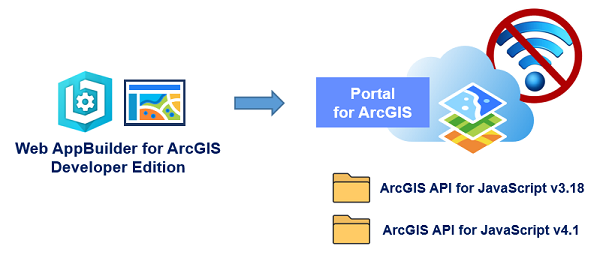
[Deploying Web AppBuilder for ArcGIS Developer Edition in a Disconnected Portal for ArcGIS Environment](https://blogs.esri.com/esri/arcgis/2016/11/02/deploying-web-appbuilder-for-arcgis-developer-edition-in-a-disconnected-portal-for-arcgis-environment/)

This blog post was contributed by Craig Cleveland, a Solution Engineer on the National Government Team in the Esri Washington, DC office.

Most users that would like to deploy [Web AppBuilder for ArcGIS Developer Edition](https://developers.arcgis.com/web-appbuilder/) with [Portal for ArcGIS](http://www.esri.com/software/arcgis/arcgisserver/extensions/portal-for-arcgis) can follow the [Get Started](https://developers.arcgis.com/web-appbuilder/guide/getstarted.htm) directions and be up and running almost immediately. However, some customers using Portal for ArcGIS operate in a completely disconnected environment (without access to the Internet), and for these users a few quick adjustments to the deployment workflow are required before they can get started.

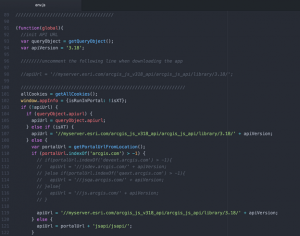
A prerequisite for successfully deploying Web AppBuilder Developer Edition v2.2 in a disconnected Portal environment is to deploy a copy of v3.18 of the [ArcGIS API for JavaScript](https://developers.arcgis.com/javascript/). [This technical support article](http://support.esri.com/technical-article/000011974) outlines how to accomplish this if you’re not already familiar with the process. (Although the article references v3.16 of the ArcGIS API for JavaScript, the process is identical for v3.18.) The article also explains how to configure the ArcGIS API for JavaScript for use with ArcGIS Server, but that process is not necessary for this use case.

[](https://blogs.esri.com/esri/arcgis/files/2016/11/Graphic.png)Once you have enabled a local instance of the ArcGIS API for JavaScript library, the following steps outline the modifications that need to be made to Web AppBuilder:

1. Browse to the unzipped location of Web AppBuilder Developer Edition and navigate to the client\stemapp directory. Open the **env.js** file in a text editor (e.g. Notepad++) and modify the apiUrl variables on lines 98, 107 and 119 to point to your local ArcGIS API for JavaScript v3.18 deployment. Once completed, save and close the file. The following are examples of what the references look like before and after the modifications have been made.

Before:  apiUrl = ‘//js.arcgis.com/’  
After:  apiUrl = ‘//myserver.esri.com/arcgis\_js\_v318\_api/arcgis\_js\_api/library/3.18/’

Note: the URL to your instance of the ArcGIS API for JavaScript may vary depending on if you’ve chosen to deploy it as outlined in the technical support article, or if you’ve made additional adjustments as desired.

[](https://blogs.esri.com/esri/arcgis/files/2016/11/Fig1.png)

2. The next change is to update the URL of the Geometry Service in your **config.json** file. In the same directory noted above (client\stemapp) open the **config.json** file in a text editor and modify the “geometryService” value to point to a local ArcGIS Server URL.

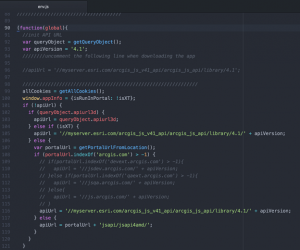
Before:  “geometryService” = ‘http://tasks.arcgisonline.com/ArcGIS/rest/services/Geometry/GeometryServer’  
After:  “geometryService” = ‘http://myserver.esri.com/server/rest/services/Utilities/Geometry/GeometryServer

To use the 3D capabilities of Web AppBuilder Developer Edition you’ll need to repeat this process once more in an additional location. FYI: The 3D capabilities in Web AppBuilder utilize version 4.1 of the ArcGIS API for JavaScript.

3. Browse to the unzipped location of Web AppBuilder Developer Edition and navigate to the client\stemapp3D directory. Open the **env.js** file in a text editor and modify the apiUrl variables on lines 96, 105 and 116 to point to your local ArcGIS API for JavaScript v4.1 deployment. Once completed, save and close the file. The following are examples of what the references look like before and after the modifications have been made.

Before:  apiUrl = ‘//js.arcgis.com/’  
After:  apiUrl = ‘ //myserver.esri.com/arcgis\_js\_v41\_api/arcgis\_js\_api/library/4.1/’

Note2: please keep in mind that the URL to your instance of the ArcGIS API for JavaScript may vary depending on how you’ve chosen to deploy it.

[](https://blogs.esri.com/esri/arcgis/files/2016/11/Fig2.png)

4. The final change is to update the URL of the Geometry Service in your **config.json** file. In the same directory noted in step 3 (client\stemapp3D) open the **config.json** file in a  text editor and modify the “geometryService” value to point to a local ArcGIS Server URL.

Before:  “geometryService” = ‘http://tasks.arcgisonline.com/ArcGIS/rest/services/Geometry/GeometryServer’  
After:  “geometryService” = ‘http://myserver.esri.com/server/rest/services/Utilities/Geometry/GeometryServer’

At this point you’ve made all the necessary changes to utilize the Web AppBuilder for ArcGIS Developer Edition v2.2 in a disconnected Portal environment. Follow the [Get Started](https://developers.arcgis.com/web-appbuilder/guide/getstarted.htm) directions from here and you should be up and running in no time!

**No SLL in Local Install**

1. Option - <https://community.esri.com/t5/arcgis-web-appbuilder-questions/ssl-in-web-app-builder/td-p/243745>
2. Option
   1. When configure the app

Graphical user interface, text, application

Description automatically generated

* 1. Reset startup

Graphical user interface, text, application

Description automatically generated