

Basic Maps Using CartoDB and JavaScript

The CartoDB JavaScript library¹ makes it easy to customize your maps. With it you can load visualizations, filter your visualization's layers, and even change the visualization's CartoCSS. This cheatsheet summarizes the basics of getting a map on a web page using JavaScript and CartoDB.

Basic Setup

There are three fundamental steps you will follow any time you use JavaScript to add a CartoDB map to your page:

1. **Load two libraries:** CartoDB.js and jQuery. Copy and paste the following near the beginning of your HTML file's head:

```
<link rel="stylesheet" href="http://libs.cartocdn.com/cartodb.js/v3/themes/css/cartodb.css" />
<!--[if lte IE 8]>
<link rel="stylesheet" href="http://libs.cartocdn.com/cartodb.js/v3/themes/css/cartodb.ie.css" />
<![endif]-->
<script src="http://libs.cartocdn.com/cartodb.js/v3/cartodb.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.0/jquery.min.js"></script>
```

These load CSS and JavaScript that will help you add your map to your page.

2. **Make a place for your map to go.** In your HTML file's body, add a `div` with an `id` that you'll remember. It's common to use “map” for the `id`:

```
<div id="map"></div>
```

This is a standard `div` element that you can style using CSS. Set the `height` and `width` for your map `div`. If you don't, the height and width will be 0, so you won't see your map! For example:

```
#map {
  height: 100%;
  width: 100%;
}
```

3. **Create the map.** At the bottom of your document's head, add a script element like this:

```
<script>
  $(document).ready(function () {
    cartodb.createVis(your_map_id, your_visualization_url);
  });
</script>
```

Replace `your_map_id` with the `id` of your map `div` and `your_visualization_url` with the url that you copy from the Share dialog of your CartoDB visualization.

What is this code doing?

Step 3 can be hard to read. Let's break it down line by line.

First, JavaScript code goes in a `script` element. So we create one (usually in the head of your document). The opening and closing tags are the first and fifth lines of the above code.

¹ <http://developers.cartodb.com/documentation/cartodb-js.html>

The second line, `$(document).ready(function () {` uses jQuery to wait for the page to load before going on. If we didn't do this, there is a chance that your code would try to create your map before the HTML is ready to show it—it would be broken! The fourth line closes the `(` and `{` on the second line with `});` without which our code would be broken since it is not closed. This is exactly like CSS and CartoCSS, where each rule must have matching `{` and `}` braces.

Finally, the third line, `cartodb.createVis(...)`, creates the map. `createVis` is a function in `cartodb` that loads visualizations. Here it is taking two parameters: the `id` of the HTML element where the visualization should go and the url of the visualization.

The screenshot shows the CartoDB 'Share settings' interface. On the left, there are toggle switches for 'Title', 'Description', 'Shareable', 'Search box', 'Layer selector', 'Scroll wheel zoom', 'Legends', and 'Fullscreen'. Below these are 'PRIVACY OPTIONS' set to 'Public'. A red arrow points from the text `your_visualization_url` to the 'API' button in the 'COPY AND PASTE THIS CODE' section. The code provided is `http://eric.cartodb.com/api/v2/viz/22a36ecc-0eb6-11e3-8f2f-5404a6a69006/viz.json`. Other buttons for 'PUBLIC URL', 'EMBED URL', and 'IFRAME' are also visible.

Configuration

Now that you have a map on your page, you can take advantage of extra configuration options that you do not have through CartoDB's Share dialog².

There are numerous settings you can change in this way. They go in a third, optional, parameter to `cartodb.createVis()`. The options are formatted in a way much like properties in a CSS or

² Under **Options** here:
<http://developers.cartodb.com/documentation/cartodb-js.html#sec-3-1>

CartoCSS rule:

```
{
  option1: value1,
  option2: value2
}
```

Add these options to `cartodb.createVis()` after your map's id and your visualization's url:

```
<script>
  $(document).ready(function () {
    cartodb.createVis(your_map_id, your_visualization_url, your_map_options);
  });
</script>
```

For example:

```
<script>
  $(document).ready(function () {
    cartodb.createVis('map', 'http://eric.cartodb.com/api/v2/viz/.../viz.json', {
      cartodb_logo: false
    });
  });
</script>
```

would create a map and set the `cartodb_logo` option to `false`. This hides the CartoDB logo from the resulting map.

Conclusion

Your code should look something like this when you're finished:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="http://libs.cartocdn.com/cartodb.js/v3/themes/css/cartodb.css" />
    <!--[if lte IE 8]>
    <link rel="stylesheet"
href="http://libs.cartocdn.com/cartodb.js/v3/themes/css/cartodb.ie.css" />
    <![endif]-->
    <script src="http://libs.cartocdn.com/cartodb.js/v3/cartodb.js"></script>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.0/jquery.min.js"></script>

    <style>
      #map {
        height: 500px;
        width: 500px;
      }
    </style>

    <script>
      $(document).ready(function () {
        cartodb.createVis('map', 'your_visualization_url', {
          cartodb_logo: false
        });
      });
    </script>

  </head>
  <body>
    <div id="map"></div>
  </body>
</html>
```

Next Steps

Once you have followed the steps above to add a CartoDB map to your page, you can start to make the map even more interactive than it is by default. This will require a bit more coding. See “**Advanced Maps Using CartoDB and JavaScript**” to learn more.