Chris Fleet[*]

# An open-source web-mapping toolkit for libraries

*Keywords*: historical maps; web publishing; *OpenLayers;* web-mapping; open-source

*Summary:* This paper describes and explains a set of open-source viewers and tools which can be used to deliver maps in an online environment. It is based on the current web-mapping technologies used by the National Library of Scotland over the last six years. The tools can be classified into those supporting search and retrieval, and those creating interfaces for historic georeferenced map layers. All of these viewers and tools have been made available on Github with commented code to encourage their onward use and future collaborative development by other libraries.

## Introduction

Over the last six years, all of the National Library of Scotland's main web-mapping applications have been based on open-source technologies, particularly *GeoServer* and *OpenLayers*. One of the advantages of this is that it fosters collaborative development of the code and applications. Another advantage is that it reduces or eliminates the costs of licensing proprietary software. Open-source technology also democratises the development of the applications themselves — the applications are not in the hands of an external "expert" or company — and the applications should be more future-proof and stable with a widespread user community supporting their ongoing maintenance.

Of course, it is necessary to qualify these broad statements, and also recognise the limitations of these technologies in order to properly appreciate their role. For NLS, the open-source technologies rest upon a bedrock of proprietary software too, both at server level (eg. *SQLServer*, *Windows IIS*), as well as in terms of desktop applications (eg. *ArcGIS*, *MapTiler*). It is fully recognised that the open-source applications described below do not provide all of the sophisticated functionality of certain proprietary or institutional applications, such as *OldMapsOnline*[1], *ArcGIS Online*[2], *MapRank Search*[3], *GeoReferencer*[4], or *CartoMundi*[5]. These applications have been developed with significant technical expertise to do comparable functions — for example, the search, retrieval, and viewing of maps — and the applications described in this paper, by way of contrast, may offer value more in terms of retaining local control and ownership over functionality, in them being free to licence and develop, or in their ability to provide the basis for other customised or bespoke applications. It is assumed that there is a place for both proprietary and open-source applications rather than regarding them as alternatives.

By early 2018, the NLS used these applications to make available 200,000 historical maps. At one level, what follows is a description of recent web-mapping technologies at NLS, but at another

[*] National Library of Scotland, Edinburgh [c.fleet@nls.uk]
[1] http://www.oldmapsonline.org/
[2] https://www.arcgis.com/home/index.html
[3] https://www.mapranksearch.com/
[4] https://www.georeferencer.com/
[5] http://www.cartomundi.fr/

level it is hoped that some may potentially be of value to other institutions, and their availability on Github means that they are easy to copy and re-use. The tools can be classified into primary functions, allowing them to be deployed independently of one another and together. They can be classified into two main two groupings: first, those that support search and retrieval of maps using bounding boxes or a marker pin, and second, those for interacting with georeferenced maps. The georeferenced map viewers can be subdivided into those presenting overlays with a transparency slider, split-screen comparison viewers, spy viewers, and 3D viewers. More specific tools for geolocation, gazetteer search, or distance and area measurement can be deployed across any of these viewers. Most of the viewers use *OpenLayers*, and some of the search viewers use *GeoServer* too for storing and displaying bounding boxes, and for Web Feature Service requests.
In addition to the NLS map interfaces, several examples are finally given of specific tailored collaborative applications that have been developed using these tools, to try to illustrate their flexibility and modular nature. GitHub is a widely used code hosting platform and all these map viewers are placed in specific repositories[6]. In each repository, it is possible to easily download all the files to create an immediate working web-mapping application, just like those illustrated. The Javascript code which drives all these applications is also commented for easy onward use.

## Basic components

This paper is not intended to provide a complete guide to *OpenLayers*, nor to the use of Javascript for web-mapping, but rather to briefly point to relevant resources on these subjects, and then focus on the more specific functions that are relevant for map collections. However, it is necessary for what follows to have a basic overview of the essential components of an *OpenLayers* map application. There are excellent guides online to this, including the *OpenLayers* Tutorials[7], the *OpenLayers* Workshop[8], and the *OpenLayers Examples*[9]. The complete API for *OpenLayers* is available for reference[10], and there are also detailed books on *Openlayers*, including beginners guides (Gratier et al, 2015), and more advanced guides too (Farkas, 2016; Langley & Perez, 2016).

### *Core components of the OpenLayers application*

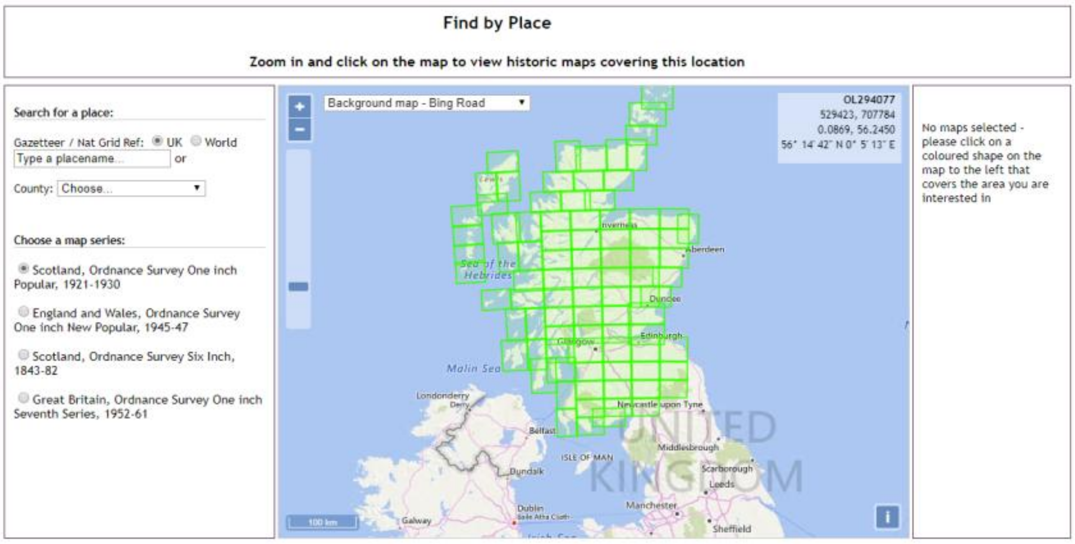
The core component of *OpenLayers* is the map (ol.Map). It is rendered to a particular target container, such as a div element on the web page that contains the map. For practical purposes, the ol.Map is usually combined with an ol.View instance, which controls the way the map is presented visually through things like its centre, zoom level and projection. For most map applications using global layers, the default projection is the Spherical Mercator (EPSG:3857), with meters as map units, but it is easy to transform between projections, and to re-project the map too. The ol.Map also needs one or more layers, allowing data from various sources to be presented visually. Three common types are the ol.layer.Tile (for layers providing a subdivided grid of pre-rendered, tiled images at specific resolutions and zoom levels), ol.layer.Vector (for point, line or polygon data that is rendered in the client), or the ol.layer.Image (for server rendered images that are available for arbitrary extents and resolutions). Each of these layers will use an

[6] The NLS Map Github repositories are at: https://github.com/NationalLibraryOfScotland
[7] https://openlayers.org/en/latest/doc/tutorials/
[8] https://openlayers.org/workshop/
[9] https://openlayers.org/en/latest/examples/
[10] https://openlayers.org/en/latest/apidoc/

ol.source subclass. Each map will also have default or customised controls, specified through the ol.control class[11] (covering things like zoom buttons, scale clines, attribution, mouse position, and rotation) as well as default or customised interactions specified through the ol.interaction class[12] (eg. different types of zooming, panning and rotation through the keyboard, mouse or screen). Combining these elements together can create a simple *OpenLayers* map[13] and is the basis of the more developed functions which follow.

### *Layer functions, Permalinks, Gazetteers, and GeoLocation*

The ability to add or remove layers from the map is another essential set of functions which are illustrated through the Github *FindByPlaceOL4* application (Figure 1). With an array of layers[14], a drop-down selection list[15] or a set of radio buttons[16] can be created, along with a function to remove or add layers based on the user interaction with these elements[17].



Find by Place

Zoom in and click on the map to view historic maps covering this location

Search for a place:

Gazetteer / Nat Grid Ref: [✓ UK] World

Type a placename... or

County: Choose...

Choose a map series:

[✓ Scotland, Ordnance Survey One inch Popular, 1921-1930]

England and Wales, Ordnance Survey One inch New Popular, 1945-47

Scotland, Ordnance Survey Six Inch, 1843-82

Great Britain, Ordnance Survey One inch Seventh Series, 1952-61

Background map - Bing Road

OL294077
529423, 707784
0.0869, 56.2450
56° 14' 42" N 0° 5' 13" E

No maps selected - please click on a coloured shape on the map to the left that covers the area you are interested in

Figure 1: An illustration of a basic *OpenLayers* map, with the ability to choose layers from radio buttons and drop-down lists[18].

Adding a Permalink suffix onto the end of the URL can be useful for defining the ol.View (with a specific zoom, lat, and lon), as well as specific layers. The *OpenLayers* Permalink example functionality[19] provides a way of doing this, although not compatible with all browsers. NLS has used functionality developed by the DataShine Census Project[20], illustrated in the Thomas Annan viewer Github Project[21]. First of all, this includes an updateUrl()function[22] that creates the

[11] https://openlayers.org/en/latest/apidoc/ol.control.html
[12] https://openlayers.org/en/latest/apidoc/ol.interaction.html#.defaults
[13] https://openlayers.org/en/latest/doc/quickstart.html
[14] https://github.com/NationalLibraryOfScotland/NLSFindByPlaceOL3/blob/master/find.js#L160
[15] https://github.com/NationalLibraryOfScotland/NLSFindByPlaceOL3/blob/master/find.js#L170
[16] https://github.com/NationalLibraryOfScotland/NLSFindByPlaceOL3/blob/master/find.js#L252
[17] https://github.com/NationalLibraryOfScotland/NLSFindByPlaceOL3/blob/master/find.js#L262
[18] http://geo.nls.uk/maps/dev/NLSFindByPlaceOL3/index.html
[19] http://openlayers.org/en/latest/examples/permalink.html
[20] http://datashine.org.uk/
[21] https://github.com/NationalLibraryOfScotland/thomas-annan-photographs
[22] https://github.com/NationalLibraryOfScotland/thomas-annan-photographs/blob/master/js/annan.js#L46

window.location.hash part of the URL with the current zoom=, &lat= and &lon= elements through an event listener on the map moveend event[23]. Second, the elements of the URL define the map view through initially being parsed on the ampersand character[24], and then specified as zoom, lat and lon variables[25], which are then integrated into the main ol.map parameters[26]. Default zoom, lat and lon parameters are defined for the initial page load if the URL suffix is not specified. It is also possible to define additional elements, such as background or overlay layers, as well as the point the user clicked on, to effectively allow all elements of the application to have a permalink quality.

Gazetteer functionality, allowing a set of names to be queried, and the map positioned on the place selected, is an intrinsic part of web-mapping operations, and can be implemented using standard *OpenLayers* ol.Map operations to centre the map on an x,y point or a bounding-box extent. At its simplest level, a set of locations for which x,y points are known can be presented as a scrollable list in alphabetical order, and upon the user selecting a particular place, a function is initiated to locate the map to the particular x,y location at a specified (relatively detailed) zoom level. More sophisticated gazetteer functionality can query larger gazetteers dynamically, with autocomplete options on the search input box, and the ability to locate on extents as well as points. The OSMNames[27] gazetteer is a good example, querying the OpenStreetMap Nominatim[28] gazetteer with autocomplete responses in an almost instantaneous manner, retrieving results that are ranked hierarchically with qualifiers for the broader geographic area and type of feature. The Google Maps Geocoding API gazetteer search functionality[29] has a broad and detailed content, especially good for postcodes and urban features, with quick autocomplete results too, but cannot be used with different map backgrounds to Google layers. There are other geocoding APIs available for gazetteers too, such as the MapBox GeoCoding API[30], or the ESRI REST API World Geocoding Service[31].

For several other functions, there are *OpenLayers* examples that make it very easy to incorporate these into another application. For example, the *OpenLayers* Geolocation[32] functionality, based on the *W3 Consortium Geolocation API Specification* (2016)[33] makes it easy to add 'Find my Location' or related geolocation tracking facilities. Further below, examples are given of the "Measure distance / area" functionality for georeferenced maps, and "Amination" functionality when moving. The important point is that simply by combining these functions together, a relatively sophisticated web-mapping application can be created. At its heart is a zoomable, pannable and rotatable map, with controls such as scalelines, mouse positions, attributions, etc. Underlying map base layers and overlaid layers can be easily specified, added and removed with radio or drop-down lists. The map details, including its centre, zoom level and layers, can be dynamically specified in a Permalink-style URL. And the map can also be positioned through a range of gazetteer operations. These combined functions form the basis of the two main generic types of web-mapping applications described in the following sections: firstly, interfaces for

[23] https://github.com/NationalLibraryOfScotland/thomas-annan-photographs/blob/master/js/annan.js#L412
[24] https://github.com/NationalLibraryOfScotland/thomas-annan-photographs/blob/master/js/annan.js#L20
[25] https://github.com/NationalLibraryOfScotland/thomas-annan-photographs/blob/master/js/annan.js#L194
[26] https://github.com/NationalLibraryOfScotland/thomas-annan-photographs/blob/master/js/annan.js#L230
[27] http://osmnames.org/
[28] https://nominatim.openstreetmap.org/
[29] https://developers.google.com/maps/documentation/geocoding/intro
[30] https://www.mapbox.com/geocoding/
[31] https://developers.arcgis.com/rest/geocode/api-reference/overview-world-geocoding-service.htm
[32] http://openlayers.org/en/latest/examples/geolocation.html
[33] https://www.w3.org/TR/geolocation-API/

searching and retrieving maps using bounding boxes, and secondly, interfaces for displaying georeferenced maps.

## Interfaces for searching and retrieving maps

From the earliest years of the development of series mapping, the paper graphic index established itself as the vital, and sometimes the only way of searching for series maps for good reason. It recognises that the primary purpose of the search is to retrieve maps of a specific place; as series maps subdivide the wider geographic territory into regular polygons, a spatial search is the most effective access method (Fleet, 2006). Web-mapping technology allows the graphic index to become more flexible and useful, as an infinitely zoomable bounding box overlay, styled with different colour, line thickness, and fill for particular purposes or at particular zoom levels, overlaid on a range of different and user-selectable base layers.

### *Bounding Box interfaces using GeoJSON files*

One of the quickest and easiest ways of presenting a map graphic index online is through a GeoJSON layer[34]. GeoJSON layers can be easily created from any other vector format, such as the ESRI Shapefile or a KML formats. The GeoJSON format allows the coding of many standard types of geometry, along with property attributes about each geometry object, and can be easily viewed and edited inside GIS or online at geojson.io. The NLSFindbyPlaceGEOJSON[35] application uses *OpenLayers* and GeoJSON graphic index files to form a geographical retrieval interface for historical maps. The GeoJSON source files are brought into ol.layer.Vector layers in the application through the ol.source.Vector class[36]. When the user clicks on the map it initiates a map.forEachFeatureAtPixel selection query[37] to the GeoJSON file, returning features that intersect with the point clicked upon. The selected feature or features can be easily styled to display prominently, and the feature properties, from the GeoJSON file, can be presented in a pop-up box or a results div element. With the steady growth in client-side computational power, there is a growing potential to do more with GeoJSON files. However, for very large sets of maps with multiple property elements, the size of the GeoJSON files can still cause problems. For this reason, for making available larger sets of maps, server-side solutions, holding the graphic index files inside *GeoServer*, for example, are more robust.

### *Bounding Box interfaces using shapefiles in GeoServer*

*GeoServer*[38] provides a convenient, easy, widely-used and open-source server technology for storing and presenting many types of geospatial data online, based on common open-standards. The background to the NLS' use of *GeoServer* as a web-mapping retrieval interface are described in Fleet & Pridal (2012), with the initial application designed and implemented by Klokan Technologies using *OpenLayers* 2. The basic application was made available as an open-source application in 2015 — NLSFindByPlaceOL3[39]. The boundaries of historic maps are held as

[34] http://geojson.org/
[35] https://github.com/NationalLibraryOfScotland/NLSFindbyPlaceGEOJSON
[36] https://github.com/NationalLibraryOfScotland/NLSFindbyPlaceGEOJSON/blob/master/find.js#L195
[37] https://github.com/NationalLibraryOfScotland/NLSFindbyPlaceGEOJSON/blob/master/find.js#L316
[38] http://geoserver.org/
[39] https://github.com/NationalLibraryOfScotland/NLSFindByPlaceOL3

shapefiles within *GeoServer*, pre-rendered using and customised for display using *GeoServer* Styled Layer Descriptors. GeoWebCache then creates a tileset of images of the shapefile at preset zoom levels for a specified coordinate system, in our case, the Google Spherical Mercator projection (EPSG:3857). As a pre-created tileset, this can be displayed very quickly, even for massive shapefiles, through the ol.layer.Tile layer, using the ol.source.XYZ source.[40] When the user clicks on the map to select a feature, it initiates a Web Feature Service (WFS) request[41] to *GeoServer*, querying the specified layer on screen for features that intersect the point clicked upon. There are various ways of specifying the query within the WFS standard, as well as the format of the results. In our case, we receive the results as a GeoJSON text string, which can be sorted, if necessary, into date order, and is then output in a Results div element, whilst the feature geometry populates a new vector layer, styled to display the feature in a highlighted style.

There are many enhancements on the basic application that are possible. A useful function is the ability to link the specific map in the right-hand results to its particular selected bounding box, where the results show maps with varying extents (Figure 2). This can be done through an ID element for each feature in the shapefile, which is returned in the GeoJSON. On the map, a map.forEachFeatureAtPixel selection query, initiated on a 'pointermove' event, can return specific details of the selected maps, which can then highlight the same map in the Result div through *jQuery*. A similar *jQuery* function works in reverse, triggered by the mouse entering the Results div and a particular returned map li element, highlighting the selected map in the main map div, through a  map.getLayers().getArray()[2].getSource(); query.

Another possible enhancement is the ability to fade the opacity of the whole bounding box layer or layerss from *GeoServer*. Having quite a dense fill (for example to show greater map coverage in specific places) can be useful, but it is helpful too in these circumstances to be able to fade the opacity. This is implemented through a *Bootstrap* transparency slider[42] and a line of code to alter the transparency of the *GeoServer* layer:

map.getLayers().getArray()[1].setOpacity(opacity);

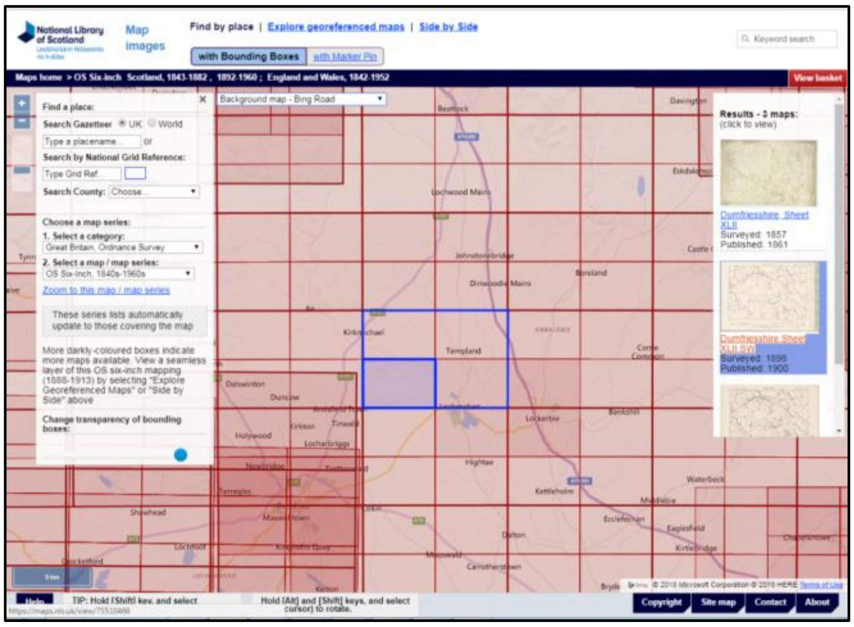

National Library of Scotland
Map images
Find by place | Explore georeferenced maps | Side by Side
with Bounding Boxes
with Marker Pin
Keyword search
Maps home > OS Six-inch Scotland, 1843-1882, 1892-1960; England and Wales, 1842-1952
View basket
Background map - Bing Road
Find a place:
Search Gazetteer UK World
Type a placename... or
Search by National Grid Reference:
Type Grid Ref...
Search County: Choose...
Choose a map series:
1. Select a category:
Great Britain, Ordnance Survey
2. Select a map / map series:
OS Six-inch, 1840s-1960s
Zoom to this map / map series
These series lists automatically update to those covering the map
More darkly-coloured boxes indicate more maps available. View a seamless layer of this OS six-inch mapping (1888-1913) by selecting "Explore Georeferenced Maps" or "Side by Side" above
Change transparency of bounding boxes:
Results - 3 maps:
(click to view)
Dumfriesshire, Sheet XLII
Surveyed: 1857
Published: 1861
Dumfriesshire Sheet XLII.SW
Surveyed: 1898
Published: 1900
Templand
Lockerbie
Lochwood Mains
Johnstonebridge
Dinwoodie Mains
Kirkmichael
Dalswinton
Holywood
Locharbriggs
Hightae
Kettleholm
Ecclefechan
Waterbeck
© 2018 Microsoft Corporation © 2018 HERE Terms of Use
Help
TIP: Hold [Shift] key, and select
Hold [Alt] and [Shift] keys, and select cursor] to rotate.
Copyright
Site map
Contact
About

Figure 2: The *Find by Place - with Bounding Box* viewer ( http://maps.nls.uk/geo/find/).

[40] https://github.com/NationalLibraryOfScotland/NLSFindByPlaceOL3/blob/master/find.js#L186
[41] http://www.opengeospatial.org/standards/wfs
[42] http://seiyria.com/bootstrap-slider/

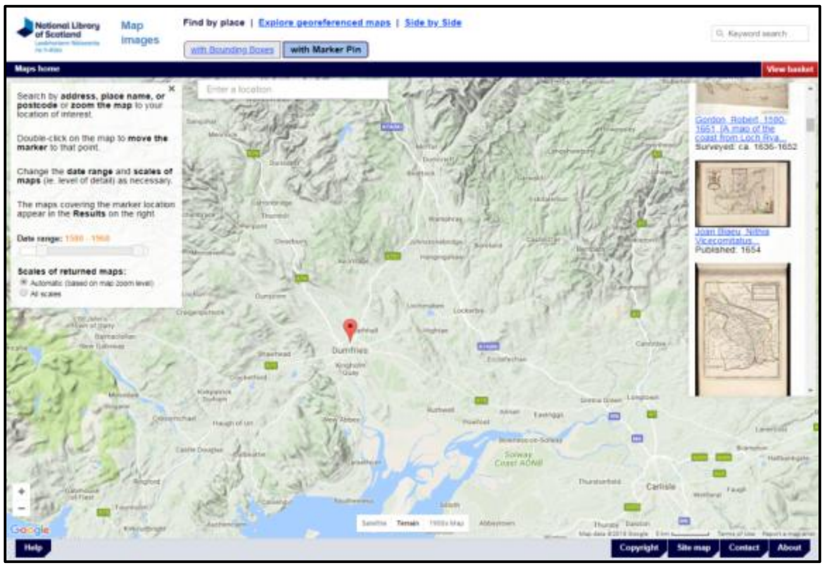
*Marker interfaces using shapefiles in GeoServer*

Bounding Box interfaces for map search are probably the most familiar and intuitive for map librarians and tend to be preferred by more expert users. However, they have the disadvantage that for large quantities of maps, the multiple indexes cannot be displayed simultaneously on screen without producing a mass of rectangular spaghetti, and so the user needs to potentially work through many different layers of bounding boxes to see maps in all layers for the same place. Some users also want instant results from queries along the lines "I want to see all the maps there are covering this place?" For these reasons, a Marker Interface is a possible alternative.

NLS introduced a *Find by Place - with Marker* application in 2017, using the Google Maps Javascript API and *GeoServer* to form a basic search and retrieval interface[43]. The idea for the application came from the Charles Close Society's *Sheetfinder* application[44], and collaborative work in 2016-17 with them to incorporate results from the NLS online collection. A similar application could easily have been created with *OpenLayers*, as the main Javascript functions are easily transferable, but we were keen to use the Google Map interface for two reasons. First the the Google Places API Web Service[45] is better for some urban features and postcode searching than the OSM Nominatim gazetteer, and second, Google Maps are instantly familiar and preferred by some users to alternative global map services. Although there have been efforts to combine Google Maps and *OpenLayers*, for example by Mapgears[46], the two different applications have quite different aims, and the resulting viewer has some limitations[47].

The *Find by Place - with Marker* application allows the user to zoom in on an area of interest, or use the gazetteer to do so, with an option to change the map base layer between Google Maps or Satellite layers, and a 1900s historic layer (Figure 3). The location of the marker initiates a Web Feature Service request to *GeoServer*, returning records for maps whose bounding boxes cover the marker location. A jQuery slider provides a way of narrowing the date range of the returned maps. By default, the maps returned reflect the approximate scale of the map (small-scale, medium-scale, and large-scale), but this can be changed by the user to return all scales. At the time of writing, NLS does not have scales recorded for all online maps, and so the three scales select particular layers in the *GeoServer* queries, rather than a direct and more precise numeric range filter.

[43] The live application is at: http://maps.nls.uk/geo/find/marker/ and the Github application is at https://github.com/NationalLibraryOfScotland/FindbyPlace---with-Marker
[44] http://sheetfinder.charlesclosesociety.org/
[45] https://developers.google.com/places/web-service/
[46] https://github.com/mapgears/ol3-google-maps
[47] https://github.com/mapgears/ol3-google-maps/blob/master/LIMITATIONS.md



Search by **address, place name, or postcode** or **zoom the map** to your location of interest.

Double-click on the map to **move the marker** to that point.

Change the **date range** and **scales of maps** (ie. level of detail) as necessary.

The maps covering the marker location appear in the **Results** on the right.

**Date range:** 1580 - 1960

**Scales of returned maps:**
Automatic (based on map zoom level)
All scales

Gordon, Robert, 1580-1661. [A map of the coast from Loch Rya...
Surveyed: ca. 1636-1652

Joan Blaeu. Nithia Vicecomitatus...
Published: 1654

Figure 3: The *Find by Place - with Marker* application ( http://maps.nls.uk/geo/find/marker/).

## Interfaces for displaying georeferenced maps

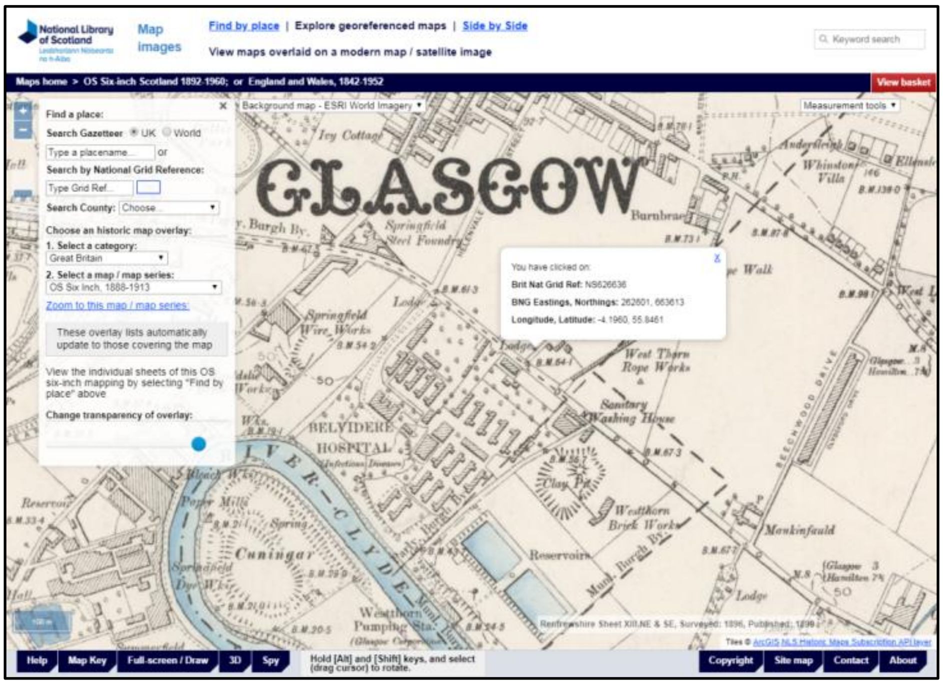
As is well known, georeferencing allows the direct and easy comparison of an historic map with either modern day satellite or image layers, or with other historic georeferenced content, and the interfaces are all intended to facilitate this comparison. In order for large georeferenced map images, or wider sets of series maps presented as a seamless layer, to display fast in web-mapping viewers, the images need to be processed. There are a number of open and proprietary ways of doing this, which are beyond the scope of this paper. The assumption is made here for the georeferenced map viewers below, that the georeferenced maps are available in an appropriate standard tiled (x,y,z) format, such as the OGC Web Map Tile Standard[48], the OSGeo Tile Map Service[49], or MapBox's TileJSON[50] or MBTiles[51] formats. For many years, the NLS has happily used Klokan Technologies' MapTiler application for this purpose, which also generates default web-mapping viewers, including an *OpenLayers* viewer, for each tileset, and these are referred to in the examples which follow.

### *Map overlay viewer*

The basic NLSExploreGeoreferencedMaps overlay viewer[52] incorporates standard features described above, including a *Bootstrap* transparency slider[53], with a line of code to alter the transparency of the *GeoServer* layer: map.getLayers().getArray()[1].setOpacity(opacity);
Various enhancements can easily be made to this viewer, using standard *OpenLayers* code available as examples. These include tools to measure distances or areas on screen[54], so that their

[48] http://www.opengeospatial.org/standards/wmts
[49] http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification
[50] https://github.com/mapbox/tilejson-spec
[51] https://www.mapbox.com/help/define-mbtiles/
[52] https://github.com/NationalLibraryOfScotland/NLSExploreGeoreferencedMapsOL3
[53] http://seiyria.com/bootstrap-slider/
[54] http://openlayers.org/en/latest/examples/measure.html

incorporation in bespoke ways is easy[55]. The basic overlay popup functionality is also useful[56], so that with a user click (or ALT-click) on a point on the map, a popup will appear with user defined content, such as locational information for easy copying and pasting (Figure 4). For larger applications, hosting many layers, a unique id for the layer and additional parameters in the layer definition can also be used. As seamless map layers are stripped of their marginalia, it is also useful to be able to display a default legend or key relating to the layer on screen[57]. The code for this looks up the map layer on screen, retrieves the default URL for the separate map key image, and displays this in a popup window.


National Library of Scotland
Map images
Find by place | Explore georeferenced maps | Side by Side
View maps overlaid on a modern map / satellite image
Keyword search
Maps home > OS Six-inch Scotland 1892-1960; or England and Wales, 1842-1952
View basket
Background map - ESRI World Imagery
Measurement tools
GLASGOW
Help
Map Key
Full-screen / Draw
3D
Spy
Copyright
Site map
Contact
About

Find a place:
Search Gazetteer [✓ UK] World
Type a placename... or
Search by National Grid Reference:
Type Grid Ref...
Search County: Choose...
Choose an historic map overlay:
1. Select a category: Great Britain
2. Select a map / map series: OS Six Inch, 1888-1913
Zoom to this map / map series:
These overlay lists automatically update to those covering the map
View the individual sheets of this OS six-inch mapping by selecting "Find by place" above
Change transparency of overlay:

You have clicked on:
**Brit Nat Grid Ref:** NS626636
**BNG Eastings, Northings:** 262601, 663613
**Longitude, Latitude:** -4.1960, 55.8461

Renfrewshire Sheet XIII.NE & SE, Surveyed: 1896, Published: 1898
Tiles © ArcGIS NLS Historic Maps Subscription API layer
Hold [Alt] and [Shift] keys, and select (drag cursor) to rotate.

Figure 4: The *Explore Georeferenced Maps* viewer (http://maps.nls.uk/geo/explore/), illustrating a popup, generated by a user click event, with locational details of the point clicked on for copying.

A closely related function is also used to display the particular sheet that the users' cursor is over for large georeferenced mosaics. This is helpful for users to view a specific map on its own, as well as for displaying specific dates of survey, revision or publication in mosaics that span a wider number of years. A Web Feature Service is initiated to *GeoServer* using very similar syntax as in the Find by Place viewers described above. The only difference is that instead of the request searching for features intersecting the point clicked on, the request searches for all features overlapping with the bounding box extents of the map view. These are brought as GeoJSON records into a new vector layer, which displays the specific sheet information in the viewer, based on a map.forEachFeatureAtPixel query, initiated by the mouse pointermove event (Figure 5).

[55] See, for example, the NLS Explore Georeferenced Maps application, which implements line and area measurement through a drop-down selection list to the upper right of the screen.
[56] https://openlayers.org/en/latest/examples/overlay.html
[57] The NLS Explore Georeferenced Maps application has a 'Map Key' tab to the lower left

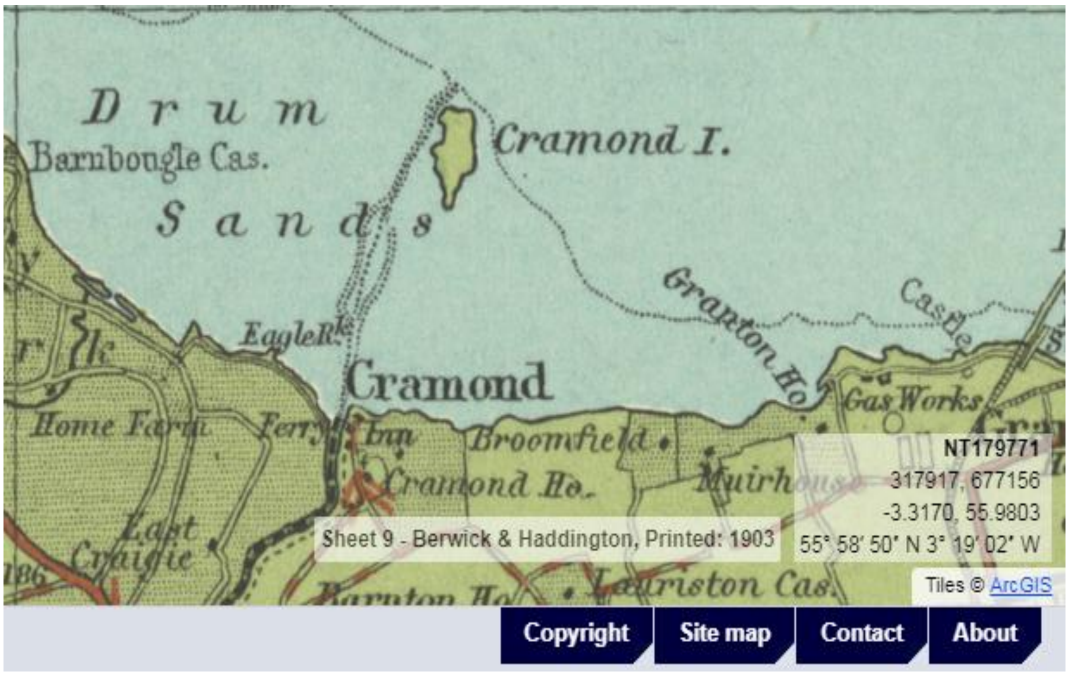


Figure 5: The *Explore Georeferenced Maps* viewer (http://maps.nls.uk/geo/explore/), showing details of the map sheet the cursor is over, using a Web Feature Service query to *GeoServer*.

### *Side-by-side / dual map viewer*

A closely related viewer displays two maps side-by-side on screen (Figure 6)[58] This makes use of two ol.Map classes in the code, here called mapleft and mapright. The two views can be integrated by not specifying a particular centre or zoom in the mapright view, but simply by having view: mapleft.getView() [59]. A slave cross-hair icon can also easily be added, matching the location of the mouse cursor in the other map window. This is a two-stage process, firstly defining the cross as a feature, and then adding this to a vector layer in both of the maps[60]. The second stage involves a pointermove event handler, recording the cursor coordinates and then adding these to the icon geometry of the cross feature in the adjacent map window[61]. A variation on the Side-by-side / dual map viewer is to implement a sliding central bar, or layer swipe. There is an *OpenLayers* example for this[62] which we have implemented in the *Side-by-side with layer swipe* viewer[63].

[58] https://github.com/NationalLibraryOfScotland/SidebySideOL3
[59] https://github.com/NationalLibraryOfScotland/SidebySideOL3/blob/master/sidebyside.js#L271
[60] https://github.com/NationalLibraryOfScotland/SidebySideOL3/blob/master/sidebyside.js#L297
[61] https://github.com/NationalLibraryOfScotland/SidebySideOL3/blob/master/sidebyside.js#L357
[62] https://openlayers.org/en/latest/examples/layer-swipe.html
[63] http://maps.nls.uk/geo/explore/side-by-side/swipe/

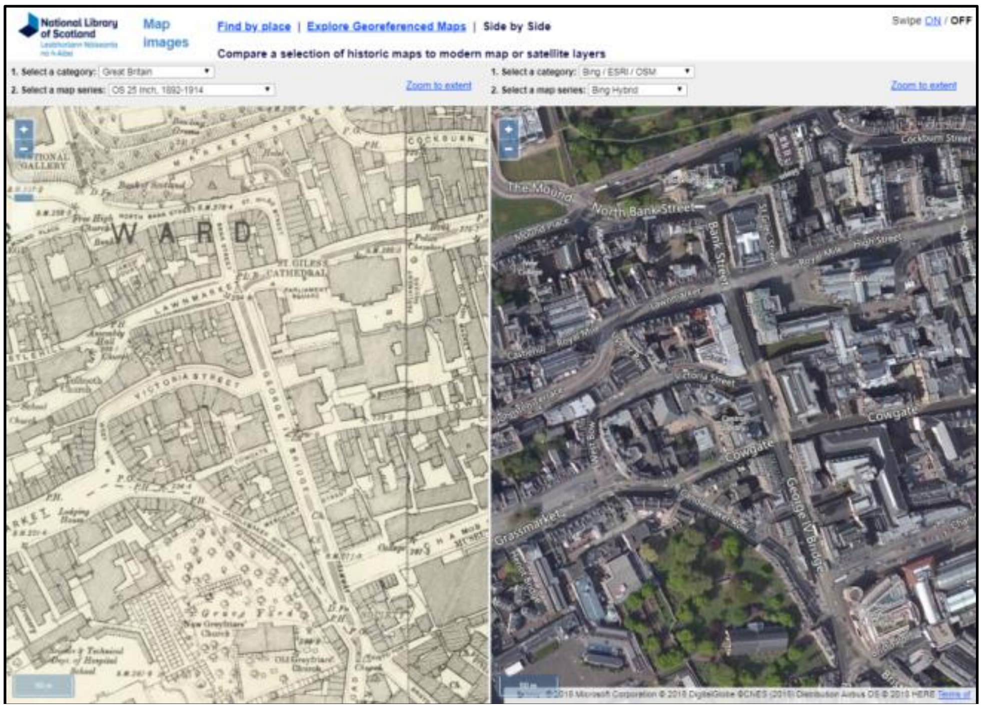

Find by place | Explore Georeferenced Maps | Side by Side
Compare a selection of historic maps to modern map or satellite layers
Swipe ON / OFF
1. Select a category: Great Britain
2. Select a map series: OS 25 inch, 1892-1914
1. Select a category: Bing / ESRI / OSM
2. Select a map series: Bing Hybrid
Zoom to extent

Figure 6: The *Side-by-sid*e / dual-map viewer ( http://maps.nls.uk/geo/explore/side-by-side/).

*Spy viewer*

This viewer is directly based on the *OpenLayers* Layer Spy viewer[64], which is a powerful and enjoyable way of visualising georeferenced map overlays (Figure 7). The only small enhancement to this viewer that NLS has made has been to add a *Bootstrap* slider (described above) to adjust the size of the radius of the spy circle[65].

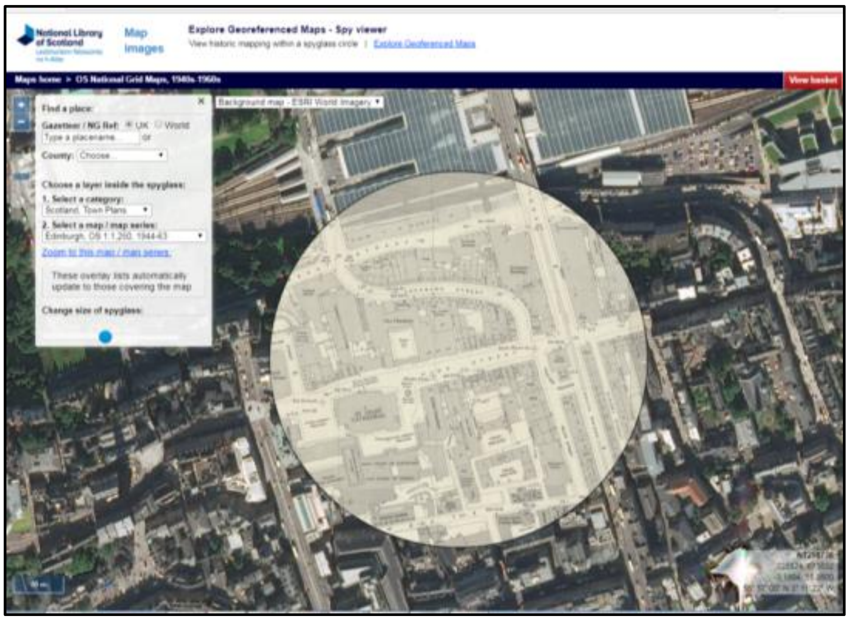
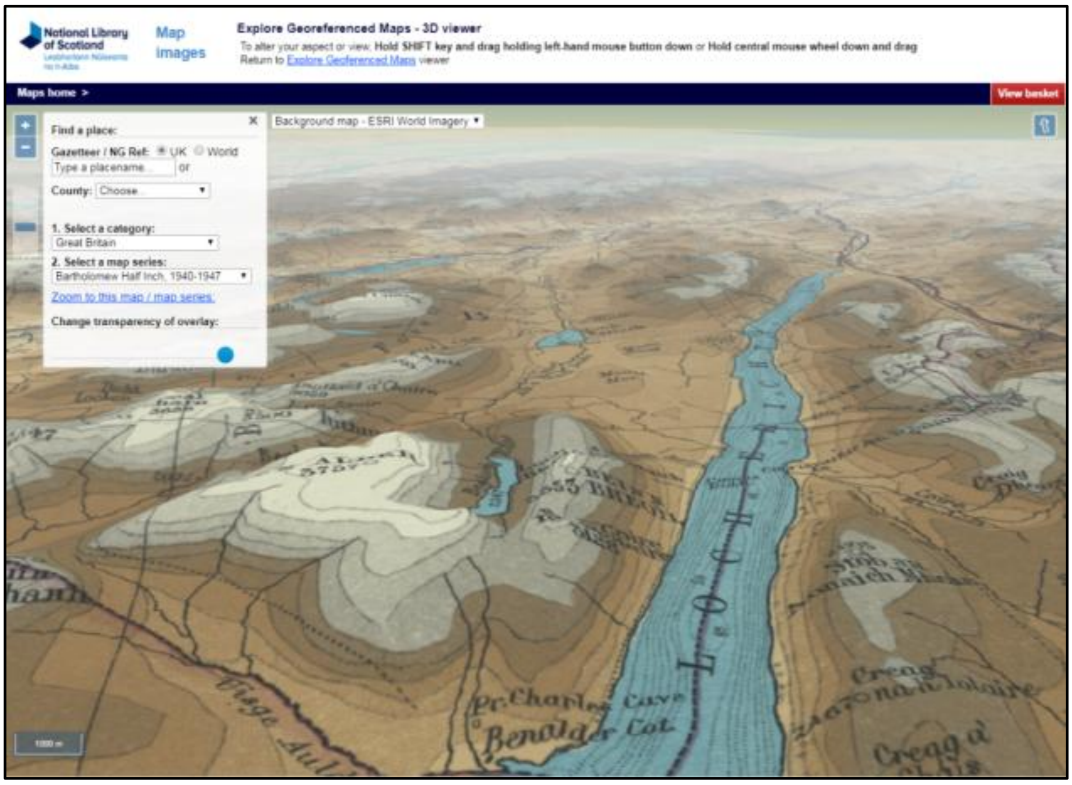

Explore Georeferenced Maps - Spy viewer
View historic mapping within a spyglass circle | Explore Georeferenced Maps
Maps home > OS National Grid Maps, 1940s-1960s
View basket
Background map - ESRI World Imagery
Find a place:
Gazetteer / NG Ref: UK World
Choose a layer inside the spyglass:
1. Select a category: Scotland, Town Plans
2. Select a map / map series: Edinburgh, OS 1:1,250, 1944-63
Zoom to this map / map series
Change size of spyglass:

Figure 7: The *Spy viewer* ( http://maps.nls.uk/geo/explore/spy/).

[64] https://openlayers.org/en/latest/examples/layer-spy.html
[65] This can be seen at http://maps.nls.uk/geo/explore/spy/ with the code for this from lines 650 of http://maps.nls.uk/geo/scripts/explore-spy.js

*3D viewer*

The ability to drape a georeferenced layer across a 3D landscape has great potential, even if the use of this in the short term can test older client computational powers and graphics. The easiest way of doing this with *OpenLayers* is through the ol-cesium integration project[66], based on the Cesium open-source Javascript library[67]. This is well documented with easy-to-follow examples, and much of the existing *OpenLayers* functions work in the same way without modification. There is a need to specify additional parameters — for example, the camera distance (height), its heading (rotation from North), and tilt (angle towards the earth) — and make minor adjustments to functions involving the map view extents and centre, and the result is a striking "helicopter view" of past landscapes (Figure 8).


Explore Georeferenced Maps - 3D viewer

To alter your aspect or view: Hold SHIFT key and drag holding left-hand mouse button down or Hold central mouse wheel down and drag

Return to Explore Georeferenced Maps viewer

Find a place:

Gazetteer / NG Ref: [✓ UK] World

Type a placename... or

County: Choose...

1. Select a category: Great Britain

2. Select a map series: Bartholomew Half Inch, 1940-1947

Zoom to this map / map series:

Change transparency of overlay:

Background map - ESRI World Imagery

Figure 8: The *3D viewer* ( http://maps.nls.uk/geo/explore/3d/).

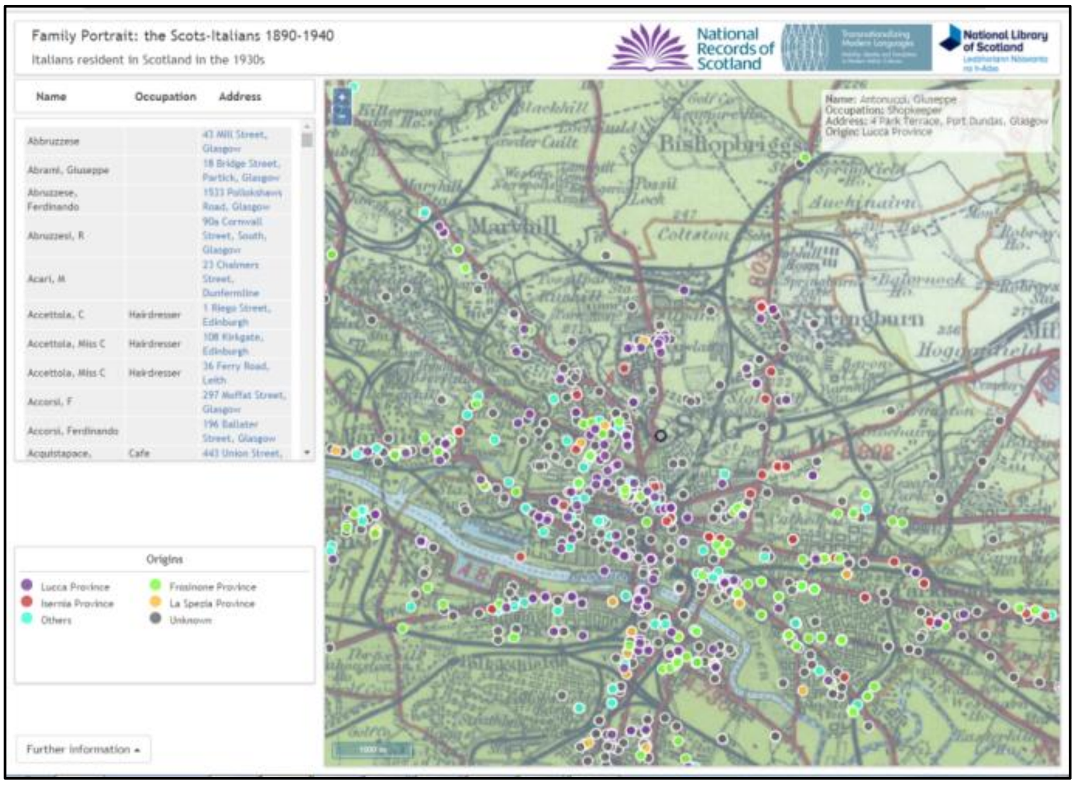
## Examples of these viewer functions in other applications

As the above has hopefully illustrated, many of the specific functions and application elements are shared across the applications, and can easily be combined in new ways for specific purposes. The following bespoke applications illustrate this:

[66] http://openlayers.org/ol-cesium/
[67] https://cesiumjs.org/

## *Italians in Scotland in the 1930s*

This application[68] tries to show where Italians lived or worked in Scotland during the 1930s (Figure 9). It was created to accompany the free exhibition *Family Portrait: The Scots Italians 1890 – 1940* at National Records of Scotland, Edinburgh (December 2015 - 29 January 2016). The names and addresses are primarily based on the Scottish entries in the 1936 edition of the *Guida Generale Degli Italiani in Gran Bretagna* / *General Guide to Italians in Great Britain*, published in London. The map tries to indicate approximately where Italians were either working or living at this period. It is possible to search the alphabetical list of Italians (upper left) and click on the address to locate the map there. Alternatively, it is possible to zoom in on the map, and at higher zoom levels, hover your cursor over the circles to view the same details of the Italians at each location. The map tries to indicate approximately where Italians were either working or living at this period. It colour-codes the top four provinces from which Italians arrived in Scotland. These were in descending order: Lucca, Frosinone, Isernia (in what was then part of the province of Campobasso), and La Spezia. People from Pistoia, Parma, Latina, Massa Carrara, Pordenone and elsewhere are grouped together as 'Other Provinces.' A sixth group consists of those whose origins are not yet known. The map indicates that over half of the Italian migrants were settled in and around Glasgow. The Glasgow Italian community was divided between those who came from the province of Lucca, especially from the commune of Barga, and other places in Tuscany. Far fewer people migrated from Frosinone in Lazio, south of Rome. The Scots Italian community living and working in Edinburgh was relatively small, and it was dominated by people from the village of Picinisco in the province of Frosinone.


Family Portrait: the Scots-Italians 1890-1940

Italians resident in Scotland in the 1930s

| Name | Occupation | Address |
|---|---|---|
| Abbruzzese | | 43 Mill Street, Glasgow |
| Abrami, Giuseppe | | 18 Bridge Street, Partick, Glasgow |
| Abruzzese, Ferdinando | | 1533 Pollokshaws Road, Glasgow |
| Abruzzesi, R | | 90a Cornwall Street, South, Glasgow |
| Acari, M | | 23 Chalmers Street, Dunfermline |
| Accettola, C | Hairdresser | 1 Riego Street, Edinburgh |
| Accettola, Miss C | Hairdresser | 108 Kirkgate, Edinburgh |
| Accettola, Miss C | Hairdresser | 36 Ferry Road, Leith |
| Accorsi, F | | 297 Moffat Street, Glasgow |
| Accorsi, Ferdinando | | 196 Ballater Street, Glasgow |
| Acquistapace, | Cafe | 443 Union Street, |

Name: Antonucci, Giuseppe
Occupation: Shopkeeper
Address: 4 Park Terrace, Port Dundas, Glasgow
Origin: Lucca Province

Origins
● Lucca Province
● Isernia Province
● Others
● Frosinone Province
● La Spezia Province
● Unknown

Figure 9: The Scots-Italians viewer ( https://maps.nls.uk/projects/italians/index.html).

[68] https://maps.nls.uk/projects/italians/index.html

The core functionality of this application is very similar to the *NLSFindbyPlaceGEOJSON* example above. The addresses of the Italians were geocoded and saved as a GeoJSON file, in this case with the features being point locations, rather than polygons. The GeoJSON details are selected by a map cursor 'pointermove' event, triggering a map.forEachFeatureAtPixel function to retrieve the name, address, occupation and origin information into a results div. The alphabetical gazetteer of names was output from the geocoded list in a structured manner, exporting the lat, lon locations which the map locates to on the user selecting the name. The structured contents are held in a table HTML markup format.

### *Scotland - Land Use Viewer*

This viewer was developed in 2016 to allow land use in the Scotland between the 1930s and 2015 to be compared in a split-screen viewer (Figure 10). In 2015 the NLS scanned, georeferenced and put online our 1930s Land Utilisation Survey maps for Scotland, the first comprehensive survey of the land use in the country. In the summer of 2015, by coincidence, a Historic Land-use Assessment map (HLAMap) was completed by Historic Environment Scotland. Placing these two different dates of maps together highlights the significant changes in the Scottish 20th century landscape, such as tree planting in Argyll and Dumfries, the damming of rivers in the Highlands, and urban expansion across the Central Belt. The split-screen viewer also allows the user to zoom-in to review more localised land use change, including changing patterns of arable and pasture land.

The code behind this application is almost identical to *SidebySideOL3* above. The main additional work involved the styling of the HLAMap vector layer to approximately match the colour coding of the original 1930s land use maps. The left-hand 1930s Land Utilisation Maps just present two different scales of georeferenced mapping, made available as ol.source.XYZ sources, within ol.layer.Tile layers. The right-hand HLAMap layer is stored within PostGIS, and delivered out of *GeoServer* (which gave better performance than storing in *GeoServer* alone). The *GeoServer* layer is delivered as an ol.source.XYZ source, within an ol.layer.Tile layer.

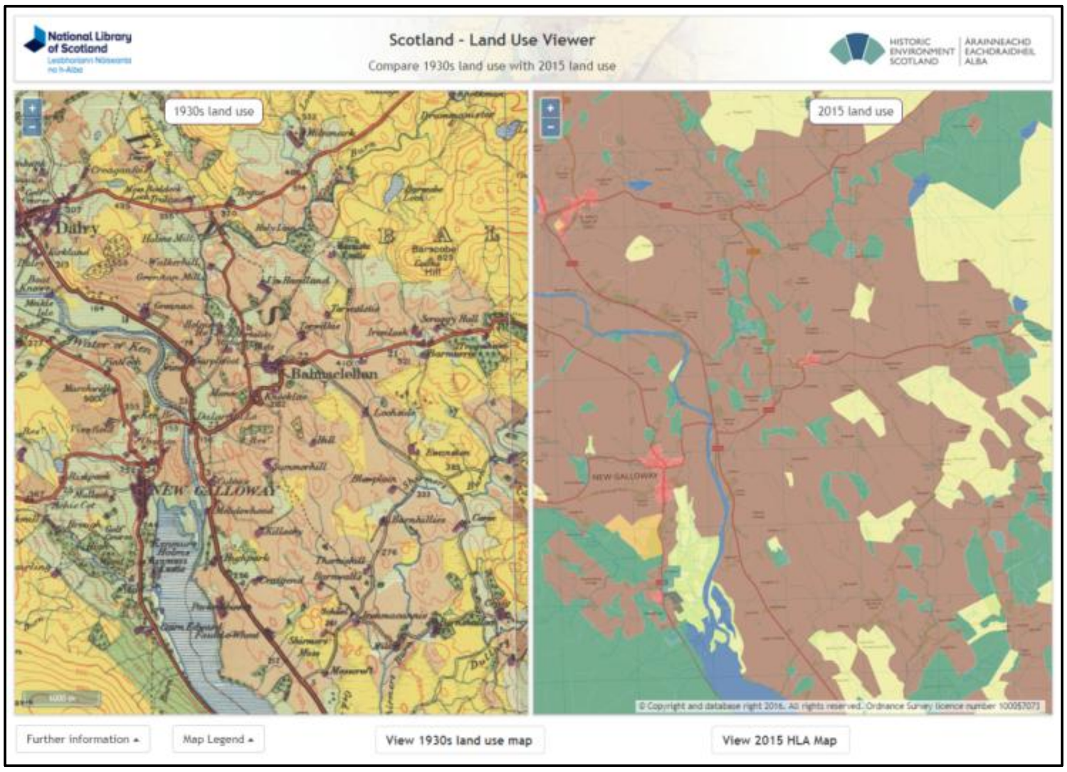


Figure 10: Scotland: Land Use viewer ( https://maps.nls.uk/projects/landuse/index.html).

*Thomas Annan photographs*

This viewer was created in early 2016 to provide a map-based search interface to a set of historical photographs of central Glasgow, dating from the late 1860s[69]. The code for the application uses components described above, and can be viewed on Github[70]. The locations of the photographs are shown using a GeoJSON file of point locations, and when these are clicked on, a map.forEachFeatureAtPixel function retrieves the specific GeoJSON, returning the results to a *Bootstrap* Popover, showing a thumbnail image and text. The URL of the application uses the Permalink code described in the *Basic components* section above, to show the particular zoom level and centre of the map, as well as if a photograph is selected. This also has the advantage of being able to link from a specific photograph to its location on the map (Figure 11).

[69] https://digital.nls.uk/learning/thomas-annan-glasgow/historical-maps/
[70] https://github.com/NationalLibraryOfScotland/thomas-annan-photographs

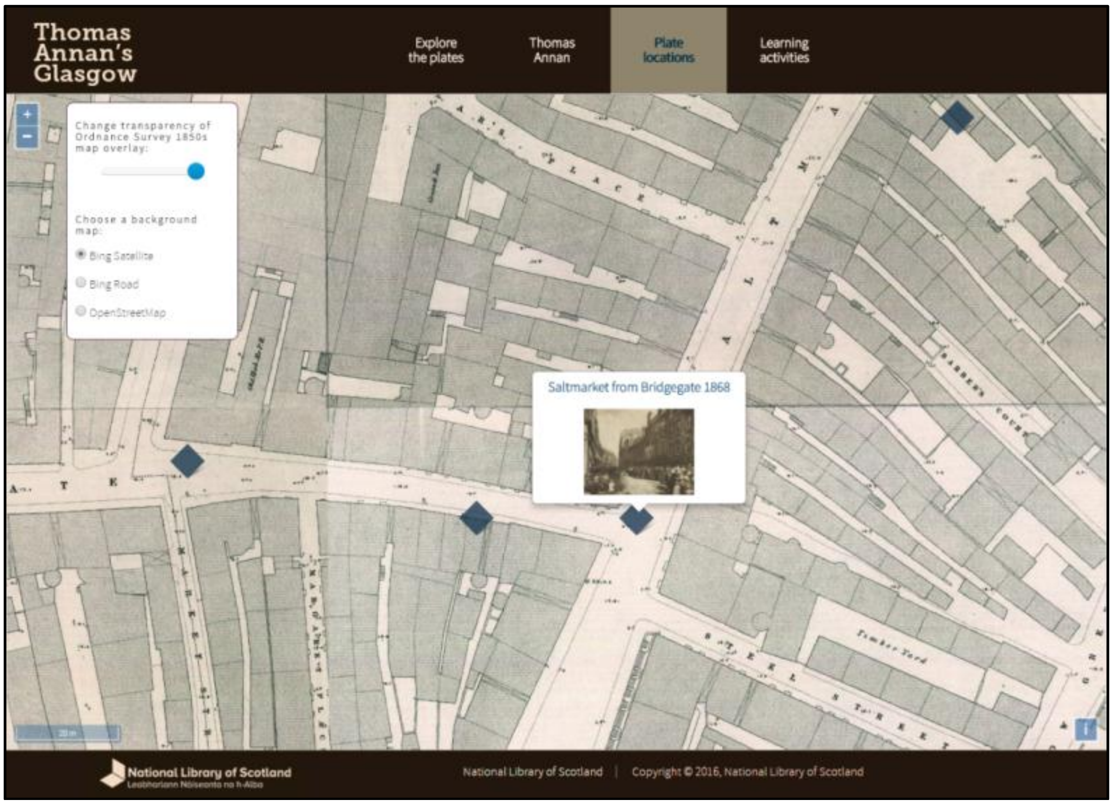

Thomas Annan's Glasgow
Explore the plates
Thomas Annan
Plate locations
Learning activities
Change transparency of Ordnance Survey 1850s map overlay:
Choose a background map:
Bing Satellite
Bing Road
OpenStreetMap
Saltmarket from Bridgegate 1868
National Library of Scotland
Leabharlann Nàiseanta na h-Alba
National Library of Scotland | Copyright © 2016, National Library of Scotland

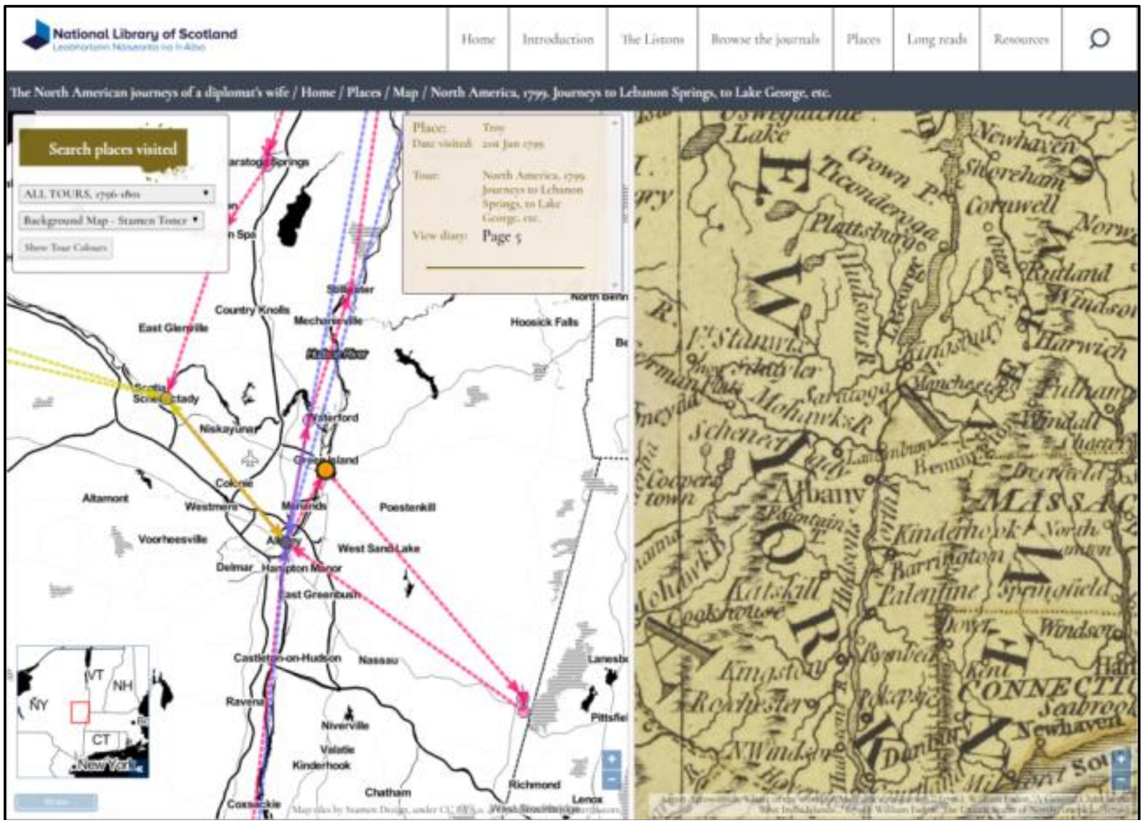
Figure 11: Thomas Annan photographs ( .https://digital.nls.uk/learning/thomas-annan-glasgow/historical-maps/).

*Henrietta Liston's diaries* - *map viewer*

This viewer was developed in early 2017 to provide a search interface to the travel diaries of Henrietta Liston in North America and Canada (1796-1801). Henrietta Liston's husband, Robert Liston, was a Scottish diplomat, and a newly appointed minister to the United States, and he travelled extensively with his wife Henrietta, who recorded these journeys in her diaries. The scanned diaries formed the central content of the website, with the map interface providing a geographic way of visualising the journeys, as well as a way of selecting particular places to read relevant diary pages. The code behind the application shares much in common with the *Italians in Scotland in the 1930s* viewer, and is available for easier viewing on Github[71]. It was felt useful to use a split-screen viewer, so that the accurate points of the itinerary could be correctly located on a modern map backdrop on the left, but with contemporary 1790s maps of the Americas on the right. These historic maps were Aaron Arrowsmith's *Chart of the world on Mercator's projection…* (1796) at lower zoom levels, and William Faden's *The United States of North America …* (1796) at higher zoom levels.The Listons' itineraries were held as GeoJSON files, with a file of point locations, and file of basic lines between the points for each journey. The easiest method of interaction is simply to zoom in to view the maps and click on any point to retrieve the details (to an upper right div) of when the Listons were there, including a link through to the relevant diary pages relating to this. This uses a map.forEachFeatureAtPixel function. Two sets of gazetteer listings were created as structured text with coordinates, one listing the places by itinerary, and the other as a continuous alphabetical list. On selecting a place, the map locates to it, this using *OpenLayers* pan and bounce animations[72], to give some dynamism to the application

[71] https://github.com/NationalLibraryOfScotland/ListonMapApplication
[72] https://openlayers.org/en/latest/examples/animation.html

as well as an impression of the geographical relationships between points. The URL of the application uses the Permalink code described in the *Basic components* section above, to show the particular zoom level and centre of the map, as well as the particular tour and if a place is selected. This allows the ability to link directly into the viewer to show a specific tour or a specific place (Figure 12).



The North American journeys of a diplomat's wife / Home / Places / Map / North America, 1799. Journeys to Lebanon Springs, to Lake George, etc.

Place: Troy
Date visited: 2nd Jun 1799
Tour: North America, 1799. Journeys to Lebanon Springs, to Lake George, etc.
View diary: Page 5

Figure 12: Henrietta Liston's diaries - map viewer ( http://digital.nls.uk/travels-of-henrietta-liston/map/).

## Conclusions

These applications should all be seen as work-in-progress, and simply as one way, amongst many others, of making available map library collections. The author is not a Javascript programmer, and the code reflects this — whilst it works, and is explained in a way that hopefully the non-programmer can understand, it is not as accomplished, efficient or bug-free as a professional programmer could produce. As has been illustrated, much of the real value and content lies in the *OpenLayers* code itself, which provides the core functionality throughout. The fact that these interfaces, particularly for georeferenced maps, share much in common with the *Mapire.eu* project[73] who also make available layers of historic mapping online, reflects the fact that both use the same core *OpenLayers* library.

This paper has focused on interfaces for historical maps, and it is useful to bear in mind that interfaces are probably the most ephemeral of digital library applications; they change fast, and need to change fast as technology and user expectations continually change. One of the ways of managing this is to use a widely supported open-source software interface, such as *OpenLayers*, migrating the code as it develops, and supporting or adopting new functionality as these are also

[73] http://mapire.eu/

developed too. It is recognized that the tools described here will change, and the images of these interfaces will quickly look dated, but the underlying application code is easy to continually update. Another useful way of managing continual change is to have a clear model or architecture of the digital map library, recognizing that components such as interfaces are clearly separate from other underlying components such as raw images and metadata, and server-side technologies that deliver these. Although interfaces will need to be updated or changed regularly, the underlying imagery or metadata should not need to change, and server technologies can be replaced independently of the viewer applications (Fleet, 2014). Many other library applications — library management systems, or content management systems, for example — are more complex integrated systems where the whole technology stack needs to be replaced as a costly migration exercise.

Geographic or map-based interfaces are a particularly useful and engaging way of making available library and archive collections online, and these tools and interfaces are offered here in the hope that other institutions may find them useful, as well as enjoyable to create, in making their own collections available online. We also welcome comments or code to develop and improve these viewers.

## References

Farkas, G. (2016). *Mastering OpenLayers 3: create powerful applications with the most robust open source web mapping library using this advanced guide.* Birmingham, UK: Packt Publishing.

Fleet, C. (2006). 'Locating trees in the Caledonian forest': A critical assessment of methods for presenting series mapping over the web. *e-Perimetron*, Vol.1, No. 2, Spring 2006 [99-112]. In digital form, http://www.e-perimetron.org/vol_1_2/fleet/fleet.pdf

Fleet, C. (2014). Old maps and new web technologies: practicalities, impact and potential. *SoC Bulletin* 48, 26-34.

Fleet, C. and P. Pridal (2012). Open Source Technologies for Delivering Historical Maps Online — Case Studies at the National Library of Scotland. *LIBER Quarterly* 22(3), 240-257. In digital form, https://www.liberquarterly.eu/articles/10.18352/lq.8052/

Gratier, T., P. Spencer, E. Hazzard (2015). *Openlayers 3 beginner's guide: get started with openlayers 3 and enhance your web pages by creating and displaying dynamic maps.* Birmingham, UK: Packt Publishing.

Langley, P.J. & Perez, A.S. (2016). *OpenLayers 3.x cookbook: over 50 comprehensive recipes to help you create spectacular maps with OpenLayers 3.* 2nd ed. Birmingham, UK: Packt Publishing.