Mapbox for Journalists — Scrollytelling Template

Some stories are best told with a map. Data journalists covering changing conditions in a population's demographics, the environment, an international conflict, or telling a simple travel story frequently provide geographic context in their graphics. As we read these stories, we scroll. The output of this template harnesses the reader's natural scrolling to tie sections of the story to a custom Mapbox map. As the story progresses, it triggers a map in the background to pan, zoom, and rotate, supplementing the text with dynamic visuals and geographic context.

This template is designed to accelerate building out a "scrollytelling" map story. The primary input is a story broken into sections, each hooked to a particular view of a map. An optional input is a custom Mapbox Studio Style with layers of data that help convey the message of the story. The output is an HTML file and a JavaScript configuration that work together to build the interactive story. These outputs can be hosted on any web-accessible location, with no extra code or infrastructure required.

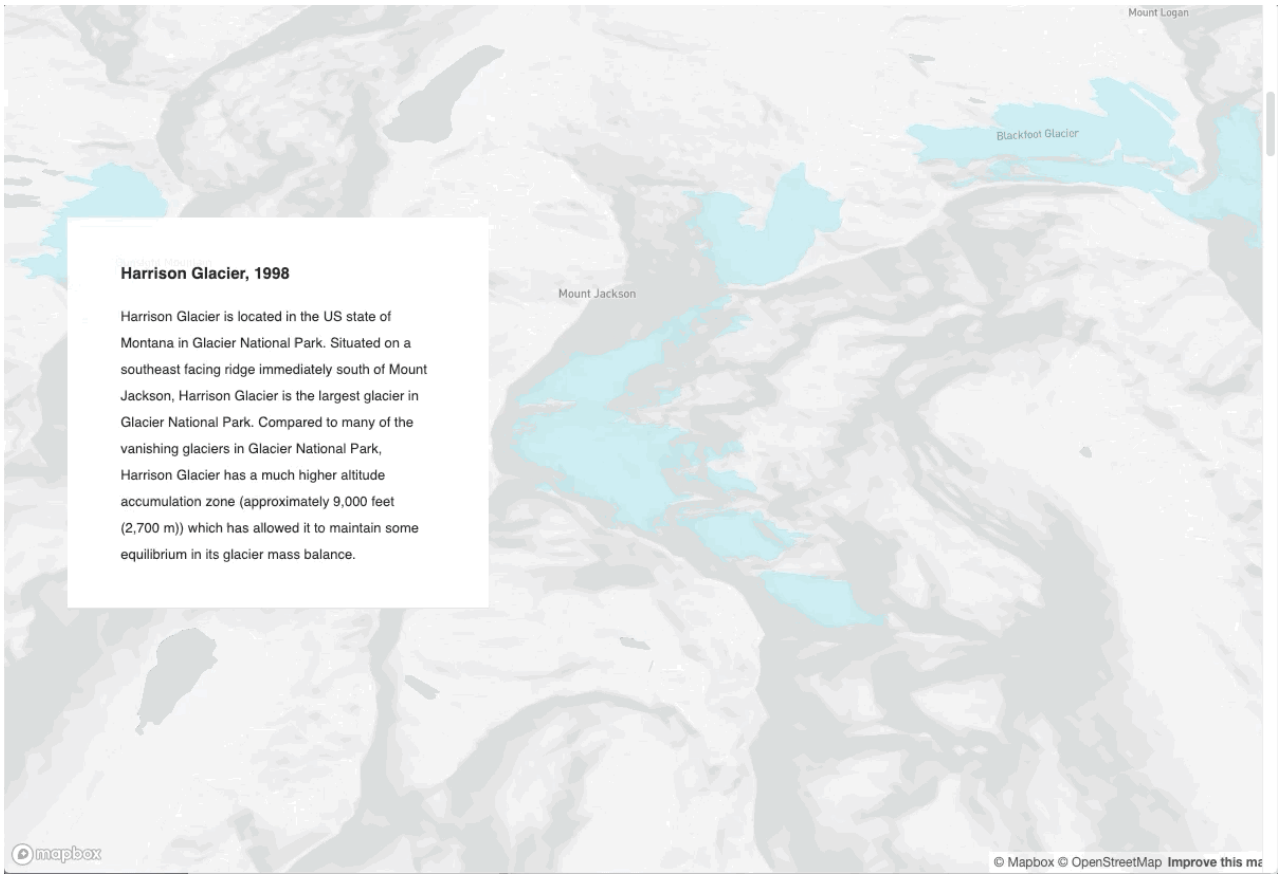
Prerequisites

This template is for data journalists and digital storytellers of any kind. No coding experience is required. To configure and publish a story, you will need:

- A Mapbox access token. Sign up for a free account at mapbox.com to get one.
- A text editor. Atom, Sublime Text, and Visual Studio Code are all fine choices. Notepad on Windows and TextEdit on Mac will also work in a pinch.
- A place to publish your work. Any service that hosts static files that can be accessed with a browser will do. This can be a web server, an AWS S3 bucket, or a service like GitHub Pages or Netlify.
- A story. This is unquestionably the hardest part. The best stories for this template will have sections that benefit from a map visual. These could be cities, addresses, regions, rivers... any place or set of places.
- Attention to detail. The configuration file does require specific syntax and punctuation. Braces, brackets, commas, and quotes are important. Follow the `config.js.template` for guidance.

- Optionally, some spatial data in your Mapbox map. The template has options to include layer names to show and hide the data as the story sections transition. You may want to highlight a neighborhood, or show satellite data from two different times.

The template does not rely on any particular CSS framework, fonts, or images. There are some basic styles in the `head` of the HTML file that can be changed, so feel free to adapt and add to these to match your site and story brand.



Getting Started

- Download this repository as a ZIP file using the button above, and unzip it. If you are using `git`, clone this repository.
- If you are new to coding in JavaScript, follow the instructions for Vanilla JS. If you are already working with React, are comfortable with the command line and build systems, and/or want bundled and minified code, choose React.

Vanilla JS

In your local copy of this repository (the unzipped file you downloaded), navigate to the `src/vanilla-js/` directory.

Make a copy of `config.js.template` and name it `config.js`. Open the new `config.js` file in your text editor. Then:

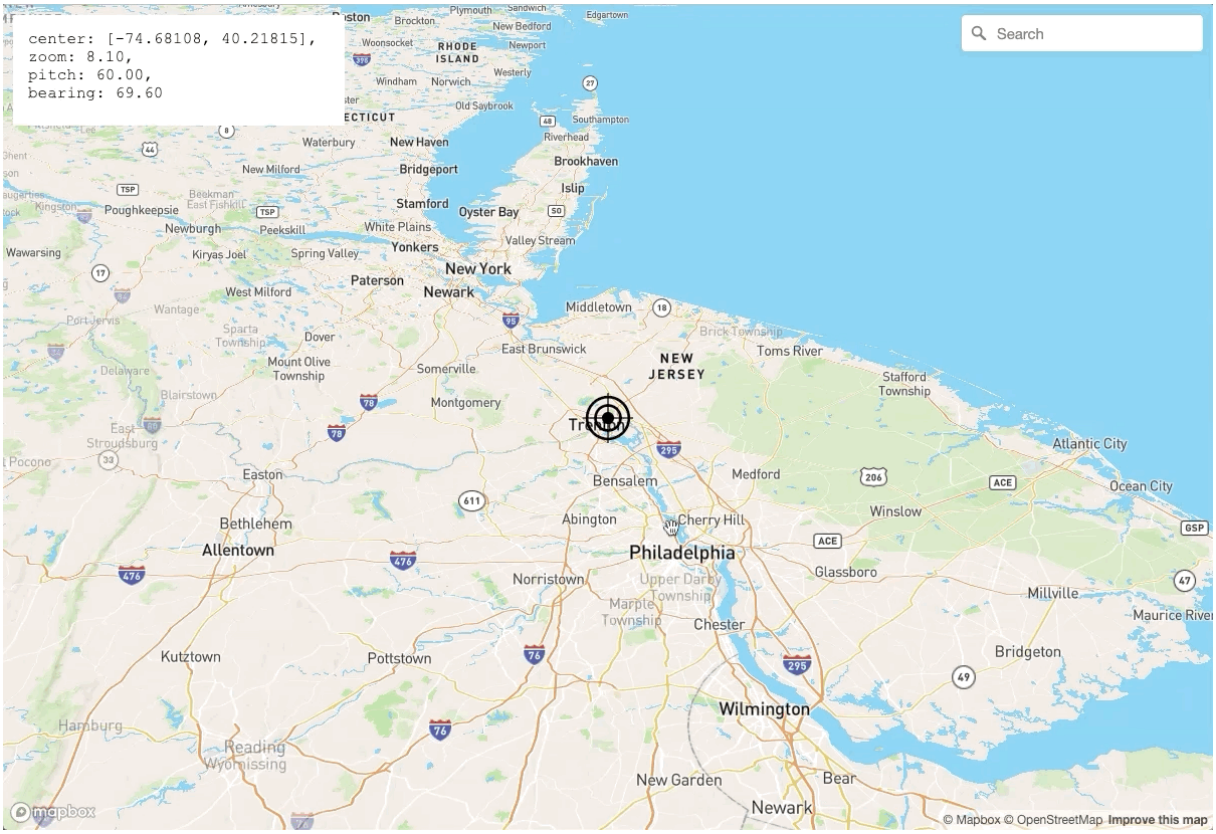
1. Select the map style you want to use (the default is Mapbox Streets, but you can find more here https://docs.mapbox.com/api/maps/#styles, or use one of your custom Studio styles).
2. Add a Mapbox access token. A good practice is to create a separate token per map to be able to track traffic to your different maps.
3. Choose whether or not to display a marker at the center of each map location.
4. Choose where your story should be aligned over the map. It can be centered or pushed to the left or right.
5. Copy and paste the sections in your template until you have enough to fill your story in. You'll need a `,` between each section, but no comma at the end. A "section" is this part of the template:

```
{
    id: 'identifier',
    title: 'Title',
    image: './path/to/image/source.png',
    description: 'Copy these sections to add to your story.',
    location: {
        center: [-77.020636, 38.886900],
        zoom: 13.5,
        pitch: 60,
        bearing: -43.2
    },
    showLayers: [],

    hideLayers: []
}
```

6. Paste in your section headings and body content into each section. Also give each section a unique name in the section `id` property. If you have an image that goes with that section of the story, add the path to the image in the `image` property.
7. Using `helper.html`, you can search for places, zoom, pan, tilt, and rotate the map until it is exactly as you want it (*Hint*: To tilt and rotate the map, right-click and drag the map). Notice the location parameters are updated in the upper left with every move of the map. Then copy the location definition from that page into the

`config.js` `location` property section for that piece of the story.



8. Repeat until you have the location entered for each of your sections.
9. Open `index.html` in a browser, and scroll. Voila!

Understanding the `config.js` file and layer settings

Here is a sample configuration:

```
var config = {
    style: 'mapbox://styles/branigan/cjz37rcb003ib1cr3s8rnkt2d',
    accessToken: 'pk.eyJ1IjoiYnJhbmlnYW4iLCJhIjoiY2p4NHVmenFrMDBnMDRibGJveXU
    showMarkers: false,
    alignment: 'center',
    mapStart: {
        center: [-113.91666, 48.66451],
        zoom: 8,
        pitch: 0.00,
        bearing: 0.00
    },
    chapters: [
        {
            id: 'glacier-np',
            title: 'Glacier National Park Glaciers',
            image: 'https://upload.wikimedia.org/wikipedia/commons/thumb/e/e
            description: 'Glacier National Park is dominated by mountains wh
            location: {
                center: [-113.91666, 48.66451],
```

```
                zoom: 8,
                pitch: 0.00,
                bearing: 0.00
            },
            showLayers: [
                {
                    layer: 'gnpglaciers-1998',
                    opacity: 0.25
                },
                {
                    layer: 'glaciernp-boundary',
                    opacity: 0.25
                }
            ],
            hideLayers: ['glaciernp-boundary']
        },
        {
            id: 'harrison1998',
            title: 'Harrison Glacier, 1998',
            image: '',
            description: 'Harrison Glacier is located in the US state of Mon
            location: {
                center: [-113.72917, 48.58938],
                zoom: 12.92,
                pitch: 39.50,
                bearing: 36.00
            },
            showLayers: [],
            hideLayers: []
        }
    ]
}
```

Configuration options:

`style` : This is the Mapbox style to use for the app. It can be a standard style, or a custom style in your Mapbox account. Use a custom style if you want to include custom data or layers.

`accessToken` : Your Mapbox access token.

`showMarkers` : This controls whether markers are shown at the centerpoint of each chapter. If `true` , the map will display a default blue, inverted-teardrop icon.

`alignment` : This defines where the story text should appear over the map. Options are center, left, and right. When the browser window is less than 750 pixels wide, the story will be center aligned.

`mapStart` : The location for where to start the map when the page loads. If this location is different than the first chapter, you will see an animated zoom-in when the page first loads.

`chapters` : This contains all of the story content and map controls for each section of the story. *Array of objects*

- `id` : A slug-style ID for the chapter. This is read by the JavaScript driving the app and is assigned as an HTML `id` for the `div` element containing the rest of the story. A best-practice format would be to use kebab case, like `my-story-chapter-1` .
- `title` : The title of the section, displayed in an `h3` element.
- `image` : The path to an image to display in this section.
- `description` : The main story content for the section. This should be aligned with what the reader is seeing on the map. In the vanilla version, this field will render as HTML. Images, links, and other items can be included as HTML.
- `location` : Details about the map display and camera view.
  - `center` : Center coordinates of the map, as `longitude, latitude`
  - `zoom` : Zoom level of the map.
  - `pitch` : Angle of the map view. `0` is straight down, and `60` is highly tilted.
  - `bearing` : Degrees of rotation clockwise from North (`0`). Negative values represent counter-clockwise rotation.
- `showLayers` : Layers to be displayed when the section becomes active. This assumes that the layer is not currently shown and that its opacity is set to `0` . *Array of objects*
  - `layer` : Layer name as assigned in Mapbox Studio.
  - `opacity` : The opacity to display the layer. `0` is fully transparent, `100` is fully opaque.
- `hideLayers` : Layers to be hidden when the section becomes inactive. Any layers listed in this array will have opacity set to `0` . *Array of strings*

Mapbox Studio Style Configuration

Add and style each custom layer in your Studio style. Before the final publish, set any layers to be hidden with `0` opacity. This will ensure that the map appears correctly when the story page loads. To turn on and off layers as the reader scrolls through the story, use the `showLayers` configuration options to set your desired opacity for the layer.

React

The output of the React version of this template is functionally identical to the vanilla version. Use this version if you prefer to use React to build and deploy your story.

Installation and dependencies

This application was built and tested using

- Node version 10.16.3
- NPM version 6.9.0

In your local copy of this repository, navigate to the `src/react-js/` directory.

Install dependencies listed in the `package.json` file:

```
yarn install
```

Run the development server:

```
yarn start
```

Follow the instructions above for setting up your configuration file and building out your story. Once the application is ready for deployment, run:

```
yarn build
```

This command will generate a `build` directory that contains everything you will need to deploy your story. As with the vanilla JavaScript version, you will only need a deployment location capable of hosting static files.

Organization

- `src` : Source code directory
  - `react-js` : Code for the React version of the template
  - `vanilla-js` : Code for the vanilla version of the template
- `example` : Example stories for the vanilla version of the template
  - `glacier` : Glaciers of Glacier National Park example
  - `bike-philly` : Philadelphia bicycle infrastructure example

Deployment

Add additional notes about how to deploy this on a live system (cloud/on-prem/etc)

Built With

List out the core pieces of technology used in this demo, with a diagram

```
Show how this all works
```

Authors

Use the `Mapbox Solutions Architecture` as point-of-contact. Use Solutions Architecture solutions_architecture@mapbox.com address for contacts.

License

This project is licensed under the XYZ License - see the LICENSE.md file for details

Acknowledgments

- Hat tip to anyone whose code was used
- Inspiration
- etc

Link back to mapbox.com/solutions landing page.