Géo Hist .ca
Canadian Historical Geographic Information Systems Partnership
Geohistory-Géohistoire Canada
Partenariat canadien en systèmes d'information géographique historiques

Development document for:
Lost Rivers of Toronto "Disappearing Rivers" web-map
Using: Leaflet + geoJSON data + Boundless Geoserver
Link to webmap: http://mercator.geog.utoronto.ca/georia/lostrivers/leaflet-storymap_webtest/disappearing.html
Link to Code posted on github: https://github.com/geohist-ca/lostrivers

REQUIREMENTS: A map of the city of Toronto showing the original hydrographic network of Toronto plotted from historical maps as early as 1802, and how those streams disappeared over time as they were buried for purposes of development, to arrive at the interrupted surface drainage pattern of today. Rivers should be symbolized by year last recorded on a map and a time slider UI tool used to show them disappearing over time. Historical map image showing surface geology for context should be incorporated with adjustable transparency.

Credits: Lost Rivers of Toronto Project, 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.

BACKGROUND REFERENCE - sites researched for approaches and ideas

https://github.com/dwilhelm89/LeafletSlider/blob/master/examples/epoch.html

http://www.digital-geography.com/filter-leaflet-maps-slider/#.WUBD8-vythE

https://blogs.kent.ac.uk/websolutions/2015/01/29/filtering-map-markers-with-leaflet-js-a-brief-technical-overview/

Filtering layers: http://duspviz.mit.edu/web-map-workshop/leaflet-javascript-interactions/

Javascript Comparison and Logical Operators: https://www.w3schools.com/js/js_comparisons.asp

Approach: Decision made to use Leaflet + jQueryUI slider for the Disappearing Rivers functionality, and to use Boundless Geoserver to serve base reference layers, including historical map image.

Functionality
Mapping functionality: Leaflet.js, Geoserver, Proj4Leaflet.js
Interface functionality: JQuery (getJSON)
JQuery UI: Time slider, Image transparency slider
Bootstrap
Historical background map image: Geoserver

Map Data:
Overlay Mapping data: Points and lines for Disappearing Rivers: geoJSON
Background Mapping data: Current rivers, Watersheds, Shoreline 1882: Group layer as Geoserver WMS
Base map: Esri World Topo
Historical background map: Georeferenced map image served as Geoserver WMS

Note re: Disappearing Rivers data compilation
The Feature layer appearing here as Disappearing Rivers is the result of a long process of research and compilation which should be credited primarily to Helen Mills of the Lost Rivers of Toronto Project, built upon the original efforts of Peter Hare, founder of Lost Rivers, and other volunteers. The first attempt at comprehensively mapping these streams resulted in a poster map: "Lost Rivers of the Downsview Lands and surrounding Toronto Area" produced by Helen Mills and Daniel Spring. The graphics files from this project were georeferenced and converted to ArcGIS shapefiles. The Map and Data Library (MDL) of the University of Toronto edited and augmented these files with information about the dates individual sections of streams were last seen on historical maps of Toronto, from various map collections, including those of Nathan Ng and the MDL itself. With the help of Mills and Wilson, further compilation and editing took place in the GIS and Cartography Office of University of Toronto. The result is the current "Best Interpreted" version of the original hydrographic network of Toronto. There are many places remaining where this network could be enhanced and improved. It is hoped that the usefulness of the current version will spur and support ongoing efforts to expand and refine these data.

PRODUCTION NOTES

OVERLAY MAPPING DATA

ArcGIS desktop

ArcMap was used to create and edit the lines for the Disappearing Rivers. 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.

LINE FEATURES AND LEAFLET SYMBOLIZATION

Line Overlay Layers

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

Disappearing Rivers
Colour scheme: Colour-coded by year last seen on map
Line file: BI_Lost_Rivers_20170705_WAMS_qgis.geojson

Colour coding is based on layer attributes and declared breaks, using methodology provided in the leaflet example: http://leafletjs.com/examples/choropleth/, using function getColor(d)
Breaks are declared based on the value for field "lastNTS" (see below)

Attributes for line features (Disappearing Rivers)

Eight attribute fields are populated in this file but the only one used for symbolization is lastNTS:

lastNTS - Year last seen on map   (Note: Integer, year - usually a NTS National Topographic System map)

STARTDATE - Set for all at 1/1/1800 (Note: formatted as Date field)

LASTDATE - Set for each line segment at 1/1/yyyy (Note: formatted as Date field, yyyy same as lastNTS)

Time slider for Disappearing Rivers (also see INTERFACE FUNCTIONALITY section below)

The Time slider was adapted from the generic jQueryUI slider:
https://api.jqueryui.com/slider/#option-range

by using the Leaflet geojson filter option:
http://leafletjs.com/reference-1.0.2.html#geojson-filter

Initially the rivers layer is added to the map:

```
$.getJSON('data/BI_Lost_Rivers_20170705_WAMS_qgis.geojson', function(json)
        rivers = L.Proj.geoJson(json, {
```

Note: See below ***Proj4Leaflet to support projection...*** for use of L.Proj.geojson

This slider does not use set breaks and required custom values. The array "valMap" is set up to contain the custom intervals for the slider, (i.e every year value for the field lastNTS). These will be used to set up the "stops" on the slider, and also to query the layer for the purposes of the filter. The slider and map filter are initialized at the first year in the time array (i.e. lastNTS=1802).

A change event captures when the the slider handle HAS MOVED location: change: function(event, ui). This will trigger the filter function which selects values greater than or equal to the updated year.

```
filter: function (feature, layer) {
         return feature.properties.lastNTS >= valMap[ui.value];
```

Note: Dynamically changing the filter option will have effect only on newly added data. It will not re-evaluate already included features. This therefore requires each time the slider registers a change event the layer is cleared from the map:

```
        rivers.clearLayers();
```

Then the rivers layer is added once again to the map, with the filter set using the slider ui value.

```
        $.getJSON('data/BI_Lost_Rivers_20170705_WAMS_qgis.geojson', function(json) {
                rivers = L.Proj.geoJson(json, {
                style: style,
                filter: function (feature, layer) {
                return feature.properties.lastNTS >= valMap[ui.value];
                }
                }).addTo(map);
                });
```

Proj4Leaflet to support projection of rivers in geoJSON in Web Mercator CRS

The disappearing rivers layer was compiled in Web Mercator projection and the decision was made not to re-project it. Leaflet expects Geographic (latitude, longitude) coordinate for geoJSON files so needs projection support to consume data in other projections, such as Web Mercator (EPSG:3857).
See: Proj4Leaflet (http://kartena.github.io/Proj4Leaflet/) which adapts the generic Proj4 projections conversion library (http://proj4js.org/)

The Proj4Leaflet files were downloaded and the path specified within the disappearing.html file

```
<!-- Proj4Leaflet to support Web Mercator CRS -->
<script src="Proj4Leaflet-leaflet-proj-refactor/lib/proj4-compressed.js"></script>
<script src="Proj4Leaflet-leaflet-proj-refactor/src/proj4leaflet.js"></script>
```

To declare a layer using the Proj4Leaflet geojson call use

```
rivers = L.Proj.geoJson
```

instead of the standard leaflet declaration:

```
rivers = L.geoJson
```

GEOSERVER

Background WMS line and historical map image from Geoserver

Geoserver is part of the Boundless/OpenGeo suite of software.
https://boundlessgeo.com/

Boundless OpenGeo Suite is built upon these individual components: PostGIS, GeoServer, GeoWebCache and OpenLayers. They now offer a Desktop version as well, which has not been tested.

The decision was made to use geoJSON for the functional Lost Rivers overlay, as detailed above, but to use Geoserver installation (OpenGeo Suite 4.6.1) on the University of Toronto Geography Department server for background layers as Web Map Services (WMS). There is a group layer for the Lost Rivers base mapping, plus a tiled image layer for the historical map. The reason was to demonstrate utility of Geoserver WMS for background mapping. The tiles generated through Geoserver also did a good job of retaining resolution and detail.

Loading Geoserver WMS layers into Leaflet map

Documentation to load layer group from Geoserver

http://leafletjs.com/examples/wms/wms.html

http://leafletjs.com/reference-1.0.3.html#tilelayer-wms

Need to set transparent option to true otherwise the tile background is white

http://leafletjs.com/reference-1.0.3.html#tilelayer-wms-transparent

Need to set format to “image/png” for layers with transparency

The Geoserver WMS layers are brought in as Leaflet layers in the final html files (disappearing.html) as Leaflet tilelayers.

Background Mapping data: Current rivers, Watersheds, Shoreline 1882

WMS group layer for background mapping features containing rivers that persist to 2017, watershed areas and shoreline ca 1882 traced from McMurrich map. These are brought in as a group WMS, but could be listed individually if layer manipulation is required.

```
        var wmsLayer =
L.tileLayer.wms('http://opengeo.geog.utoronto.ca:8080/geoserver/lostrivers_wksp/wms', {
                layers: 'lostrivers_wksp:Lost_Rivers_base_layer_group',
                transparent: true,
                format: 'image/png'
        }).addTo(map);
```

Sources:
Current rivers persisting to 2017: City of Toronto / Open Data / Data catalogue/ Toronto Centreline (TCL)
https://www1.toronto.ca/wps/portal/contentonly?vgnextoid=9acb5f9cd70bb210VgnVCM1000003dd60f89RCRD&vgnextchannel=1a66e03bb8d1e310VgnVCM10000071d60f89RCRD
Watersheds: Toronto Region Conservation Authority, Map and Data Library University of Toronto
https://mdl.library.utoronto.ca/collections/geospatial-data/toronto-region-conservation-authority-data
Shoreline 1882 (McMurrich): GIS and Cartography Office, Dept of Geography, University of Toronto
Traced from http://oldtorontomaps.blogspot.ca/2013/02/1882-city-engineers-office-plan-of-city.html

Historical background map (see sources below)

WMS layer used to load georeferenced historical background geological map image

```
        var mytile =
L.tileLayer.wms('http://opengeo.geog.utoronto.ca:8080/geoserver/lostrivers_wksp/wms', {
                layers: 'lostrivers_wksp:G_3501_C5_Coleman_1932_rebuilt',
                transparent: true,
                opacity: 0.4,
                format: 'image/png'
        }).addTo(map);
```

BASE MAPS

Reference Base Map: Esri World Topo map

Selected Esri World Topo Map as the most useful option to provide topographic reference for drainage pattern and changes over time, as elevation contours extremely helpful.

```
var Esri_WorldTopoMap =
        L.tileLayer('http://server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer/t
ile/{z}/{y}/{x}'
```

Historical background map georeferenced raster overlay

Lost Rivers project were eager to include the 1932 geological map of the area around Toronto which was based on the 1931 National Topographic map at 1:63360 (A.P. Coleman. "The Pleistocene of the Toronto region: Including the Toronto interglacial formation" created by the Province of Ontario Department of Mines in 1932. Map No. 41G. Source Map and Data Library, University of Toronto.

http://maps.library.utoronto.ca/cgi-bin/datainventory.pl?idnum=851&display=full

Map was georeferenced in ArcMap and exported as a GeoTIFF. However Geoserver was not able to consume the output file directly, as georeferencing information which was in a separate .TFW file needed to be integrated into the image file. Web research discovered that it was necessary to use GDAL utilities to rebuild the image to make it acceptable to Geoserver.
GDAL - Geospatial Data Abstraction Library http://www.gdal.org/
GDAL is a translator library for raster and vector geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation (http://www.osgeo.org/). It is part of the same consortium that originally developed GeoServer. A Windows version is downloadable from https://trac.osgeo.org/osgeo4w/

The gdal_translate command must be used:

```
gdal_translate -of GTiff -a_srs EPSG:3857 G_3501_C5_Coleman_1932.tif
G_3501_C5_Coleman_1932_rebuilt.tif
```

The output file may then be brought into a Geoserver data store of type RASTER, and published as a cached tile layer. The map service is specified thus:

```
http://opengeo.geog.utoronto.ca:8080/geoserver/lostrivers_wksp/wms?service=WMS&version=1.1.0&request=GetMap&layers=lostrivers_wksp:G_3501_C5_Coleman_1932_rebuilt
```

It can be consumed by Leaflet as a tile layer in the L.tileLayer.wms form referenced above.

For more detailed information on the use of Geoserver for mapping and image generation, please contact the authors.

INTERFACE FUNCTIONALITY

jQuery Slider - Transparency slider for historical base map

Created a jQuery slider tied to the tiles generated using GeoWebcache in Geoserver 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.