Iowa State University
Extension and Outreach

GIS
Training and Services

**Geospatial Technology
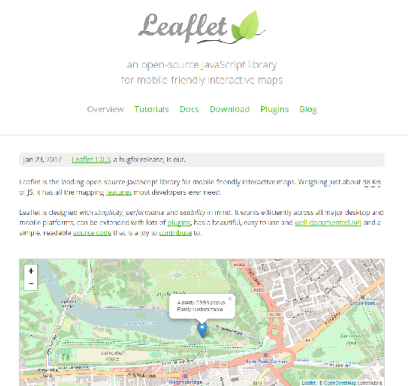Training Program**

Mapping API's: Leaflet - Getting Started

Welcome to the Essential ArcGIS Task Sheet Series. This series supplements the Iowa State University GIS Geospatial Technology Training Program short course series. The task sheets are designed to provide quick, easy instructions for performing mapping tasks.

Mapping API's allow developers to integrate Web content with a vendor's mapping services to provide rich, interactive Web applications. These applications can be customized to fit the specific dimensions of the website, display certain locations, utilize local content, provide customized tools, or allow for interchangeable basemap. This task sheet will take you through the first steps of creating a map using the Leaflet API and will serve as the foundation for future task sheets.

1. Introduction

a. There are several Mapping API's available, including Google Maps, Leaflet, Mapbox, Bing, ModestMaps, OpenLayers 3, ESRI ArcGIS, Polymaps and a few others. Each of these services have their own pros and cons. Some have a steep learning curve but provide an extensive set of features while others are easy to learn and can produce a simple slippy map for a website with minimal effort.

b. In all cases, the API, or Application Programming Interface is a specification that is used by the services software components to communicate with each other and describes how a task is to be done. In the case of the Mapping API's listed above, all of them utilize JavaScript.

2. What you need to complete this tutorial

a. A copy of the code used in this task sheet can be accessed from https://github.com/ISUEOGTP/GISTaskSheets/blob/master/Leaflet-Tutorials/helloLeaflet.htm.

b. In order to share the Web map you create on the Internet, you will need to host the files on a Web server. For the purposes for this task sheet you can just create the file on your local hard drive and open it with your Web browser or use an Online editor such as jsbin.com or jsfiddle. These Online editors are easy to use and include colored code syntax and other helpful features. *Hint: this would be the preferred method for beginners.*

3. Setting up the Web page for a map

a. Start with a basic HTML file containing **<!DOCTYPE html>**, **<html>**, **<head>**, **<style>** and **<body>** tags.

```
<link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
```

b. The Leaflet API requires that you link to both the Leaflet CSS and the Leaflet JavaScript Library. The CSS file can be linked to as a relative link and should appear in the document **<head>**.

c. After linking to the leaflet CSS, add a **<style>** tag defining the size and width for an element with a CSS id selector of **#map**. *Note: more information on CSS selectors can be found here - https://www.w3schools.com/cssref/css_selectors.asp*

```
<style>
    #map {
        height: 400px;
        width: 400px;
        margin: 0;
        padding: 0;
    }
</style>
```

d. In the HTML **<body>** tag add a new **div** and assign it an **id = "map"**. This is the div that the Leaflet map will be displayed in.

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

e. Before the closing **</body>** tag add a **script src** link to the leaflet JavaScript library. The libarary can be hosted locally or retrieved from http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js. *Note: this is no longer the most up to date library version. Check http://leafletjs.com/ for updates.*

```
<script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
```

f. Next, add a **<script>** and closing **</script>** tag before the closing **</body>** tag.

```
<script >
  // Leaflet JavaScript goes here
</script>
```

4. Adding and customizing the map display location

a. Under the **<script>** tag add the line of code shown in the image below. This will initialize the variable **map**. Change the latitude and longitude values of **[43, -93]** to the coordinates you wish. The value **10**, at the end of the line, is the zoom level. The higher the number the greater the level of zoom.

```
<script >
  var map = L.map('map').setView([43, -93], 10);

  // add an OpenStreetMap tile layer
  L.tileLayer('http://{s}.title.osm.org/{z}/{x}/{y}.png', {
    attribution: 'The map contributors',
    maxZoom: 18
  }).addTo(map);
</script>
```

b. The next step is to define the map tile layer (basemap) used and add it to the map. For this example the OpenStreetMap tile layer is used and shown in the code block above. Add this code following the initialization of the map variable.

c. The map tile layer contains two options. The first is map **attribution** that can be used to cite the map data source and copyright information. The second is **maxZoom** option, users will not be allowed to zoom into the map any closer than this level selected.

d. Save the file as **helloLeaflet.htm** and open the map in a Web browser. Your web map should display! Once it does, go back and change the zoom level and map coordinates to see the impact. *Note: right-click on the html file and select open with (your web browser).*

5. Additional Resources

a. When learning how to develop Leaflet maps jsbin.com is a highly recommended tool to use. The Online text editor automatically updates the map display as you work. Thus as soon as you change the coordinates or zoom, the map updates.

b. Additional Leaflet tutorials and examples found on the Leaflet website: http://leafletjs.com/examples.html.

Contact:

Bailey Hanson bahanson@iastate.edu, 515-520-1436 or Professor Christopher J. Seeger, ASLA, GISP cjseeger@iastate.edu, 515-509-0651 for more information about the Geospatial Technology Program. This task sheet and more are available at www.extension.iastate.edu/communities/gis