Nooksack Salmon Enhancement Association

GeoMapping Project

Taichen Rose & Zach Cooper & Juniper Still

Repository

https://github.com/NSEA-Whatcom/NSEA-GeoMapping

Roles

Developer: Taichen Rose (jatr812@gmail.com) | Fall 2020

Taichen's Documentation Here

Developer: Zach Cooper (zach@raincitysolutions.com) | Spring/Summer 2020

Zach's Documentation Here

Designer: Juniper Still (junistill@gmail.com) | Spring 2020

Login Information

Github Information

Username: NSEA-Whatcom

Password: Chinook3057!

Mapbox Information

Username: NSEAMapping

Password: Nsea4Mapbox!

MongoDB Information

Username: ajohnson@n-sea.org

Password: Nsea4Mongo!

MongoDB User

Username: Admin

Password: Nsea4Admin

App Login

This is used when locally logging into the application

Username: NseaAdmin

Password: Nsea4Map!

Project Information

Goal

Amy Johnson envisions an interactive map that can be embedded into the NSEA website showcasing all their restoration projects. Each restoration project will display a different icon whether that be a fishery project, planting project, etc. When users click on these icons, a small tooltip will display project information. Some of these tooltips will display a URL, which will lead to a short page of project information in depth.

Progress

The first interns to work on this project, Zach and Juniper implemented a React native application. This application uses React as the front-end framework, utilizes MongoDB as a NoSQL database, and express/Node.js for server-side processing. They were able to get map visuals and icons designed using MapBox. Creating a local React application, they have a database which has a range of different columns.

When running the program (can see how to run the program looking at the README.md in GitHub repo), you can see the progress that shows. Icons pop up and users can click on those icons for a small tooltip of information to display.

Taichen was given the task to update the database and add more columns to it. She also redocumented a lot of the documentation for it to be easier to read. Taichen started creating watershed lines, but later stopped since that project was scrapped. Her task was to get the React application onto the website. She realized that you wouldn’t be able to just lift the React application without using some sort of third-party application. After speaking with Zach, it was envisioned that they would host this application using Azure (Microsoft’s cloud computing system). Once this is hosted on Azure, you would embed that hosted website into NSEA’s square space website.

Taichen was not able to dig deeper to see if this was fully possible or start hosting it on Azure. She investigated what Mapbox has to offer, since initially we were using Mapbox to grab the map visuals. Tai was able to use Mapbox's database and map GUI to create a dataset of the map that is necessary. This is currently embedded into NSEA's square space website working fine. She also has made a template for Amy to add customer information efficiently into Mapbox.

Future Improvements

If you continue going down the 'just use Mapbox' method, figure out how to maintain this better and faster if possible.

If continue going through the React and Azure method, investigate Azure pricing. Figure out how to host this React app onto their web application, then investigate square space website embedding to add the map into NSEA's website.

Diagrams

To get UML, Site Navigation, MongoDB Entity Relationship Diagram, Back End Routes Diagram, Database Schema, and User Case UML Diagram check Zach's documents in the repository.

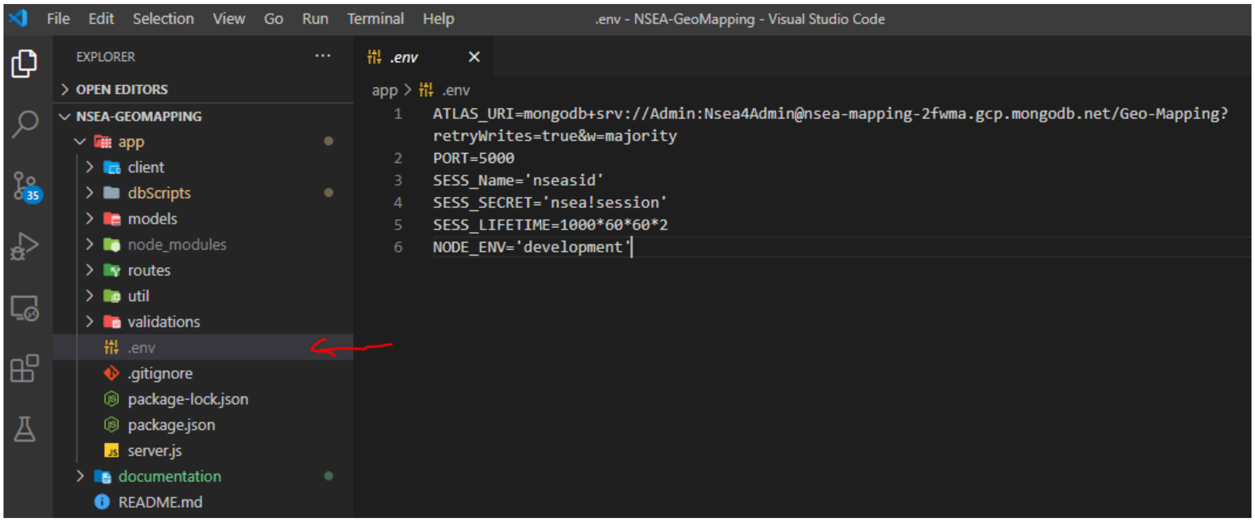
React Application

Setup

Step 1: To get this setup, follow the GitHub repo README. Before you run the nodemon and start server stuff you need to add your .env files

Step 2: In the app folder create an .env file. Paste this into that .env file

```
ATLAS_URI=mongodb+srv://Admin:Nsea4Admin@nsea-mapping-2fwma.gcp.mongodb.net/Geo-Mapping?retryWrites=true&w=majority
PORT=5000
SESS_Name='nseasid'
SESS_SECRET='nsea!session'
SESS_LIFETIME=1000*60*60*2
NODE_ENV='development'
```

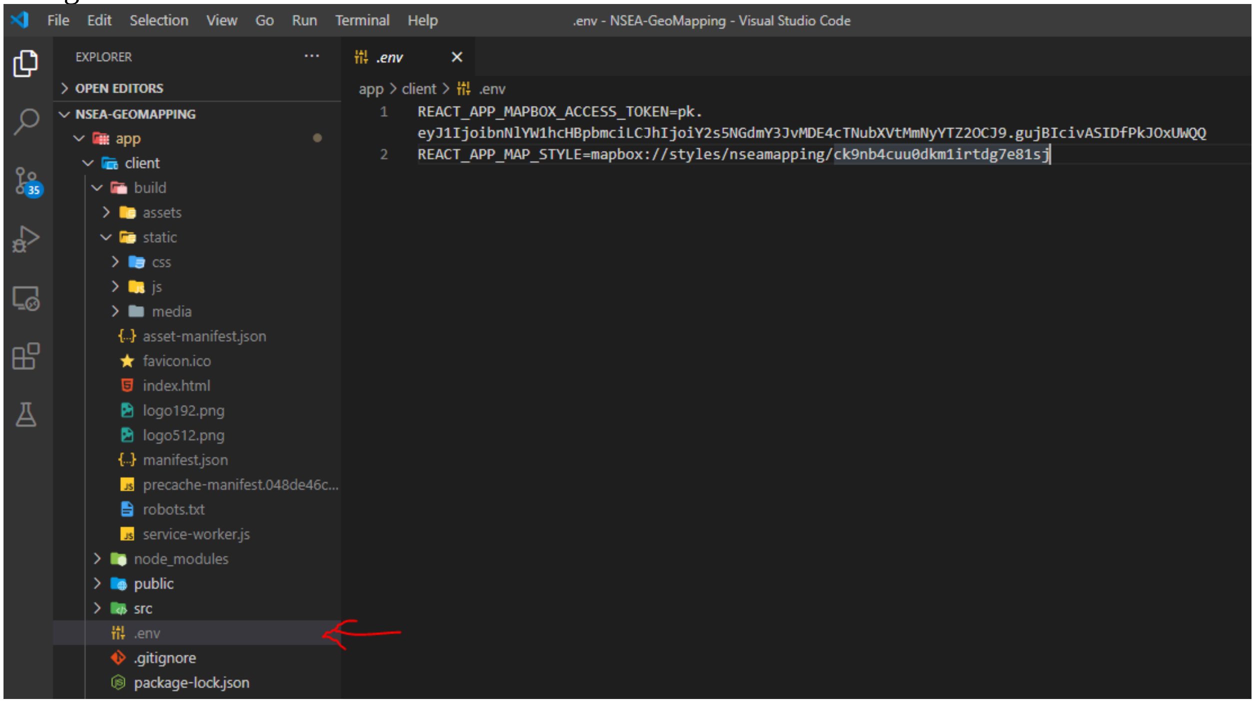


```
ATLAS_URI=mongodb+srv://Admin:Nsea4Admin@nsea-mapping-2fwma.gcp.mongodb.net/Geo-Mapping?retryWrites=true&w=majority
PORT=5000
SESS_Name='nseasid'
SESS_SECRET='nsea!session'
SESS_LIFETIME=1000*60*60*2
NODE_ENV='development'
```

Step 3: In the app/client folder create an .env file. Paste this into that .env file

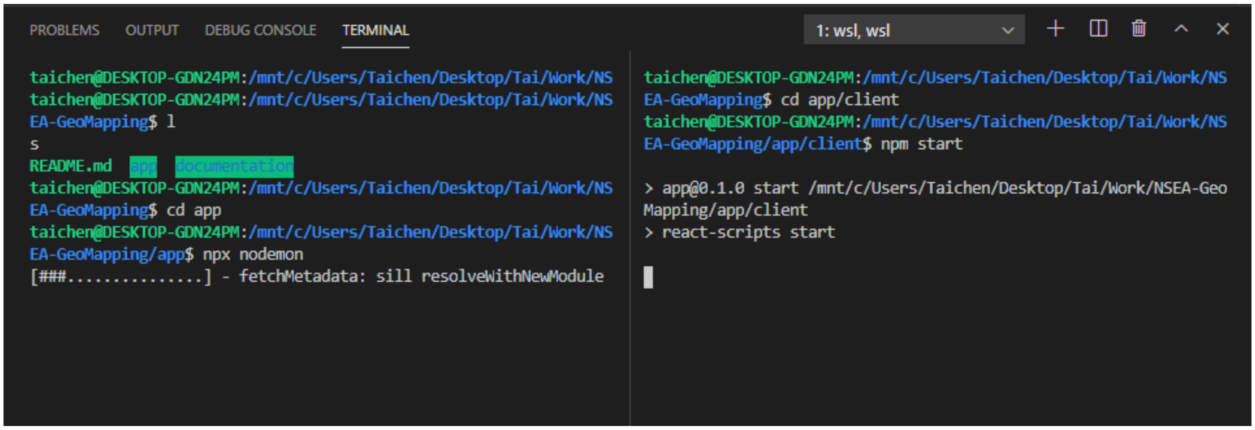
```
REACT_APP_MAPBOX_ACCESS_TOKEN=pk.eyJ1IjoibnNlYW1hcHBpbmciLCJhIjoiY2s5NGdmY3JvMDE4cTNubXVtMmNyYTZ2OCJ9.gujBIcivASIDfPkJOxUWQQ
REACT_APP_MAP_STYLE=mapbox://styles/nseamapping/ck9nb4cuu0dkm1irtdg7e81sj
```

Note: The MapAccess Token and Map Style can be found under the "Styles" tab next to the Datasets/Tilesets. Click on the map Details you want, and then click share (don't share it) but you will see the map token and map style ID. So, if you ever want to change the map, you will change the information.



```
REACT_APP_MAPBOX_ACCESS_TOKEN=pk.eyJ1IjoibnNlYW1hcHBpbmciLCJhIjoiY2s5NGdmY3JvMDE4cTNubXVtMmNyYTZ2OCJ9.gujBIcivASIDfPkJOxUWQQ
REACT_APP_MAP_STYLE=mapbox://styles/nseamapping/ck9nb4cuu0dkm1irtdg7e81sj
```

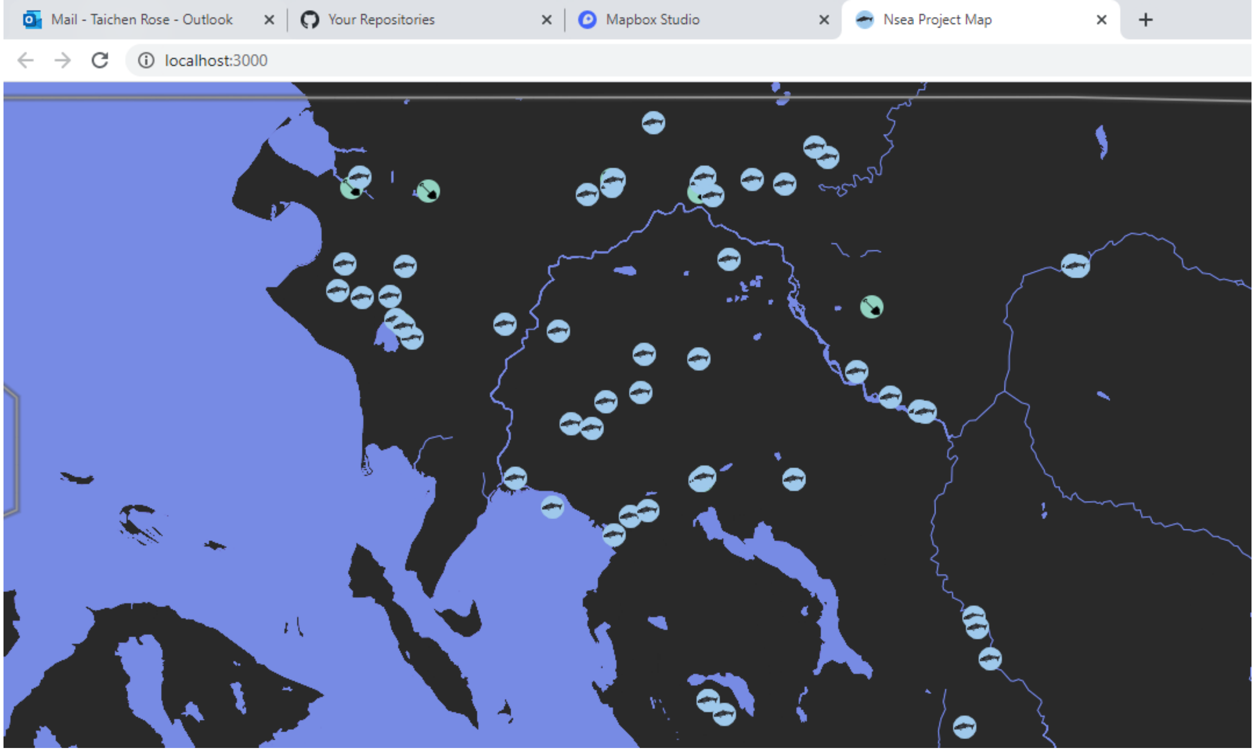
Step 4: As discussed in the GitHub repo README.md, after you have everything ready. Open two terminals. In one terminal go to /app folder. Type in npx nodemon (or the other command as show in the README.md) in the other terminal go to app/client and type in npm start.



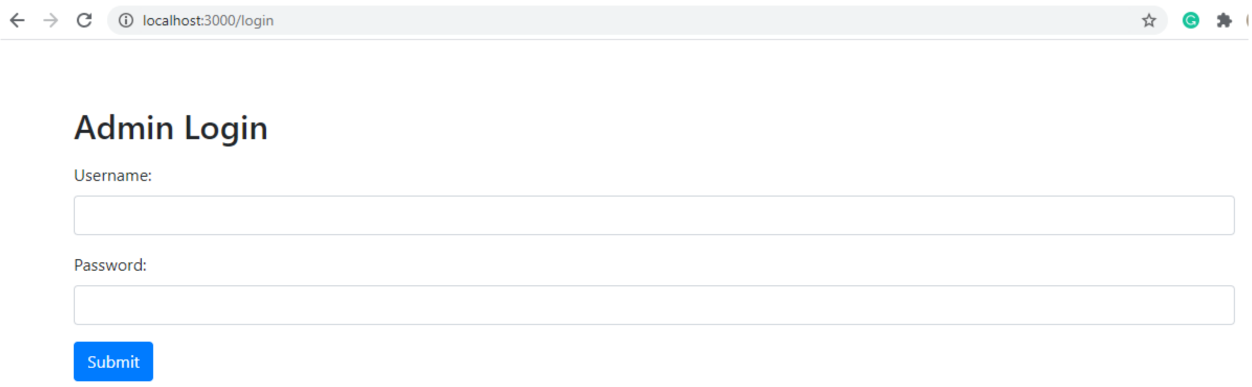
```
taichen@DESKTOP-GDN24PM:/mnt/c/Users/Taichen/Desktop/Tai/Work/NS
taichen@DESKTOP-GDN24PM:/mnt/c/Users/Taichen/Desktop/Tai/Work/NS
EA-GeoMapping$ l
s
README.md  app  documentation
taichen@DESKTOP-GDN24PM:/mnt/c/Users/Taichen/Desktop/Tai/Work/NS
EA-GeoMapping$ cd app
taichen@DESKTOP-GDN24PM:/mnt/c/Users/Taichen/Desktop/Tai/Work/NS
EA-GeoMapping/app$ npx nodemon
[###..............] - fetchMetadata: sill resolveWithNewModule
```

```
taichen@DESKTOP-GDN24PM:/mnt/c/Users/Taichen/Desktop/Tai/Work/NS
EA-GeoMapping$ cd app/client
taichen@DESKTOP-GDN24PM:/mnt/c/Users/Taichen/Desktop/Tai/Work/NS
EA-GeoMapping/app/client$ npm start

> app@0.1.0 start /mnt/c/Users/Taichen/Desktop/Tai/Work/NSEA-Geo
Mapping/app/client
> react-scripts start

▌
```

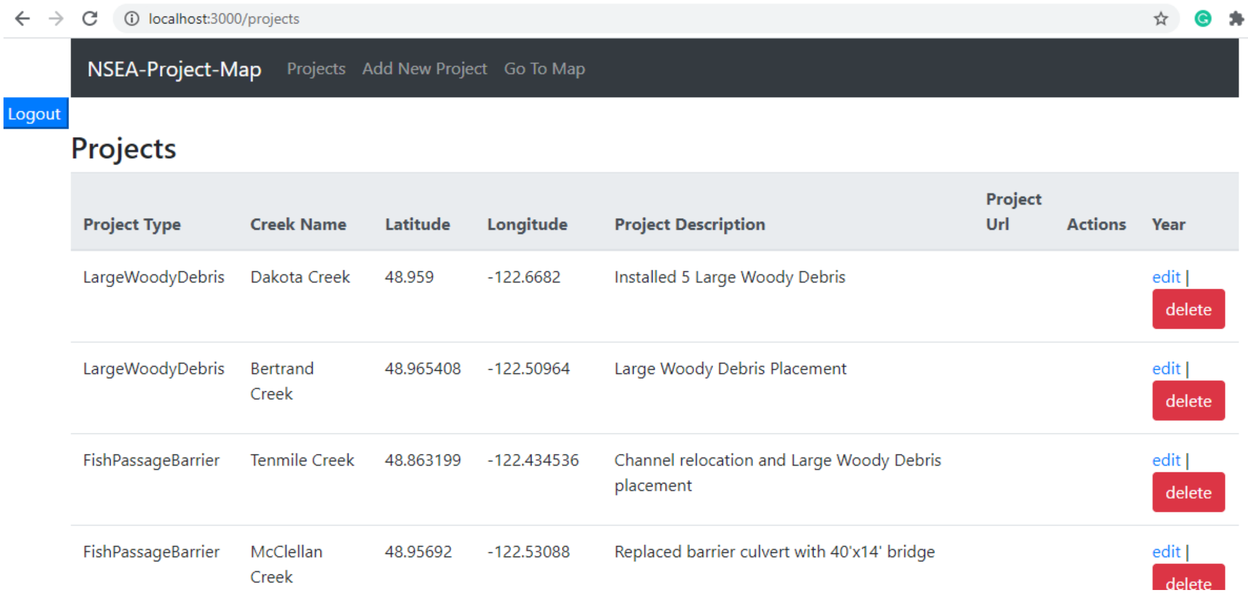
Step 5. When the window pops up, you can see the map with fish icons. This is the basic design Zach and Juniper had. Taichen only added the year field to the database for this.



To see the app login, at the URL for localhost:300/ manually type in localhost:3000/login



Once doing so it will take you to the username/password. You will use the credentials above on this document for the App Login. When logged in you, will see all the projects.

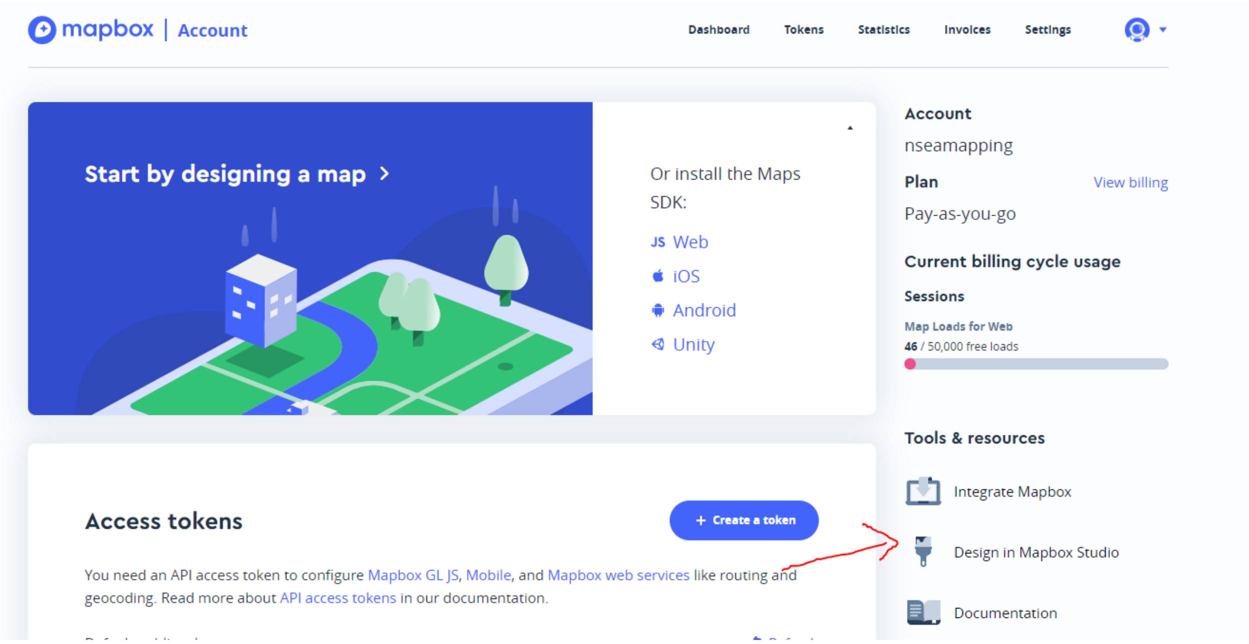


Project Type	Creek Name	Latitude	Longitude	Project Description	Project Url	Actions	Year
LargeWoodyDebris	Dakota Creek	48.959	-122.6682	Installed 5 Large Woody Debris			edit | delete
LargeWoodyDebris	Bertrand Creek	48.965408	-122.50964	Large Woody Debris Placement			edit | delete
FishPassageBarrier	Tenmile Creek	48.863199	-122.434536	Channel relocation and Large Woody Debris placement			edit | delete
FishPassageBarrier	McClellan Creek	48.95692	-122.53088	Replaced barrier culvert with 40'x14' bridge			edit | delete

The add new projects section has not been finished yet. That part is still a work in progress.

Mapbox Application

Login to Mapbox, then click on My Account. It will lead you to a page like this. For any design changes, or any data updates, click on “Design in Mapbox Studio” button in the lower right-hand corner.

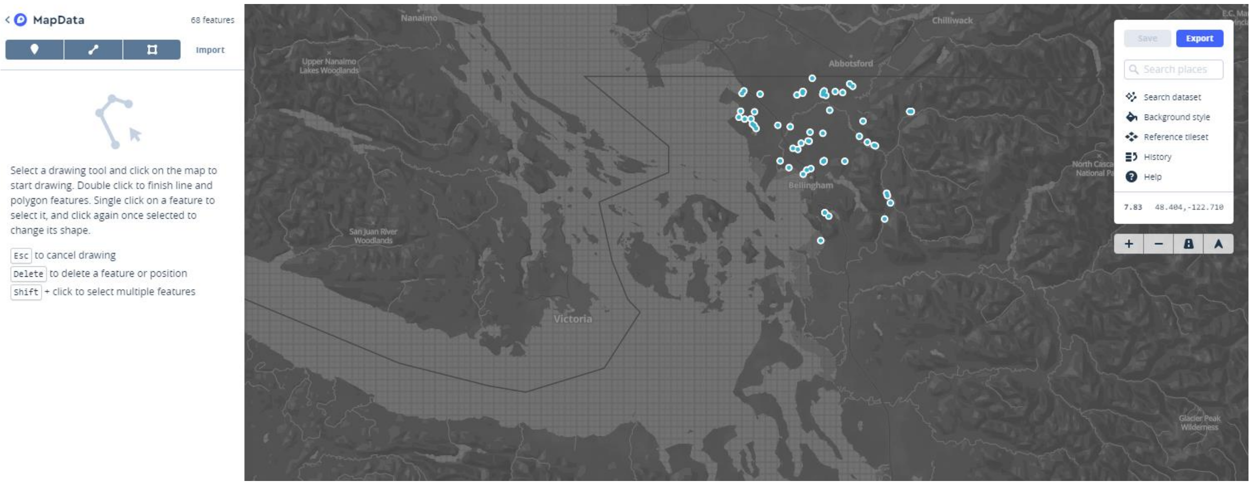


You need an API access token to configure Mapbox GL JS, Mobile, and Mapbox web services like routing and geocoding. Read more about API access tokens in our documentation.

Once you click on "Design in Mapbox Studio" it will take you to Styles page. But in the top right corner of the screen, you will see "Styles" "Tilesets" and "Datasets" I will explain what each mean below.

Datasets

Data sets is where users add their user data. They can upload new data using GeoJSON, JSON, or .csv format. Currently we keep adding data to our MapData dataset. When clicking on MapData this screen will show up.



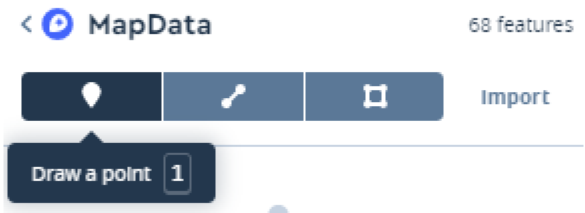
Select a drawing tool and click on the map to start drawing. Double click to finish line and polygon features. Single click on a feature to select it, and click again once selected to change its shape.

Esc to cancel drawing
Delete to delete a feature or position
Shift + click to select multiple features

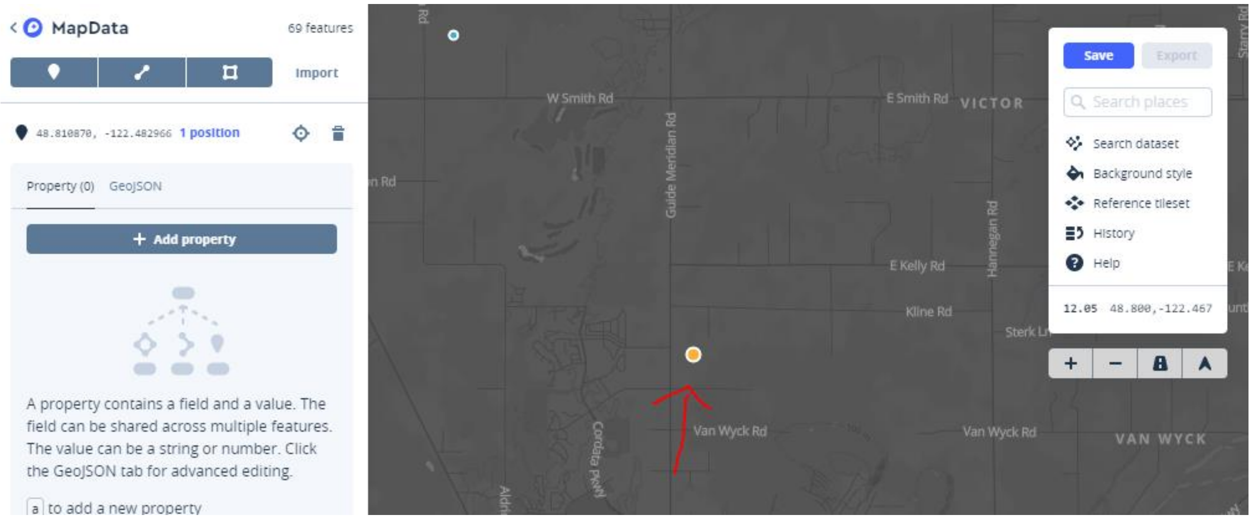
Adding Data Manually (One Point)

If we want to add data in manually, in the top right corner, there is a box. We can search places or enter the latitude and longitude. In the same white box, you can see when you move your curser, the latitude and longitude points change accordingly.

Once you find the exact location, in the left-hand corner of the screen click on the “Draw a point” button.

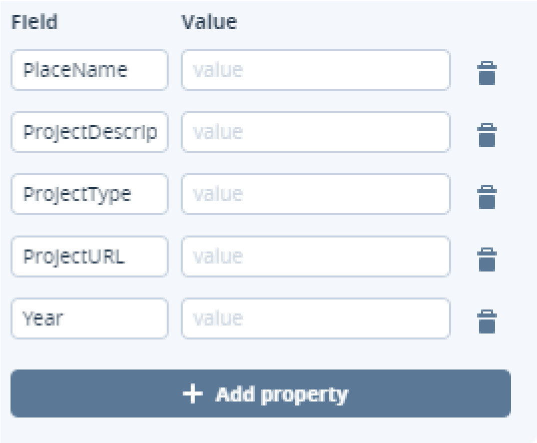


Clicking on that, go to the area and click on the map where the new data point is.

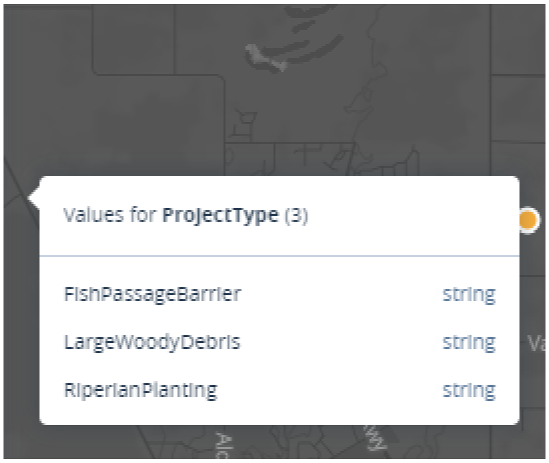


A property contains a field and a value. The field can be shared across multiple features. The value can be a string or number. Click the GeoJSON tab for advanced editing.

Once you enter a new datapoint manually, to the left of the screen more information pops up. Click on the “+ Add Property” button in the left side of the screen. Click the “+ Add all” button that pops up. Enter in the information for the place name, description, project type, project URL (if applicable), or year. You can leave some of these empty. For ProjectType, this is case sensitive to keep same. You can click on project type, and some values will show up which you can click on. You can add a new project type if they aren’t listed.



Field	Value
PlaceName	value
ProjectDescrip	value
ProjectType	value
ProjectURL	value
Year	value



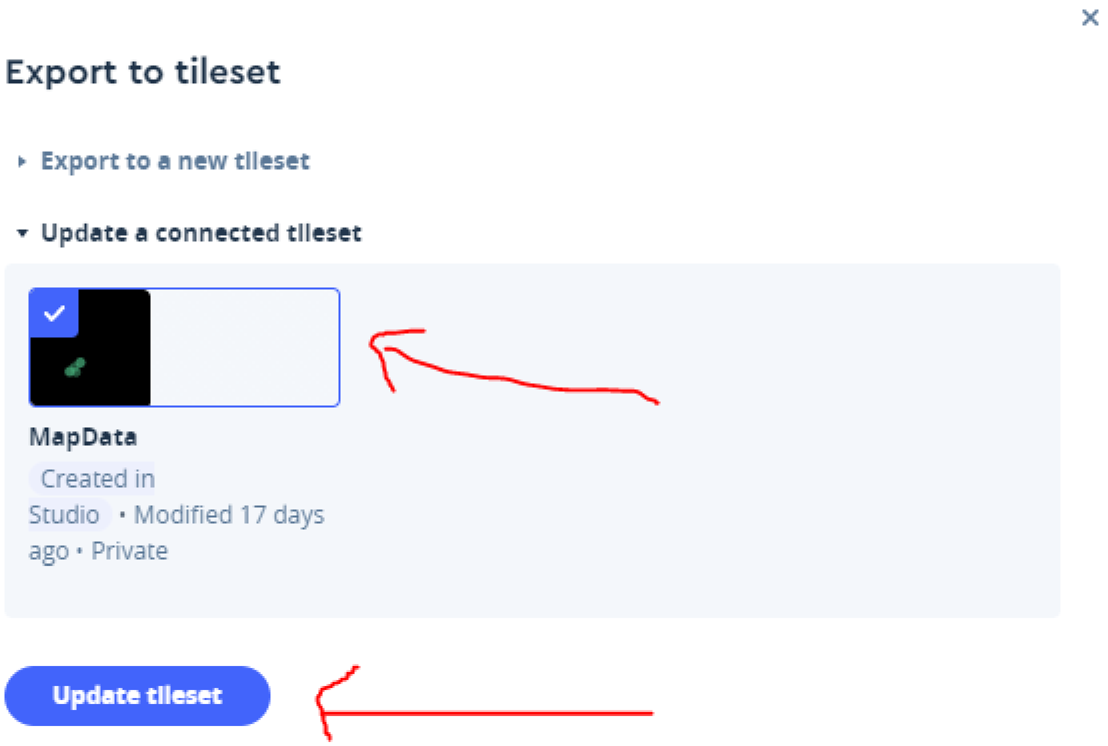
Values for ProjectType (3)

FishPassageBarrier	string
LargeWoodyDebris	string
RiperianPlanting	string

To delete the point, click on the trash can icon above. Once you have finished adding all points, click on Export in the top right corner, then click on MapData. You are NOT creating a new

Tileset, you are updating the one we already have. Once you have clicked on MapData, click on Update Tileset. (Please note that we have a free account and can only export “x” amount of time a month. I believe it is currently 25, so don’t export more than 25 times a month, which shouldn’t be a problem.) You can find this payment information on the MapBox studio payment website.

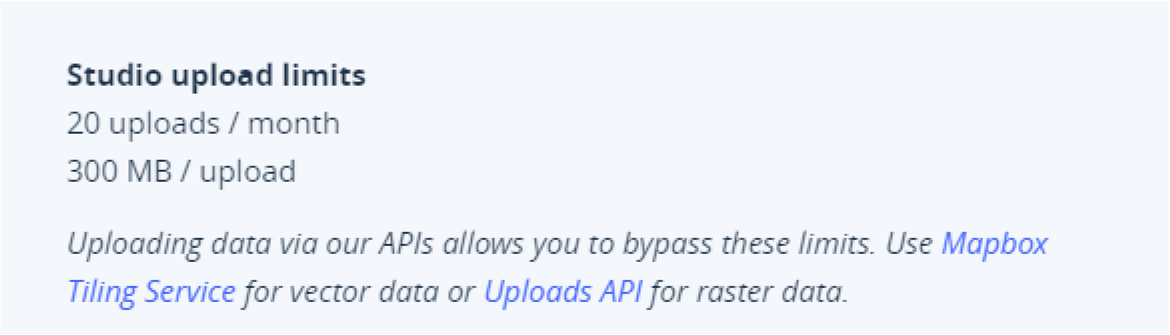
You can always reupdate the information that is already there. If you need to change a datapoint, click on one of the points and you can manually type in different information.



Export to tileset

MapData

Created in Studio • Modified 17 days ago • Private



Studio upload limits

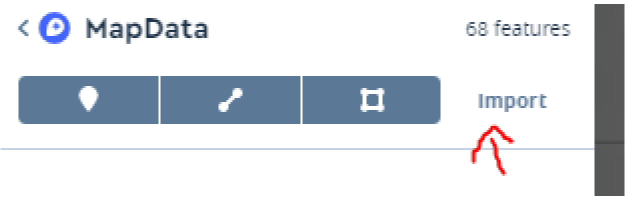
20 uploads / month

300 MB / upload

Uploading data via our APIs allows you to bypass these limits. Use Mapbox Tiling Service for vector data or Uploads API for raster data.

Adding Data – Through a File (Multiple points at once)

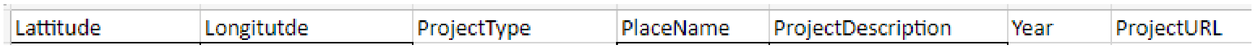
You can add multiple data points at a time if you have the correct file and format. In the same window, you can click Import.



When importing data, you can either have a GeoJSON file, a JSON file, or a CSV file. Enter your file and it will populate all values.

Correct File Format for Import

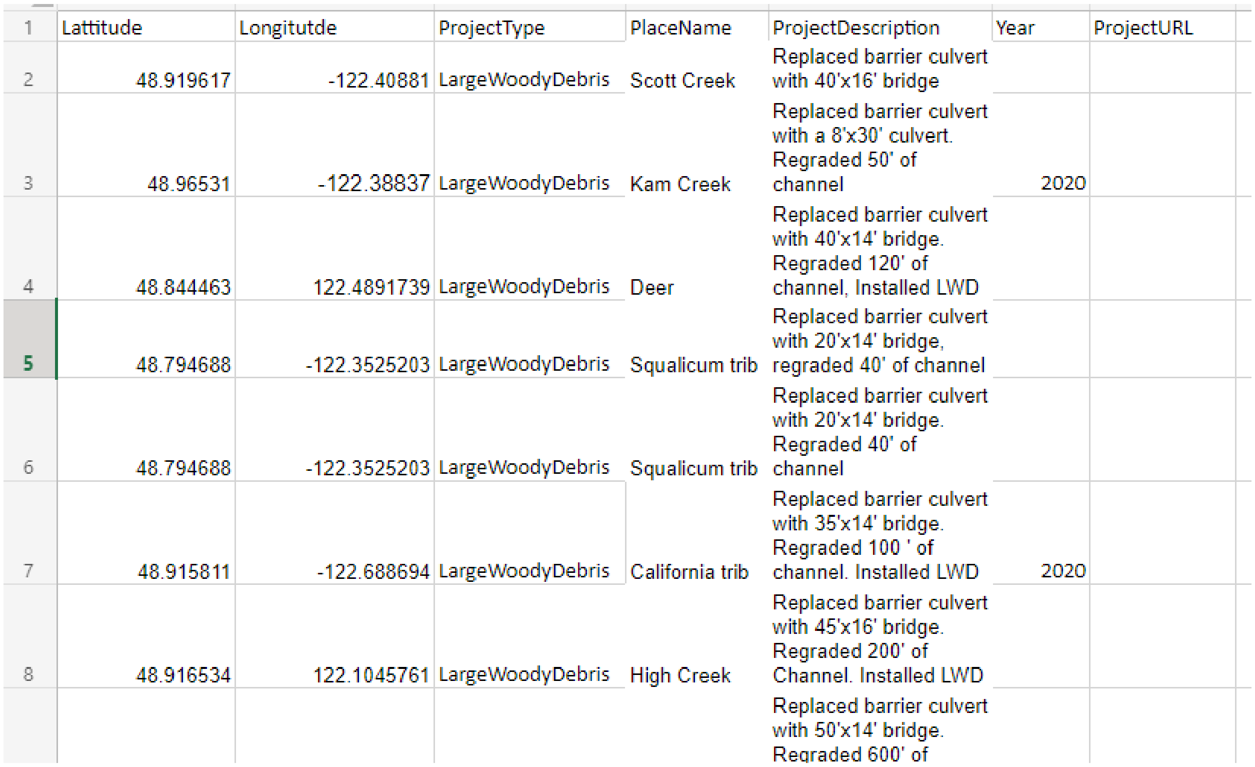
I have created a template in which you can export your files to. Open the "Template_For_Dataset_Import" excel sheet. This should be in the Tai, Documentation portion of the GitHub repo. Create a copy, make sure that this template empty so you can make a copy of it each time. In your copy it should have these fields.



Lattitude	Longitutde	ProjectType	PlaceName	ProjectDescription	Year	ProjectURL

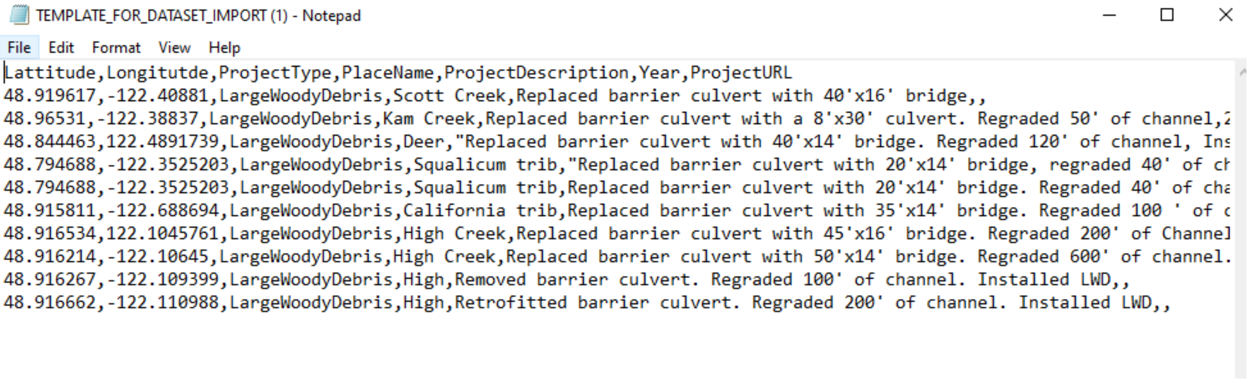
With the data you are given, copy and paste that information in the correct column. If you don't have Year/Project URL you can leave it blank. For ProjectType, sometimes it won't be given. Read the description and see what type of project it was. "LargeWoodyDebris?" "RiperianPlanting?"

Once you have added everything, your file might look something like this.



1	Lattitude	Longitutde	ProjectType	PlaceName	ProjectDescription	Year	ProjectURL
2	48.919617	-122.40881	LargeWoodyDebris	Scott Creek	Replaced barrier culvert with 40'x16' bridge		
3	48.96531	-122.38837	LargeWoodyDebris	Kam Creek	Replaced barrier culvert with a 8'x30' culvert. Regraded 50' of channel	2020	
4	48.844463	122.4891739	LargeWoodyDebris	Deer	Replaced barrier culvert with 40'x14' bridge. Regraded 120' of channel, Installed LWD		
5	48.794688	-122.3525203	LargeWoodyDebris	Squalicum trib	Replaced barrier culvert with 20'x14' bridge, regraded 40' of channel		
6	48.794688	-122.3525203	LargeWoodyDebris	Squalicum trib	Replaced barrier culvert with 20'x14' bridge. Regraded 40' of channel		
7	48.915811	-122.688694	LargeWoodyDebris	California trib	Replaced barrier culvert with 35'x14' bridge. Regraded 100 ' of channel. Installed LWD	2020	
8	48.916534	122.1045761	LargeWoodyDebris	High Creek	Replaced barrier culvert with 45'x16' bridge. Regraded 200' of Channel. Installed LWD		
					Replaced barrier culvert with 50'x14' bridge. Regraded 600' of		

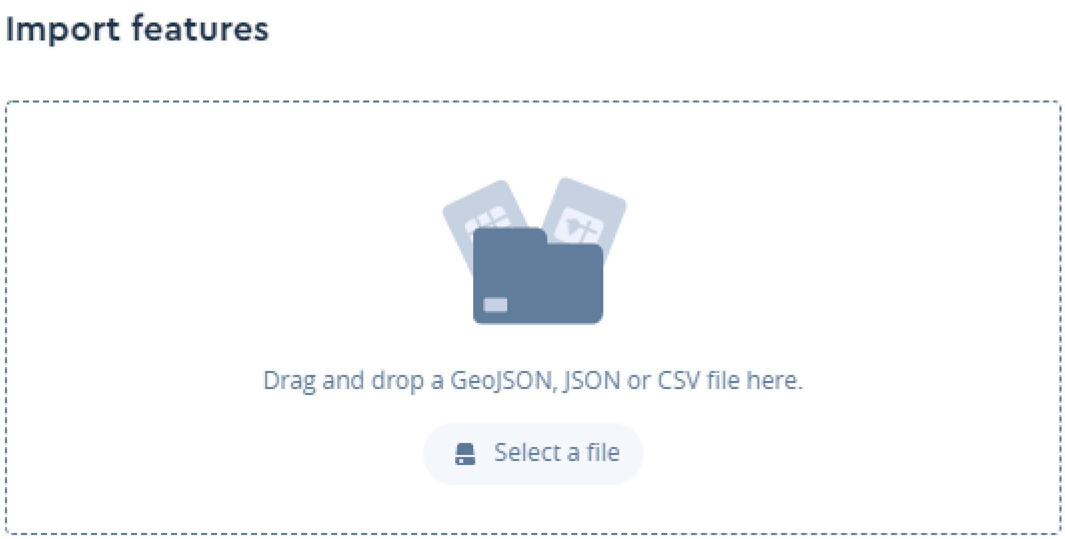
Once you have added all elements to the excel sheet, you will want to export it to a .csv file. Now that you have converted the .xlsx file (excel) to .csv it will look like this.



TEMPLATE_FOR_DATASET_IMPORT (1) - Notepad
File Edit Format View Help

```
Lattitude,Longitutde,ProjectType,PlaceName,ProjectDescription,Year,ProjectURL
48.919617,-122.40881,LargeWoodyDebris,Scott Creek,Replaced barrier culvert with 40'x16' bridge,,
48.96531,-122.38837,LargeWoodyDebris,Kam Creek,Replaced barrier culvert with a 8'x30' culvert. Regraded 50' of channel,2
48.844463,122.4891739,LargeWoodyDebris,Deer,"Replaced barrier culvert with 40'x14' bridge. Regraded 120' of channel, Ins
48.794688,-122.3525203,LargeWoodyDebris,Squalicum trib,"Replaced barrier culvert with 20'x14' bridge, regraded 40' of ch
48.794688,-122.3525203,LargeWoodyDebris,Squalicum trib,Replaced barrier culvert with 20'x14' bridge. Regraded 40' of cha
48.915811,-122.688694,LargeWoodyDebris,California trib,Replaced barrier culvert with 35'x14' bridge. Regraded 100 ' of c
48.916534,122.1045761,LargeWoodyDebris,High Creek,Replaced barrier culvert with 45'x16' bridge. Regraded 200' of Channel
48.916214,-122.10645,LargeWoodyDebris,High Creek,Replaced barrier culvert with 50'x14' bridge. Regraded 600' of channel.
48.916267,-122.109399,LargeWoodyDebris,High,Removed barrier culvert. Regraded 100' of channel. Installed LWD,,
48.916662,-122.110988,LargeWoodyDebris,High,Retrofitted barrier culvert. Regraded 200' of channel. Installed LWD,,
```

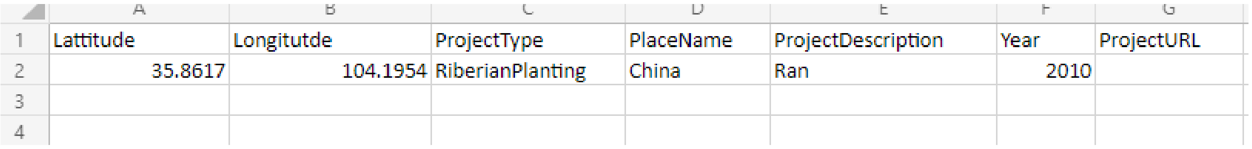
Go back to Dataset in Mapbox, and click on the import button.



Add in the .csv file. Then done! You've added in your new values. Every time you want to add new data, start with a fresh new excel sheet template, add in the information then import it into the dataset. Be sure to export your dataset at the end!

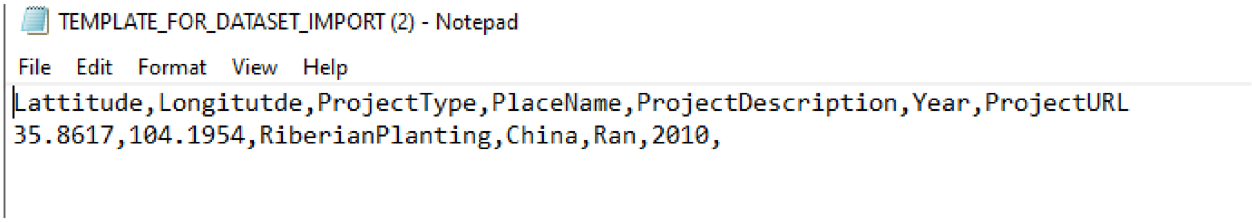
Another Example

Here I'll show another example of importing. Here's my excel sheet, I added a datapoint somewhere in China.



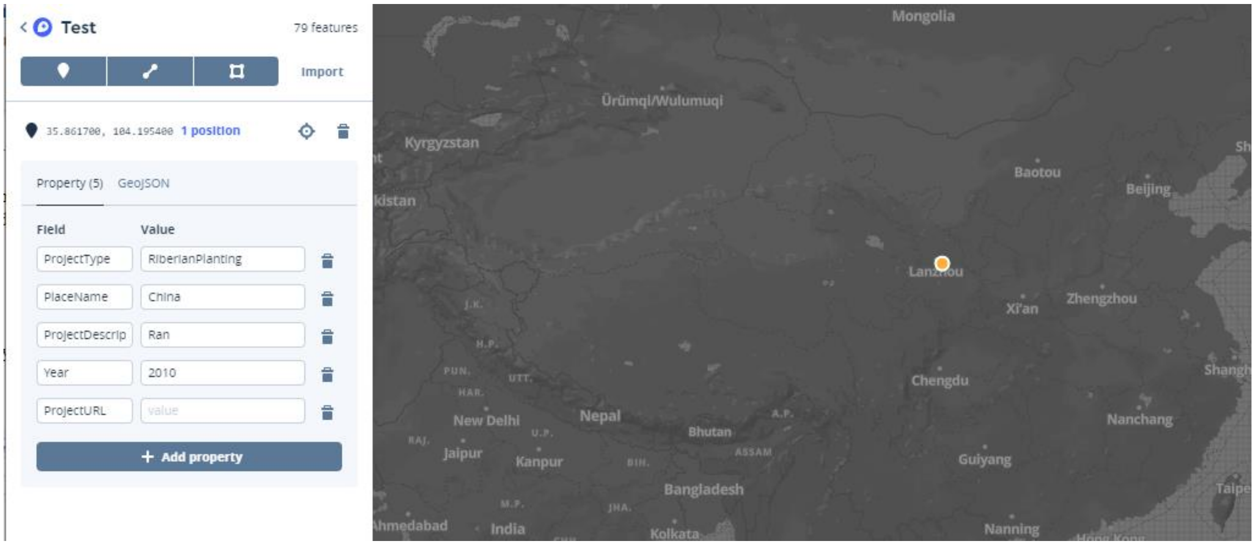
	A	B	C	D	E	F	G
1	Lattitude	Longitutde	ProjectType	PlaceName	ProjectDescription	Year	ProjectURL
2	35.8617	104.1954	RiberianPlanting	China	Ran	2010	
3							
4							

I am going to export this or convert it to a .csv file. It now looks like this.


TEMPLATE_FOR_DATASET_IMPORT (2) - Notepad

```
Lattitude,Longitutde,ProjectType,PlaceName,ProjectDescription,Year,ProjectURL
35.8617,104.1954,RiberianPlanting,China,Ran,2010,
```

Going to import this into my Dataset. There my Dataset was uploaded in China!


Test 79 features

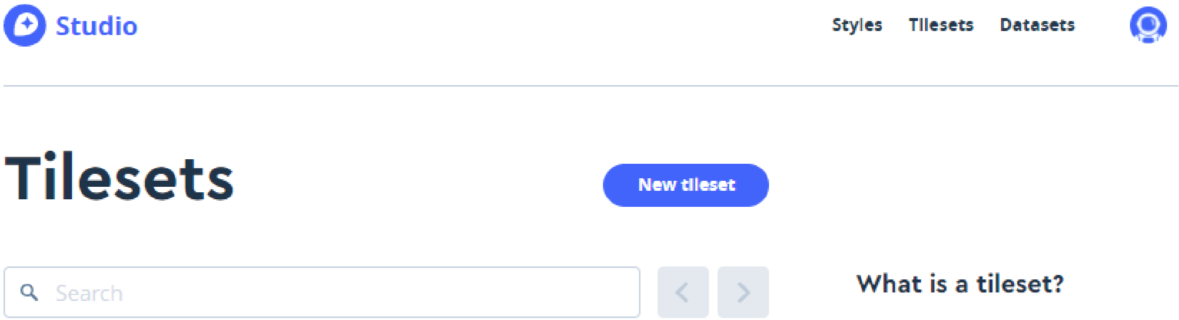
35.861700, 104.195400 1 position

Field	Value
ProjectType	RiberianPlanting
PlaceName	China
ProjectDescrip	Ran
Year	2010
ProjectURL	value

If I export this, it will update on all the Maps (including the one in Square Space) so for the sake of this example, I'm going to delete this point.

Tilesets

Going to the Tilesets page, you will initially see Default Tilesets. Scroll all the way to the bottom and you will see our Custom Tileset.



Whenever we export our Datasets, it will populate up our Tileset which is this MapData. You won't ever need to touch or change this. You can read more about what a Tileset is on the MapBox website. Think of a Tileset as a layer you can put on your map. We could have a Tileset of just schools in the area. We could have a Tileset of Hospitals in the area. So, if we have two different Tilesets, we can add these layers to a single map.

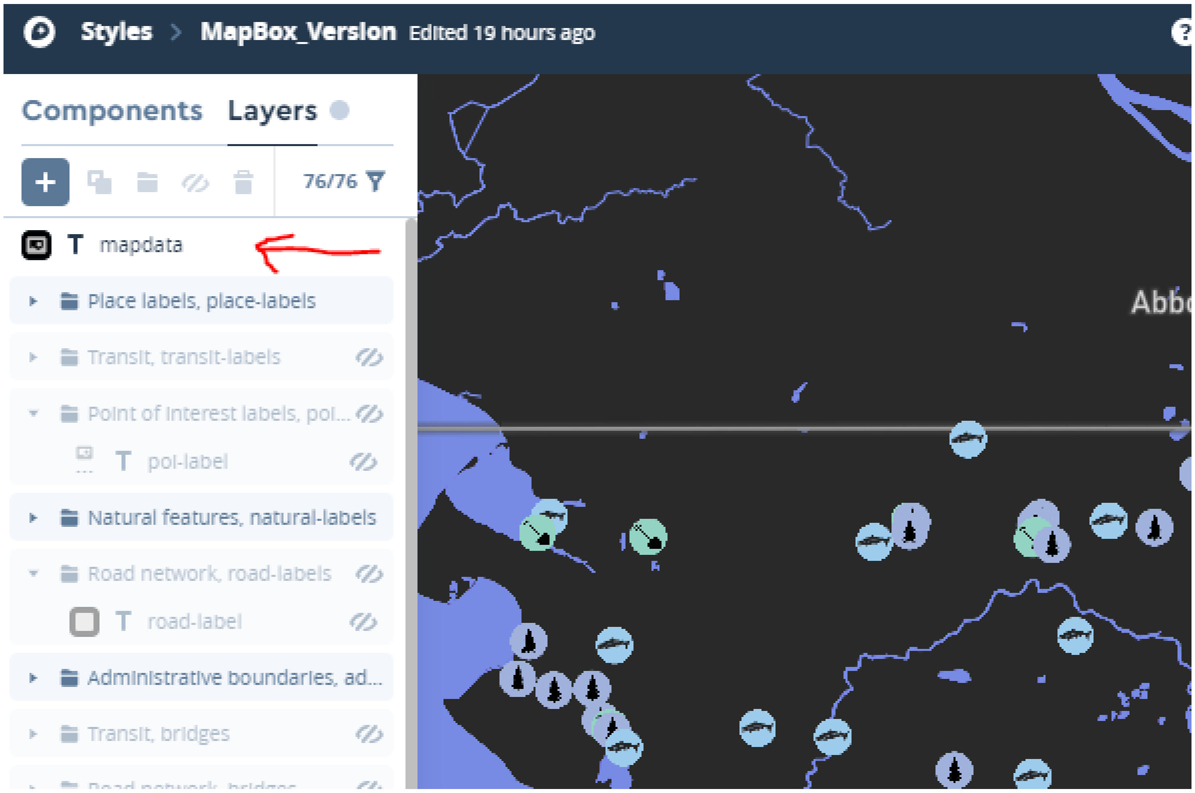
Styles

This is where all the magic happens. Let's recap what we learned though. A Dataset is where we can add our data points or import our data points. We export our Dataset to create a layer called a Tileset. Now we are at the Styles tab.

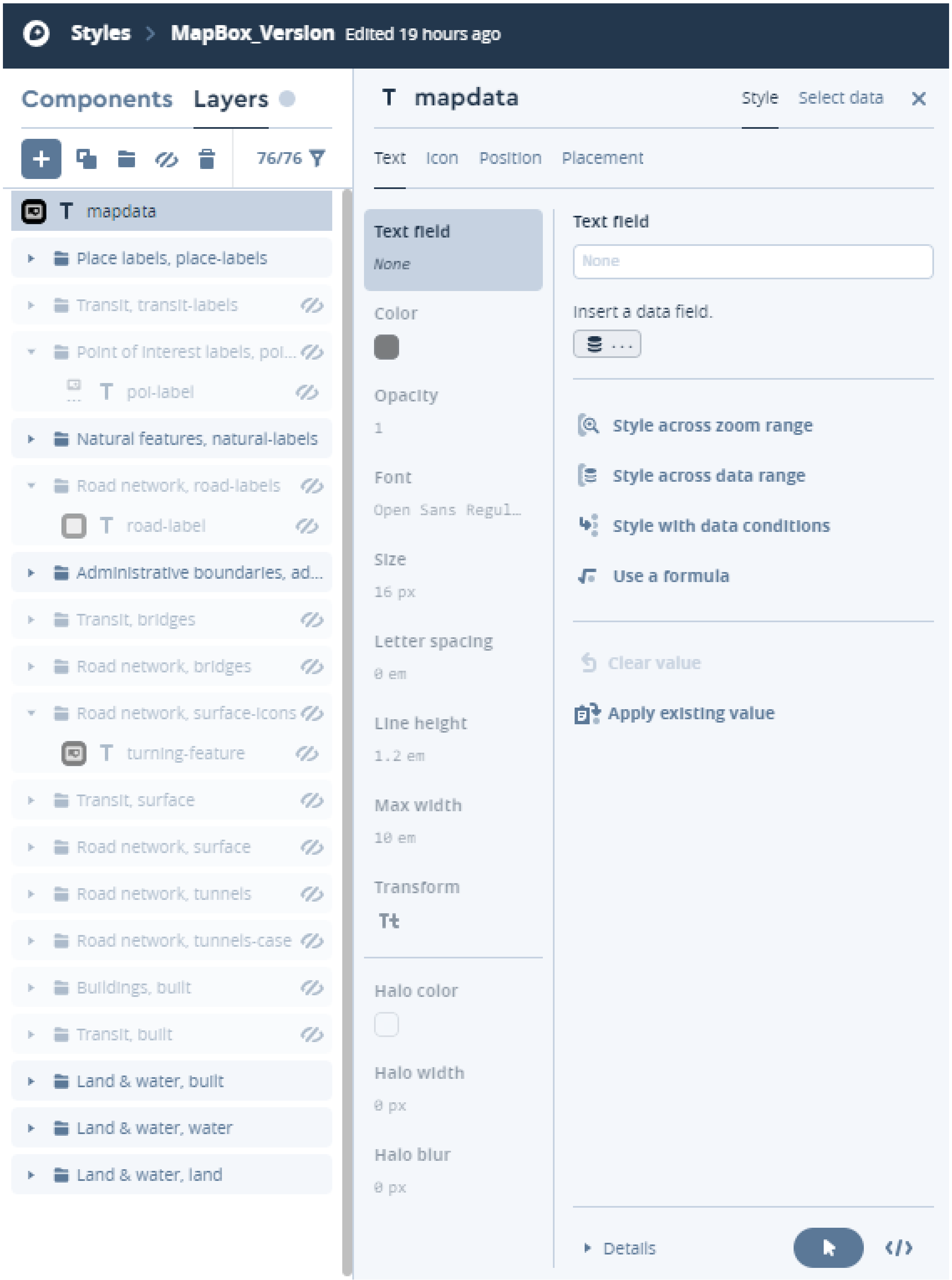
In our Styles you will see four separate maps currently. The Blank one, is never used, it's just there. The Blank_Copy will be used as a backup map in case one of the other maps gets deleted or something. The ReactApp_Version is the map that the React Map uses. This is currently empty, because remember with the React App it populates this map from the MongoDB. Then finally we have the MapBox_Version style. This is populated because it has the MapData layer from the Tileset.

Let's investigate how to manage the MapBox_Version of this.

Opening this version, we see the following. On the left column you can see various folders of information. You can also see mapdata. This is a layer that I added manually from the Tileset. Without this layer, you wouldn't see ANY icons.


Styles
MapBox_Version
Edited 19 hours ago
Components
Layers
76/76
mapdata
Place labels, place-labels
Transit, transit-labels
Point of interest labels, poi...
poi-label
Natural features, natural-labels
Road network, road-labels
road-label
Administrative boundaries, ad...
Transit, bridges
Abb

The way I added this Tileset layer is I clicked the "+" sign in the top left corner. Now lets talk a little about what's going on with this layer. Click on that layer. Another window opens as follows.



Text field
None

Color

Opacity
1

Font
Open Sans Regul…

Size
16 px

Letter spacing
0 em

Line height
1.2 em

Max width
10 em

Transform
Tt

Halo color

Halo width
0 px

Halo blur
0 px

Text field
None

Going through this, you will only be touching the Icon tab.

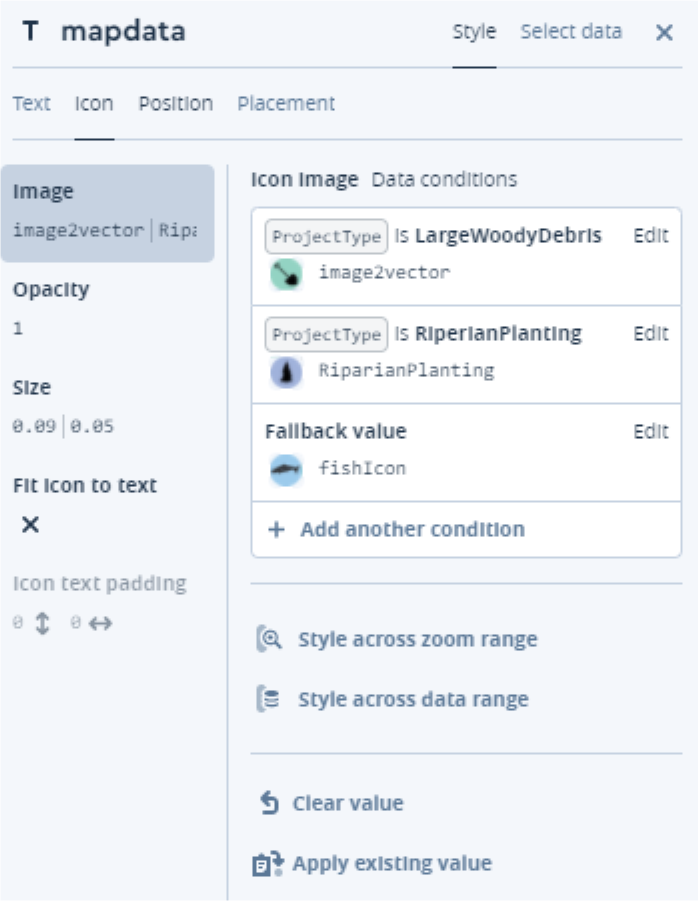


Image
image2vector | Ripa

Opacity
1

Size
0.09 | 0.05

Fit icon to text
✕

Icon text padding
0 ↕ 0 ↔

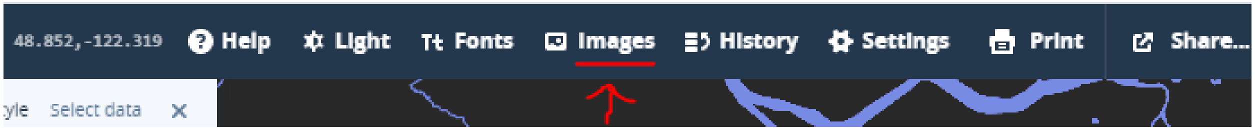
ProjectType is LargeWoodyDebris
image2vector

ProjectType is RiperianPlanting
RiparianPlanting

Fallback value
fishIcon

If you add another ProjectType, you will have to add a new icon to the map. Looking at the Data conditions we have “LargeWoodyDebris” show up as a shovel icon, “RiperianPlanting” show up as a tree icon, and “Fallback value” show up as fish icon. Now the Fallback value means for any other thing that isn’t RiperianPlanting or LargeWoodyDebris. Currently it shows the fish icon for the Fisheries. Think of this as an if/else if/else statement. To add another condition, click on the “Add another condition”, you’ll input the new ProjectType, and then you’ll add the icon.

To upload an icon, you can click on the “Images” button at the top of the screen.



Once you click on that, click on the Custom tab. Then click Upload Images, done. Some images might be bigger than others, so you will have to scale them if necessary. To scale, go back to the Icon section. Instead of clicking Image, click on Size.

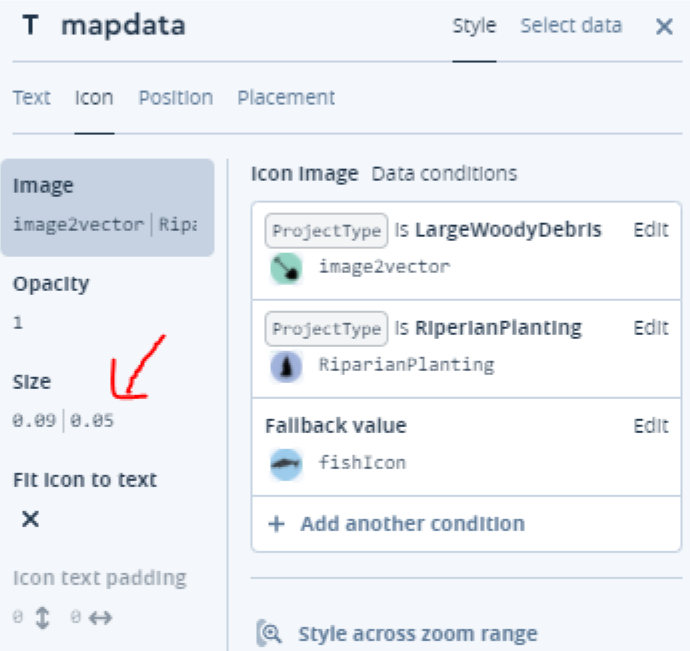


Image
image2vector | Ripa

Opacity
1

Size
0.09 | 0.05

Fit icon to text
×

Icon text padding
0 ↕ 0 ↔

Icon image Data conditions

ProjectType is LargeWoodyDebris
image2vector

ProjectType is RiperianPlanting
RiparianPlanting

Fallback value
fishIcon

Here you can enter in different sizes depending on the image. Go back to the if/else if/else idea. This is how that works. Add another condition and show which condition it is.

Dataset/Tileset/Style End Thoughts

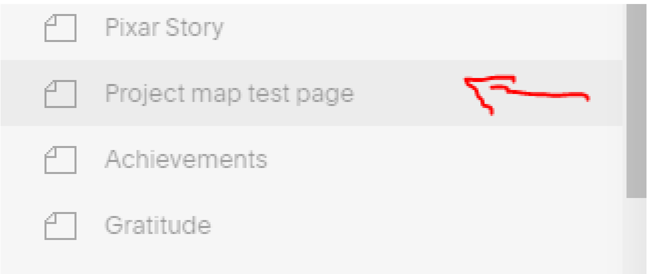
Since all of this has been set up, all you will ever really need to touch is the Dataset portion. That is because you will only be adding new Data. You might touch Styles if you add a new project that requires a new logo.

Once you export your data from Dataset, it AUTOMATICALLY updates the Tileset, and since in Styles, the Tileset is added “mapdata” it will AUTOMATICALLY update your Style. Thus, you will only ever need to touch Dataset.

Square Space – NSEA Website

Now I’m going to explain how this map was embedded into the Square Space website. Currently it’s a hidden file (this may change for the next person in charge of this project) but go to the page that the map is currently on.

For me the page is here:



I have embedded this map using JavaScript. If you open the code, and click Edit, you will see all the JS code. Here is where I apply the tooltip click options. For when a user clicks on one of the icons, the tool tip will open.

If there needs to be new information that will show up, here is where you will edit this. I have attached the map style (found in Styles) and the token for mapbox in this script so it will get the map.