

Development document for:

Lost Rivers of Toronto "Ashbridges Walks Maptour" - "Story map" web-maps

Using: Leaflet + geoJSON data

Link to webmap: http://mercator.geog.utoronto.ca/georia/lostrivers/leaflet-storymap_webtest/AshbridgeBay.html

Link to Code posted on github: <https://github.com/geohist-ca/lostrivers>

REQUIREMENTS: A map with numerically ordered point symbols linked to a story panel containing corresponding text and images. Suitable for Lost Rivers walking tour data for 3 walking tours: Ashbridge's Bay Watershed Walk, Ashbridge's Creek Walk and Tomlin's Creek Walk. Historical map image showing original river pattern should be incorporated with adjustable transparency.

Credits: Lost Rivers of Toronto Project (John Wilson), Nathan Ng, University of Toronto Map and Data Library, Canadian Historical GIS Partnership Development project, GIS and Cartography Office Dept. of Geography University of Toronto. Programming assistance: Jonathan Critchley.

Approach: Decision made to use a Leaflet + jQuery story map template

Web-based research into various options discovered a template that is available that has the type of functionality that is required for the project:

<https://github.com/JackDougherty/leaflet-storymap>.

Creates a map that follows your text. Annotate each paragraph and place a map alongside it. Then you can zoom/pan/add marker etc to the map as the reader reads through the text.

Demo: <http://jackdougherty.github.io/leaflet-storymap/index.html>

Note: This github site also has a "Compare with" section listing other "story map" programming options.

Functionality

Mapping functionality: Leaflet.js

Interface functionality: JQuery (getJSON) (scrollTop)

JQuery UI: Image transparency slider

Bootstrap

Historical background map image: QTiles

Map Data:

Overlay Mapping data: Points and lines for map tours: geoJSON

Base map: Humanitarian OpenStreetMap

Historical background map: Georeferenced map image tiled using QGIS Qtiles

PRODUCTION NOTES

OVERLAY MAPPING DATA

ArcGIS desktop

ArcMap was used to create the points and lines for the map tours. Any GIS software (eg. QGIS) could be used for this purpose.

Conversion to GeoJSON

GeoJSON (geoJSON.org, JSON.org) is a standardized text-based GIS file format for encoding a variety of geographic data structures. It has become a standard for use in Leaflet and other light-weight web-mapping applications. You can generate a geoJSON file from a .CSV using the OS tool provided by Mapbox at: <http://geojson.io>. That is the method used by the story map template referenced above, and they provide a .CSV template they have used. The CSV that is saved is using Geographic coordinates (latitude and longitude) in the WGS84 CRS.

ArcGIS can export geoJSON directly from a shapefile or geodatabase feature class. This worked for the points file. However, the ArcGIS line file directly exported to geoJSON will not appear when used with Leaflet. To create a generic geojson the line file was imported to QGIS and then re-exported as a GeoJSON layer. There are many alternative ways of getting around this issue, including brute force text editing, online Open Source converters, or commercial data conversion software like FME.

Point and Line Overlay Layers

All the geoJSON files used for Point and Line overlay layers are put into the local "data" subdirectory and called using the JQuery getJSON() method <http://api.jquery.com/jquery.getjson/>. See comments in htm.

Ashbridge's Bay Watershed Walk Colour scheme: Orange

Point file: AB_Watershed_Points_WGS.geojson

Line file: AB_Watershed_Path_WGS_qgis.geojson

Ashbridge's Creek Walk Colour scheme: Green

Point file: AB_Creek_Points_WGS.geojson

Line file: AB_Creek_Path_WGS_qgis.geojson

Tomlin Creek Walk Colour scheme: Purple

Point file: TomlinCreekWalk_Points_WGS.geojson

Line file: TomlinCreekWalk_Path_WGS_qgis.geojson

Map-specific customization: main path and two other paths appearing below it

The layers are graphically laid on top of each other in the order that they are loaded in the map. Therefore the path that is the focus of the specific tour map is loaded in last so that the other two do not cover it. This is accomplished by declaring the other two supplementary paths layers first in the code and declaring the main path layer last. The supplementary paths are also given 60% transparency to put them below the main path in the visual hierarchy ("opacity": 0.4).

Reference Line layers

The following two layers were included for reference and can be toggled on or off in the Layers Control.

Rivers plotted by Lost Rivers Project - historical river courses for Ashbridge's Bay area provided by John Wilson of the Lost Rivers Project

Line file: lost_rivers_john_fc.geojson

Shoreline 1882 (McMurrich) - Tracing of shoreline of historical base map McMurrich 1882 (see below)

Line file: shoreline_fc.geojson

POINT FEATURES AND LEAFLET TEMPLATE

Attributes for point features (map tour stops)

The template runs through the features in the order that they appear in the points file, sequencing them in the right hand panel this way. Ensure that they are ordered in the correct ascending order by NUMBER in the points file in the sequence that you want them to appear. The following fields are required.

NUMBER - Number field for ordering required to be in numeric order one after another

NAME – Title of feature

DESCRIPTION – Main text description

PIC_URL – URL for picture of feature

SHORT_DESC – Image caption

DESC2 – Second text description if required

PIC_URL_2 - second picture URL

We also added a corresponding number in the NAME (title) field so that it was easier to visually relate the contents in the side panel to the corresponding point on the map.

Added “1 -” to “Holy Cross Roman Catholic Church” resulting in “1 - Holy Cross Roman Catholic Church”

Adding quotation marks in the titles of features

Some of the features' titles required quotation marks. In order for them to appear in the browser and so that they did not break any code they had to be reformatted in the geojson file. Quotation marks (and most special characters) needed to have a backslash before them to be correctly interpreted:

```
\ "Sand dune\"
```

Customization of template for second image

To satisfy the requirements of the Lost Rivers Project the story map template needed to be customized to accommodate a second image and description for some stops, where desired. The fields DESC2 and PIC_URL_2 were created and populated with the URL and secondary text wherever they existed.

The javascript also had to be updated to read and append the second image. We used an "If" statement: If the json feature has a property 'pic_url_2' then display contents.

```
if (feature.properties['PIC_URL_2']){
    var image2 = $('<img>', {
        src: feature.properties['PIC_URL_2'],
    });
    var imgHolder2 = $('<div></div>', {
        class: 'img-holder'
    });
    imgHolder2.append(image2);
};
```

ISSUES WITH LEAFLET TEMPLATE

There were some issues with the template that caused it not to work correctly. The issues were present in the example:

<http://jackdougherty.github.io/leaflet-storymap/index.html>

Line layer does not update when using flyTo (scrolling down list)

Line layers appear to not update until after the map view has adjusted when either clicking on a point or scrolling down the list. Upon clicking on a point or scrolling down the list the Leaflet flyTo method is used to adjust the map viewer and centre on a point of interest. When the flyTo method is animated the map view will readjust, however for a short period the line layers will appear to stay in the same location in the viewer. The line layer will only update after the map viewer has finished the flyTo animation. This is a known issue with GeoJSON lines when using flyTo animation in Leaflet.

<https://gis.stackexchange.com/questions/168781/leaflet-vector-layers-not-rendering-during-flyto-animation>

Issues with right hand panel

An issue is that the right hand panel does not scroll all the way down enough to select the bottom container. This means that the item at the very end of the list does not get selected.

To resolve this we increased the height of the .space-at-the-bottom div to 600px. This ensures that the bottom item gets pushed all the way up allowing it to get selected.

Clicking on a point to center it on the map and display its associated data in the right div was also not working correctly. It would pan to center the point on the map and scroll to the associated container div but it would not set that container div class to inFocus. Resolved by updating the function that handled the behavior, adding "+ 1". This added an extra pixel that allowed the div to correctly update to inFocus based on position.

```
$("#div#contents").animate({scrollTop: areaTop + 1 + "px"});
```

There was also an issue that we encountered where the height of the divs in the side panel that contain the attribute was being calculated before images were loaded. This would cause an issue where the side panel was not synced up to the point being displayed and the div was being turned in focus prematurely. To resolve this the function updateScrollTop() was created. The function contains the code to calculate the height of the divs. The function is called in two events:

Initialize the scrolltop function when all images finish loading:

```
$("#img").on("load", updateScrollTop);
```

Initialize the scrolltop function if the browser window changes size:

```
window.addEventListener('resize', updateScrollTop);
```

Updating the template for mobile devices

The template covered the map with the scrolling side panel with a transparency. This made it difficult to see anything on the map when viewing from a mobile device.

The template was updated with responsive design @media queries to change the page layout on mobile devices from tablets to phones.

<https://v4-alpha.getbootstrap.com/layout/overview/#responsive-breakpoints>

The first @media (max-width: 991px) forces the navigation buttons to collapse on a resolution smaller than 991px. The issue was that on tablet devices the buttons were too large and started to be pushed down.

The second @media (max-width: 768px) deals primarily with the map and scrollable information div placements: map on top, scrollable area in the bottom.

BASE MAPS

Reference Base Map: Selected from Leaflet Providers' base maps

<https://leaflet-extras.github.io/leaflet-providers/preview/>

using Humanitarian OpenStreetMap tiles to provide contrast with overlay tiles

```
var OpenStreetMap_HOT = L.tileLayer('http://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png',
    maxZoom: 19,
    attribution: '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>,  
    Tiles courtesy of <a href="http://hot.openstreetmap.org/" target="_blank">Humanitarian  
    OpenStreetMap Team</a>'  
}).addTo(map);;
```

Historical background map georeferenced raster overlay

The original image was obtained from the Historical Maps of Toronto website by Nathan Ng, provided courtesy of Toronto Public Library:

<http://oldtorontomaps.blogspot.ca/2013/02/1882-city-engineers-office-plan-of-city.html>

The image was georeferenced in ArcMap. It was then brought into QGIS for generating web tiles, using the QTiles plugin. This was done to test the efficacy of this method.

http://www.qgistutorials.com/en/docs/creating_basemaps_with_qtiles.html

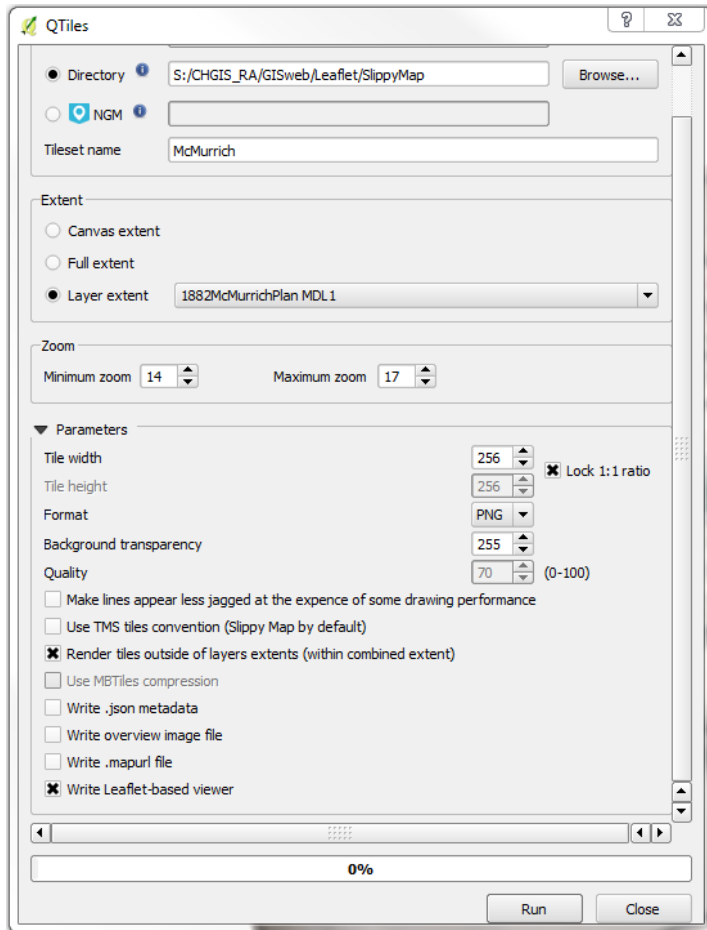
<https://plugins.qgis.org/plugins/qtiles/>

Generated tiles using QTiles plugin:

Min zoom: 14

Max zoom: 17

See image next page



ISSUES WITH TILES

Quality of image

QTiles rendered the georeferenced image tiles easily. The available image was somewhat coarse and degraded from the original artifact through a lossy compression process. QTiles essentially maintained the quality that was in the original, when generated with the default 70% Quality value.

Areas that have no tiles

Some browsers will display missing image icon in areas outside of the tile layer coverage or if zoomed to a level that tiles were not generated for. Use the image provided by ESRI at the link below and the `errorTileUrl` parameter to prevent the icon from appearing.

<http://support.esri.com/technical-article/000010678>

<http://leafletjs.com/reference-1.0.2.html#tilelayer-errortileurl>

(Alternative methods not used - Loading the image as an overlay

<http://leafletjs.com/reference-1.0.3.html#imageoverlay>

<http://leafletjs.com/reference-1.0.3.html#latlngbounds>)

INTERFACE FUNCTIONALITY

jQuery Slider - Transparency slider for historical base map

Created a jQuery slider tied to the tiles generated using QTile in QGIS to control tile opacity.

<http://api.jqueryui.com/slider/>

Initial placement of the handle on the slider matching the filter specified in the layer properties, setting here 50% opacity:

value: 0.5,

Set the minimum slider value:

min: 0,

Set the maximum slider value:

max: 1,

Provide the number of steps/intervals required between the max and min values:

step: 0.1,

Navigation bar

Using Bootstrap (<http://getbootstrap.com/>)

The Bootstrap navigation bar is a navigation header that can be placed at the top of a page to add branding using a logo image and a navigation menu to link to other pages on a site. With Bootstrap the navigation bar is responsive and can extend or collapse depending on the screen size, making it functional and accessible on mobile devices with little effort. A standard navigation bar is created with `<nav class="navbar navbar-default">` however a template was used to create a navigation bar with a increased logo size and navigation buttons on the right. The template used was retrieved from Example 2 on the following codepen.io page:

<https://codepen.io/bootstrapped/pen/KwYGwq>

Only the HTML and CSS code identified as Example 2 are required from the codepen.io example. The code was further modified to assign unique colour to each button, and button text was edited to reflect the purpose of the button with hyperlinks setup to link to corresponding pages.

GENERAL ISSUES

Opening Leaflet .html map documents when stored locally on a computer

If you are opening the map html files using Leaflet stored locally on a computer the geojson layers will not load when using Chrome. A solution to this is to setup a local server on your computer and view the files. However, much the easiest and quickest solution is to use Firefox.