

Introduction to



Hands-On Workshop

Lab 5 - Charts

Overview

In the first two labs of this workshop you created a MongoDB cluster and loaded your data. How nice would it be to now visualize that data to gain valuable insights? Well, with [MongoDB Charts](#) that's now entirely possible.

We've been asked to provide a dashboard that displays:

- The number of cuisine types in each NYC borough
- The Top 5 Cuisines in Manhattan

Let's see how easy this is to accomplish...

Prerequisites

You've completed Labs 1 and 2 of this workshop, so you have a MongoDB cluster deployed in Atlas and you've loaded the restaurant data set.

Hands-On Exercises

Exercise 1 - Visualize Your Data

MongoDB Charts allows you to quickly and easily unlock the value in your data. Let's do some analysis on our NYC restaurant data:

Activate Charts

Click the **Charts** menu under SERVICES on the left on the left:

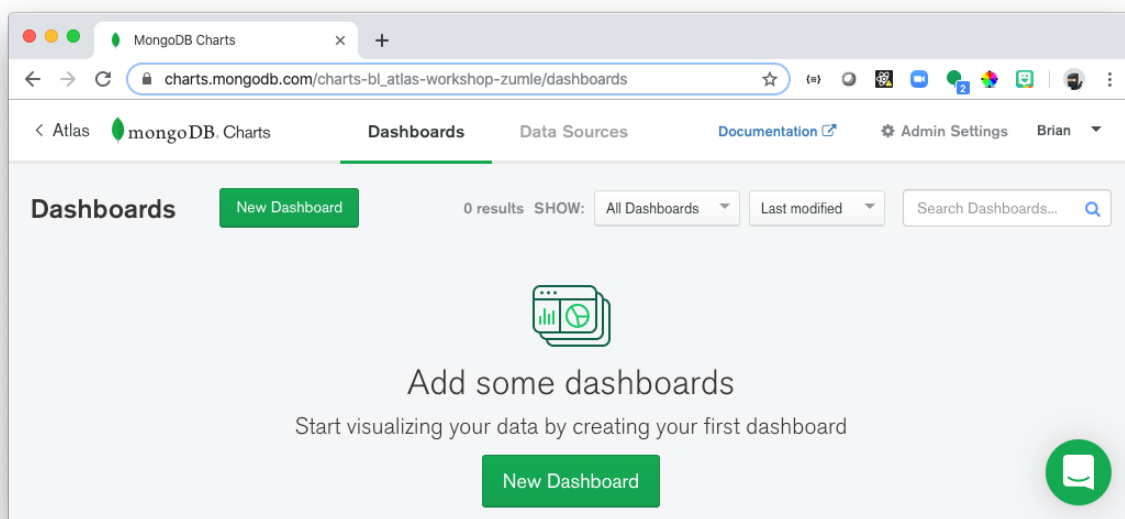


Get Started with MongoDB Charts

The fastest and easiest way to create visualizations from MongoDB Atlas

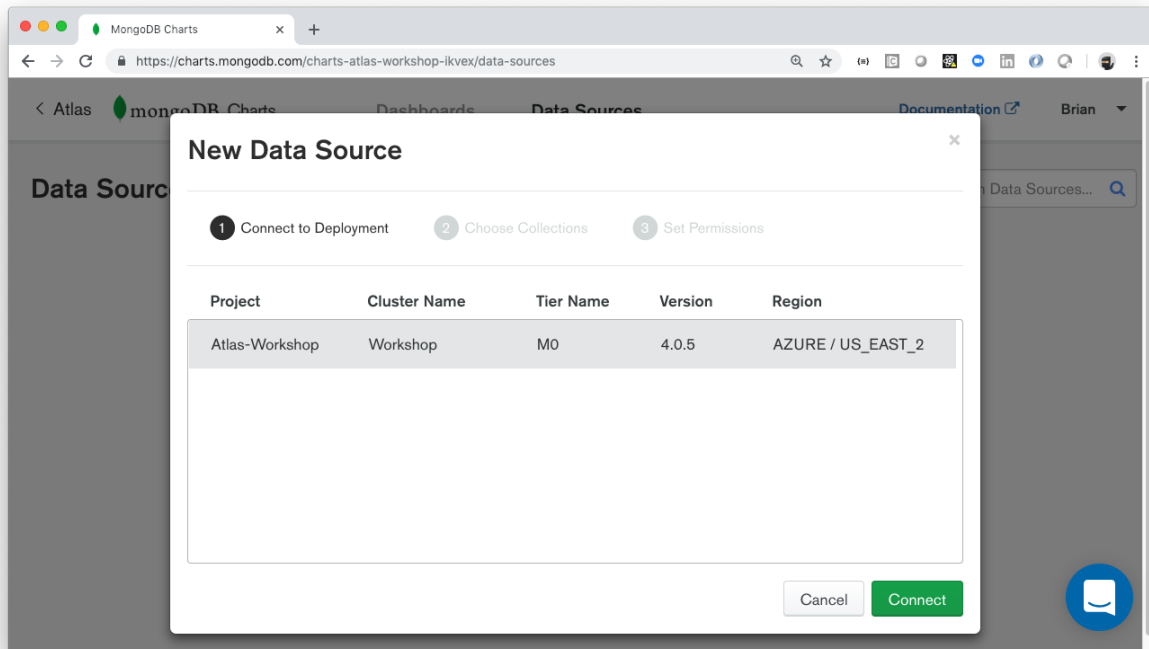
Activate MongoDB Charts

And **Activate MongoDB Charts**.



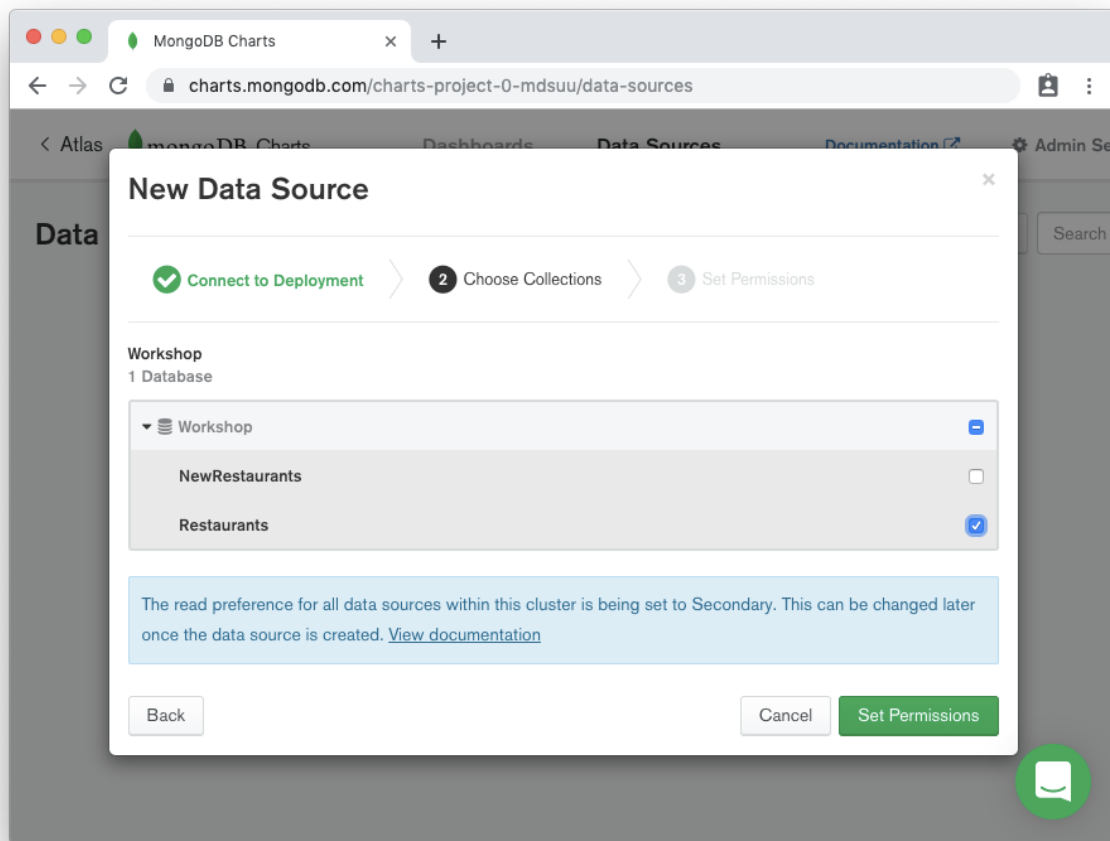
Define a Data Source for the Charts

Select **Data Sources** from the menu at the top of the page and add a **New Data Source**. As we only have one cluster in our project, there's only one data source to select:

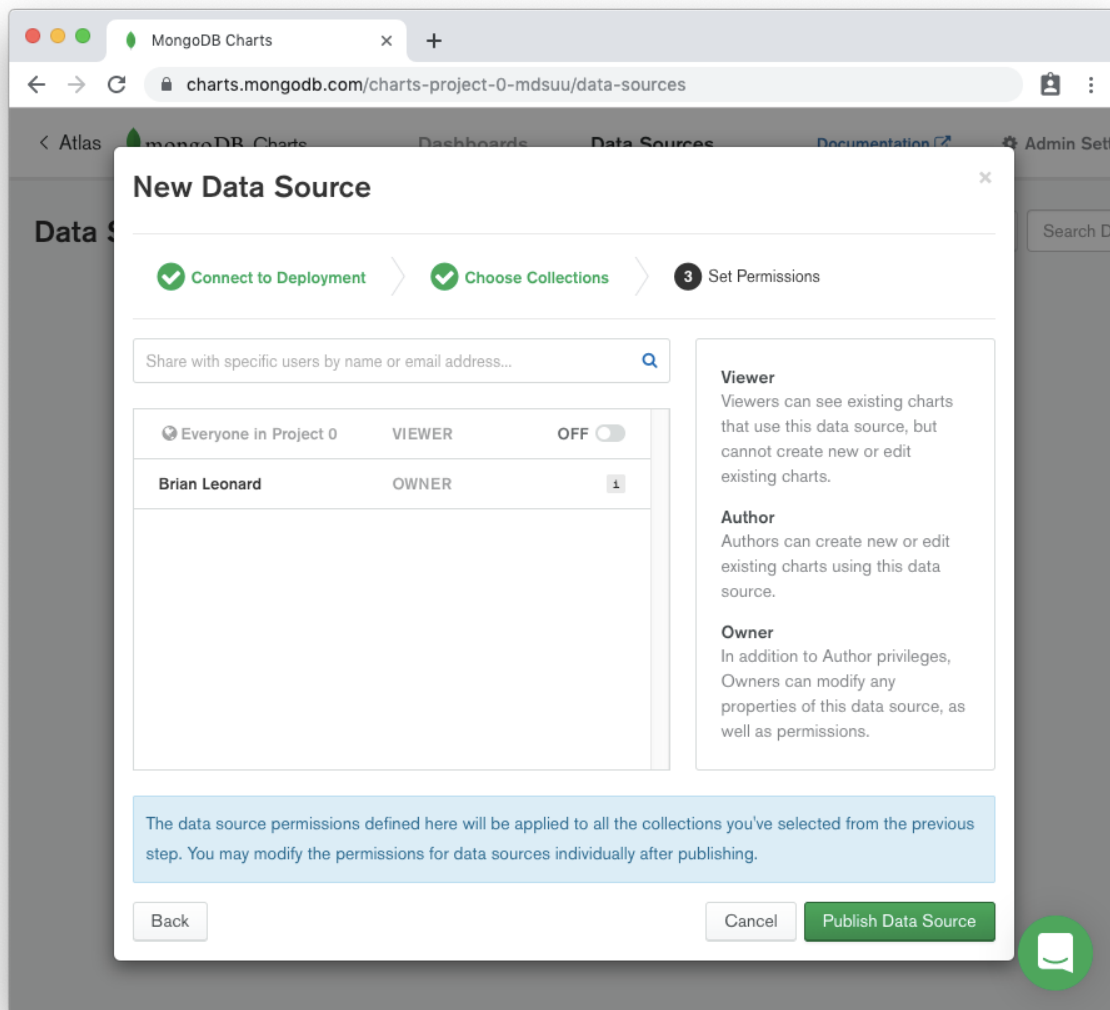


Click **Connect**.

Next we have to select the collection(s) from the data source that we want to work with. Select the **Restaurants** collection:



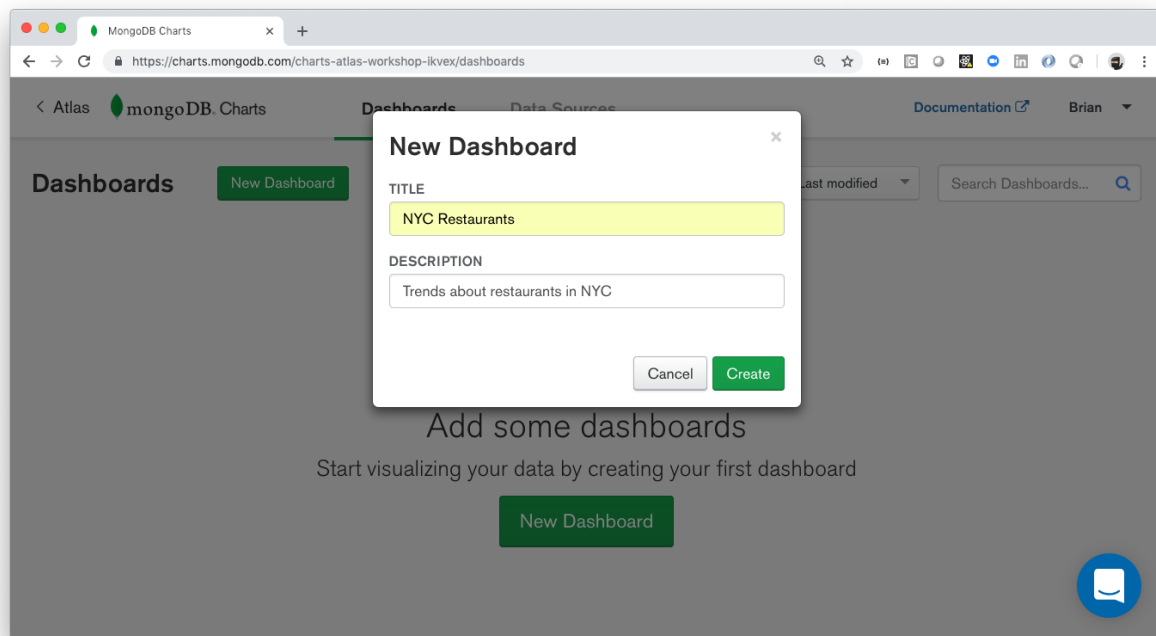
Click **Set Permissions**:



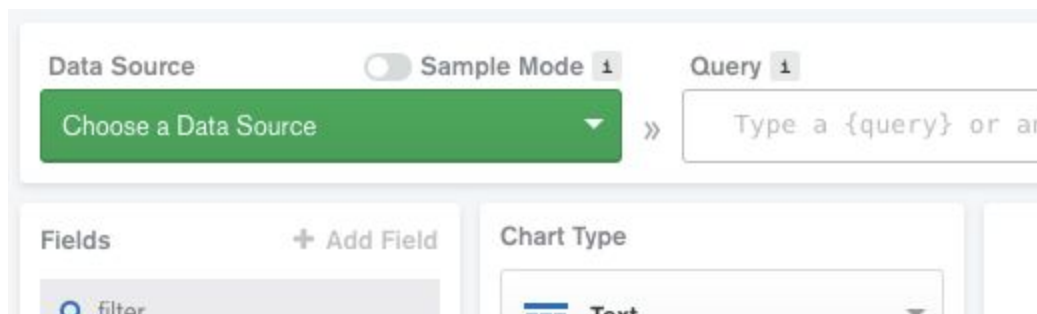
And finally, **Publish Data Source**.

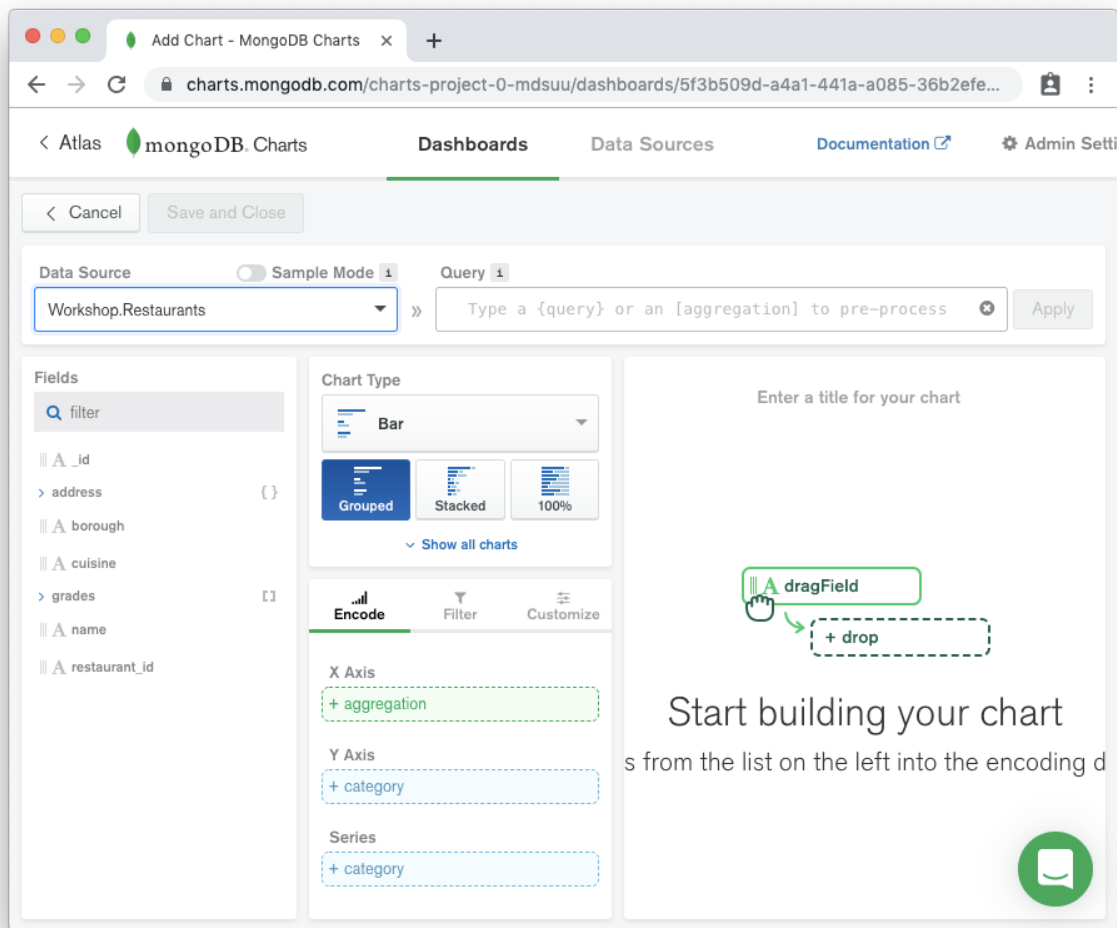
Create a Dashboard and Charts

With our data source published, we're now ready to start building some dashboards. Click the **Dashboards** menu and **New Dashboard**. Name the dashboard **NYC Restaurants** and describe it as **Trends about Restaurants in NYC**:



Click **Create**. The next step is to add charts to the dashboard. Click **Add Chart** to load the chart builder. Where it says “Choose a Data Source”, select the **Workshop.Restaurants** data source we just established:





For this lab, we'll just add one simple chart. Actually, a report of cuisines by borough. Set the Chart Type to **Text**. Then drag **borough** and **cuisine** to the Groups category. You'll see the chart update to reflect our inputs:

MongoDB Charts

https://charts.mongodb.com/charts-atlas-workshop-ikvex/dashboards/7e9b6bf7-ca89-4325-b720-c7bba7c4b2c8/c...

< Atlas **mongoDB. Charts** Dashboards Data Sources Documentation Brian

< Cancel Save and Close

Data Source Workshop.Restaurants Sample Mode Filters Apply

Fields

- filter
- id
- address
- borough
- cuisine
- grades
- name
- restaurant_id

Chart Type

Text

Table 100 Number

Groups

- borough
- cuisine
- + category

Values

- + aggregation

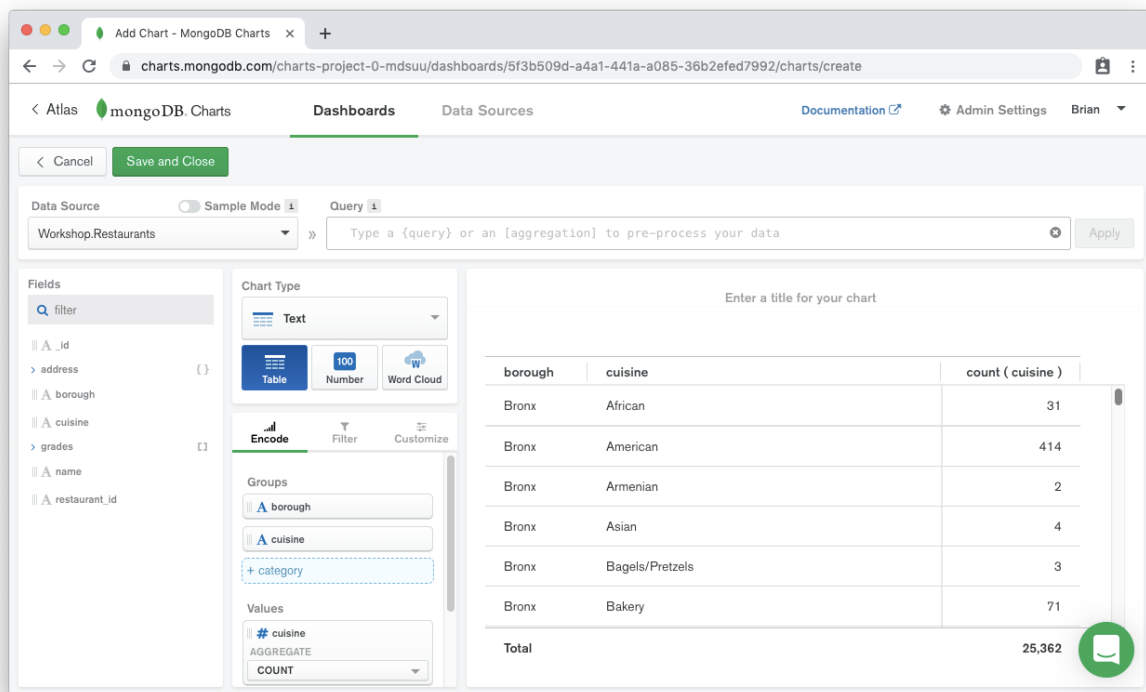
Dynamic Columns

- + category

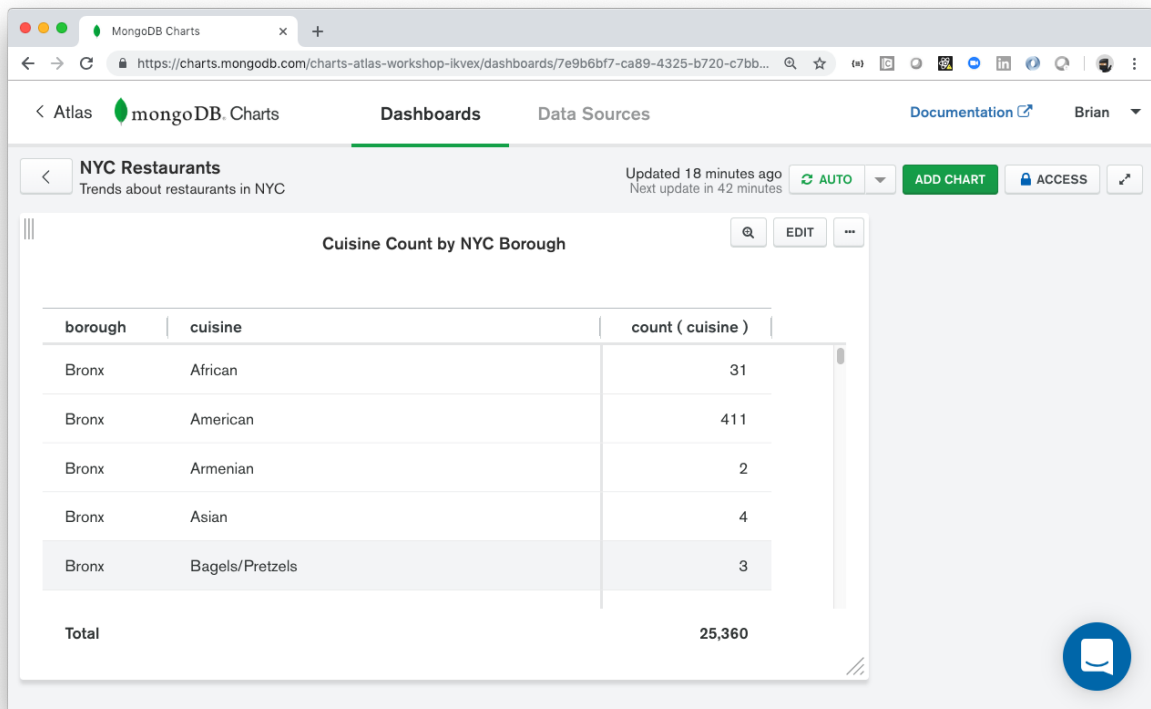
Enter a title for your chart

borough	cuisine
Bronx	African
Bronx	American
Bronx	Armenian
Bronx	Asian
Bronx	Bagels/Pretzels
Bronx	Bakery
Total	

Then drag **cuisine** to the Values aggregation:



Finally, title your chart **Cuisine Count by NYC Borough** then **Save and Close**. You have your first chart on the dashboard:



Embed a Chart

Charts can be [embedded](#) into your application. Let's quickly create a new chart that shows the 5 most popular cuisines in Manhattan.

Click **Add Chart** and select the **Workshop.Restaurants** data source.

Drag **cuisine** to the Y Axis (+ category) and **restaurant_id** to the X Axis (+ aggregation). Limit the results of the Y Axis (cuisine) to 5:

Y Axis

A cuisine

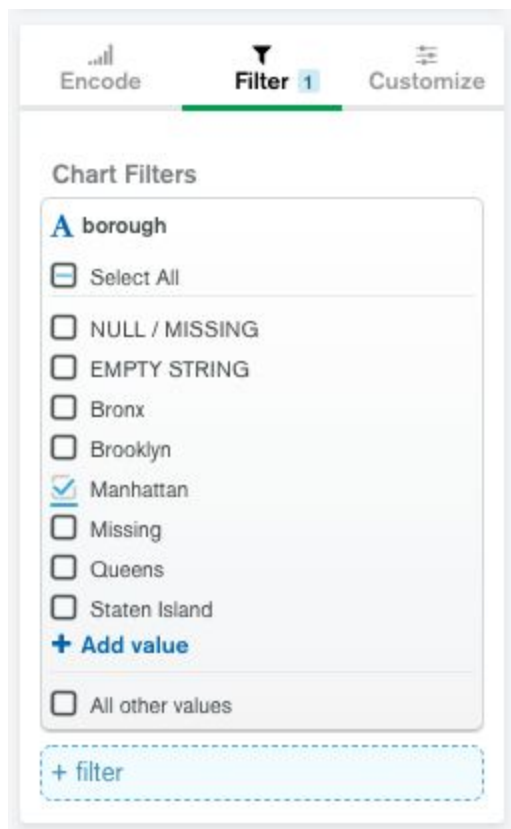
SORT BY

VALUE

☒ LIMIT RESULTS

SHOW: 5

Next, select the **Filter** tab. Drag borough to the filter (+ filter). Choose **Deselect All** and then select **Manhattan**:



Finally, title the chart “**Top 5 Cuisines in Manhattan**”.

Save and Close the chart to return to your dashboard.

Select the ellipses and **Embed Chart**.



Charts can be embedded unauthenticated or with a [verified signature](#). For the purpose of this workshop we'll add a simple unauthenticated chart to our application.

Embed Chart

UnauthenticatedVerified Signature

The data source for this chart does not have unauthenticated external sharing enabled. [Configure external sharing](#).

Enable unauthenticated accessOFF

[View embedding documentation](#)[Close](#)

Click the **Configure external sharing** link, then turn on **Enable external sharing for charts that use this data source** and select **Unauthenticated or Verified Signature**:

External Sharing Options

Enable external sharing for charts that use this data source ON

☐

 Verified Signature only ⓘ

☒

 Unauthenticated or Verified Signature ⓘ

< Go Back

Finally, click **< Go Back** and turn on **Enable unauthenticated access**:

Embed Chart

Unauthenticated

Verified Signature

Enable unauthenticated access

User Specified Filters (optional)

Specify a whitelist of fields that will be used to determine what filters a viewer can apply to this embedded chart.

Enter a field

Iframe

Javascript SDK

Auto refresh

Set a time for this chart to automatically refresh

None

Theme

Select a color theme that matches the colors of the surface that your chart will be rendered on.

Light

EMBED CODE:

```
<iframe style="background: #FFFFFF;border: none;border-radius: 2px;box-shadow: 0 2px 10px 0 rgba(70, 76, 79, .2);" width="640" height="480" src="https://charts.mongodb.com/charts-jj-generalpurpose-pnbde/embed/charts?id=b52d019c-a7ea-4ca5-9780-449d4c3ebac5&theme=light"></iframe>
```

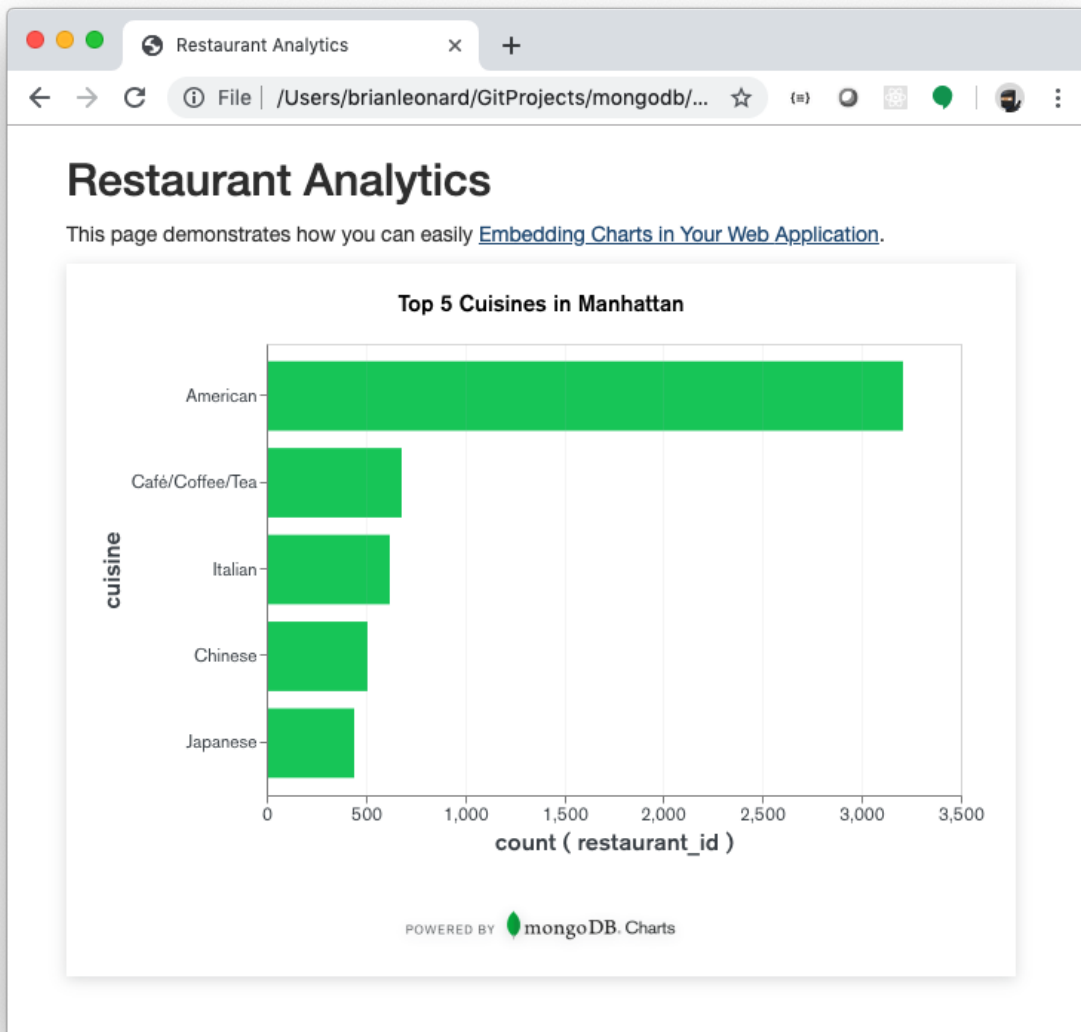
View embedding documentation

Close

Copy the EMBED CODE.

Download this [analytics.html](#) file and open it in your browser. It should work as is because it's currently pointing to a pre-existing chart.

Open the analytics.html file in an editor and familiarize yourself with the contents. Then replace the iframe with the copy of your chart embed code. Save and test the UI. Then host the UI in Stitch and test again.



Congratulations!

You've completed this lab on using MongoDB Charts! You've now seen how easy it is to create and share beautiful visualizations of data stored in MongoDB.