

Building an Integrated Web Mapping and Incident Reporting Platform for Volunteer Organizations Using Open Source Technology



David Todd

Capstone Project

Pennsylvania State

University

GEOG 596B: Individual

Studies – Capstone Project

Spring 2 Term 2014

Advisor: Jan Oliver

Wallgrün

5/5/2014

Table of Contents

Abstract	2
Introduction.....	2
Motivation	4
Goal.....	5
Research Results.....	5
Methodology	8
Process.....	8
Development.....	14
Results	15
Conclusions.....	21
Bibliography	23
Literature References.....	23
Web References	23

Abstract

Across the United States, trails are being managed by local organizations powered by volunteers. Given the current fiscal environment, these organizations often have a core volunteer base providing labor, but considerably smaller amounts of funding. Trail data is usually stored in random files and spread across multiple volunteers. Organizations are lucky if they are able to provide updated trail maps and timely notes on current trail status. What if there was a better way to focus the volunteers' time on areas that needed the most attention, while also getting the public more involved? Using open source and cloud technologies, we are able to build a platform that allows volunteer organizations to manage their data and make it available to the public anytime, anywhere. Additionally, the public is encouraged to get involved by providing the location and description of any trail issues right from the trail on their mobile device. This project explores the process of providing an integrated open source geospatial publishing and incident reporting platform that allows volunteer organizations to get up and running quickly by simply importing their data. This project targets the Bay Circuit Trail system, but the platform could easily be utilized by any number of volunteer organizations.

Introduction

Trails are vital links to communities everywhere. Some are steeped in history and others provide opportunities for both recreation and transportation to all levels of people. Trails benefit communities by being a tourism boon and by providing safe locations for healthy activities for the citizens. Given the importance of trails to both the economy and citizen well-being, it might be surprising to find out that many trail systems are being managed by volunteer organizations.

Volunteer organizations often have little to no budget and funding for projects can be extremely tight. Members of volunteer trail organizations generally fall into two categories – trail liaisons and trail users. Trail liaisons are a group of administrators who come in both global and regional flavors. They want the ability to create, read, update and delete trail information. Trail users on the other hand, consist of hikers, bikers, and other outdoor enthusiasts. Users generally want the ability to visualize the trail, get their current location, and report any trail issues they come across, e.g. tree down over trail, hornets' nest, etc.

Within these two member categories, organizations often have a core volunteer base who gives free labor in the form of building and maintaining trails. What limited funding is available usually goes toward deflecting costs associated with building materials and tools. In addition to labor, organizations will sometimes have members with the technical skills to build and host a trail or organization website.

Given the general lack of funding, websites are usually very basic. Trail data, if it exists, are usually stored in random files and spread across multiple volunteers. More often than not, trail data is stored on a member's personal global positioning system (GPS) receiver. Organizations are lucky if they are able to provide updated trail maps, never mind giving timely notes on current trail status.

Today, there exists a wide range of privately developed web mapping applications for trail users. These range from simple trail displays to trails with built in loops with user generated ratings and skill or ability levels that one would need to possess. As seen in Figure 1, the International Mountain Bike Association's (IMBA) MTB Project is one of the better trail mapping sites, because it gives users a lot of information in a clean layout (IMBA, 2014). Their trails are a combination of IMBA provided and crowd sourced. Having said that, the vast majority of organizations only provide very basic options, if at all, for users to contribute to trail information or report issues on the trail. I was unsuccessful in finding a single trail trouble reporting site that paired event reporting with a trail map. The sites that provided a mechanism for reporting trouble issues mostly used simple web forms that ask users to describe what and where the issue is located. The most basic sites simply have an email link. If

the user doesn't forget to report the issue, which happens often, reporting at a later time gives a much greater possibility for incorrect data.

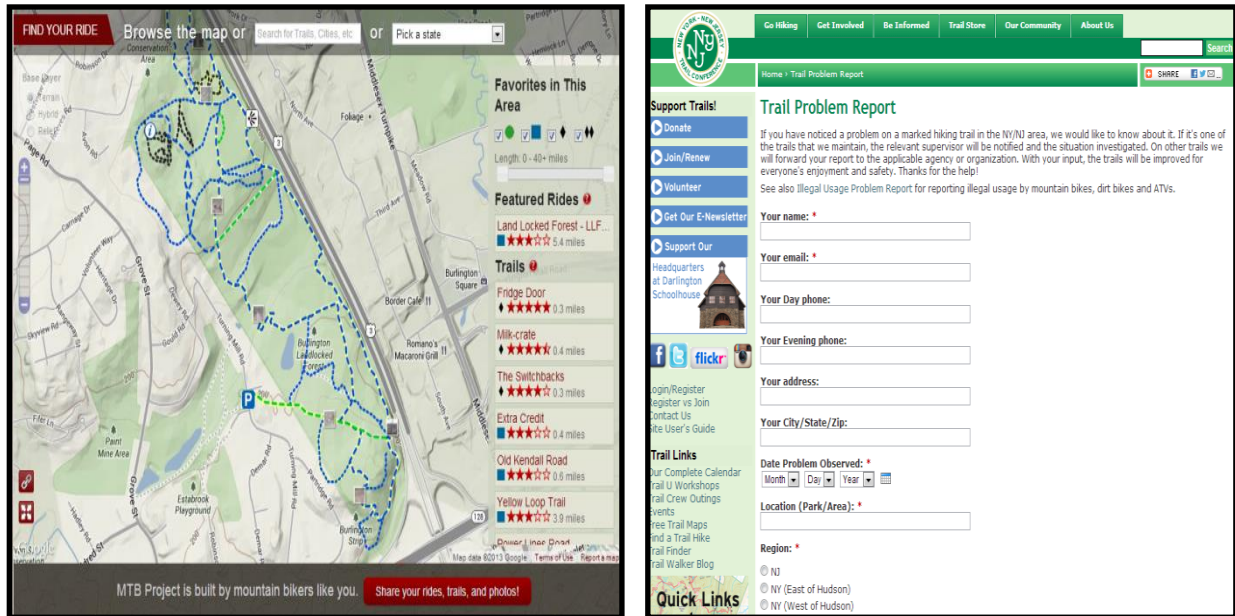


Figure 1: IMBA MTB Project web application (IMBA, 2014) and typical Trail Problem Report system (NYNJTC, 2014).

Motivation

As an avid outdoor enthusiast and volunteer, I am motivated to give something back to the outdoor community and the volunteer organizations that build and maintain our trails. With cost being an obvious factor for these organizations, open source tools often provide the freedom of customizability and wide community support. With this in mind, there are several questions that motivate this project:

Is it possible to reduce the costs and skills associated with trail maintenance by using open source technology?

Is there a better way to focus the volunteers' time on areas of trail that need the most

attention?

Is it possible to get the public and communities more involved in the process?

Goal

The primary goal of the project is to provide an integrated open source geospatial publishing and reporting platform that allows volunteer organizations to get up and running quickly.

Research Results

During the proposal phase, this project looked at several possible options for accomplishing its ultimate goal. These options ranged from a full geographic information system (GIS) stack to a complete cloud stack solution to a more piecemeal web based solution. Like the options, the requirements for each varied tremendously. A full GIS stack, as seen in Figure 2, required a big investment in infrastructure, including up front server hardware costs. All in one cloud based solutions were lighter on the initial hardware investments, but most would require monthly subscription fees for the life of the organization. The fees were usually based on the amount of data an organization hosted and how much traffic the data would receive. The final option was taking a piecemeal approach that combined several existing applications and ties them together using yet to be developed code. The final option has the lowest costs for organizations, but has the highest demand placed on technical skills.

The full GIS stack option was an open source desktop to server package that consisted of Quantum GIS (QGIS) as the desktop application for working with geospatial data, PostGIS for the backend database used to store geospatial data, and Boundless OpenGeo as the server application to publish geospatial data to the web (Boundless, 2014). The end goal would have

been to create a simple single install to lighten the technical load on organizations. Even with the simple install, this option was extremely hardware heavy and required a broad range of technical skills to manage and maintain. Since most people already have a personal computer, server hardware and maintenance now becomes the major barrier. What if the hardware barriers could easily be removed? Eliminating the heavy hardware requirements by taking advantage of cloud infrastructure leads to the cloud implementation option.

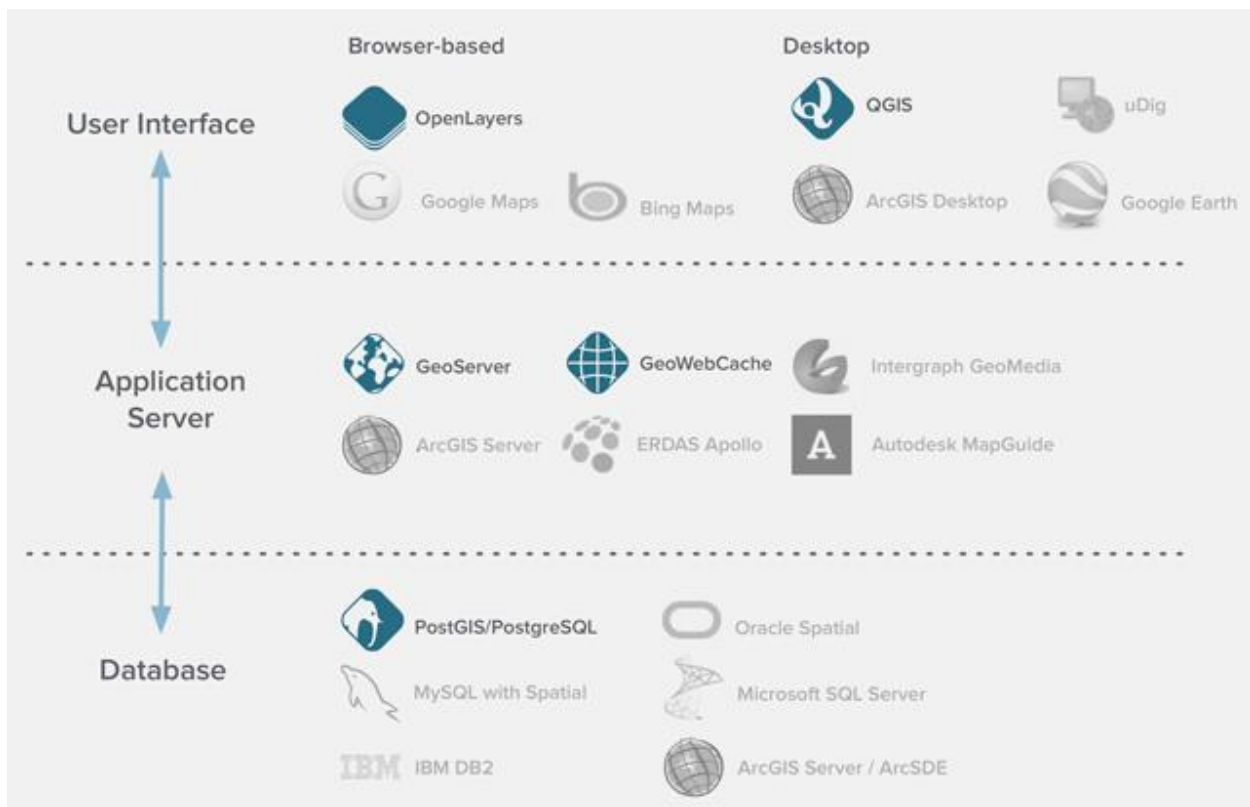


Figure 2: Typical full GIS open source stack (Boundless, 2014).

Many GIS cloud solutions currently exist, e.g. ArcGIS Online, GIS Cloud, MangoMap, etc., but far fewer are open source, e.g. CartoDB and Mapbox. An all in one cloud solution removes the OpenGeo server software from the initial option, which also eliminates the need to buy and maintain server hardware. The downside to the cloud option is an organization must be willing

to pay a monthly fee. For example, CartoDB, which is a cloud hosted PostGIS database with a very user friendly front end (Figure 3), will cost \$29 per month for their most basic plan (CartoDB, 2014). Because CartoDB and Mapbox are open source, organizations could eliminate the monthly fees by self-installing one of the open source projects on a cloud service like Amazon Web Services (AWS) or Windows Azure. While this option is appealing, it requires specialized skills in virtual machines and resource management. While this project didn't utilize the cloud option as a final solution, it certainly offers simple web mapping with the ability to do advanced SQL queries. Depending on the organizations financial and skills standing, both CartoDB and Mapbox are highly recommended for their advanced mapping options and friendly user interface (UI).

cartodb_id - number	the_geom - geometry	bronze - number	censusarea - number	count - number	division - string	drawseq - number	gold - number	region_1 - string	region_id - string	silver - number	state - string
1	Polygon	0	6422.628	0	9	1	0	West	4	0	HI
2	Polygon	2	66455.521	4	9	2	0	West	4	2	WA
3	Polygon	2	145545.801	4	8	3	0	West	4	2	MT
4	Polygon	0	30842.923	0	1	4	0	Northeast	1	0	ME
5	Polygon	0	69000.798	0	4	5	0	Midwest	2	0	ND
6	Polygon	0	75811	0	4	6	0	Midwest	2	0	SD
7	Polygon	1	97093.141	2	8	7	1	West	4	0	WY
8	Polygon	0	54157.805	5	3	8	2	Midwest	2	3	WI
9	Polygon	0	82643.117	0	8	9	0	West	4	0	ID
10	Polygon	0	9216.657	0	1	10	0	Northeast	1	0	VT

Figure 3: CartoDB PostGIS data table UI (CartoDB, 2014).

The final piecemeal option has the potential to be the cheapest and easiest to maintain. It attempts to eliminate the heavy hardware requirements, lighten the recurring monthly fees, and make installation quick and simple. It takes into account that volunteer organizations may not have members who are familiar with GIS concepts, but are quite web savvy. This option combines state of the art open source web technologies for data management and editing, while utilizing low cost shared web hosting services for publishing and drag and drop technology for easy installation. Finally, all data is stored in open source formats rather than proprietary data structures, as to not require a single tool or package.

Methodology

The project was broken into two stages, process and development. The process stage involved documenting how to use specific options of open source web technologies to create and manage trail data. The development stage involved coding a web mapping template for publishing trail data and reporting trouble issues.

Process

Collecting and maintaining trail data can be tough for volunteer organizations. The process stage is about documenting the tools and steps to make the procedure easy and independent of traditional GIS skills. Traditionally, trail data has been collected and managed by state or regional authorities that are responsible for managing parks. These days, trail systems are more often collections of trails that span multiple parks and communities or are located in unmanaged wilderness and forest areas.

The first step is to obtain or collect the trail data. If the trails are within park property, trail data may often be obtained by contacting the park directly. If the park service doesn't have recently updated trail data, it is up to the organization to collect the data on their own. This can be accomplished by traversing the trails while carrying a modern GPS, preferably in spring or fall when the leaves are off the trees. Manufacturers like DeLorme, Garmin, and Magellan offer consumer GPS units that are capable of recording routes in GPS exchange (GPX) format. If an organization is unable to obtain or collect trail data, a final option exists to refer to a map and draw the trails. In all three of the data scenarios, the same tool can be used to import, convert, edit, draw, or export the data. This tool is a new web application called geojson.io.

[Geojson.io](http://geojson.io) is an open source web application for creating, viewing, and sharing map data (geojson.io, 2014). It features drag and drop functionality for quick import and editing of many spatial file formats, e.g. comma separated values (CSV), GPX, ESRI shapefile (SHP), Keyhole Markup Language (KML), etc. If trail data is not pre-existing, geojson.io also provides a mechanism for drawing lines, as well as placing points or drawing polygons, as seen in Figure 4. Because geojson.io utilizes basemap data from OpenStreetMap (OSM), it's possible somebody outside the organization has already added the trail data to OSM. This means instead of drawing the trails, an organization could trace the existing trails and add whatever information they deemed appropriate.

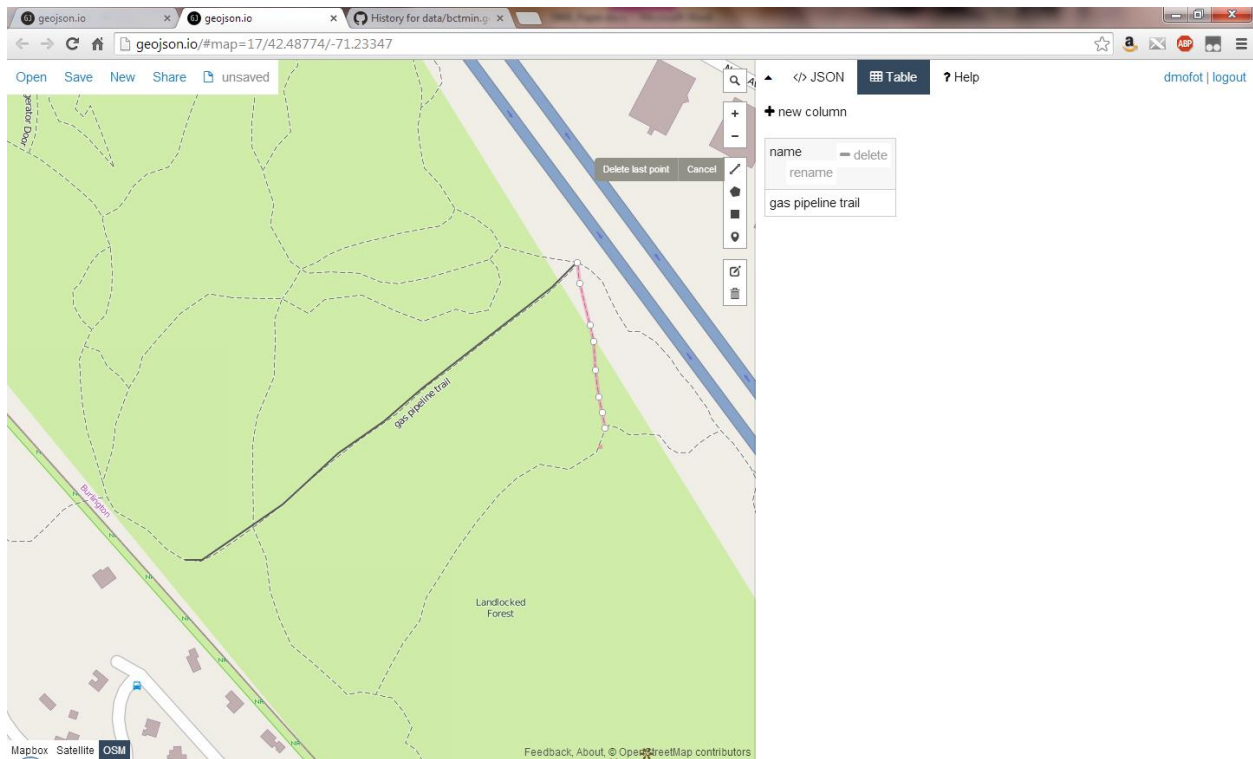


Figure 4: Trail drawing example using geojson.io with OSM basemap (geojson.io, 2014).

Data on geojson.io is presented in a simple table format, which will be familiar to anybody who has used a spreadsheet in Microsoft Excel, Apple Numbers, or Google Docs. While visually the table format is easy to understand, geojson.io also gives the user access to the underlying GeoJSON data format. GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON) and is extremely web friendly (geojson.org, 2014). As seen in Figure 5, data properties in GeoJSON are stored as 'key value pairs'. For example, if the user wanted to give each feature a name, they would simply type 'name' in the first table field, and the name of the feature in the second. Once the data has been imported or drawn and edited, the user clicks 'Save' to export the data in one of many formats, but preferably in GeoJSON. This first step is as a single user, but leads to the second step of data management.

```
▲ </> JSON Table ? Help dmfot | logout
1 {
2   "bbox": [
3     -71.50838,
4     41.986548,
5     -70.711176,
6     42.798879
7   ],
8   "type": "FeatureCollection",
9   "features": [
10    {
11     "type": "Feature",
12     "properties": {
13       "desig": "Dedicated",
14       "name": "Bay Circuit Trail",
15       "other": null,
16       "leng_miles": 6.27855,
17       "created_at": "2013-12-30T16:14:11Z",
18       "updated_at": "2013-12-30T16:14:11Z",
19       "cartodb_id": 1
20     },
21     "geometry": {
22       "type": "LineString",
23       "coordinates": [
24         [
25           -71.27589,
26           42.514792
27         ],

```

Figure 5: GeoJSON code example (geojson.io, 2014).

Proper data management is necessary for an organization that will have multiple users editing data. The second step utilizes a web application called Github, a popular repository for open source collaboration, code review, and code management (Github, 2014). It is a powerful tool for organizations, because it allows multiple user access and keeps track of all data changes referred to as commits (Figure 6). While Github has traditionally been geared towards code, it has recently implemented several key features for keeping track of geospatial data in the GeoJSON format.

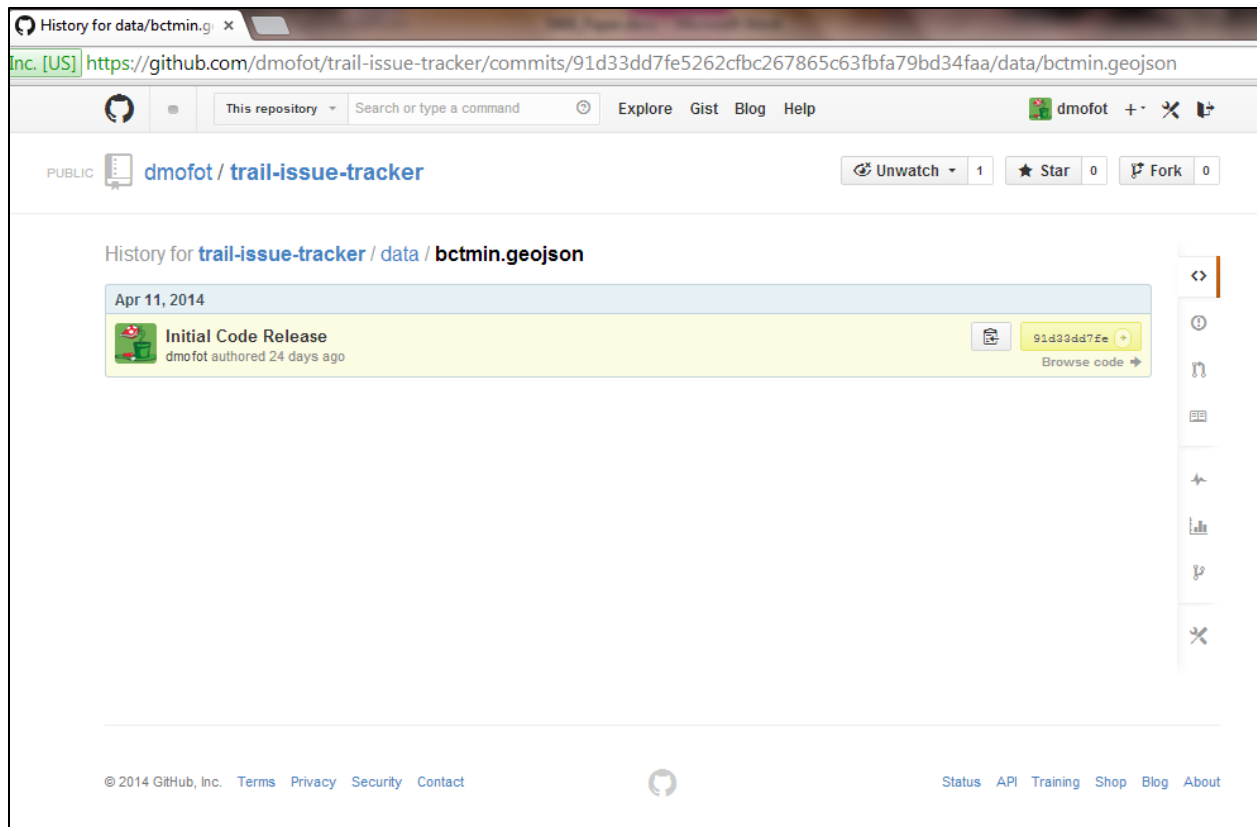


Figure 6: Github history of commits (Github, 2014).

GitHub allows for both user and organizational accounts for free. Signing up for a user account and creating the organizational account is as simple as signing up for webmail. Organizations can set user access permissions to control who can actually edit the trail data. GitHub also offers an application programming interface (API) that allows other tools to easily integrate. As seen in Figure 7, geojson.io takes advantage of GitHub's API to allow users to connect directly to their data stored on GitHub, make changes, and save the edited data back to GitHub with comments.

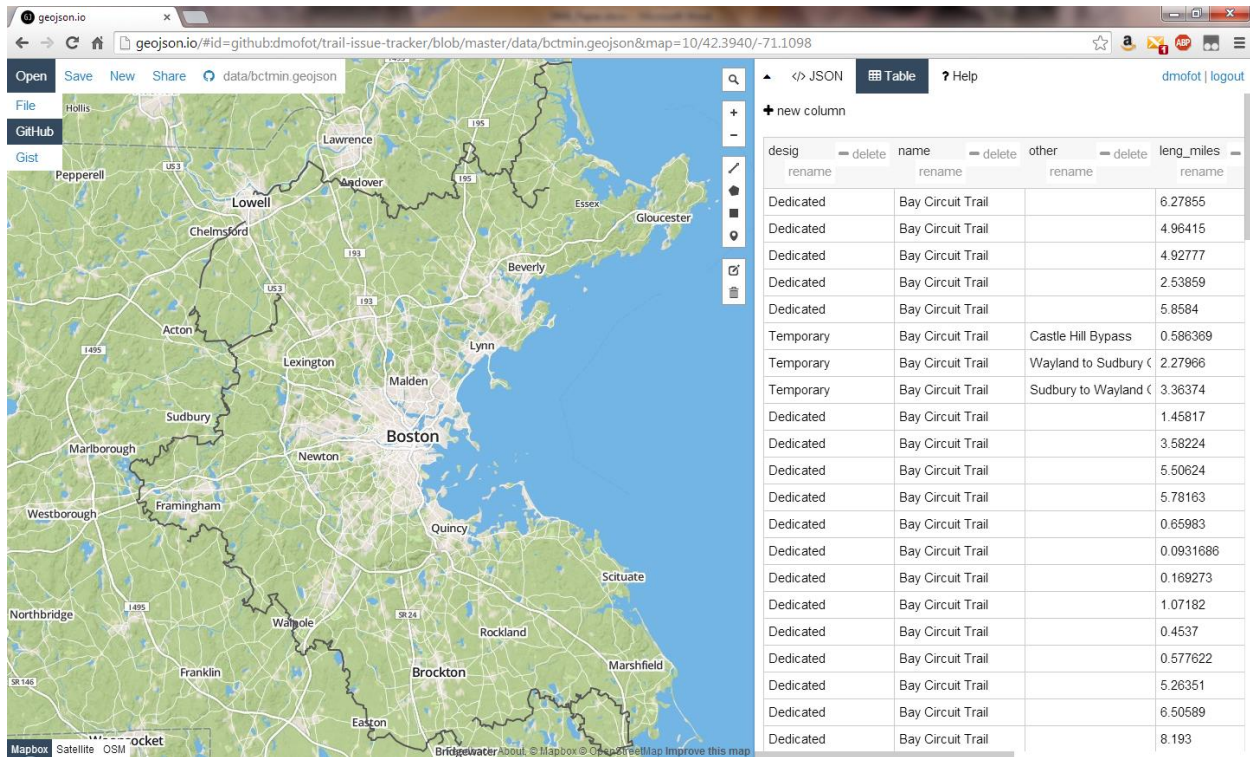


Figure 7: Example of opening and editing a GeoJSON file from Github on geojson.io (geojson.io, 2014).

If for some reason an error in the data was committed, the data differences can easily be compared (Figure 8) and stepped back in time on Github.

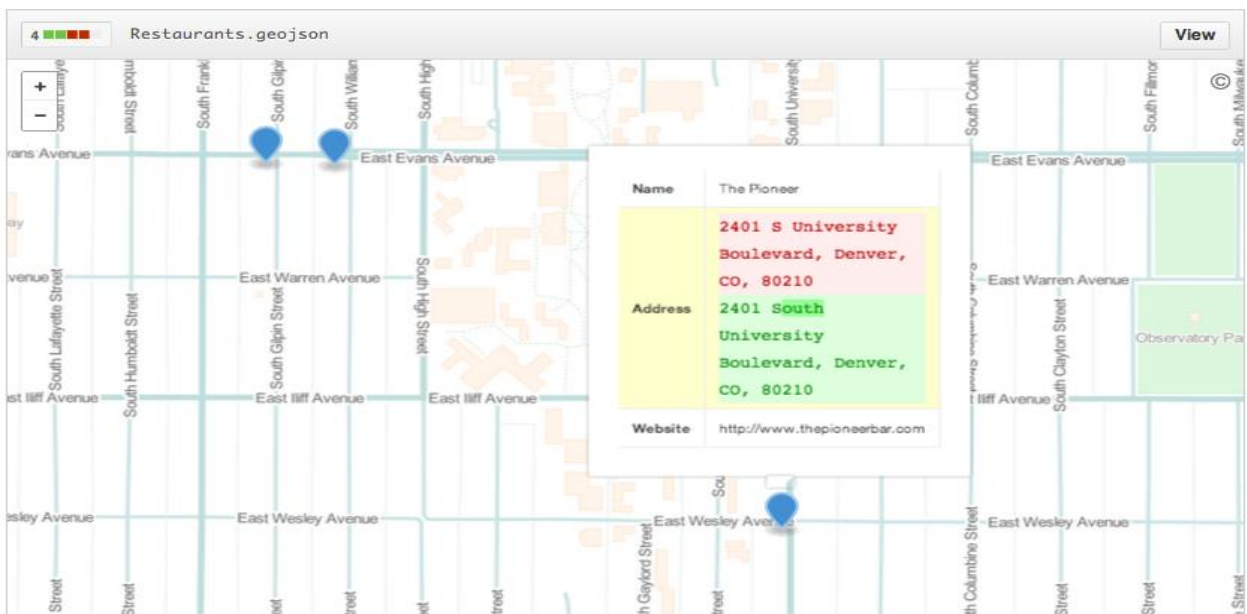


Figure 8: Github data commit comparison. Red is previous data and green is current (Github, 2014).

Development

In order to continue the lightweight piecemeal option, like the process stage, the development stage looked to current web technologies to build a web mapping template for publishing geospatial data. Several factors influenced the design of the template, from site navigation on desktop browsers and mobile phones to presentation and storage of data. The site template needed to be visually appealing and easy to navigate, no matter the device. The base of the template was coded in HTML5 and Javascript in addition to integrating several open source Javascript plugins.

Bootstrap is the front-end framework for developing responsive, mobile first projects and the navigation power behind sites like Twitter (Bootstrap, 2014). It allows a site to display differently depending on the device without having separate sites with device specific code. On mobile devices, site navigation collapses into a single button to save on valuable real estate.

The mapping is provided by an open source javascript library for mobile friendly interactive maps called Leaflet (Leaflet, 2014). Because Leaflet is lightweight, weighing in at only 33 KB of Javascript, it can easily be hosted on any shared web hosting service and quickly transmitted over cell phone networks. It makes ingesting and visualizing GeoJSON formatted data a snap and is the mapping power behind sites like Foursquare, Flickr, and Pinterest. It also has a developer friendly API, which has resulted in a long list of third party plugins. This template takes advantage of Leaflet's plugins to add a geolocation control for providing the current location of the user, marker clustering for keeping the map clean when there are a lot of trouble issue reports, and a URL hash plugin for keeping track of the map view to make sharing easy.

While there are several possible database options for storing the user contributed trouble reports, only one meets the lightweight theme. SQLite is a completely self-contained open source file database (SQLite, 2014). Unlike other options like MySQL, SQLite doesn't require a server or special configuration. In fact, a SQLite database file is completely portable and can be sent via email. This makes it possible to include the trouble reporting database with the site template without any further changes from the user.

Results

The final result is a mobile friendly trail mapping and trouble reporting site template called [trail-issue-tracker](#) (Trail-issue-tracker, 2014). First and most importantly, the template was built to be responsive, so it works different, but equally well on both a desktop and mobile environments. Bootstrap allows for a different user experience on mobile devices by collapsing elements like navigation, layer management, and the legend. The collapsing of elements provides more visible screen real estate for the map itself, as seen in Figure 9.

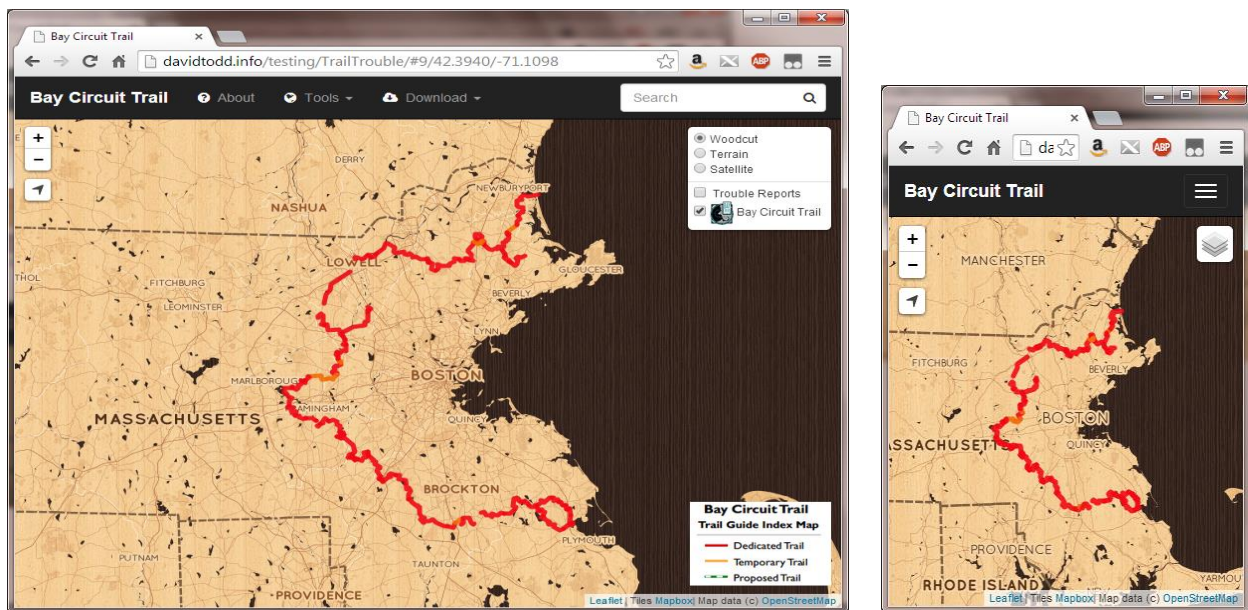


Figure 9: Desktop vs. mobile environment - example of collapsed navigation (Todd, 2014).

Second, the template comes with a search box that has the ability to search for places via the GeoNames service or search through feature attributes in the trail data. The search box integrates typeahead, which will start to autocomplete the search results after at least three characters are typed (Figure 10).

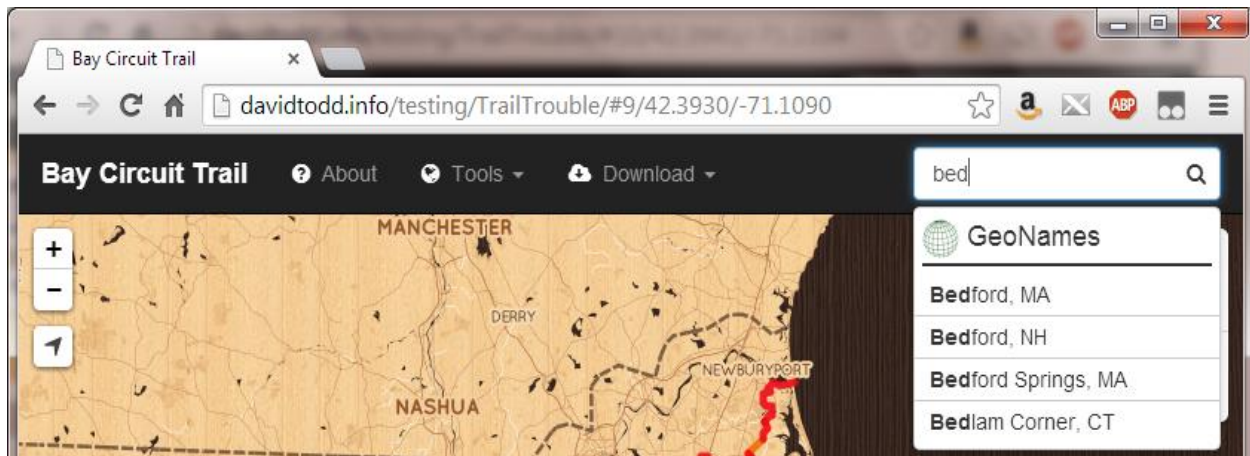


Figure 10: Autocomplete search box showing results (Todd, 2014).

Third, the template includes several elements to provide a friendly user experience. One example of these elements is a single button “zoom to current location” that changes color when activated. The map zooms to the current location provided by the device and shows a point with circle buffer that represents the accuracy error of the location (Figure 11). Another user friendly element seen in Figure 12 is the addition of zoom level and center latitude/longitude to the uniform resource indicator (URI), which allows users to email or bookmark specific map views. Other elements make the map experience cleaner for the user. A prime example is the use of marker clustering to avoid markers that appear to fall on top of other markers at certain zoom levels. The marker clusters break apart to show individual markers when the map is zoomed in or the marker cluster is clicked (Figure 13).

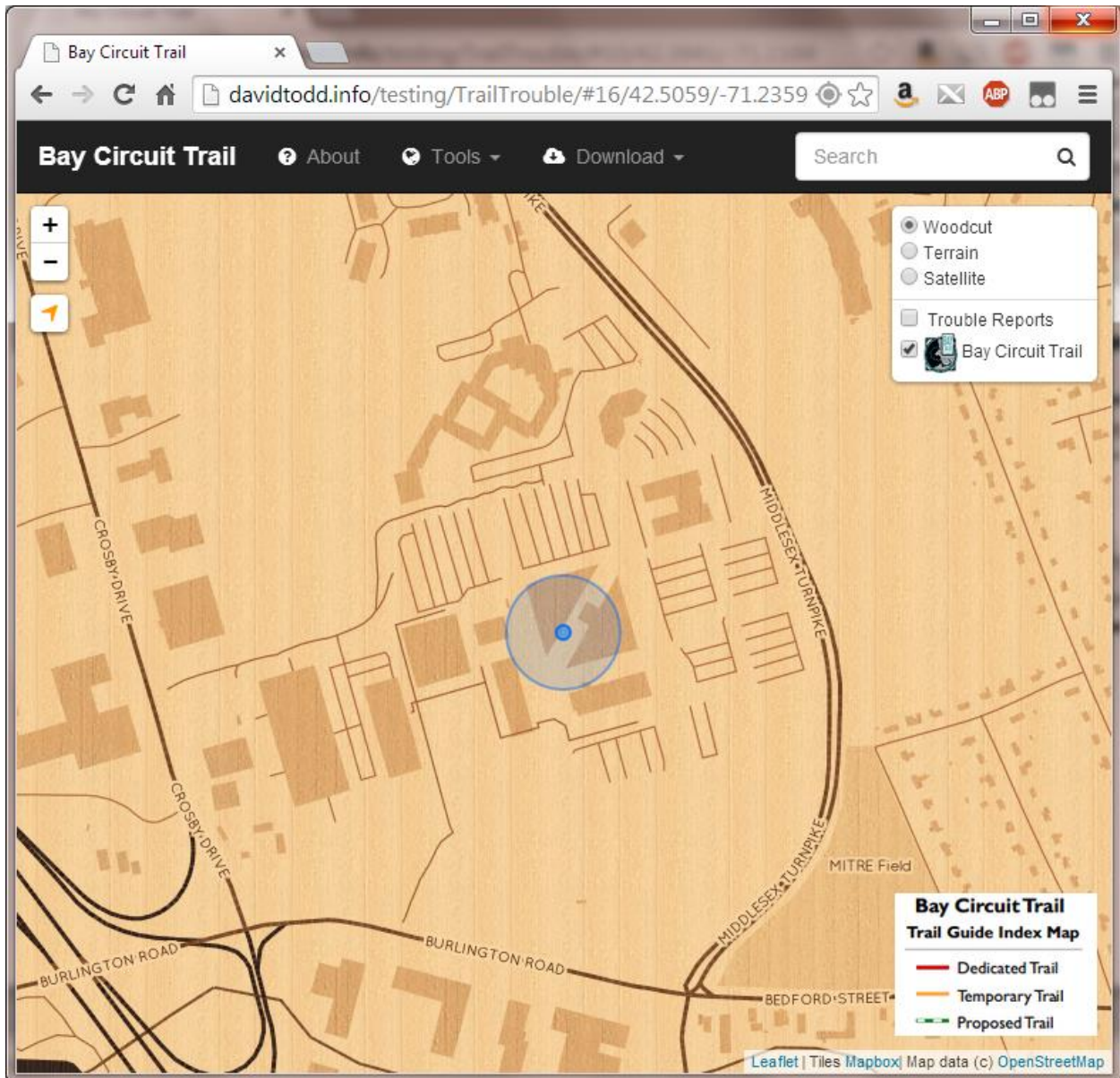


Figure 11: Zoom to current location, especially useful adding new trouble reports (Todd, 2014).

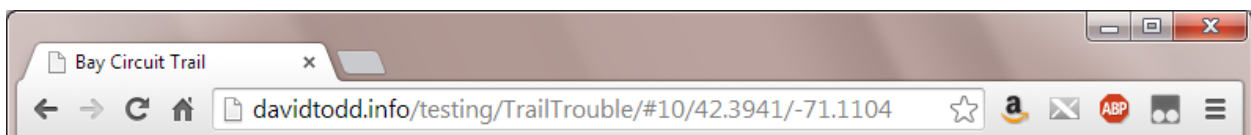


Figure 12: Custom URI that includes current map zoom level, latitude, and longitude (Todd, 2014).

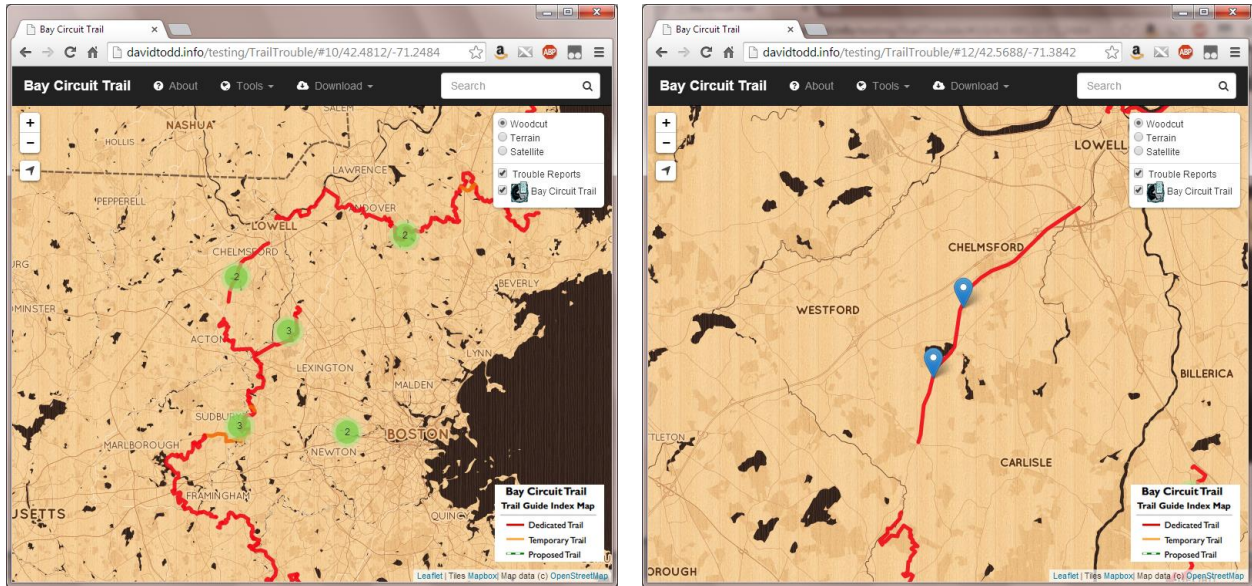


Figure 13: Trouble report marker clustering zoomed out vs. individual markers zoomed in (Todd, 2014).

Finally, the template includes an easy mechanism for both adding and removing user generated trouble reports. The Add and Remove Issue controls are under the Tools dropdown (Figure 14).

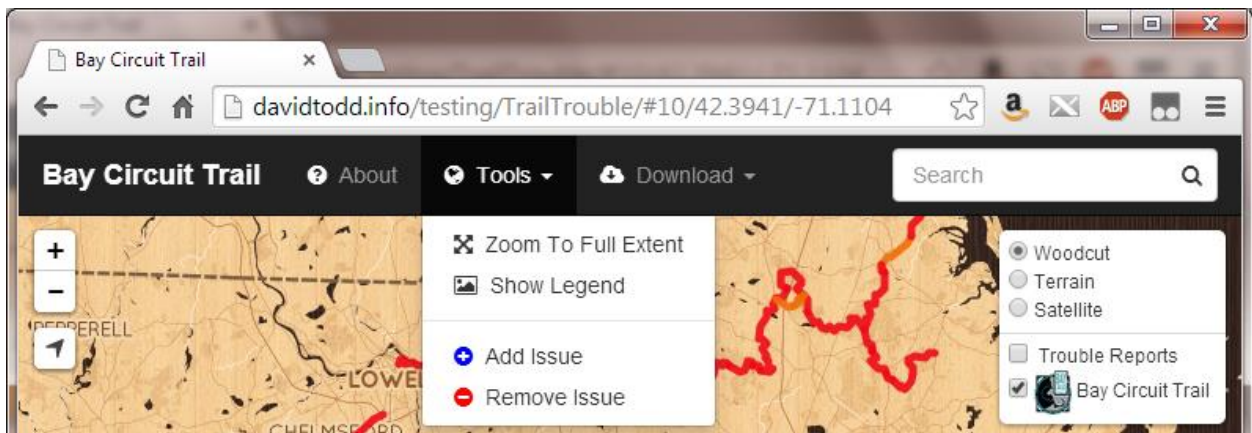


Figure 14: Add and Remove Issue located under the Tools dropdown in the navigation bar (Todd, 2014).

As seen in Figure 15, clicking Add Issue brings down a modal that explains the process of dropping a marker.

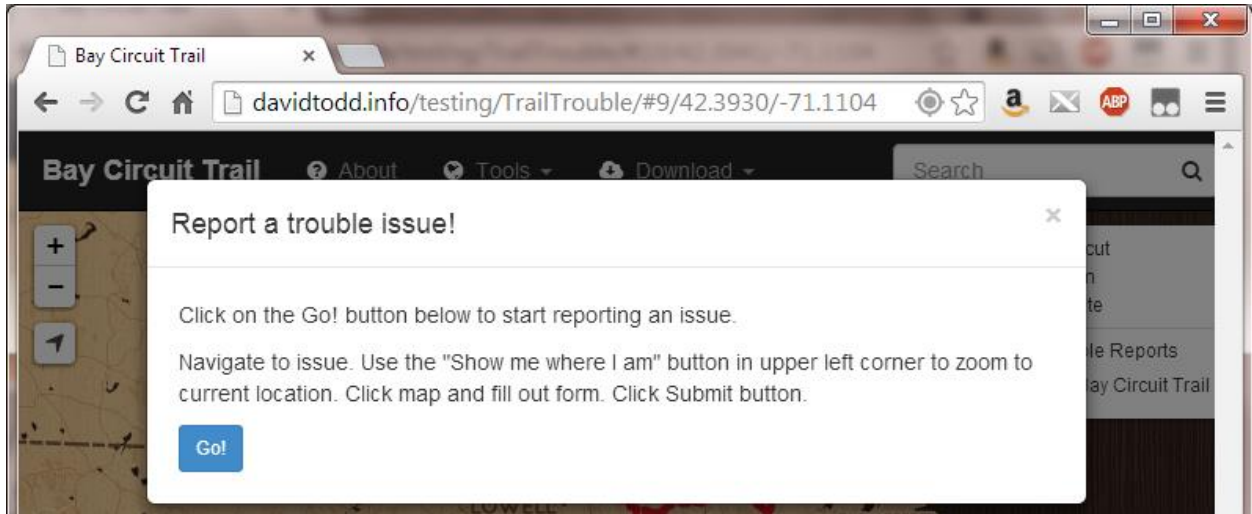


Figure 15: Add Issue modal explaining how to submit a trouble report (Todd, 2014).

The user then clicks or touches the map to place their trouble report on the map. The marker is draggable, so the user doesn't have to worry about placing the marker in the exact spot the first time. After placing a marker on the map, a popup appears that contains the trouble report form (Figure 16).

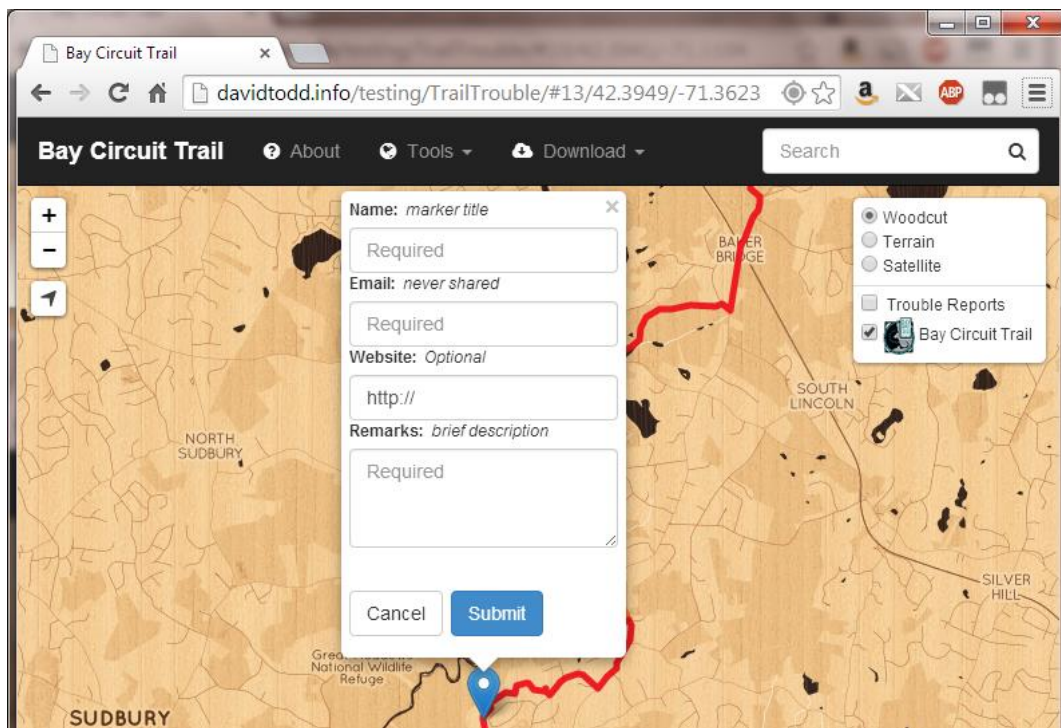


Figure 16: Popup containing trouble report form (Todd, 2014).

Filling in the trouble report form and clicking Submit places the trouble report into the SQLite database, sends the user a thank you email that includes a random token for removing their issue, and adds the report immediately to the map. If the user clicks on the marker representing their trouble report, the popup now contains the information they submitted (Figure 17).

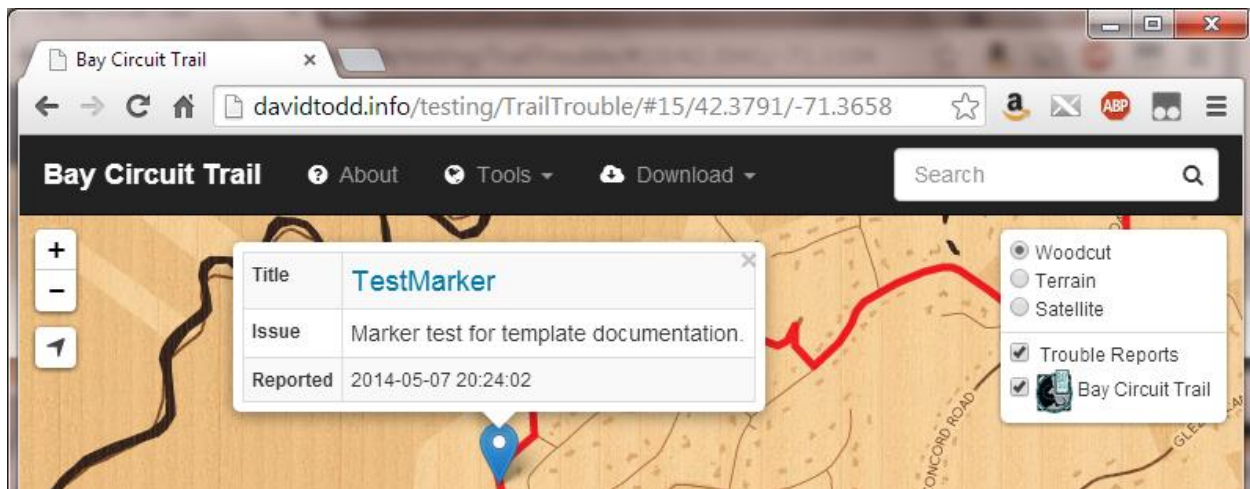


Figure 17: Newly added trouble report with new popup containing user submitted content (Todd, 2014). Removing an issue is just as easy. Selecting Remove Issue from the Tools dropdown brings down the Remove Issue modal. The modal asks the user for the email that was used to submit the issue and the random token that was sent to that email (Figure 18). Clicking Submit removes the issue from the SQLite database and immediately removes the issue from the map (Figure 19). The trail-issue-tracker, like many other open source projects, is available on Github at <https://github.com/dmofot/trail-issue-tracker> (Trail-issue-tracker, 2014). Any organization can download the template, add their data, upload the files to their own shared web hosting service, and start spreading the word about their trails and collecting user generate trouble reports.

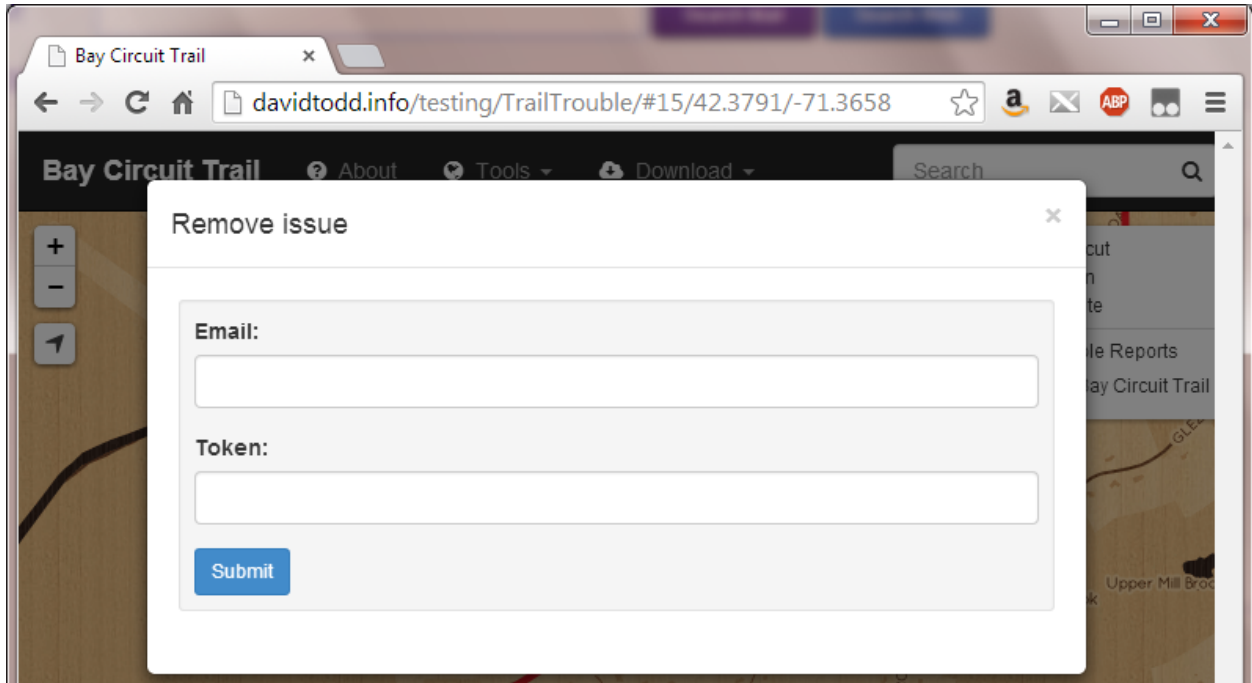


Figure 18: Remove Issue modal (Todd, 2014).

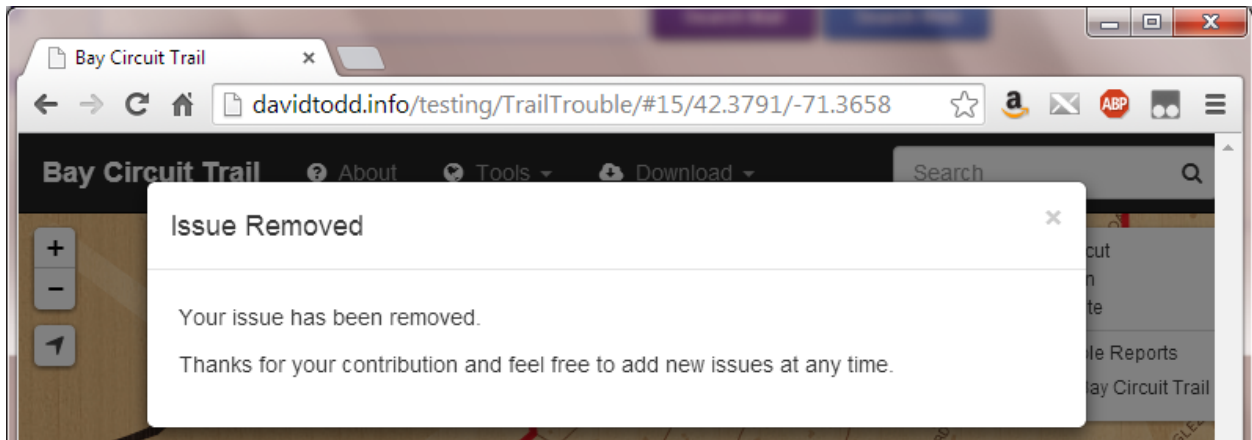


Figure 19: Issue removed modal (Todd, 2014).

Conclusions

While the initial trail-issue-tracker is complete, there are a couple of future enhancements that are being considered. Currently, the template is a full page map with all navigation and features built into the map. It could be beneficial to provide multiple example templates for varying the visual appearance. For example, visually separating the map and the

trouble reporting form, but keeping the two elements connected in the background could be advantageous. This would be useful for embedding a smaller version of the map into an existing site for the sole purpose of trouble reporting and dropping the navigation aspect. Additional future enhancements involve improving the trouble reporting aspect. Adding an authentication method for validating trouble reports would give the ability to restrict the map to showing just validated trouble reports, while the others wait in the database. Trouble reporting is simply a form for accepting text reports, but it could be an extra touch to add the ability to submit photos with the reports.

Researching and building the trail-issue-tracker project provided several lessons learned. First, while there are many proprietary solutions for managing and publishing geospatial data, there are also many cutting edge open source options that can reduce both the cost and skills required to implement. Using open data formats ensures data access to a wider audience. Second, geo-collaboration is not easy or intuitive. Considering how long code collaboration has been around, few options exist that target collaborative editing and versioning of geospatial data. Even less open source options exist, but hopefully this will change as places like Github and Boundless are working on Git based distributed versioning of geospatial data like GeoGit (GeoGit, 2014). Third, current trouble reporting systems are lackluster and require users to jump through hoops for basic information. Based on feedback from user testing of the trail-issue-tracker template, users said they initially used the site to simply see their current location on the trail to aid with navigation. Already being on the site for navigation, users found they were more likely to use the trouble reporting system, especially since it was easy and intuitive. This takes advantage of user generated content, also known as volunteered geographic

information (VGI), to focus a volunteer's time on areas that need immediate attention. As with most public facing projects, the Keep It Simple Stupid (KISS) method helps to obtain public participation. The trail-issue-tracker template project encourages organizations to take control of their trail data and take advantage of trail users willing to generate information.

Bibliography

Literature References

Bishr, M. and Janowicz, K. Can we Trust Information? - The Case of Volunteered Geographic Information. In Towards Digital Earth: Search, Discover and Share Geospatial Data, Workshop at Future Internet Symposium, September 20th, 2010, Berlin, Germany, Volume Vol-640. CEUR-WS, 2010.

Sui, D. and Elwood, S. and Goodchild, M. Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice. Web. Dordrecht; New York: Springer, August 10, 2012.

Web References

BootLeaf. <https://github.com/bmcbride/bootleaf> (Last accessed May 5, 2014)

Bootstrap. <http://getbootstrap.com/> (Last accessed May 6, 2014)

Boundless, formerly OpenGeo. <http://boundlessgeo.com/> (Last accessed May 5, 2014)

CartoDB. <http://cartodb.com/> (Last accessed May 5, 2014)

GeoGit by Boundless. <http://geogit.org/> (Last accessed May 5, 2014)

geojson.io. <http://geojson.io> (Last accessed May 5, 2014)

geojson.org. <http://geojson.org/> (Last accessed May 5, 2014)

GeoStories – by National Geographic. <http://www.geostories.org/> (Last accessed May 5, 2014)

GitHub. <http://github.com> (Last accessed May 5, 2014)

IMBA. MTB Project. <http://www.mtbproject.com/> (Last accessed May 5, 2014)

Kebler, Carsten and de Groot, Rene. Trust as a Proxy Measure for the Quality of VGI in the Case of OSM. <https://speakerdeck.com/crstn/trust-as-a-proxy-measure-for-the-quality-of-vgi-in-the-case-of-osm> (Last accessed May 5, 2014)

Leaflet. <http://leafletjs.com/> (Last accessed May 5, 2014)

MapBox. <http://www.mapbox.com/> (Last accessed May 5, 2014)

NYNJTC. New York-New Jersey Trail Conference Trail Problem Report.

<http://www.nynjtc.org/webform/trail-problem-report> (Last accessed May 5, 2014)

Olsen. Mobile App vs. Site Infographic. <http://www.pinterest.com/pin/566961040560470850/>
(Last accessed May 5, 2014)

OpenStreetMap. <http://www.openstreetmap.org/about> (Last accessed May 5, 2014)

OSGeo.org. <http://www.osgeo.org/> (Last accessed May 5, 2014)

OSM In A Box. <http://2010.foss4g.org/presentations/3877.pdf> (Last accessed April 11, 2014)

Peddi, Ashok. Free Open Source GIS.
http://www.academia.edu/460275/Free_Open_Source_GIS (Last accessed May 5, 2014)

Portable GIS. <http://www.archaeogeek.com/portable-gis.html> (Last accessed May 5, 2014)

PostGIS. <http://postgis.net/> (Last accessed May 5, 2014)

QGIS. <http://qgis.org> (Last accessed May 5, 2014)

Roth, Robert E., Donohue, Richard G., Sack, Carl M., Wallace, Timothy R., and Buckingham, Tanya, M.A. A Process for Assessing Emergent Web Mapping Technologies.
<http://www.slideshare.net/roth/a-process-for-assessing-emergent-web-mapping-technologies> (Last accessed May 5, 2014)

SQLite. <https://sqlite.org/> (Last accessed May 5, 2014)

TileMill | Mapbox. <https://www.mapbox.com/tilemill/> (Last accessed May 5, 2014)

Todd, David. <http://www.davidtodd.info/testing/TrailTrouble/> (Last accessed May 5, 2014)

Trail-issue-tracker. <https://github.com/dmofot/trail-issue-tracker> (Last accessed May 5, 2014)