2/11/2019 4:21 PM	File folder
setting	2/11/2019 4:21 PM	File folder
config.json	2/11/2019 9:04 AM	JSON File
manifest.json	2/12/2019 8:06 AM	JSON File
ShowVertexFeatureAction	2/11/2019 9:04 AM	JavaScript File
Widget	2/11/2019 9:04 AM	HTML File
Widget	2/11/2019 9:04 AM	JavaScript File



3. Open the  images folder and make a copy of the of the **icon.png** file. You can create your own icon if you choose, otherwise you can use the ESRI default.



Default



Custom

4. Open the  nls folder, then the  strings file in a text editor and make the following changes and save:

```
1 define({
2   root: {
3     widgetLabel: "Demo",
4     _featureAction_ShowVertex: "Show Vertex Count",
5     label1: "I am a demo widget.",
6     label2: "This is configurable."
7   }
8 },
```

```
1 define({
2   root: {
3     widgetLabel: "Night Mode",
4     _featureAction_ShowVertex: "",
5     label1: "This widget toggles between day and night mode.",
6     label2: "There are no configurations for this widget."
7   }
8 },
```

→ 1. Add Widget Label

→ 2. Add widget labels





Web App Builder Custom Widgets

Getting started, let's build a simple widget (Night Mode):

5. Open the manifest.json file in a text editor and perform the following changes, then save:

From:

```
1 {  
2   "name": "Demo",  
3   "platform": "HTML",  
4   "version": "2.11",  
5   "wabVersion": "2.11",  
6   "author": "Esri R&D Center Beijing",  
7   "description": "This is the widget used in developer guide",  
8   "copyright": "",  
9   "license": "http://www.apache.org/licenses/LICENSE-2.0",  
10  "featureActions": [{  
11    "name": "ShowVertex",  
12    "uri": "ShowVertexFeatureAction"  
13  }]  
14 }
```

```
1 {  
2   "name": "NightMode",  
3   "platform": "HTML",  
4   "version": "2.11",  
5   "wabVersion": "2.11",  
6   "author": "City of Edmond, Oklahoma - Information Technology",  
7   "description": "This widget toggles between day and night mode.",  
8   "copyright": "",  
9   "license": "http://www.apache.org/licenses/LICENSE-2.0",  
10  "properties": {  
11    "inPanel": false,  
12    "hasUIFile": false,  
13    "hasConfig": false  
14  }  
15 }
```

1. Change widget name

2. Update author and title

3. Change featureActions to Properties, and add properties

6. Open the Widget file and let's observe the following code blocks (no modifications)

```
1 define(['dojo/_base/declare', 'jimu/BaseWidget'],  
2 function(declare, BaseWidget) {  
3   //To create a widget, you need to derive from BaseWidget.  
4   return declare([BaseWidget], {  
5     // DemoWidget code goes here  
6  
7     //please note that this property is be set by the framework  
8     //templateString: template,  
9  
10    baseClass: 'jimu-widget-demo',  
11  
12    postCreate: function() {  
13      this.inherited(arguments);  
14      console.log('postCreate');  
15    },  
16  
17    startup: function() {  
18      this.inherited(arguments);  
19      this.mapIdNode.innerHTML = 'map id: ' + this.map.id;  
20      console.log('startup');  
21    },  
22  
23    onOpen: function(){  
24      console.log('onOpen');  
25    },  
26  });
```

```
27 onClose: function(){  
28   console.log('onClose');  
29 },  
30  
31 onMinimize: function(){  
32   console.log('onMinimize');  
33 },  
34  
35 onMaximize: function(){  
36   console.log('onMaximize');  
37 },  
38  
39 onSignIn: function(credential){  
40   /* jshint unused:false*/  
41   console.log('onSignIn');  
42 },  
43  
44 onSignOut: function(){  
45   console.log('onSignOut');  
46 },  
47  
48 showVertexCount: function(count){  
49   this.vertexCount.innerHTML = 'The vertex count is: ' + count;  
50 }  
51 });  
52
```





Web App Builder Custom Widgets

Getting started, let's build a simple widget (Night Mode):

7. Let's do some JavaScript and introduce new code for the widget in the **widget.js** file

- Add in Classes and Comments

```
1 define(['esri/layers/FeatureLayer', 'esri/renderers/UniqueValueRenderer', 'esri/renderers/SimpleRenderer', 'esri/symbols/SimpleLineSymbol',
2 'dojo/base/Color', 'dojo/base/declare', 'jimu/BaseWidget'],
3 function (FeatureLayer, UniqueValueRenderer, SimpleRenderer, SimpleLineSymbol, Color, declare, BaseWidget) {
4     //To create a widget, you need to derive from BaseWidget.
5     return declare([BaseWidget], {
6         // DemoWidget code goes here
7         //
8         // Night Mode Widget - for Web App Builder Developer Edition
9         // Created By: Ian Peebles, GISP
10        // Organization: City of Edmond
11        // Department: Information Technology
12        // Updated: 08/29/2018
13        // Description: This widget is used for toggling between night and day mode
14        //                  in the map for the Web App Builder
15        //
16        //please note that this property is be set by the framework when widget is loaded.
17        //templateString: template,
18        //
19        // Declare the baseClass
20        baseClass: 'jimu-widget-NightMode',
21    });
22 }
```

1. Add in required ESRI JavaScript classes

2. (Optional) Add in comments for the widget

3. Rename the jimu-widget-demo class to jimu-widget-NightMode





Web App Builder Custom Widgets

Getting started, let's build a simple widget (Night Mode):

8. The widget works when it is **opened** and when it is **closed** only during the cycle.

```
onOpen: function(){  
    console.log('onOpen');  
},
```

```
onClose: function(){  
    console.log('onClose');  
},
```

9. Add for the **onOpen: function()**:

```
// Night Mode - On Click (When Widget is Open)  
onOpen: function(){  
    console.log('onOpen');  
    var mapbgcolor = document.getElementsByClassName("esriMapContainer");  
    mapbgcolor[1].style.background = "black";  
    console.log("Map changed to night time mode (black).")  
  
    // *****  
    // * START - Streets Symbol Rendering *  
    // *****  
    // Set Default Renderer if no value is present - Default Symbol is required as specified in the API documentation  
    var defaultStreetsSymbol = new SimpleLineSymbol().setColor(new Color([0, 0, 0, 0]));  
  
    // Create a New Symbol Renderer  
    var rendererStreets = new UniqueValueRenderer(defaultStreetsSymbol, "LINETYPE");  
    rendererStreets.addValue("1", new SimpleLineSymbol().setWidth(2).setColor(new Color([255, 255, 255, 1])));  
  
    // Add in Streets Feature layer  
    streetsFL = new FeatureLayer("https://arcgis03/arcgis/rest/services/Base/Base/MapServer/16", {  
        id: "Streets Night Mode",  
        visible: true  
    });  
  
    // Apply Streets Renderer and Streets Layer to Map  
    streetsFL.setRenderer(rendererStreets);  
    this.map.addLayer(streetsFL);  
  
    // *****  
    // * END - Streets Symbol Rendering *  
    // *****
```

4. Define Map Background as black

5. Set Default Renderer if no value is present (Streets)

6. Define symbol renderer for Streets

7. Define Streets Feature Layer

8. Set Renderer for Feature Layer and Add in Feature Layer





Web App Builder Custom Widgets

Getting started, let's build a simple widget (Night Mode):

9. Add for the **onOpen: function():continued**

9. Define the City Limits symbol

```
// *****  
// * START - City Limits Symbol Rendering *  
// *****  
// Set Default Renderer if no value is present - Default Symbol is required as specified in the API documentation  
var defaultCityLimitsSymbol = new SimpleLineSymbol(SimpleLineSymbol.STYLE_SHORTDASH).setWidth(4).setColor(new Color([255, 255, 0]));
```

10. Define the symbol renderer
to be used for City Limits

```
// Create a New Symbol Renderer  
var rendererCityLimits = new SimpleRenderer(defaultCityLimitsSymbol);
```

11. Define the City
Limits Feature Layer

```
// Add in Streets Feature layer  
cityLimitsFL = new FeatureLayer("https://arcgis03/arcgis/rest/services/Base/Base/MapServer/18", {  
  id: "City Limits Night Mode",  
  visible: true  
});
```

12. Set Renderer for
Feature Layer and Add in
Feature Layer

```
// Apply Streets Renderer and Streets Layer to Map  
cityLimitsFL.setRenderer(rendererCityLimits);  
this.map.addLayer(cityLimitsFL);
```

```
// *****  
// * END - City Limits Symbol Rendering *  
// *****
```

```
},
```





Web App Builder Custom Widgets

Getting started, let's build a simple widget (Night Mode):

10. Add for the **onClose: function()**:

```
// Day Mode - On Click (When Widget is Closed)
onClose: function(){
  console.log('onClose');

  var mapbgcolor = document.getElementsByClassName("esriMapContainer");
  mapbgcolor[1].style.background = "white";
  console.log("Map changed to night time mode (white).")

  // Remove the Streets Feature Layer after the widget is closed
  this.map.removeLayer(streetsFL);

  // Remove the City Limits Feature Layer after the widget is closed
  this.map.removeLayer(cityLimitsFL);
},
```

13. Change map background back to white for daytime mode

14. Remove feature layers with renders when daytime mode is activated

11. Go to the following location:

...\WebAppBuilderForArcGIS\client\stemapp\sample-configs and open the **config-demo.json**

From:

```
"widgets": [{
  "label": "Demo",
  "uri": "widgets/samplewidgets/Demo/Widget"
},{
  "label": "UsejQuery",
  "uri": "widgets/samplewidgets/UsejQuery/Widget"
},{
  "label": "Simple",
  "uri": "widgets/samplewidgets/Simple/Widget"
},{
  "label": "CustomWidgetTemplate",
  "uri": "widgets/samplewidgets/CustomWidgetTemplate/Widget"
}, {
  "uri": "widgets/samplewidgets/WidgetCommunication/WidgetB/Widget"
}]
```

To:

```
"widgets": [{
  "label": "Night Mode",
  "uri": "widgets/samplewidgets/NightMode/Widget"
},{
  "label": "Demo",
  "uri": "widgets/samplewidgets/Demo/Widget"
},{
  "label": "UsejQuery",
  "uri": "widgets/samplewidgets/UsejQuery/Widget"
},{
  "label": "Simple",
  "uri": "widgets/samplewidgets/Simple/Widget"
},{
  "label": "CustomWidgetTemplate",
  "uri": "widgets/samplewidgets/CustomWidgetTemplate/Widget"
}, {
  "uri": "widgets/samplewidgets/WidgetCommunication/WidgetB/Widget"
}]
```

15. Insert following (widget) code, then save



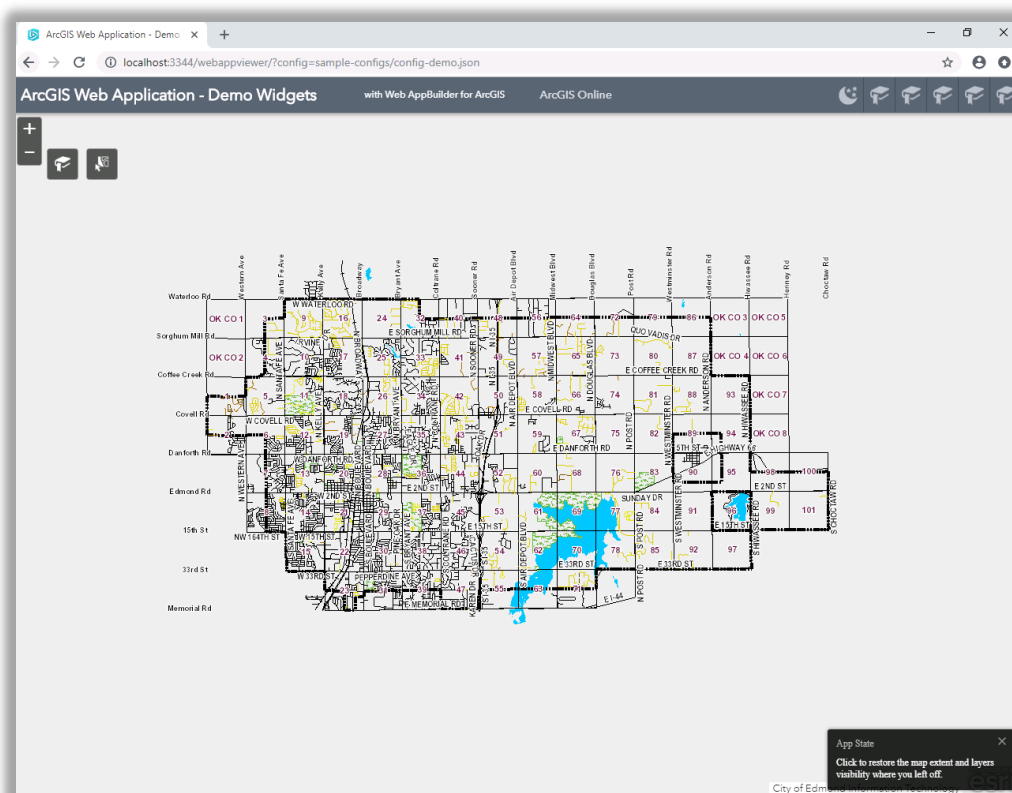


Web App Builder Custom Widgets

We are done, now it is time to test before deployment

11. Open the following link from the machine where the WAB is installed:

<http://localhost:3344/webappviewer/?config=sample-configs/config-demo.json>





Web App Builder Custom Widgets

DEMO

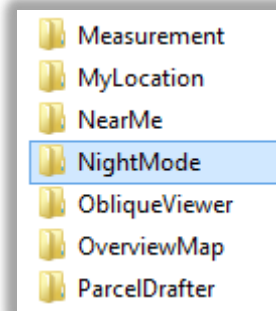




Web App Builder Custom Widgets

The tool works in the test environment, so how do I deploy?

1. Copy the NightMode widget (entire folder) from:
...\\WebAppBuilderForArcGIS\\client\\stemapp\\widgets\\samplewidgets



To:
...\\WebAppBuilderForArcGIS\\client\\stemapp\\widgets

2. Open **config.json** file located at:
...\\WebAppBuilderForArcGIS\\client\\stemapp

Look for the block of code:

```
170  "widgets": [{
171    "label": "Bookmark",
172    "uri": "widgets/Bookmark/Widget"
173  }, {
174    "label": "Legend",
175    "uri": "widgets/Legend/Widget"
176  }, {
177    "label": "LayerList",
178    "uri": "widgets/LayerList/Widget"
179  }, {
```



1. Add in these lines of code and save

```
170  "widgets": [{
171    "label": "NightMode",
172    "uri": "widgets/NightMode/Widget"
173  }, {
174    "label": "Bookmark",
175    "uri": "widgets/Bookmark/Widget"
176  }, {
177    "label": "Legend",
178    "uri": "widgets/Legend/Widget"
179  }, {
180    "label": "LayerList",
181    "uri": "widgets/LayerList/Widget"
182  }, {
```





Web App Builder Custom Widgets

DEMO





Web App Builder Custom Widgets

Resources to help get started

ESRI Resources

Web App Builder for ArcGIS Developer Edition:

<https://developers.arcgis.com/web-appbuilder/>

Web App Builder for ArcGIS Developer Edition – Creating a Custom in-panel widget:

<https://developers.arcgis.com/web-appbuilder/>

ArcGIS API for JavaScript:

<https://developers.arcgis.com/javascript/>

Non ESRI Resources

W3schools.com – for coding, HTML, CSS, and much much more!:

<https://www.w3schools.com/>





Web App Builder Custom Widgets

Closing Statements and Questions

City of Edmond GIS Successes:

- Organization has been using the WAB near the first release and using latest version
- Learned how to navigate the structure and develop custom widgets (night mode, zoom to scale, and feature list)

Future:

- Work towards converting python GP tools to JavaScript widgets for the WAB
- Continue to assess mapping needs and desired functionality for staff members
- New ESRI Builder coming....Experience Builder coming BETA summer 2019? 2020
- Experience Builder built on the 4.x JavaScript API

Challenges:

- The Developer Edition of the WAB is very extensive...thousands of files, hard to find settings and configurations at times
- Unique and sometimes hard to follow coding structure
- Documentation limited on how to perform modifications. It is important to document

QUESTIONS

