

2018 Feb 24

A Simple Leaflet Web Map Example

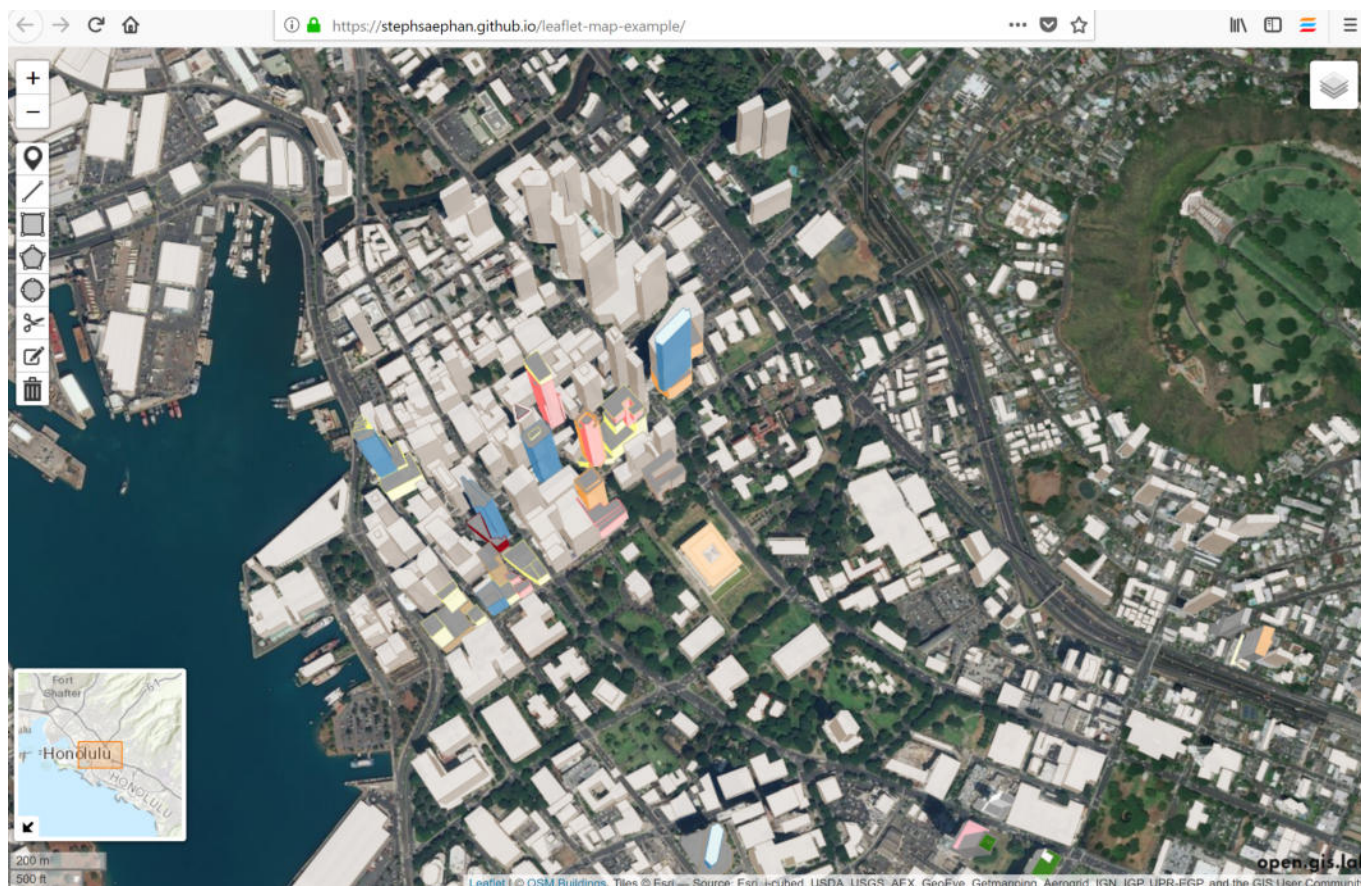
By now I think everyone has heard of [Leaflet](#). I've been playing around with it on and off over the last year. For this post, I thought I'd write about a simple Leaflet map example, using Leaflet and [ESRI Leaflet plugin](#). I am not a coding expert by any means, just someone who likes to learn, so I try to annotate/comment in my code as much as I can. This post is not a step-by-step guide in creating a Leaflet Maps but rather comments and notes on what I've learned in the process of using/learning Leaflet. However, feel free to download the source files and use it however you like.

The Leaflet example uses several plugins and calls layers from the [Hawaii Statewide GIS Program's ArcGIS REST](#) service and the [Pacific Islands Ocean Observing System \(PacIOOS\) Geoserver](#). You can view and download the source files on my GitHub account and also view the demonstration site as well.

Demo Web Map: <https://stephsaephan.github.io/leaflet-map-example/>

For the demo web map all of the data layers are turned off by default with exception of a basemap. I also include OSM 2.5D buildings to the map as well.

Source files: <https://github.com/StephSaephan/leaflet-map-example>



3 Lessons I Learned:

plugins

Not all plugins will work well together, even if they come from the same source. Case in point, ESRI Render Plugin will render layers using the default symbology as defined by ArcGIS REST service. This plugin though doesn't currently work with the ESRI Cluster Plugin

Legends in Leaflet

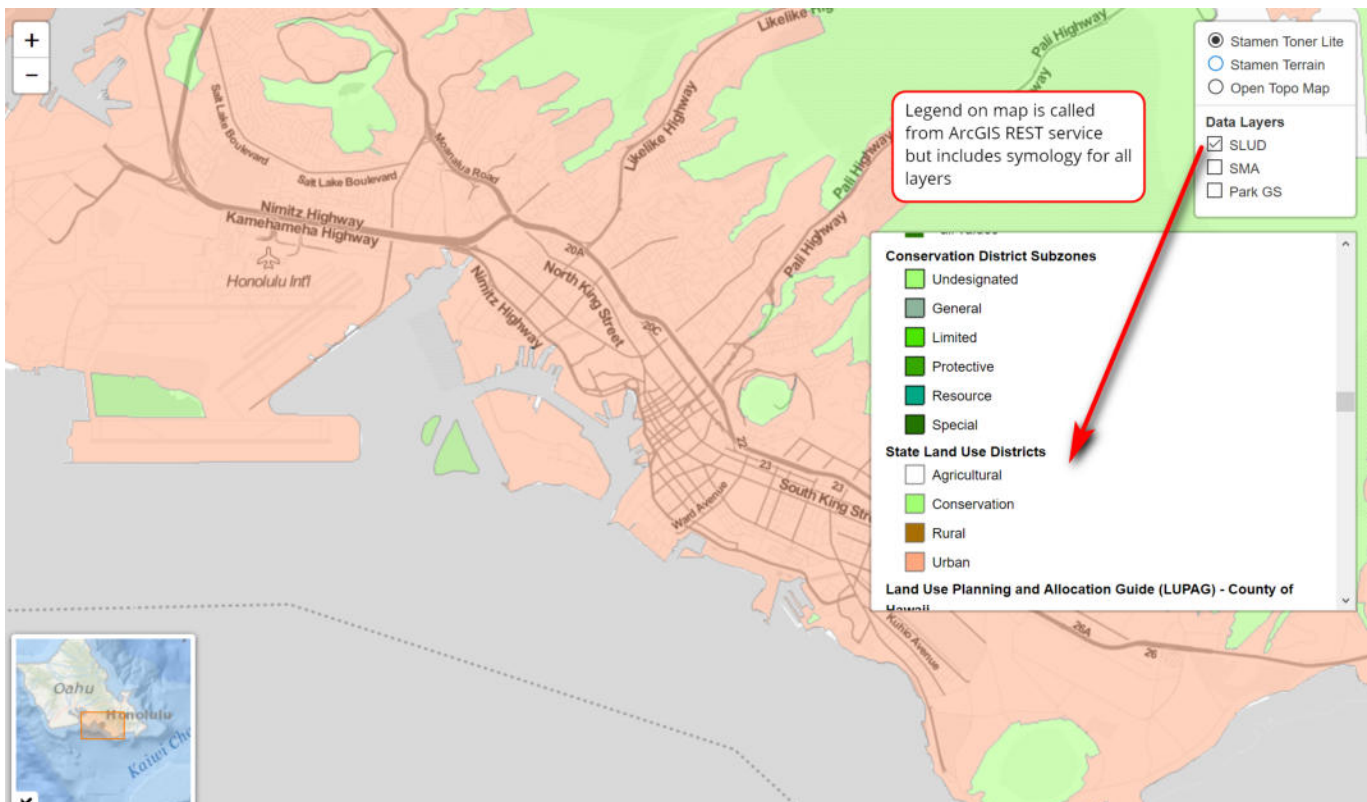
Legends are harder to implement than it seems. I didn't include legends on the demo web map, but I have tried adding legends from ArcGIS and GeoServer. See images below.

I've used this [ESRI Leaflet Legend plugin](#) to call a legend for a feature service. Below is a example code snippet for calling ArcGIS REST Legend. I didn't go any further than just trying to see if it'll work.

```

111 // Hawaii State GIS Program Map Server Layers
112
113 var SLUD = L.esri.dynamicMapLayer({
114   url: 'http://geodata.hawaii.gov/arcgis/rest/services/ParcelsZoning/MapServer',
115   layers: [20],
116   opacity: 0.5,
117   useCors: false
118 });
119
120 SLUD.bindPopup(function (error, featureCollection) {
121   if(error || featureCollection.features.length === 0) {
122     return false;
123   }
124   else {
125     return 'Code: ' + featureCollection.features[0].properties.ludcode;
126   }
127 });
128
129 SLUD.legend(function(error, legend){
130   if(!error) {
131     var html = '<ul>';
132     for(var i = 0, len = legend.layers.length; i < len; i++) {
133       html += '<li><strong>' + legend.layers[i].layerName + '</strong><ul>';
134       for(var j = 0, jj = legend.layers[i].legend.length; j < jj; j++){
135         html += L.Util.template('<li><span>{label}</span></li>', legend.layers[i].legend[j]);
136       }
137       html += '</ul></li>';
138     }
139     html += '</ul>';
140     document.getElementById('legend').innerHTML = html;
141   }
142 });

```



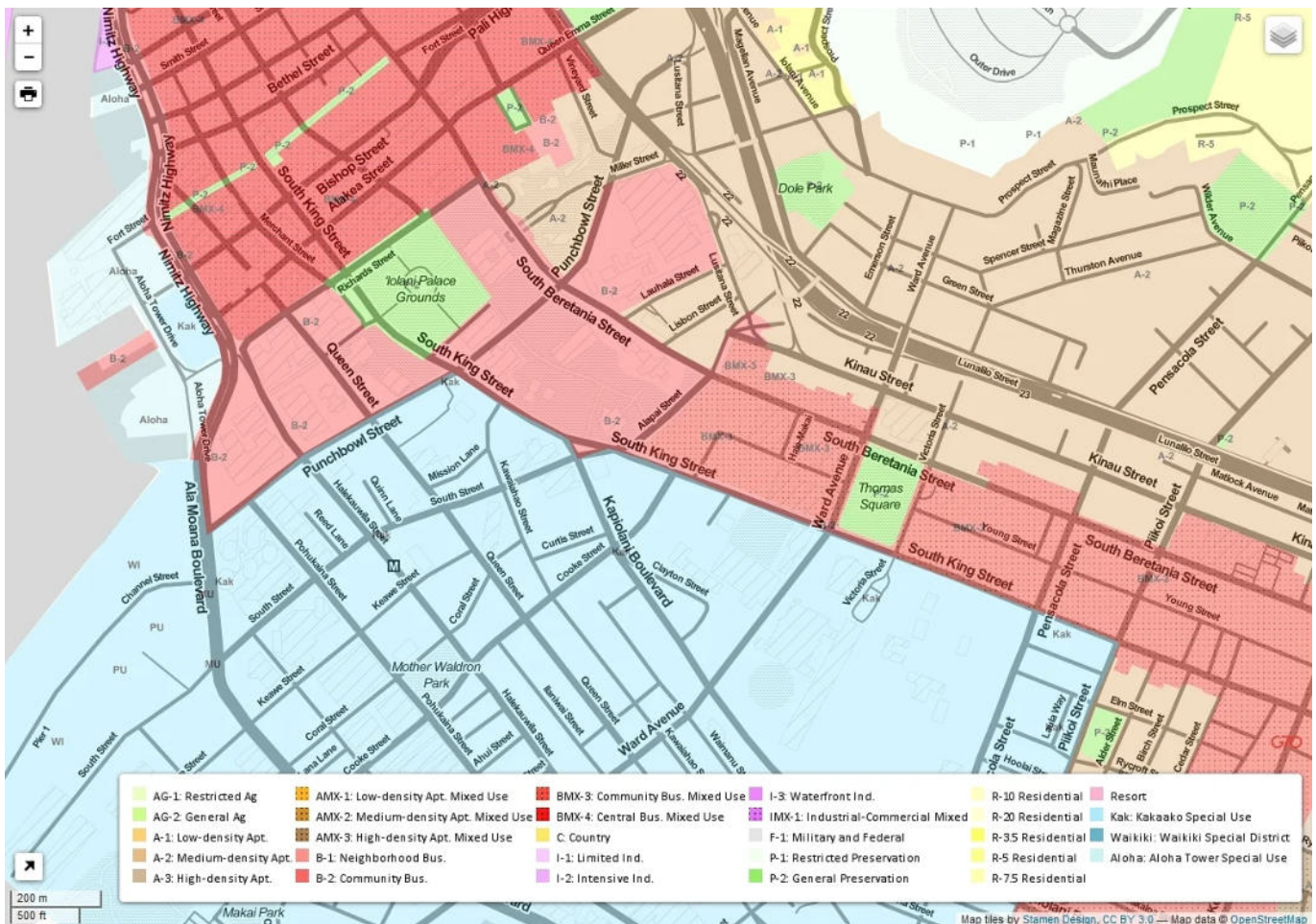
ArcGIS REST Legend displayed on map, but not exactly working the way I want it to.

Here is another code snippet for displaying a legend from a Geoserver WMS using the Get Legend Graphics request. This is calling from an in-house geoserver I set up.

```

109 // G70 layers from GeoServer
110 var Zoning = L.tileLayer.wms('http://[redacted]/geoserver/ows?', {
111   layers: 'OahuGIS:Zoning',
112   format: 'image/png',
113   transparent: true,
114   opacity: 0.5,
115   tiled: true
116 });
117
118 //Overlay grouped layers
119 var overlay = {
120   "Data Layers": {
121     "Zoning": Zoning
122   }
123 };
124
125 //Get Legend from GeoServer
126 var legend = L.control({position: 'bottomright'});
127 legend.onAdd = function (map) {
128
129   var div = L.DomUtil.create('div', 'info legend');
130
131   uri = 'http://[redacted]/geoserver/wms?
132   REQUEST=GetLegendGraphic&VERSION=1.3.0&FORMAT=image/png&WIDTH=20&HEIGHT=20&LAYER=OahuGIS:Zoning&legend_options
133   =fontName:Calibri;fontAntiAliasing:true;fontSize:12;layout:vertical;columns:6',
134
135   L.wmsLegend(uri);
136   return div;
137 }; legend.addTo(map);

```



Secured (https) and unsecured (Http) items

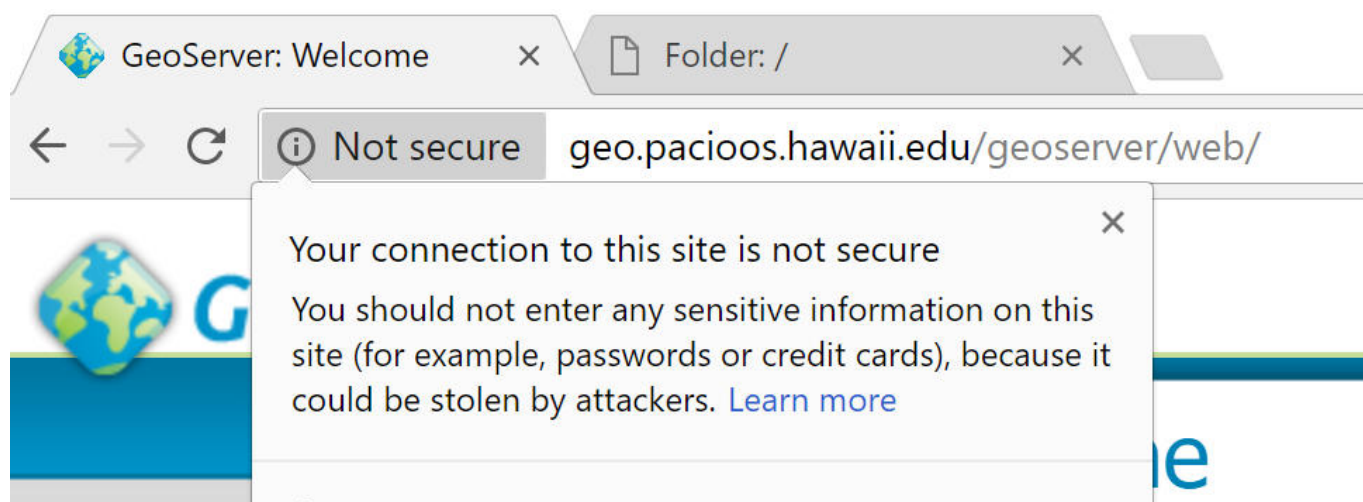
I have the demo leaflet web map hosted on GitHub - which is free to use if the repository is public, and by default will be secured. When I access the demo site, I get the warnings:

Internet Explorer: This site contains both secure and nonsecure items. Do you want to display the nonsecure item?

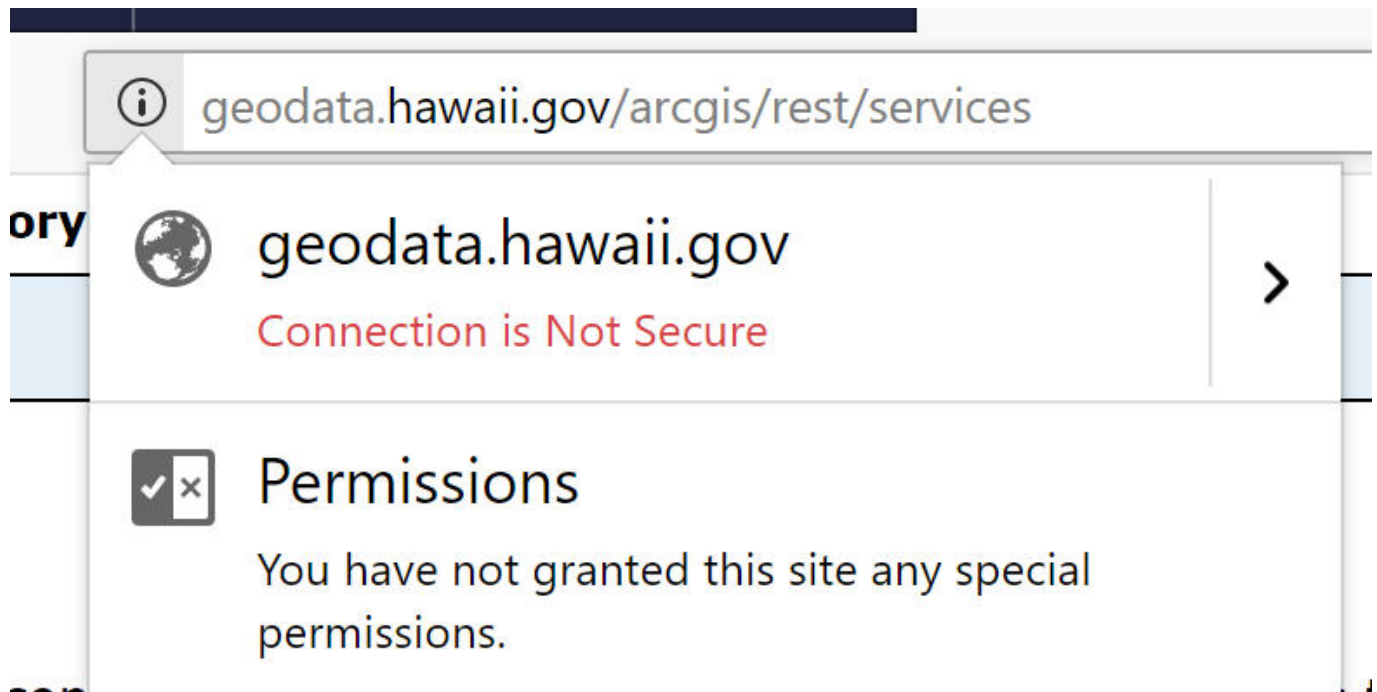
Firefox: Parts of this page are not secure (such as images)

Chrome: This page includes other resources which are not secure.

This warnings occurs when items on the following web site (frames, images and other objects) are referenced in a non-secure manner (http). For example the page code might include an image or code that pulls from a non-secure URL. Both the PacLOOS GeoServer and Hawaii Statewide GIS ArcGIS REST services unsecured - on http , so hence the warning messages. However, I noticed though that features from the PacLOOS unsecured geoserver will still display on the web map but not features on the ArcGIS REST service - snapshots are shown below.



PacLOOS Geoserver is unsecured (http)



Hawaii Statewide GIS ArcGIS REST service is unsecured (http)

Leaflet web map hosted on secured site (https) but referencing items from unsecured (http) sites (PacIOOS and Hawaii State GIS).

Solution is to use one of the following:

- Link to https explicitly:

Some websites will use both http and https. If you can access the items by using https then use that instead. Testing explicit https, I was able to access the Hawaii Statewide ArcGIS REST services (<https://geodata.hawaii.gov/arcgis/rest/services>) but not the PacIOOS Geoserver (<https://geo.pacioos.hawaii.edu/geoserver>)

- Use relative linking if on your own domain
- Use protocol relative linking to use images/items from other domains. This is the option I used in my Leaflet example, like so:

This works: `'//geodata.hawaii.gov/arcgis/rest/services/Terrestrial/MapServer/34'`

This does not work (at least not from GitHub https, but it works locally when I do live preview locally from Brackets): `//geo.pacioos.hawaii.edu/geoserver/wms?`

So I kept the URL for the PacIOOS geoserver WMS as http as it displayed on the web map.

That's it for this post. Thanks for reading.

Comments (2)

Newest First

gis consulting 5 years ago · 0 Likes

This is great information. Providing an example of the Leaflet web map is helpful. Thanks for sharing.

Bill Koerner 6 years ago · 0 Likes

I inherited an application that is currently running on Windows Server 2008 R2 and uses leaflet and leaflet.snap to allow a user to click an icon, and drag it along the displayed pipe centerline on a Google Earth map. The icon will snap to the pipe centerline even if the mouse location is not exactly on the centerline. I

have to migrate this application to Windows Server 2019, and suddenly the snap function does not work, it follows the location of the mouse location. Have you run into this?

< 3D DEM Visualization in QGIS
3.0

Bloggging Offline Using
Markdown and Brackets >

open.gis.lab



© 2026. open.gis.lab