This dot appears on summary slides

Making a Interactive Map Story - “storymap”

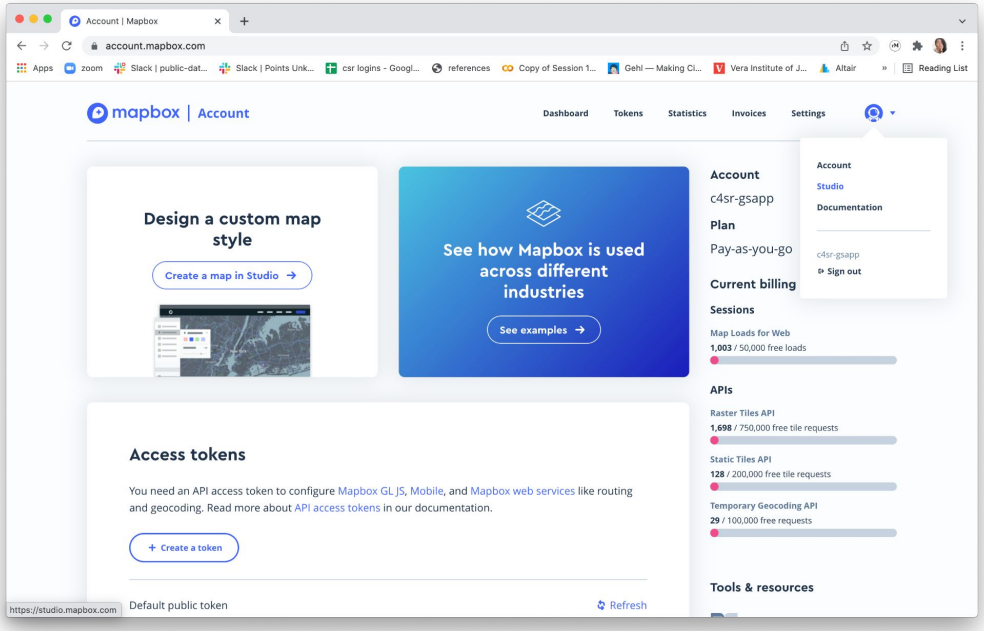
Steps	*Tools*
Research and writing	
Analyse and make map data	
Set up local development environment	*Download Software*
Upload datasets and style map data layers	*Mapbox Studio*
Open and run the storytelling template on your computer Putting your writing into story template Configure your story’s interactivity	*Visual Studio* *Or* *Local Server*
Publish your story	

This tutorial session covers

Setup: Mapbox Studio

Mapbox Studio

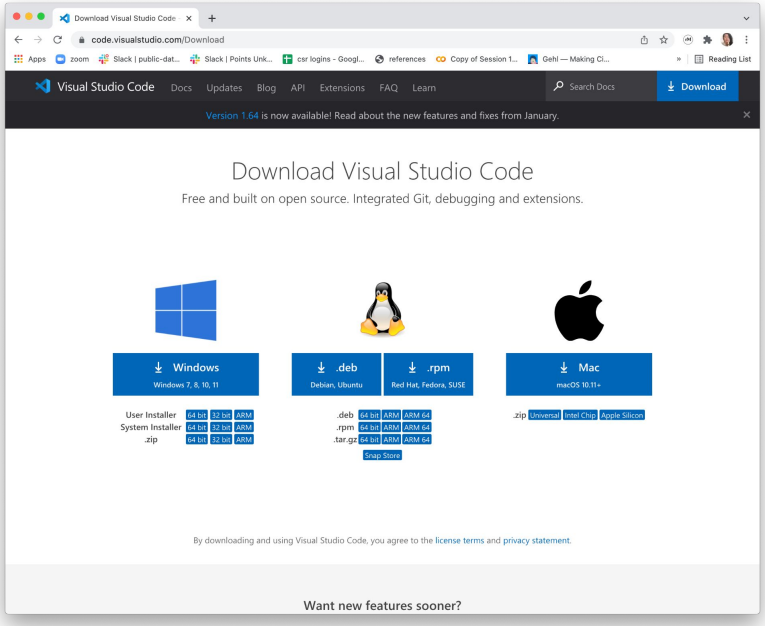
Create an account at mapbox.com

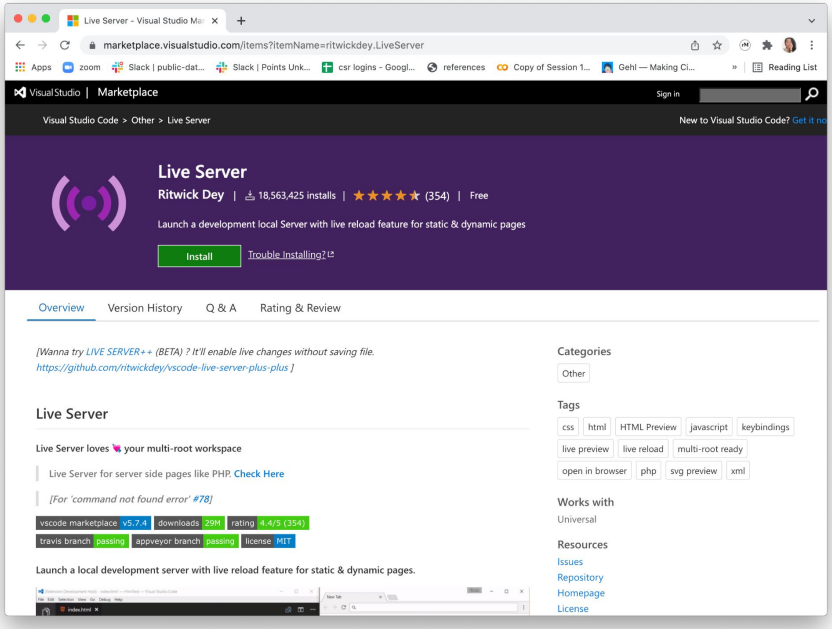


Setup: Code Editor and Development Environment

Download and Install

1. Visual Studio Code: https://code.visualstudio.com/Download
2. Live Server (extension for VS): https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer



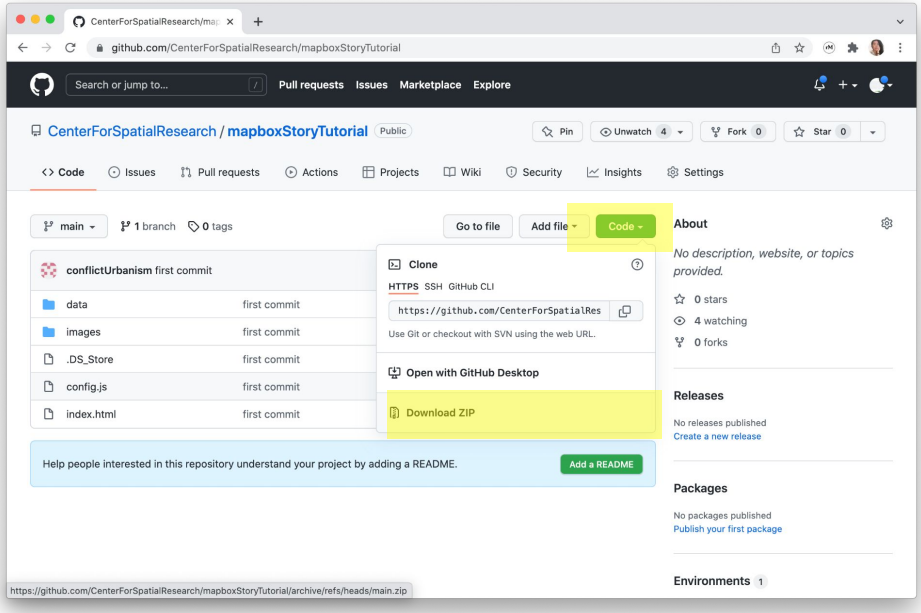


Setup: Download Example Files

Go to https://github.com/CenterForSpatialResearch/mapboxStoryTutorial

Click on the green “Code” button and select “Download ZIP”

Unzip the downloaded contents and save to somewhere you will be storing your project (usually Documents).



Mapbox Studio: Data

- Mapbox Studio uses data in .geojson format.
- You can export geographic data in .geojson format from ArcGIS, QGIS, or download directly from an open data portal.

The data we use in this example are:

Point Data
- Site location
- Points of Interest
- Amenities

Path Data

Polygon Data
- Institution Boundaries
- Census Boundaries - blocks, block groups, tracts, counties

Mapbox Studio: Create a style

Steps

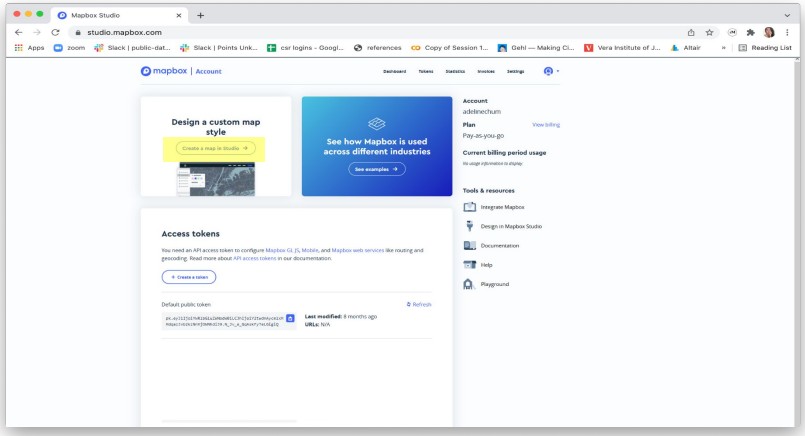
Create a new mapbox style

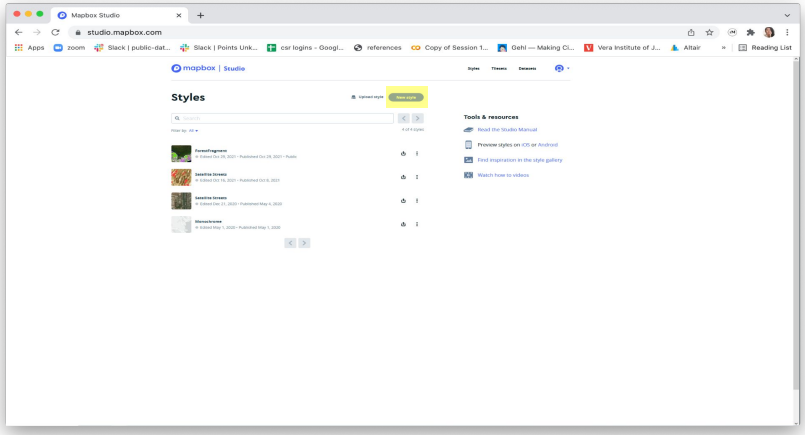
For each layer of data:
- Add a new map layer
- Upload new data or use an existing dataset(within mapbox)
- Filter data if necessary
- Style data with conditional formatting
- Rename layer with something descriptive if auto-generated layer name is not useful
- save(publish)

For this tutorial we will add layers for simple polygons, text labels, symbols, and a choropleth

Share your map by using your
- Style URL
- Access Token

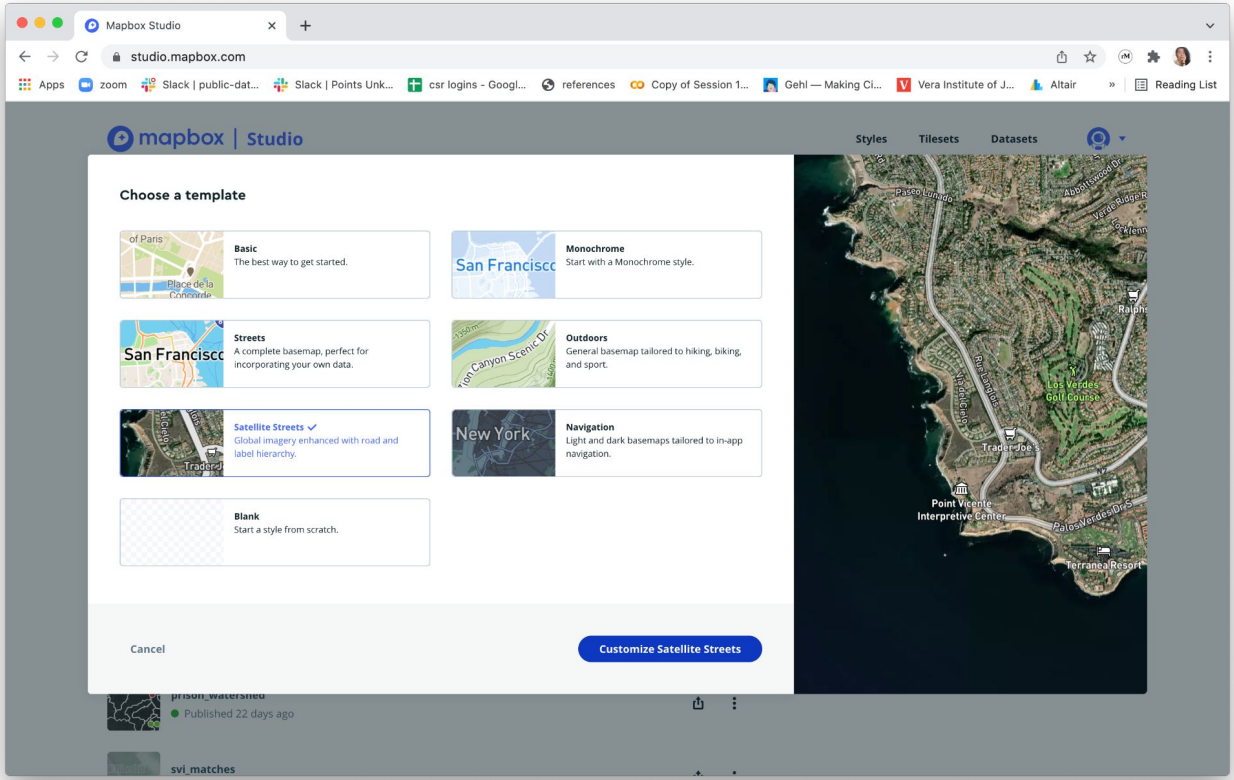
Create a new map style with a basemap





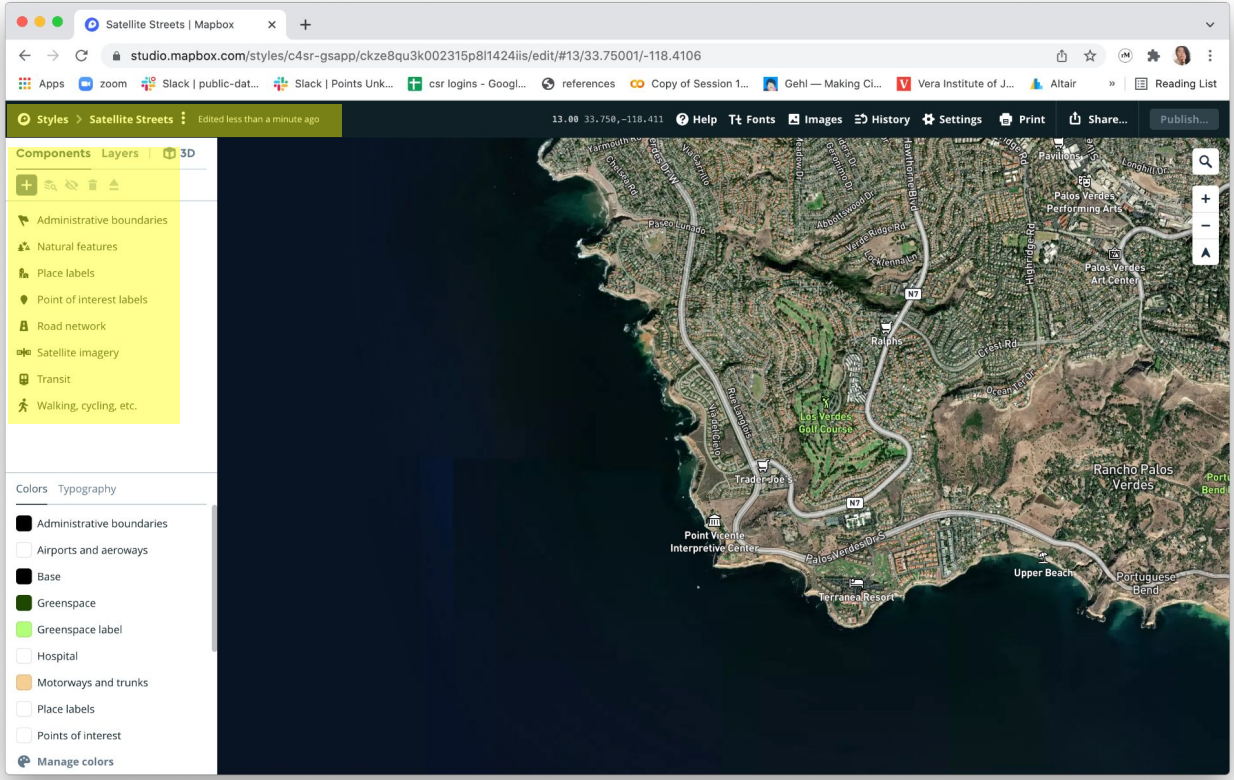
We will start by clicking “Create a map in Studio” and then click on the icon “New style”.

Create a new map style with a basemap



For this example, we will start with the preexisting basemap called “Satellite Streets”.

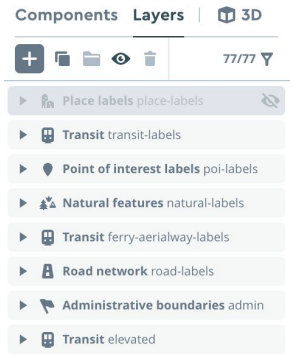
The satellite basemap



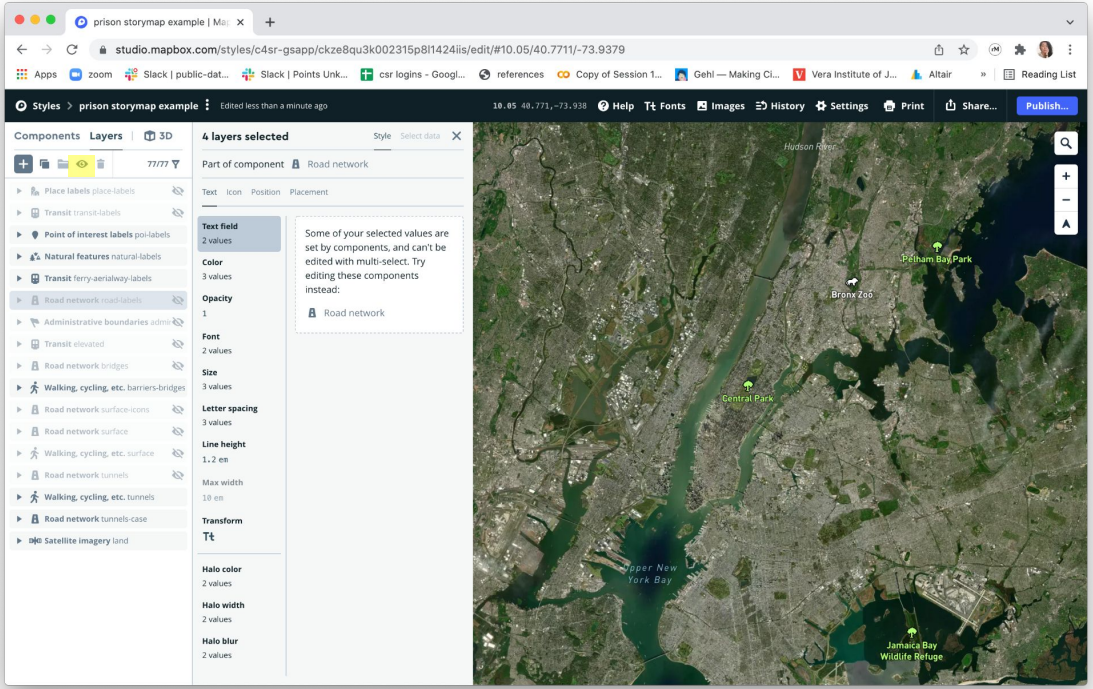
Once you have this new style:

- Click on the top left corner where the map is currently called “Satellite Streets” and rename your map

- There are 2 tabs that list the elements of the map - “Components” and “Layers” We will be adding to the “Layers” tab with our own data.



Toggling layers on and off



Pan the map to our area of interest - the Fulton facility that we will work on is in the Bronx.

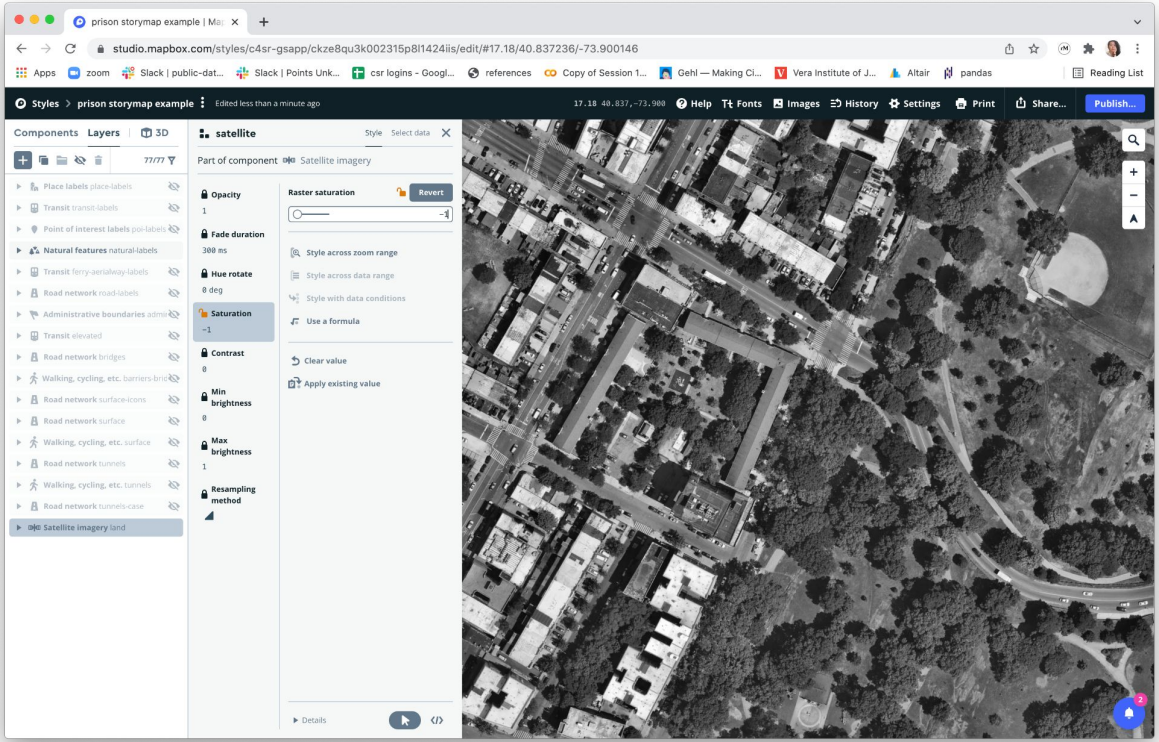
In the “Layers” tab, toggle layers on and off by clicking the eye icon.

When working it helps to toggle layers off to reduce visual noise.

The layers that are off will not appear in the final map, but can be turn back on with code on the client side.

Click “Publish” after changing the visibility of layers to see the changes reflected in your map

Change the saturation of the Satellite layer



You can change the hue, contrast, saturation, and lightness of the raster satellite layer as you would a photograph.

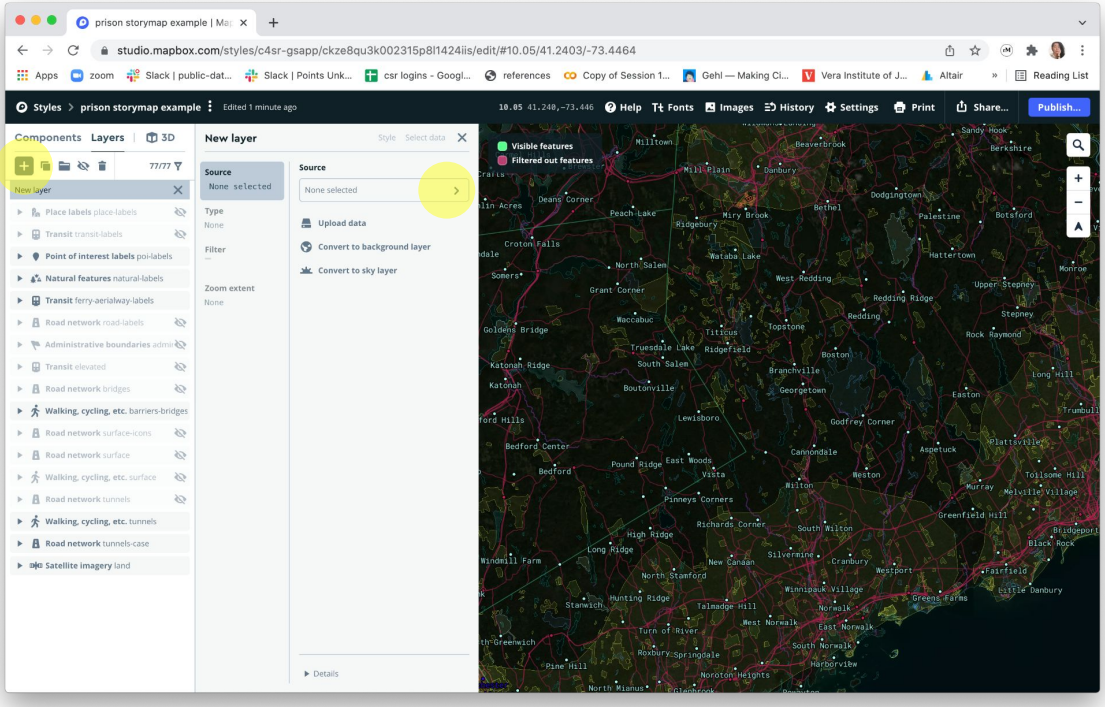
Select the layer called “Satellite imagery”

Because this was a layer we inherited from the mapbox style template, the settings for components are locked.

Click on Saturation and click on the lock icon to make changes.

Set the saturation of the layer to -1, which turns the satellite images into black and white.

Adding a new layer



Click the “+” sign on the left panel to start a new layer.

Familiarize yourself with the interface:

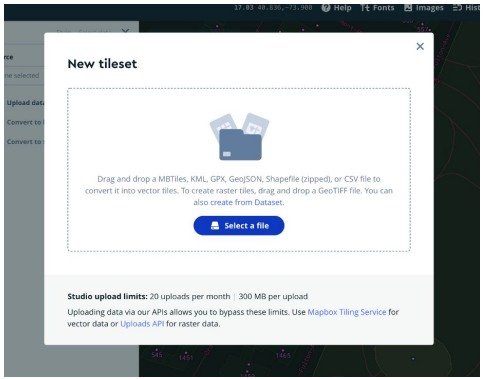
- The map that is shown to the right displays the many data layers that already exist within Mapbox and can be added.
- There are 2 tabs in the New Layer panel, “Select data”, and “Style”.
- The Select data panel allows you to chose one of the pre-existing datasets or upload a new one.

Adding the prisons footprints layer

Click “Upload data” under “Source”

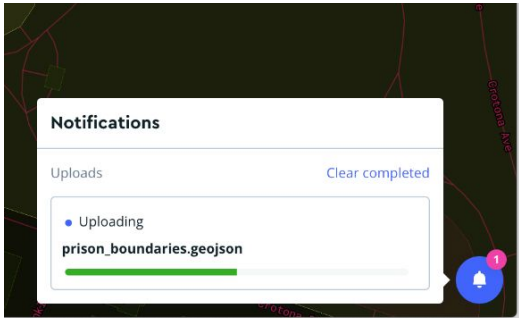
Click “Select a file” from the popup window

Navigate to the tutorial folder on your computer, go to the subfolder “data” and choose the file named “newYorkState_censusData.geojson” to upload.

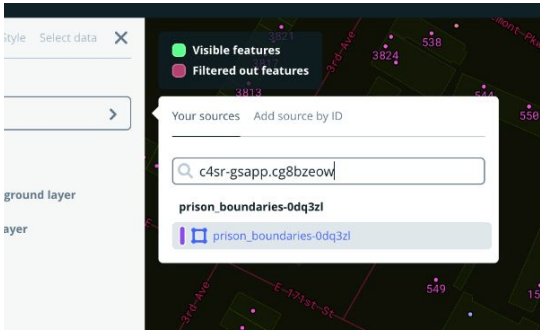


A “Notifications” window will pop up on the lower right corner.

Once the data has been uploaded, the status will update to show that it succeeded with a green dot.

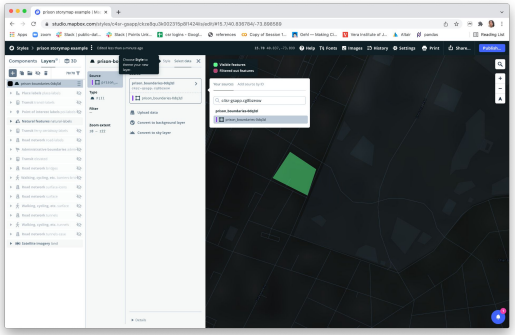


Select the dataset by copying the id from “Notifications” window the and pasting it into the “Source” window.

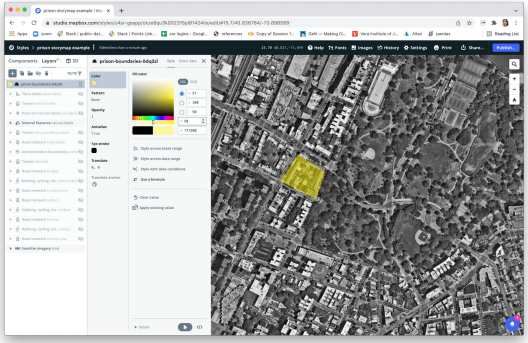


Styling and filtering the prisons footprints layer

The uploaded polygons will appear as a green highlight on the map.



Switch over to the “Style” tab of the layer and set a color and opacity for the polygon.

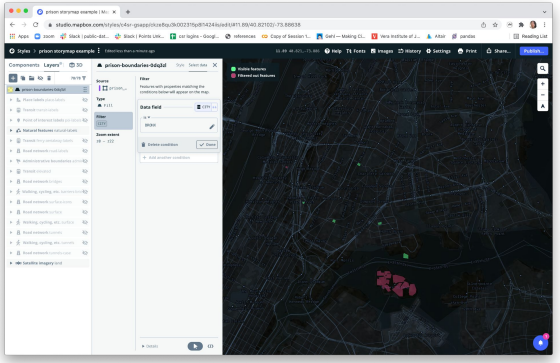


This dataset includes facilities all over the country, filter this dataset to the Bronx.

Click back to the “Select data” tab and select “Filter”

“Create filter” condition where the data field is “CITY” and the value is “BRONX”

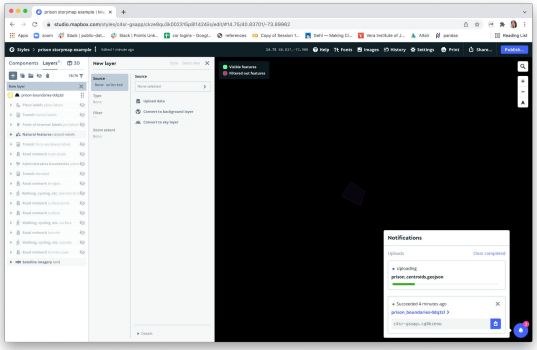
Zoom out a bit so you can see that some footprints that are outside of the Bronx have turned red. These red footprints will now be hidden in the when you switch back to the “Style” tab.



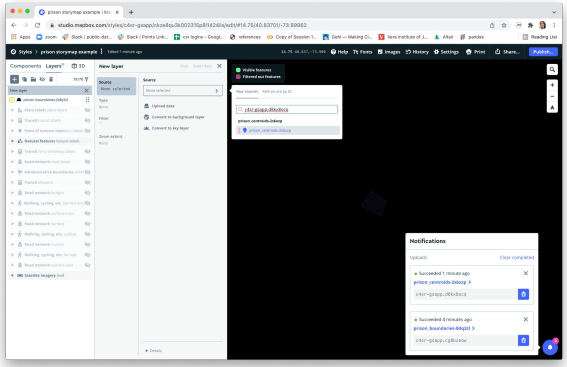
Add text

Create another layer and select upload data again.

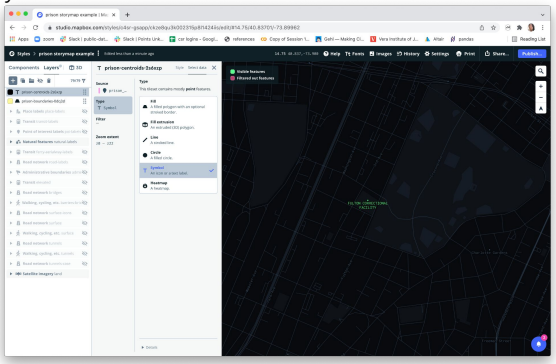
Upload the dataset named “prison_centroids.geojson” from the data folder



Repeat the process of pasting the dataset Id from the bottom right corner “Notifications” window to the layer data window.



Under Select data tab, change the layer “Type” to “Symbol”



Adding a label to the Prisons footprints

Create another layer and select upload data again.

Upload the dataset named “prison_centroids.geojson” from the data folder

Repeat the process of pasting the dataset Id from the bottom right corner Notifications window to the layer data window.

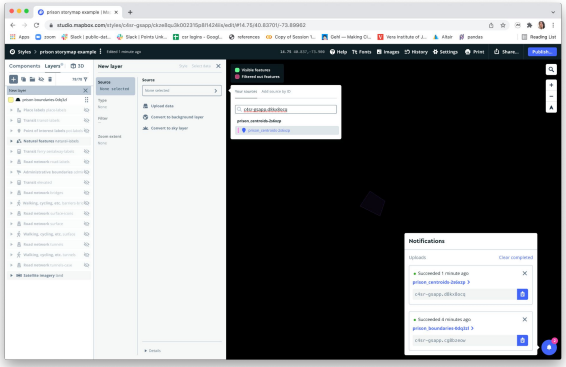
Under “Select data” tab, change the layer “Type” to “Symbol”

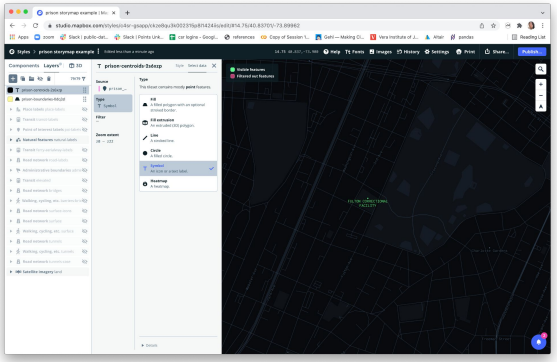
Switching to “Style”, in the “Text” tab, select “Text Field”

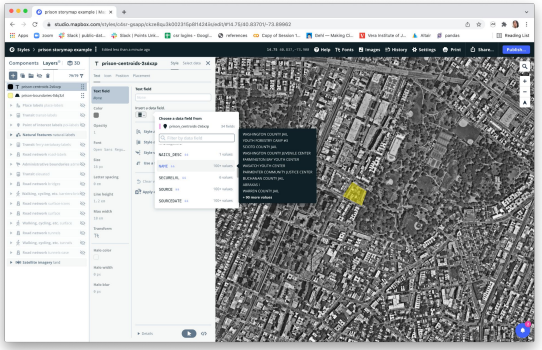
Click on the “insert a data field” icon and choose “NAME”

The name of your facility will now be visible and placed in the center of the polygon.

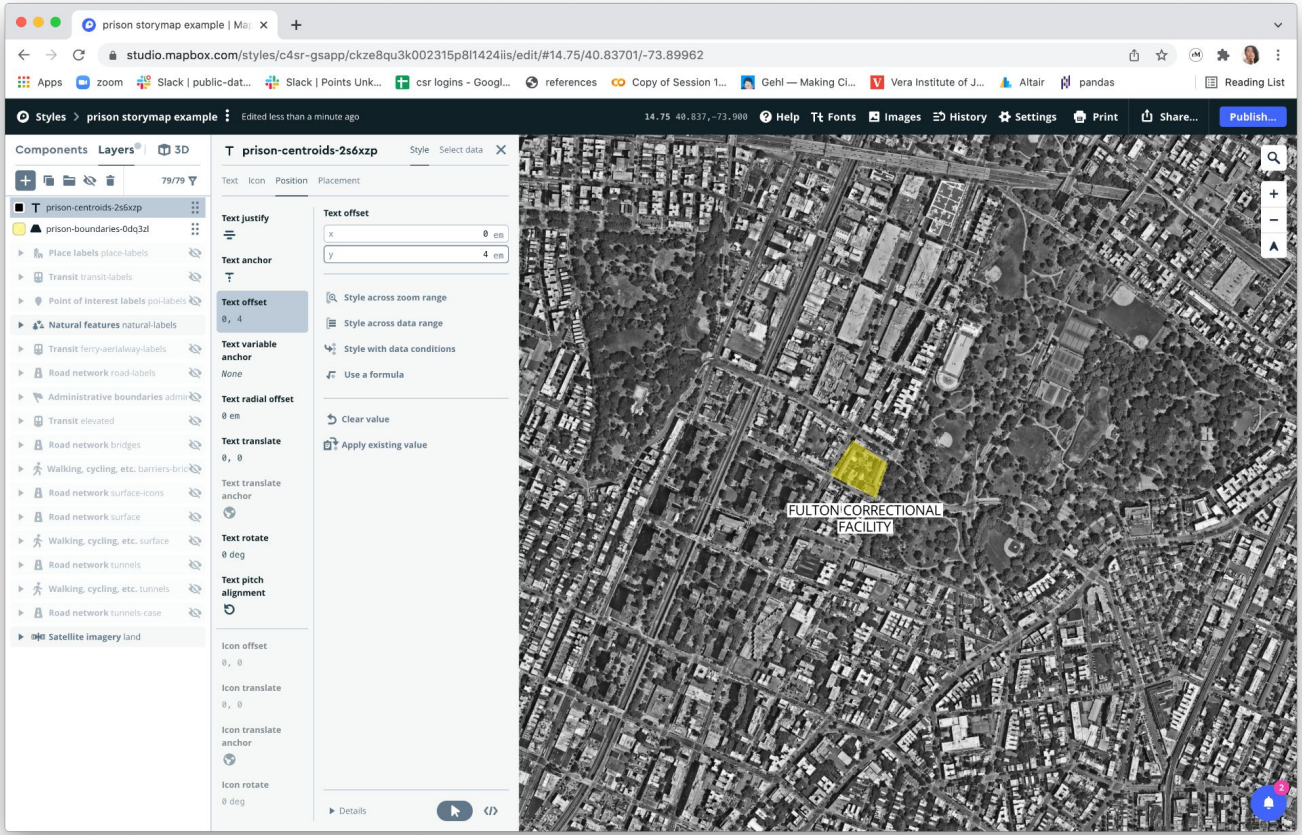
Style the text color, size, font, and placement to make the label more legible - try the different options under each tab.







Our map so far



Add another polygon layer and make a choropleth

Click “Upload data” under “Source”

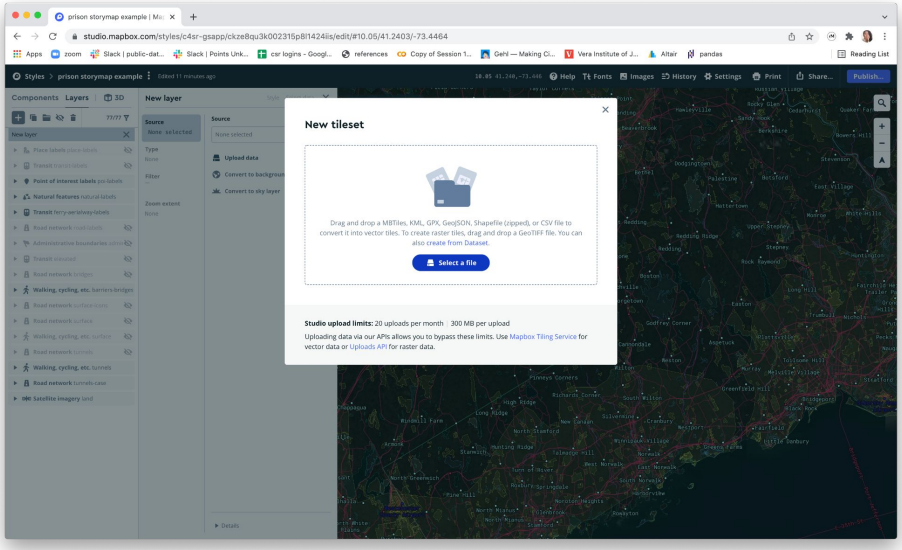
Click “Select a file” from the popup window

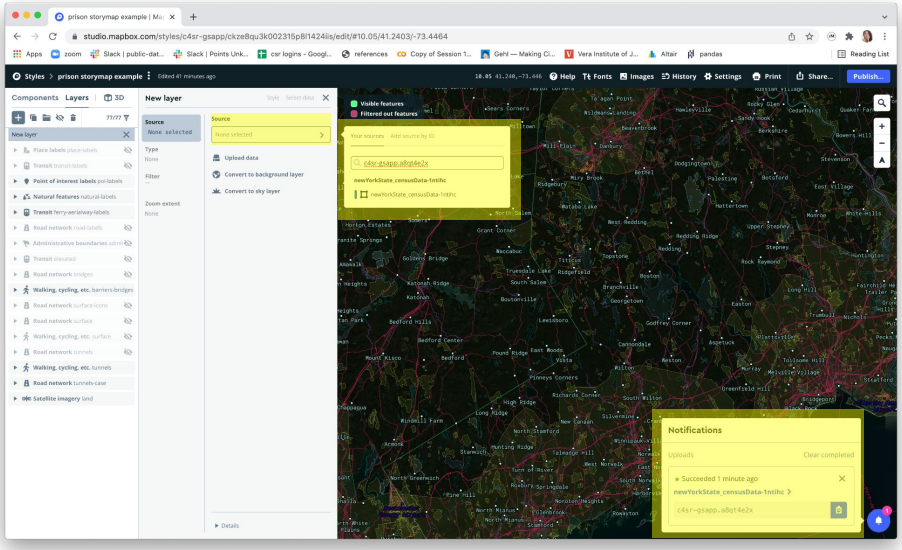
Navigate to the tutorial folder on your computer, go to the subfolder “data” and choose the file named “newYorkState_censusData.geojson” to upload.

A “Notifications” window will pop up on the lower right corner.

Once the data has been uploaded, the status will update to show that it succeeded with a green dot.

Select the dataset by copying the id from “Notifications” window the and pasting it into the “Source” window.





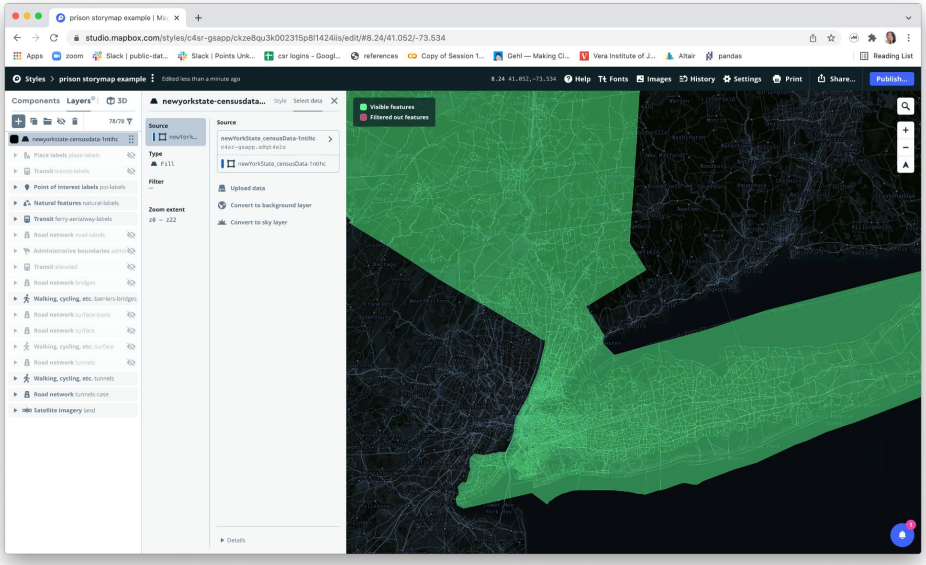
Style layer with data

The data layer will show as a highlighted shape.

Set the layer type to “Fill”

There are currently no filters set

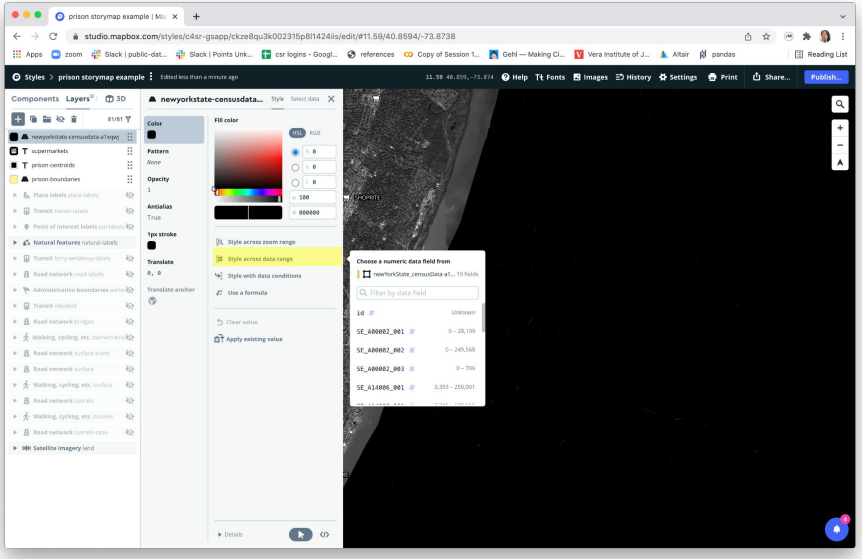
Click to the “Style” tab.



In the “Style” tab view, the polygons will turn black as a default.

Click on “Color” and then select “Style across data range”

In the popup window select “SE_A00001_002” which is the Social Explorer Census code for population density. You can find the definition of each code in the file named “newYorkState_censusDataDictionary.txt” in the same data folder



Style polygon layer with data

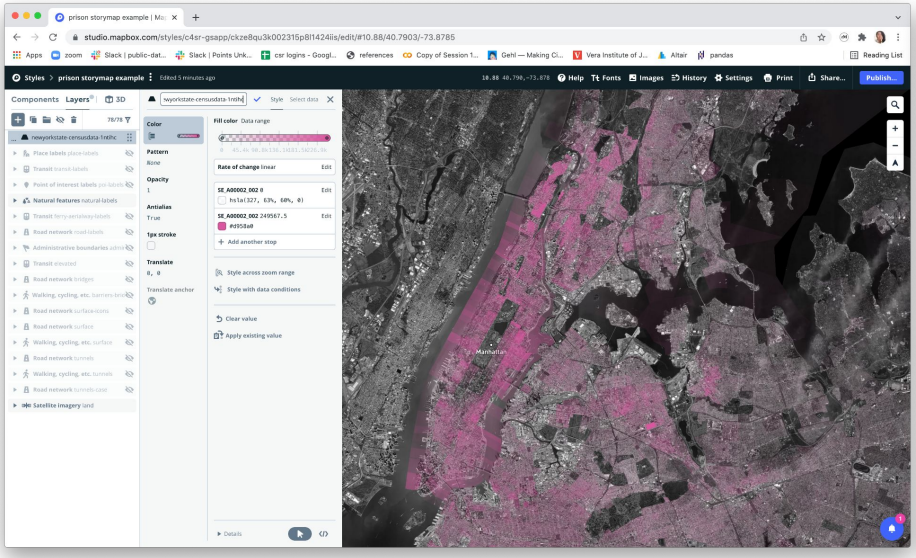
Once you select “SE_A00001_002” as your data column, mapbox will automatically set up a gradient with 2 stops to color your map.

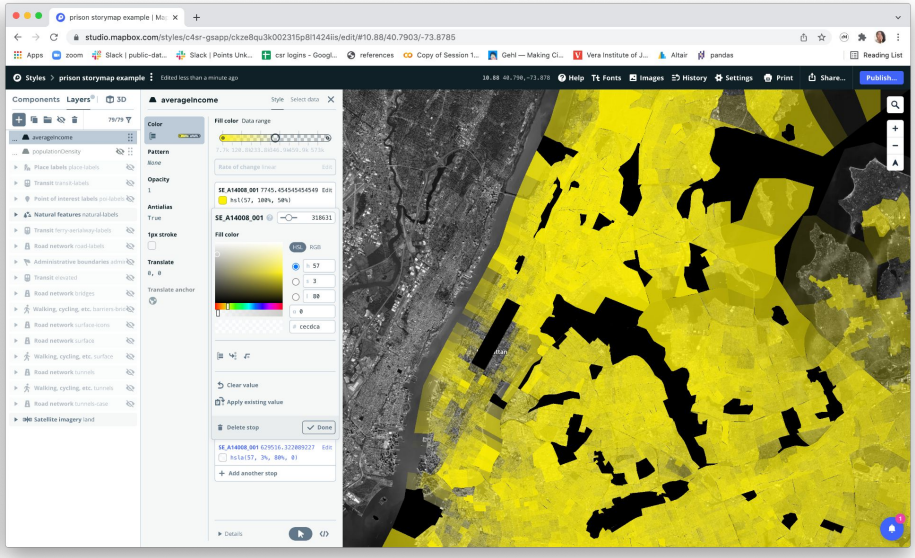
The maximum for population density in this dataset is 249567.5 persons per sq mile, and the minimum is 0.

Click edit on each stop and set the maximum stop to a different color and opacity than the minimum stop.

Repeat the same process with a different data column.

The average household income column the intensity of color might be reversed to show concentrations of low income census tracts.





On your own: Add a layer of Points

Try adding a new layer using the dataset named “RetailFoodStores.geojson”

Follow the same steps as the previous layers.

Set the layer type to “Symbol” and add the field for the name of the store.

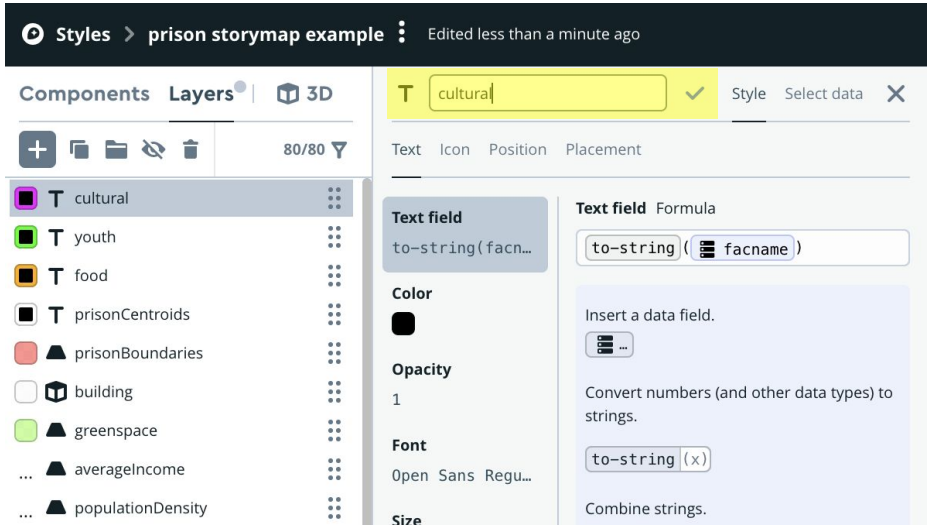
Rename and check the name of all visible layers

Data layer names are automatically generated by mapbox and will contain the name of the original file you uploaded followed by a unique string.

We will be accessing the layers by their name and need to make sure each layer name is descriptive and easy to reference.

Click on the the layer name field highlighted in yellow to change.

Toggle any temporarily hidden layers to be included in your final map style.

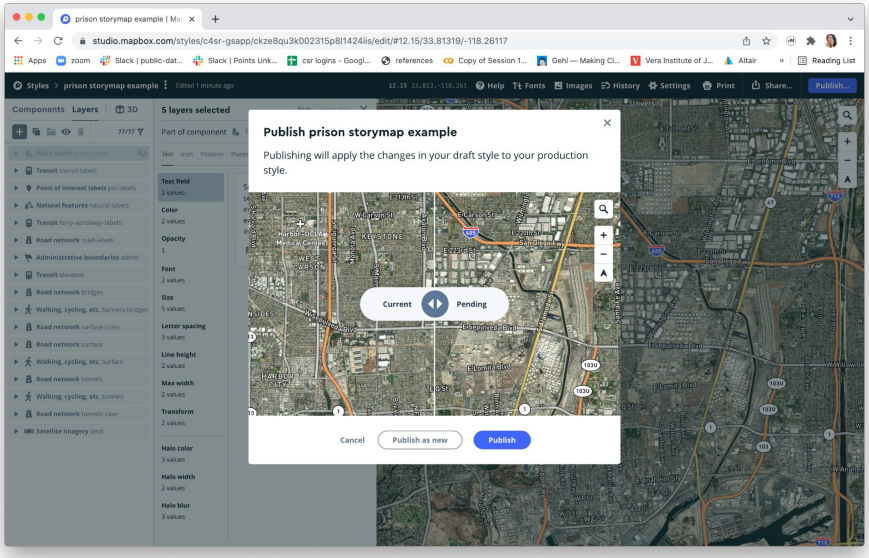


Publish your map

When you have finished editing your map, publish it by clicking on “Publish” on the upper right corner of your window.

Check within the popup window that the changes are what you expect by swiping between the current and pending views.

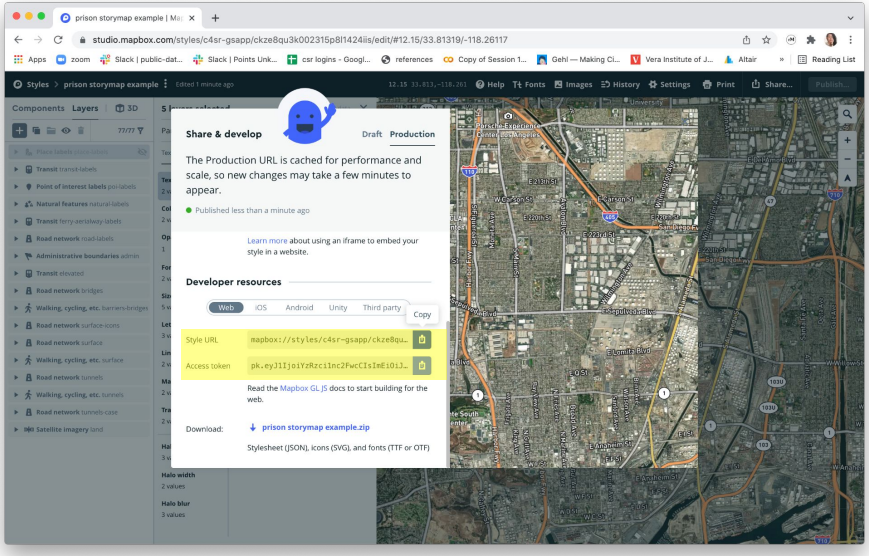
Click “Publish” again, this popup will close automatically.



Now click on “Share” just to the left of the Publish button on the upper right corner of the window.

Under the “Share style” section you will see a link to preview your map, an option to embed your map in an iframe.

Scroll down to the “Developer resources” section, you will see a “Style URL” and an “Access token”. Copy the URL and the token to your javascript(.js) configuration file.



Switching over to your local development environment: Visual Studio

- You can go back to mapbox to make changes as you go, add or erase layers, change color codings, etc..

- Always republish the map after each change you make.

- Your “Style URL” and “Access token” will be the same, and do not need to be copied over again as long as you are using the same map style and account.

Adding content and interactivity with Visual Studio Code

Steps

- Open project template with Visual Studio
- Run live version on browser

- Within the config.js file
  - Update link to mapbox style and access token
  - Update text content for each chapter
  - Update map settings for each chapter
  - Test on browser iteratively

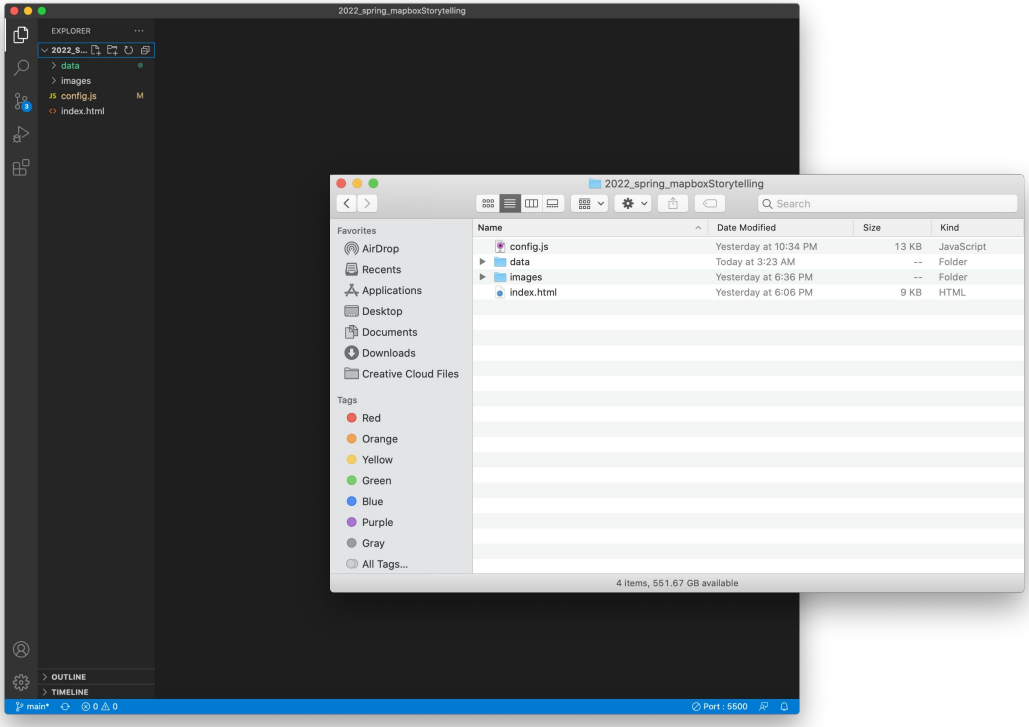
Open your project in Visual Studio Code

Open Visual Studio and click on “Open Folder” or go to “File” and select “Open Folder”

Navigate to the tutorial folder that you downloaded and unzipped earlier.

Click open.

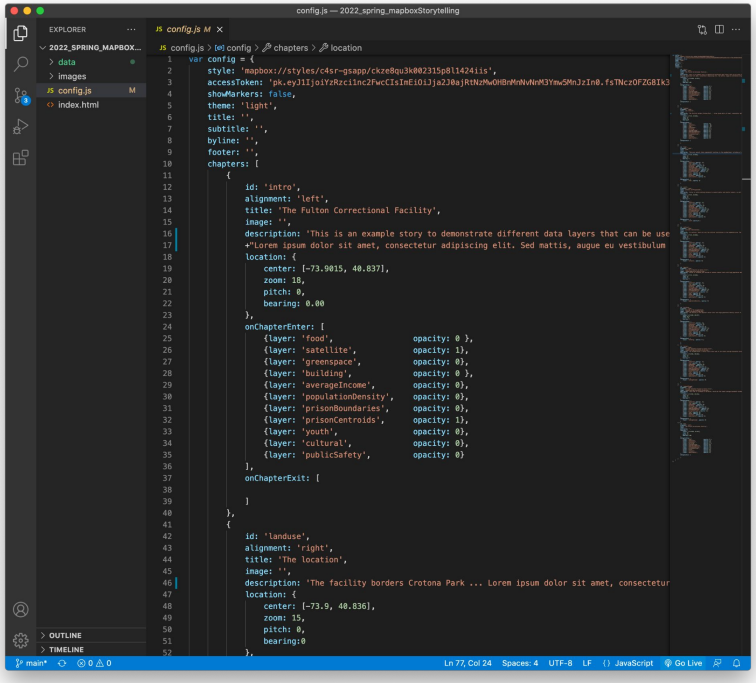
The file browser panel on the left hand side of your Visual Studio window should list the same items as if you view the project folder in the finder

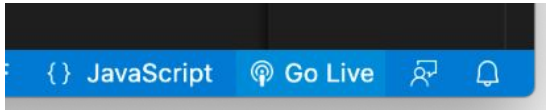


Run your code in the browser

Open the HTML file called “config.js”

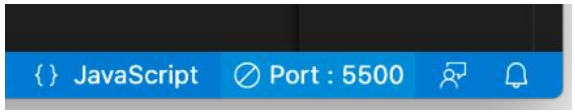
Click “Go Live” on the blue bar at the bottom of the window





The Go Live text will change to “Starting” and then to show a port number.

This is the local address where your webpage will be shown



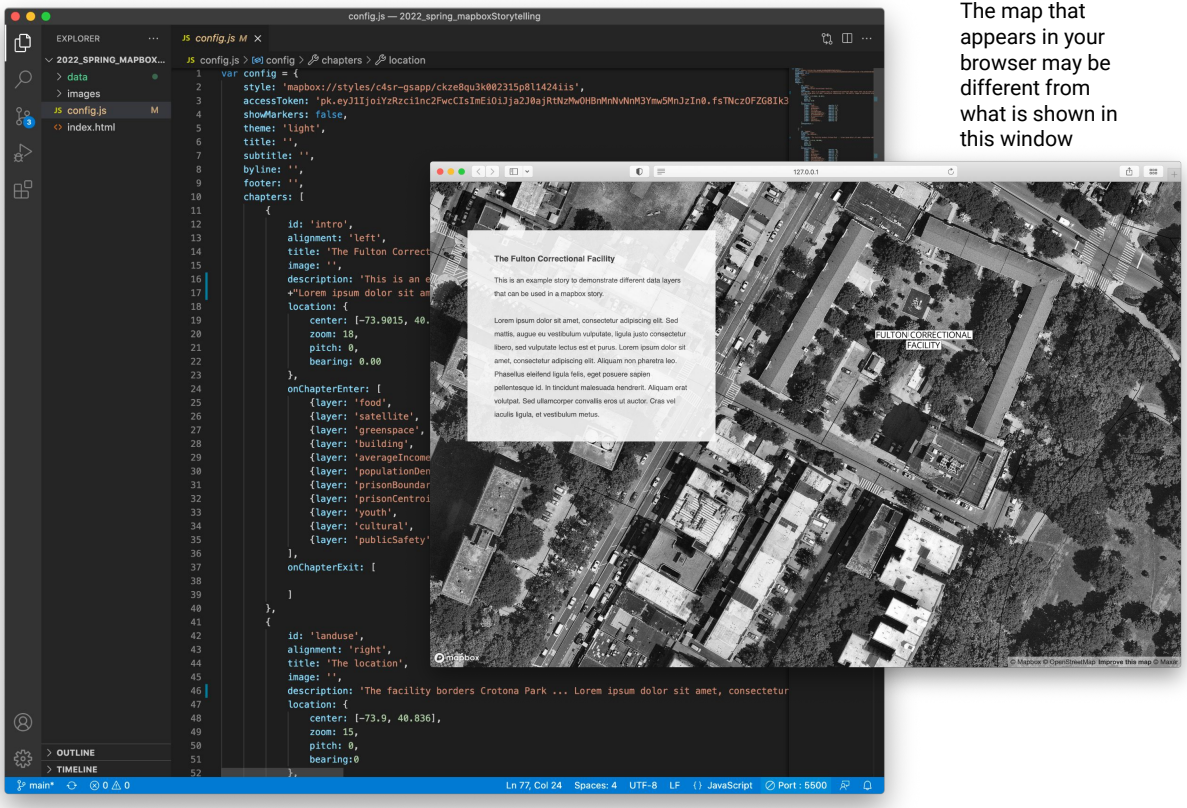
A browser window will popup automatically to show our webpage.

You can also view the webpage by opening a new browser window and going to “http://localhost::” and the port number displayed in your window. In this case http://localhost:5500/

Our workspace so far

We will need these 2 windows open:

1. code which we will edit
2. browser to view our changes



The map that appears in your browser may be different from what is shown in this window

Add your mapbox style URL and access token

The config.js file consists of a single variable called “config” which is a javascript object.

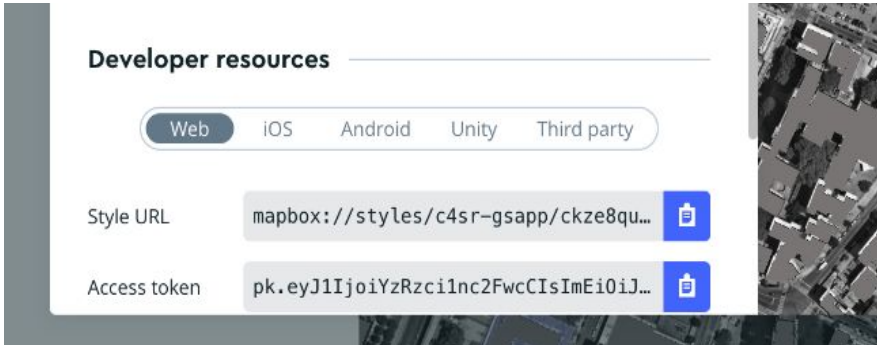
A javascript object is a set of key and value pairs.

For example in the screenshot below the keys are “style” and “accessToken” and the values are the link to a map style and access token.

```
style: 'mapbox://styles/c4sr-gsapp/ckze8qu3k002315p8l1424iis',
accessToken: 'pk.eyJ1IjoiYzRzci1nc2FwcCIsImEiOiJja2J0ajRtNzMwOHBnMnNvNr
```

Update these 2 values by copying and pasting the URL and access token of the map we just created.

If you have already closed the mapbox window, go to the style again and click share to view the 2 values.



Add your written content to the chapters

As stated previously - The config.js file consists of a single variable called "config" which is a javascript object. A javascript object is a set of key and value pairs.

Another example of this is the structure of the chapters in our story seen at right

You can add as many chapters as your story requires, each chapter is structured the same and contains keys which we can change the values of to create our story:

id - use a unique id that is descriptive of what is contained in the chapter, for example: "intro", "populationDensity", etc..

alignment - your text box for each chapter can be aligned to the left, center or right.

title - this will appear as the large title text in the text box for each chapter

image - is optional field where you can insert photographs, charts, or legends and will appear between the title and the paragraph text

description - is the main text in your chapter, copy and paste your prewritten text here and check to make sure you preserve the beginning and end quotation marks

location - the values that can be set within location determine the viewport of the map, you can set the size and angle of the map here.

```
chapters: [
    {
        id: 'intro',
        alignment: 'left',
        title: 'The Fulton Correctional Facility',
        image: '',
        description: 'This is an example story to demonstrate differ
        +"Lorem ipsum dolor sit amet, consectetur adipiscing elit. S
        location: {
            center: [-73.9015, 40.837],
            zoom: 18,
            pitch: 0,
            bearing: 0.00
        },
        onChapterEnter: [
            {layer: 'food',                opacity: 0 },
            {layer: 'satellite',           opacity: 1},
            {layer: 'greenspace',          opacity: 0},
            {layer: 'building',            opacity: 0 },
            {layer: 'averageIncome',       opacity: 0},
            {layer: 'populationDensity',   opacity: 0},
            {layer: 'prisonBoundaries',    opacity: 0},
            {layer: 'prisonCentroids',     opacity: 1},
            {layer: 'youth',               opacity: 0},
            {layer: 'cultural',            opacity: 0},
            {layer: 'publicSafety',        opacity: 0}
        ],
        onChapterExit: [

        ]
    },
    {
        id: 'landuse',
        alignment: 'right',
        title: 'The location',
        image: '',
        description: 'The facility borders Crotona Park ... Lorem ip
        location: {
            center: [-73.9, 40.836],
            zoom: 15,
            pitch: 0,
            bearing:0
        },
```

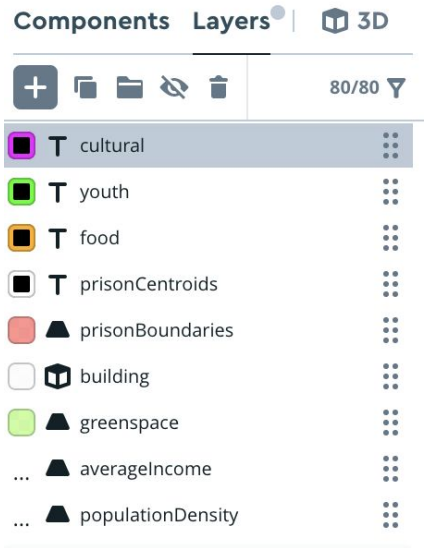
Ln 77, Col 24 Spaces: 4

Add layer transitions to the chapters

onChapterEnter and **onChapterExit** - These 2 final keys take a list of layer settings each. Put layers by their name and set their opacity for each chapter here.

Opacity is any value between 0 and 1(inclusive), for example .5 is 50% transparent.

I have listed all the layers that are visible in my mapbox style on the right under “onChapterEnter” and set their opacity to 0 except for the “satellite” and “prisonCentroids” layers. The layer names I have listed on the right match the ones in my style exactly(below).




```
chapters: [
    {
        id: 'intro',
        alignment: 'left',
        title: 'The Fulton Correctional Facility',
        image: '',
        description: 'This is an example story to demonstrate differ
        +"Lorem ipsum dolor sit amet, consectetur adipiscing elit. S
        location: {
            center: [-73.9015, 40.837],
            zoom: 18,
            pitch: 0,
            bearing: 0.00
        },
        onChapterEnter: [
            {layer: 'food',                opacity: 0 },
            {layer: 'satellite',           opacity: 1},
            {layer: 'greenspace',          opacity: 0},
            {layer: 'building',            opacity: 0 },
            {layer: 'averageIncome',       opacity: 0},
            {layer: 'populationDensity',   opacity: 0},
            {layer: 'prisonBoundaries',    opacity: 0},
            {layer: 'prisonCentroids',     opacity: 1},
            {layer: 'youth',               opacity: 0},
            {layer: 'cultural',            opacity: 0},
            {layer: 'publicSafety',        opacity: 0}
        ],
        onChapterExit: [

        ]
    },
    {
        id: 'landuse',
        alignment: 'right',
        title: 'The location',
        image: '',
        description: 'The facility borders Crotona Park ... Lorem ip
        location: {
            center: [-73.9, 40.836],
            zoom: 15,
            pitch: 0,
            bearing:0
        },
```

As you add content …

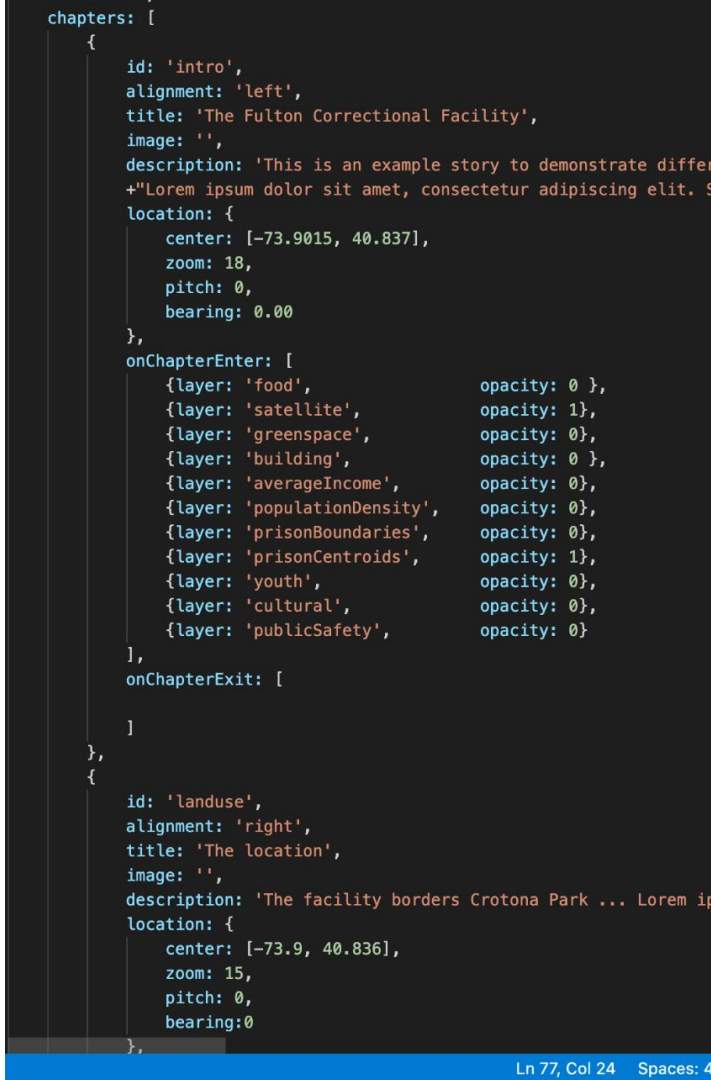
Save your work as you go

You will see the changes you make reflected on the browser immediately - as you type words will appear on the webpage, although some changes may require saving.

If you make a syntax error - such as missing a comma, quotation mark, or parentheses, your first clue of the error may be that the browser page will go blank because it is unable to render the page as you have written it in its current state.

Look for errors:

- The text editor helps you by color coding your writing. If something is not their assign type color, you may have missed punctuation somewhere.
  You can see that on the page to the right
  - all the keys are in light blue
  - all the number values such as the zoom level and the latitude and longitude are green
  - all the text paragraphs and sentences(string) are brown
  - and the punctuation is white
- The text editor also shows some errors with a red underline, this is similar to the automatic spell check function in a word document. Look for those underlines.
- You can undo your changes and save to go back to a stage where the page was working. It is good to check that your page is rendering correctly frequently so that you do not miss the error and write for too long after an error occurs.



Publishing your story

Method is tbd

Mapbox references

https://www.mapbox.com/solutions/interactive-storytelling
https://labs.mapbox.com/education/impact-tools/interactive-storytelling/
https://www.mapbox.com/webinars/storytelling-template-live-demo

Other Tutorials

https://centerforspatialresearch.github.io/methods-in-spatial-research-sp2022/
https://pointsunknown.nyc/