

# Citi Bike

---

A large part of this class is going to be devoted to working on a mini-project in Leaflet. We will utilize the Citi Bike API in order to retrieve the statuses and locations of all of the Citi Bike stations in New York. It's highly recommended to complete the **Basic Version** before attempting the **Advanced Version**.

## Instructions

---

### Basic Version

- Use the [Citi Bike Station Information Endpoint](#) to retrieve information about station names and locations. Take a moment to study the data sent back by the endpoint in your browser.
  - Each object in the `stations` array has `station_id`, `name`, `capacity`, `lat`, and `lon` properties.
  - The `logic.js` file in the `skeleton` folder contains coordinates that can be used to position a Leaflet map over New York City.
- Create a function called `createMap` that will take in `bikestations` as an argument. This function will create both the tile layer and an overlay layer with the pins for each station.
- Create a second function `createMarkers` that will take `response` as an argument.
  - Using the response from a future `d3` call loop through the stations and create a marker to represent each station.
  - Give each marker a popup to display the name and capacity of its station.
- In the `createMarkers` function pass the result the bike makers into the the `createmap` function as an `layerGroup`.
- Perform a GET request using `D3` to the [Citi Bike Station Information Endpoint](#) that will call the `creatMarkers` function.
  - Remember to pass in your unique Mapbox token.



## Advanced Version

After completing the **Basic Version**, write code to complete as much of the **Advanced Version** as possible.

- Write code to perform a second API call to the [Citi Bike Station Status Endpoint](#). Take a few moments to study the data being returned. In particular we are concerned with the `station_id`, `num_bikes_available`, `is_installed`, and `is_renting`.
- Using the data retrieved from the second API call, try to add the following functionality:
  - \*. Display the number of bikes available inside the popup for each marker.

- Add a layer control and split the markers up into the following overlay layers:
  1. **Coming Soon:** If a station is not installed.
  2. **Empty Stations:** If a station has no bikes available.
  3. **Out of Order:** If a station is installed but not renting.
  4. **Low Stations:** If a station has less than 5 bikes available.
  5. **Healthy Stations:** If a marker does not fall into any of the previous layer groups.
- Utilize a Leaflet plugin to create different types of markers to represent each layer. The example shown below is using [Leaflet.extra-markers](#), but you are free to use another plugin if you prefer.
- Add a legend to your map to explain the different markers.



- Deploy the app to Github Pages when complete.

## Hints

- Make sure that you are running `python -m http.server` in the folder where your files are located. Since all the work is being done on the front end of your application, there will be no need to restart the router for changes made.
- Some helpful links:
  - [Leaflet Usage Example](#)
  - [Citi Bike Station Information API EndPoint](#)

- [Leaflet Popup Doc](#)
- [Citi Bike Station Status API EndPoint](#)
- [Leaflet Layer Groups Doc](#)
- [Leaflet Extra Markers](#)
- [Leaflet Legend Doc](#)