Mapping API’s: Leaflet - Creating Polylines and Polygons

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
http://lib.dr.iastate.edu/extension_pubs/171

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 Users are hereby notified that the content may be inaccurate, out of date, incomplete and/or may not meet the needs and requirements of the user.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 - Creating Polylines and Polygons

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.

The code for this task sheet (polyLeaflet.html) and the previous task sheets can be found on the ISU Geospatial Technology Program GitHub page at https://github.com/ISUEOGTP/GISTaskSheets. This task sheet demonstrates how to add simple polyline and polygon overlays to a Leaflet map and then modify the options to change how these overlays display.

1. Introduction

a. First, you will need to start with a basic Leaflet map setup. Reference the task sheet: Mapping API’s: Leaflet - Getting Started PM2082-14r to learn how to get this set up, or get the starter code from our GitHub page at https://github.com/ISUEOGTP/GISTaskSheets/blob/master/Leaflet-Tutorials/helloLeaflet.htm.

2. Creating Polylines

a. Start by using the basic Leaflet setup to display a map and adjust the zoom and map center as desired.

b. The first step is to instantiate a polyline or polygon object and to then provide it as an array of geographical coordinates and options describing how to display the object.

c. A simple example of this is as follows and can be added to the end of the script after the map is created. Note: the latlngs are currently an empty variable.

```
var polyline = L.polyline(latlngs, {color: 'blue'}).addTo(map);
```

d. Because the latlngs variable has not been defined, no line will appear on the map. Latitudes and longitudes can be added using a variety of syntaxes. Three different syntax examples are shown here and all examples have three pairs of coordinates passed that result in a V shaped polyline being drawn.

e. In the example shown to the right, the coordinates are added directly to the constructor. However, this can make your code messy if you have a lot of data points.

```
var polyline = L.polyline([ [43.01, -93.1], [43.1, -93.18], [43.05, -93.7] ], {color: 'blue'}).addTo(map);
```
3. Creating a Polygon

a. Switching the polyline to a polygon is very simple, just change \texttt{L.polyline} to \texttt{L.polygon}. Note that the first word \texttt{polyline} is a variable name and can be anything you wish e.g. \texttt{myPolygon}.

\begin{verbatim}
  var polygon = L.polygon(latlngs, {color: 'blue'}).addTo(map);
\end{verbatim}

b. Polygons do not require any additional data points to close the polygons they just draw a line between the last and first point and fill the defined area with the specified fill color.

c. Polylines and polygons can use a variety of options as defined by the Path class. As with the coordinates, options are best placed in a variable that is put before the polyline or polygon constructor. Options shown to the right. \textbf{Note:} additional polylines and polygons can be created in a map and make use of the same options, however they would need to connect to a unique data set of points and coordinates.

\begin{verbatim}
  var pathOptions = {
    stroke: true, //stroke visibility
    weight: 1,  //stroke thickness
    color: 'blue', //stroke color
    opacity: 0.5, //stroke opacity
    fill: true,   //polygon fill
    fillColor: 'yellow', //fill color
    fillOpacity: 0.5, //fill opacity
    //other options
    dashArray: '5,3' //dash pattern used by stroke
  };
  var polygon = L.polygon(latlngs, pathOptions).addTo(map);
\end{verbatim}