Connecting to ShareGIS Web Services in JavaScript and HTML

Updated January 30, 2023

INTRO

Many government agencies make map layers available through ArcGIS Server map services. The NYS GIS Program Office (GPO) has a catalog of web services (including civil boundaries, streets, address points, etc.) and a geocoding service, all of which can be connected to through JavaScript.

Connecting to web services in JavaScript is achieved through accessing an API for JavaScript, for example the ESRI standalone API or the ESRI-Leaflet API. Neither of these approaches require any cost for the developer as both the APIs and ShareGIS web services are free to the public.

This document covers adding a web service via:

1. the **ESRI API for JavaScript**; and

2. the **ESRI-Leaflet API for JavaScript**;

ESRI API for JavaScript

The ESRI API for JavaScript is located here: [Esri Support ArcGIS API for JavaScript](#)

Notes:

- This document covers the most current API version; version 4.25.
- A free ArcGIS developer account is required, however an API key is not required to access ShareGIS web services.

Generating a Map

ESRI provides an introductory tutorial on using the JavaScript API to generate a map. Start with this [tutorial](#) prior to attempting to add ShareGIS web services.

Adding ShareGIS Web Services to the Map

Adding a web service is achieved through ESRI's **FeatureLayer class** which is used to display a feature layer on top of all other layers including the basemap.

The **FeatureLayer class** can be used for both map and feature services.

Adding a web service is achieved through two steps:

1) First add the FeatureLayer module to the require statement in the script:

```
require([
        "esri/Map",
        "esri/views/MapView",
        "esri/layers/FeatureLayer"
    ],

    function(Map, MapView, FeatureLayer)
```

2) Then use the FeatureLayer class to define the url of the features you want added to the map

The url below is for NYS county boundary polygons clipped to the state shoreline.

```
// This is where we name the web servce we want to add and call its url with "new FeatureLayer"
counties = new FeatureLayer({
url: "https://gisservices.its.ny.gov/arcgis/rest/services/NYS_Civil_Boundaries/FeatureServer/3"
});

//Now add the layer to the map
map.add(counties);
```

You can copy the code below and paste it into a blank html document for an ESRI JavaScript map displaying the NYS Civil Boundaries web service.

```html
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no">
  <title>Adding ShareGIS Web Services with the ArcGIS API for JavaScript</title>
    <link rel="stylesheet" href="https://js.arcgis.com/4.25/esri/themes/light/main.css">
    <script src="https://js.arcgis.com/4.25/"></script>

  <style>
   html, body, #viewDiv {
     padding: 0;
     margin: 0;
     height: 100%;
     width: 100%;
    }
  </style>

  <script>

    require([
        "esri/Map",
        "esri/views/MapView",
        "esri/layers/FeatureLayer"
      ],

      function(Map, MapView, FeatureLayer)

      {
        var map = new Map({
        basemap: "topo-vector"
      });

      var view = new MapView({
        container: "viewDiv",
        map: map,
        center: [-73.75521, 42.65155], // longitude, latitude
        zoom: 8 //sets the starting zoom level
      });

  // This is where we name the web servce we want to add and call its url with "new FeatureLayer"
        const counties = new FeatureLayer({
         url: "https://gisservices.its.ny.gov/arcgis/rest/services/NYS_Civil_Boundaries/FeatureServer/3"
        });
```

```
  //Now add the layer to the map
        map.add(counties);
  });

</script>
</head>
<body>
  <div id="viewDiv"></div>
</body>
</html>
```

ESRI-Leaflet API for JavaScript

> "Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. Leaflet is designed
> with *simplicity*, *performance* and *usability* in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of plugins, has a beautiful, easy to use and well-documented API and a simple, readable source code." – Leaflet Development Website

ESRI has extended its developer resources to include an API for linking ESRI and Leaflet resulting in "Esri Leaflet," a light-weight, open source Leaflet plug-in for accessing ArcGIS services. You can use the ESRI-Leaflet API to display interactive maps and data, and to access services to perform operations such as geocoding, routing, and spatial analysis.

Notes:

- This document covers the most current API version; version 3.0.10
- A free ArcGIS developer account is required, however an API key is not required to access ShareGIS web services.

Generating a Map

Developer documentation for getting started with ESRI-Leaflet is available here: https://developers.arcgis.com/esri-leaflet/   Please refer to this resource for the prior to adding ShareGIS web services.

Web services are added in ESRI-Leaflet through the "L.esri.featureLayer" class. This class accepts a url for the desired web service. The full code for generating a map with the ShareGIS web service for county boundaries is provided below.

Full Code:

```html
<!doctype html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <title>ESRI-Leaflet Map with a Basemap</title>
 <meta name='viewport' content='initial-scale=1,maximum-scale=1,user-scalable=no' />

 <!-- Load Leaflet's .js and .css-->
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.9.3/leaflet.css">
        <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.9.3/leaflet.js"></script>

 <!-- Load Esri Leaflet from CDN.  it has no .css stylesheet of its own, only .js -->
        <script src=" https://cdnjs.cloudflare.com/ajax/libs/esri-leaflet/3.0.10/esri-leaflet.js"></script>

 <style>
  html,
  body,
  #map {
    height: 100%;
    width: 100%;
    margin: 0;
    padding: 0;
  }
 </style>
</head>
<body>

 <div id="map"></div>

 <script>
        const map = L.map('map', {
                center: [42.75, -75.23],
                zoom: 8
        });

        <!—Add a Basemap-->
        const esriStreets = L.esri.basemapLayer('Gray').addTo(map);

        <!—Add NYS County Boundaries Web Service-->
        var county = L.esri.featureLayer({
        url: 'https://gisservices.its.ny.gov/arcgis/rest/services/NYS_Civil_Boundaries/FeatureServer/3'
        });
```

```
    var assembly = L.esri.featureLayer({
        url: 'https://gisservices.its.ny.gov/arcgis/rest/services/NYS_Assembly_Districts/FeatureServer/0',
        color: 'red'
    }).addTo(map);

    // Notice that the Civil Boundaries service above defaults to a blue color.  The Assembly
    // districts would default to blue also, however the color option is set to draw in red.
    // Both feature services and map services can be styled

    // Create an object to contain my services as a toggled item in the top right layer box.  By default assembly
districts are on at the start
    var webservices = {
      "NYS Assembly Districts": assembly,
      "NYS County Boundaries": county
    };

    // Add a layer control where the feature services are overlays that can be toggled on and off
    L.control.layers(webservices).addTo(map);

    </script>
</body>
</html>
```