



CIT 255 - Training Module Project

ArcGIS JavaScript API

By Neal Sabin & Colin Haroutunian



Overview

- ArcGIS JavaScript API
 - Enables you to create interactive web maps, rich with features for visualizing and analyzing spatial data
 - Developed and maintained by ESRI, the creators of industry standard GIS mapping software. They also provide interfaces and APIs to create web mapping applications.



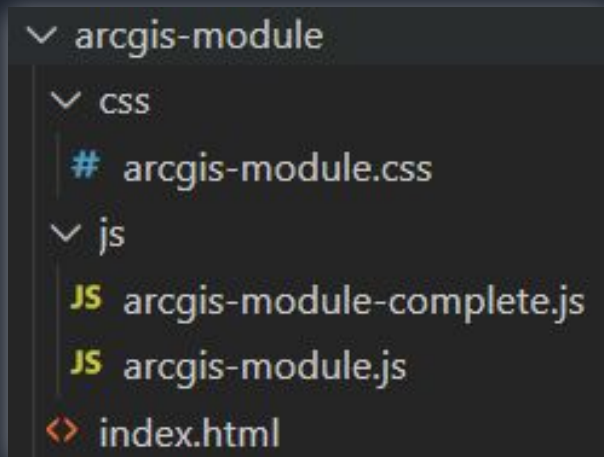
Learning Outcomes

1. Students are able to set up the Visual Studio Code environment to organize the files and folders of the map page.
2. Students are able to implement the ArcGIS JavaScript API script and modules into their pages.
3. Students are able to create multiple, detailed layers for their maps.
4. Students are able to create shapes on their maps.
5. Students are able to modify the features of their maps with effective styling.



Environment / File Setup

- Use your preferred code editor, we'll be using VS Code
- File structure
 - index.html
 - arcgis-module.css
 - arcgis-module.js
- You can also choose to use internal styling and JavaScript




index.html

arcgis-module > index.html > html

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no" />
6      <title>ArcGIS JavaScript API Training Module</title>
7      <link rel="stylesheet" href="css/arcgis-module.css">
8      <link rel="stylesheet" href="https://js.arcgis.com/4.21/esri/themes/light/main.css" />
9      <script src="https://js.arcgis.com/4.21/"></script>
10     <script src="js/arcgis-module-complete.js"></script>
11   </head>
12   <body>
13     <div id="viewDiv"></div>
14   </body>
15 </html>
```



arcgis-module.css

```
arcgis-module > css > # arcgis-module.css >  html
1  html,
2  body,
3  ▼ #viewDiv {
4      padding: 0;
5      margin: 0;
6      height: 100%;
7      width: 100%;
8  }
```

arcgis-module-complete.js

```
arcgis-module > js > JS arcgis-module-complete.js > require() callback > popupInfo > content > fieldInfos
```

```
1 v require(["esri/Map", "esri/views/MapView", "esri/layers/FeatureLayer", "esri/widgets/LayerList"], (Map, MapView, FeatureLayer, LayerList) => {
```

```
30   const map = new Map({
31     basemap: "topo-vector"
32   });
33
34   const view = new MapView({
35     container: "viewDiv", // Reference to the view div
36     map: map, // Reference to the map object
37     zoom: 6, // Sets zoom level based on level of detail (LOD)
38     center: [-84.930666, 44.317797] // Sets center point of view using longitude, latitude
39   });
```



Widgets

Widgets are reusable user-interface components and are key to providing a rich user experience. The ArcGIS for JavaScript API provides a set of ready-to-use widgets.

Two widgets we'll be using in our demo are:

1. Popup
2. Layer List



FeatureLayer

A FeatureLayer is a single layer that can be created from a Map Service or Feature Service. Services allow you to serve feature data over the internet. This makes your data available for use in web clients, desktop apps, and field apps. The layer can be spatial or non-spatial.

In our demonstration, we are creating an object from the FeatureLayer class.



Adding shapes using GraphicsLayer

Another way to display spatial data is to render graphics client side. Unlike a `FeatureLayer`, a `GraphicsLayer` has no associated geometry that is coming from a service, meaning you specify the geometry in the instance of the object.

We will be creating an object from the `Graphic` class and assigning properties to it to create the graphic.



DEMONSTRATION



THANK YOU

QUESTIONS OR COMMENTS