11-2016

Mapping API’s: Leaflet - Pre-Built Fill Patterns (Basic)

Bailey A. Hanson  
Iowa State University, bahanson@iastate.edu

Christopher J. Seeger  
Iowa State University, cjseeger@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/extension_pubs

Part of the Architectural Technology Commons, Landscape Architecture Commons, Other Architecture Commons, and the Urban, Community and Regional Planning Commons

Recommended Citation
Hanson, Bailey A. and Seeger, Christopher J., "Mapping API's: Leaflet - Pre-Built Fill Patterns (Basic)" (2016). Extension and Outreach Publications. 129.  
http://lib.dr.iastate.edu/extension_pubs/129

Iowa State University Extension and Outreach publications in the Iowa State University Digital Repository are made available for historical purposes only. Users are hereby notified that the content may be inaccurate, out of date, incomplete and/or may not meet the needs and requirements of the user. Users should make their own assessment of the information and whether it is suitable for their intended purpose. For current publications and information from Iowa State University Extension and Outreach, please visit http://www.extension.iastate.edu.
Mapping API’s: Leaflet - Pre-Built Fill Patterns (Basic)

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

Displaying multiple thematic polygon layers in a Leaflet.js map is often accomplished by providing a layer control that allows the user to toggle between overlays. However, in some situations it may be necessary to display two overlapping thematic maps at the same time. Unfortunately, this can create a situation where one polygon blocks the underlying feature. One method that can be utilized to visualize both layer properties is to fill the bottom layer with color and the top with a pattern fill. This task sheet will demonstrate how to use the leaflet.pattern plugin to add a basic line pattern fill to style a feature.

1. Leaflet - Fill Pattern Setup

a. Start with the basic Leaflet map file and add in the `<head>` a link to the leaflet.pattern plugin. This plugin is available at https://github.com/teastman/Leaflet.pattern.

b. Before drawing any features that will use a pattern, provide a map constructor and then define and construct the pattern. To do this add the following code.

```javascript
var myPattern = new L.StripePattern({});
myPattern.addTo(map);
```

c. After creating the pattern, add two yellow circles to the map. The following example will create two overlapping yellow circles with a red outline at the provided coordinates. (Figure 2)

```javascript
var circle1 = new L.Circle([42.0, -93.79], 750, {
    color: 'red'
    fillColor: 'yellow'
    fillOpacity: 0.75
});
circle1.addTo(map);

var circle2 = new L.Circle([42.0, -93.80], 750, {
    color: 'red'
    fillColor: 'yellow'
    fillOpacity: 0.75
});
circle2.addTo(map);
```
d. To access the fill pattern, on the second circle comment out the `fillColor` and add a `fillPattern` option that references the pattern `myPattern`.

e. The circles should now be filled with the default black and transparent horizontal line pattern. The pattern will be displayed using default settings. (Figure 3)

f. This pattern can be modified by specifying the maximum pattern width or height and adding an angle, weight, color and opacity options to the pattern constructor. For the purpose of this basic tutorial you will focus on the last four options, but note the default width and height of 8.

```javascript
var myPattern = new L.StripePattern({
    angle: 45,
    weight: 2,
    color: 'blue',
    opacity: 0.5,
});
myPattern.addTo(map);
```

g. Modify the pattern constructor to look as follows. Note that the outline of the circle is not controlled by the pattern options and is instead part of the `circle options`. In this case, it has `color: red`. (Figure 4)

```javascript
var myPattern = new L.StripePattern({
    angle: 15,
    weight: 7,
    color: 'black',
    opacity: 0.75,
});
myPattern.addTo(map);
```

h. Continue to modify the settings to modify the pattern. Note that the maximum weight is currently 8. Modifying the width and changing the color, weight, and opacity of the space between the line is covered in Mapping API’s: Leaflet - Pre-Built Fill Patterns (Advanced) PM2082-16v. (Figure 5)

```javascript
var myPattern = new L.StripePattern({
    angle: 15,
    weight: 7,
    color: 'black',
    opacity: 0.75,
});
myPattern.addTo(map);
```