Customizing Explorer apps for imagery

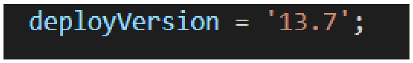
Files for the Explorer apps for imagery are provided in GitHub for users who want to download, customize, and host similar apps themselves. Below are guidelines for customizing these apps.

For more details about widget development and theme development, please refer to Web AppBuilder for ArcGIS documentation.

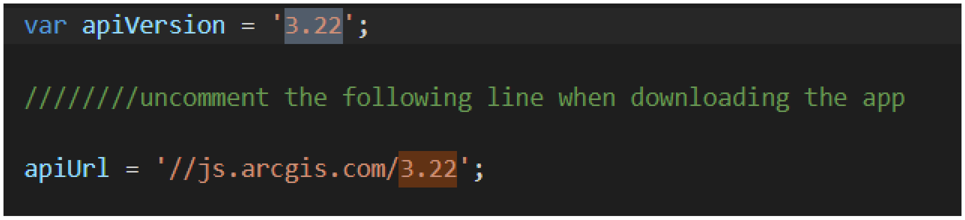
Important files to update

When deploying a customized version of an Explorer app for imagery, be sure to update the following files:

1. **env.js**—Update the ArcGIS JS API version and app deploy version number.



```
deployVersion = '13.7';
```

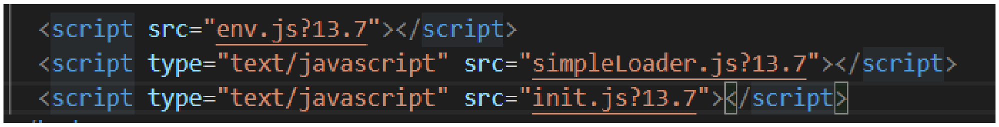


```
var apiVersion = '3.22';

////////uncomment the following line when downloading the app

apiUrl = '//js.arcgis.com/3.22';
```

2. **index.html**—Append the app deploy version number set in env.js to the JS files loaded below so that the old cached files are not loaded once you modify the app.



```
<script src="env.js?13.7"></script>
<script type="text/javascript" src="simpleLoader.js?13.7"></script>
<script type="text/javascript" src="init.js?13.7"></script>
```

3. **config.json**—Update app parameters as needed. Likely examples include:
   - Portal URL
   - App ID
   - Web map item ID
   - Different widget positionings
   - URLs to widgets' HTML and JS files

Important folders to review

1. **Configs**—All the config files are arranged according to the corresponding widgets, each in its own separate folder. The paths to these config files need to be provided in the **config.json** file mentioned above.

   All configurations that your widgets need are saved here, including service URLs, pixel values, important field names, raster function templates, etc. Review these closely and update for your custom app as necessary.

2. **Themes**—This is where all the JS and CSS files related to your theme are located. There are widgets, panels, and layouts, each of which define how your app looks. You can modify the files here to create your own custom theme (learn more about creating custom themes).

3. **Widgets**—Each widget in your application is arranged in its own folder structure with a JS file, HTML file, manifest.json file, CSS folder, localization folder, and images folder. If a widget needs to have settings configured, there is a Settings folder as well, again with corresponding HTML, JS, and CSS files. There is also an optional config.json file in each widget folder.

   The **manifest.json** file is very important when you are creating a new widget. This is the file where you will define if your widget has a settings page or not; if it has a local config file or not; and if it has localization support or not.

   The most important thing to keep in mind is that **the name parameter in the manifest.json file must the same as the widget folder name**. For example, for the About Widget, the name parameter in the manifest.json file is set to “About”, same as the folder name. If the name doesn’t match the folder, the widget will not come up in the app.