

Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

Overview

In this MongoDB Workshop, participants will get an introduction to MongoDB and the MongoDB Atlas managed cloud database service, along with a high level overview of JSON, Node.js and Javascript.

In the lab, you will create a database with a collection, add some data, and then build a simple Node.js application to connect to your database to read and add data. During the lab, you will learn the create, read, update and delete capabilities using a UI provided by MongoDB.

Don't worry if you don't have experience with Node.js or Javascript, our lab will provide all the code you need to successfully create Node.js RESTful services to read and add data.

What is Mongo DB?

MongoDB is a **document**-oriented NoSQL database. But, what does this really mean?

NoSQL (or originally known as non-SQL or non-relational) databases allow for storage and retrieval that is modeled differently than traditional tabular relationships used in relational databases. There are four types of NoSQL Databases. MongoDB is known as a Document Database.

Data is stored in documents in MongoDB. This shouldn't be confused with documents we store in document management systems. A document in MongoDB is similar to a JSON (JavaScript Object Notation) object and is analogous to a record in a traditional RDBMS (Relational Database Management System).

A **document** is a data structure composed of name (field)/value pairs. The value of a field may contain other documents (embedded documents), arrays or arrays of documents.

Some advantages to storing data as documents are:

- Documents correspond to native data types in many programming languages and most programming languages provide code to create and parse JSON.
- Embedded documents and arrays reduce the need for expensive joins that are needed in RDBMS.

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

- Dynamic Schema - the schema does not need to be defined before you can start to store data, as is required in RDBMS. This allows for development in a more agile fashion.

MongoDB Atlas is a fully-managed cloud database developed by the same people that build MongoDB. Atlas handles all the complexity of deploying, managing, and healing your deployments on the cloud service provider of your choice (AWS, Azure, and GCP).

MongoDB vs. Relational Database Management Systems (RDBMS)

This table shows the relationship of traditional RDBMS terminology with MongoDB^{**}.

RDBMS	MongoDB
Database	Database: Database is a physical container for collections. A single MongoDB server can have multiple databases.
Table	Collection: A group of MongoDB documents that exist within a single database. Collections do not enforce a schema, so documents within a collection can have different fields. However, typically all documents in a collection are of similar or related purpose.
Row	Document: A set of name/value pairs. Documents have dynamic schema and can be easily modified over time. Documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.
Column	Field: The name part of name/value pair. The value of a field can be a standard data type like String, Integer, Datetime. It can also be an embedded document, an array or an array of arrays.

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

Table Join	<p>Embedded Document: A multiple name/value pair structure embedded in a document. An example might be storing a Customer with their Address. In a Relational Database, you would probably have a customer record on the customer table and their address record on an address table. In MongoDB, you could have a Customer document that contains their Address, where Address would be an embedded document. This also allows you to store all the Customer's addresses within a single Customer document.</p> <p>Reference: A reference to another document. For example, if we had a Book document we might want to store the publisher information. However, we would probably have many book documents that have the same publisher. To avoid storing all the publisher information repetitively in all book documents, we could instead store a reference to a specific publisher document with each book document.</p>
Primary Key	<p>Primary Key: MongoDB automatically provides <code>_id</code>, which assures the uniqueness of every document. You can provide <code>_id</code> when storing a document or MongoDB will provide it for you.</p>

** Taken from [Tutorialspoint MongoDB - Overview \(https://www.tutorialspoint.com/mongodb/mongodb_overview.htm\)](https://www.tutorialspoint.com/mongodb/mongodb_overview.htm)

JSON

JSON stands for JavaScript Object Notation. It is a light-weight format that is easy to read and write. It is a language-independent format. Although it is derived from JavaScript, many programming languages have the ability to generate and parse JSON. JSON isn't bloated like XML is with all the opening and closing tags.

JSON is built on two structures:

- A collection of name/value pairs.
- An ordered list of values.

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

In JSON, an object is an unordered set of name/value pairs contained between curly braces. Each name is followed by a colon and then the value. Each name/value pair is separated by a comma. Here is an example:

```
{ item: "journal", qty: 25, status: "A" }
```

An array is an ordered collection of values contained between square braces. Each value is separated by a comma. Here we have added tags as an array to the example above:

```
{ item: "journal", qty: 25, status: "A", tags: [ "blank", "red" ] }
```

Node.js and the rise of JavaScript

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Historically, JavaScript was used for creating client-side scripts that were embedded in a web page's HTML and run by a JavaScript engine in a user's browser. Node.js allows a developer to now write server-side scripts which run and create dynamic web page content before the page is sent to the user's browser.

- JavaScript everywhere.
- Unifies web application development around a single programming language.

Node.js uses **npm** as its default package manager. npm is used to install, share and distribute code. It is used to manage dependencies in code projects. npm is also the world's largest software registry and hosts the largest collection of reusable code. It gives a developer access to hundreds of thousands of building blocks of code (packages).

Now for the fun part...

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

MongoDB Lab

- Part 1: Create a new Database using MongoDB Compass
- Part 2: Insert a Document
- Part 3: Delete a Document
- Part 4: Insert Multiple Documents
- Part 5: Query Documents
- Part 6: Update a Document
- Part 7: Connect to MongoDB Atlas from a simple Node.js application
- Part 8: Add HTML GET and POST Endpoints

This lab will provide a high-level review of some common capabilities in MongoDB. We will start by creating a database with a collection that we can then add documents to. We will introduce you to **MongoDB Compass Community Edition**, which is a free UI provided by MongoDB. It allows you to visualize and interact with your data with full CRUD functionality.

We will touch on each of the CRUD (Create, Read, Update, Delete) operations. After this lab, if you are interested in learning more about performing CRUD operations in MongoDB, you can check out the [MongoDB CRUD Operations](https://docs.mongodb.com/manual/crud/) (<https://docs.mongodb.com/manual/crud/>) page.

We have pulled together this lab based on some of the tutorials on the [MongoDB Getting Started Tutorial](https://docs.mongodb.com/manual/tutorial/getting-started/) (<https://docs.mongodb.com/manual/tutorial/getting-started/>) site.

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



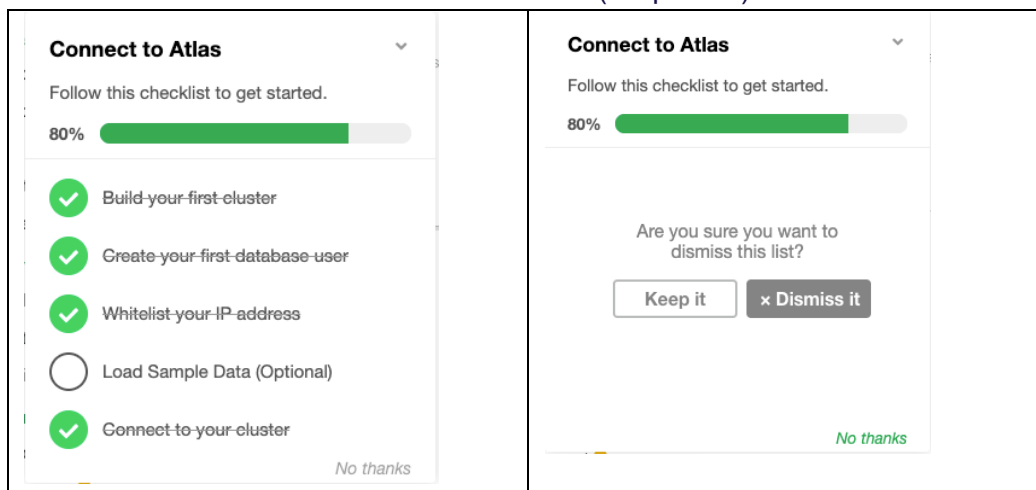
Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

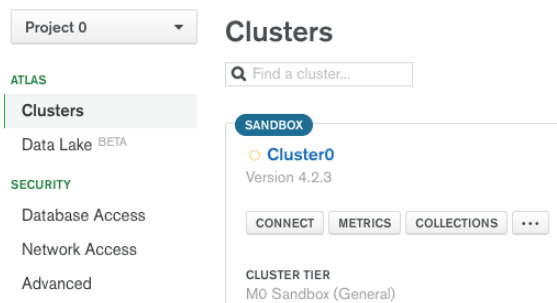
Part 1: Create a new Database using MongoDB Compass

We will use **MongoDB Compass Community** create a Database named **retail** for this lab, along with a collection named **inventory**. We will use this database and collection throughout the lab.

1. Navigate to [MongoDB Atlas](https://cloud.mongodb.com/user#/atlas/login) (<https://cloud.mongodb.com/user#/atlas/login>) and login using the email and password you used to setup your free MongoDB Atlas in Part 1 of the setup instructions.
2. Once logged in, if you see **Connect to Atlas** (1st picture), you can select **No thanks** at the bottom and then select the **Dismiss it** button (2nd picture).



3. Select the **CONNECT** button under **Cluster0** in the **SANDBOX** section:



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

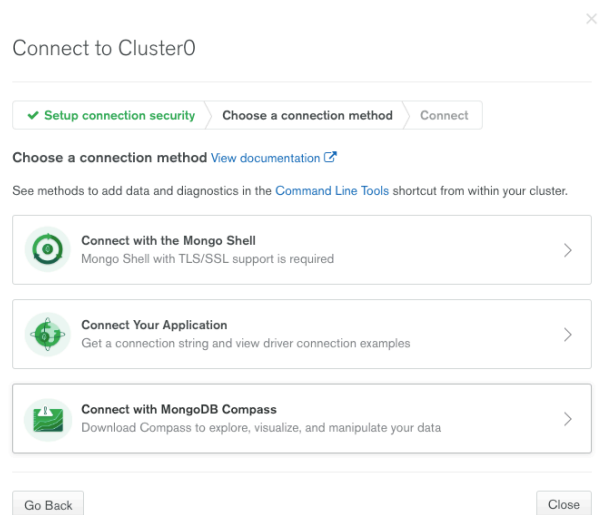
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



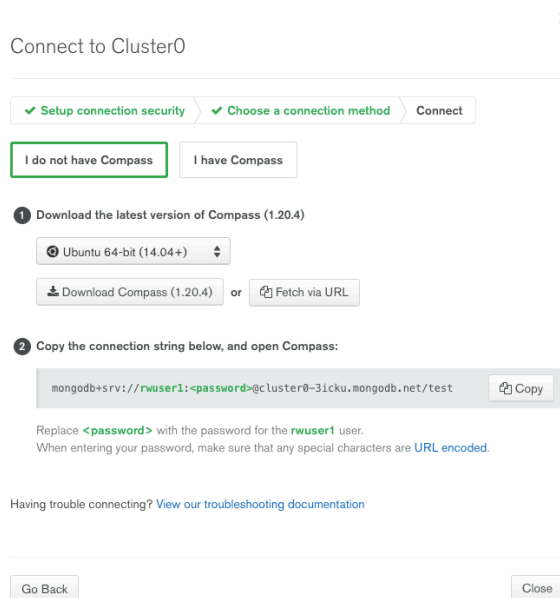
Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

4. Select **Connect with MongoDB Compass**:



5. Select the **Copy** button in section 2 to copy the connection string:



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

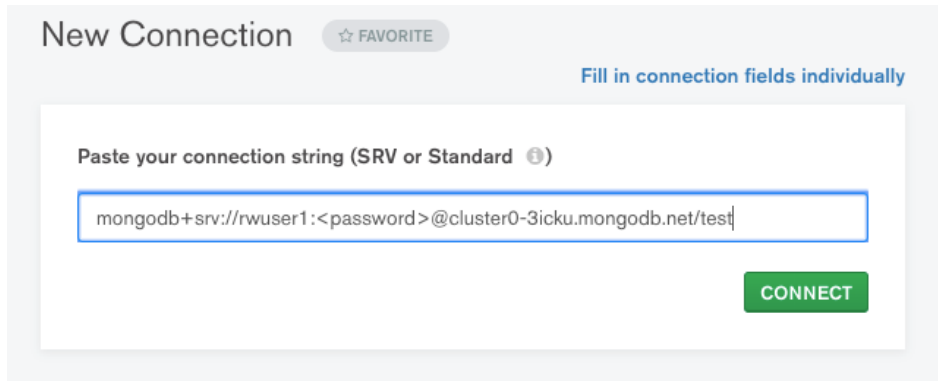
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

- Launch **MongoDB Compass Community** (installed in Part 2 of the setup). Paste the connection string and replace **<password>** with the password you set for the **rwuser1** ID during the setup work. Select the **CONNECT** button:



New Connection ☆ FAVORITE

Fill in connection fields individually

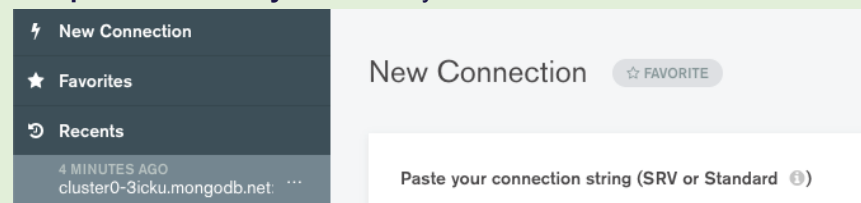
Paste your connection string (SRV or Standard ⓘ)

mongodb+srv://rwuser1:<password>@cluster0-3icku.mongodb.net/test

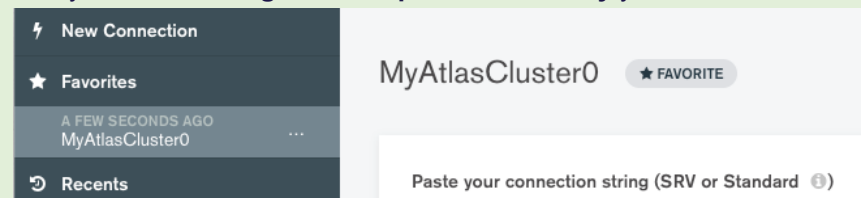
CONNECT

TIP: MongoDB Connection

Once you have connected successfully, you will notice the connection appears under **Recents**. In the future, you can connect without having to log into MongoDB Atlas first. Just launch **MongoDB Compass Community** and select your connection under **Recents**.



When your connection string is populated in the connection string field, you can select the **FAVORITE** button next to **New Connection**, then enter a name to save the connection as. Next time you launch **MongoDB Compass Community** you will see it under **Favorites**:



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

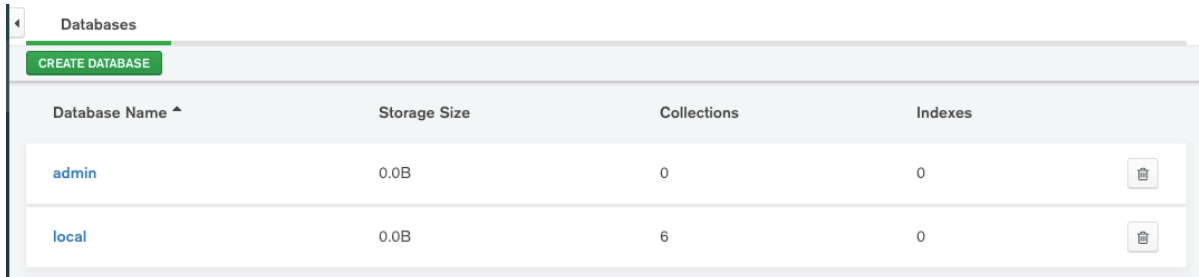
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

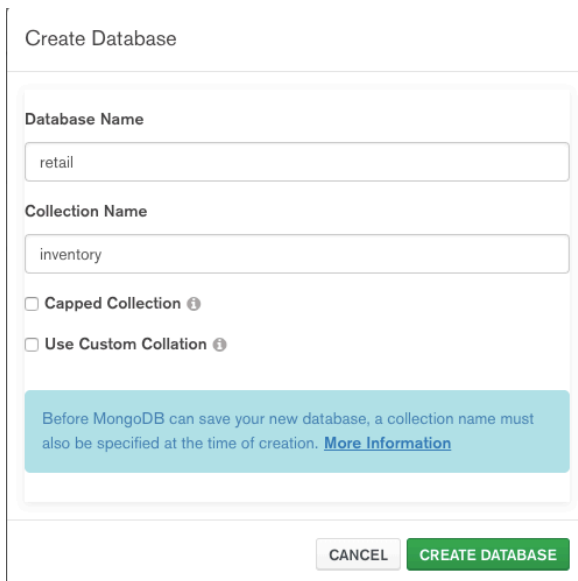
MongoDB Atlas Workshop – Presentation & Lab

7. **MongoDB Atlas** starts with 2 databases; however, we will create a new one for our lab. Select the **CREATE DATABASE** button:



Database Name ^	Storage Size	Collections	Indexes
admin	0.0B	0	0
local	0.0B	6	0

8. When you create a new database, you must also create a collection. Enter **retail** in the **Database Name** field and enter **inventory** in the **Collection Name** field. Select **CREATE DATABASE**:



Create Database

Database Name

retail

Collection Name

inventory

☐ Capped Collection ⓘ

☐ Use Custom Collation ⓘ

Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)

CANCEL CREATE DATABASE

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

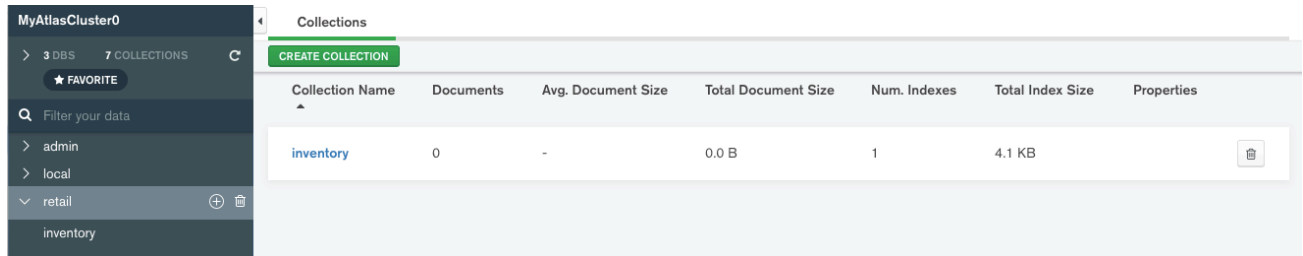
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

9. In the left navigation panel, select the **retail** database. It will expand and you will see the **inventory** collection underneath it. If you wanted to create a new collection, you would do it here by selecting the **CREATE COLLECTION** button. However, for our purposes, we will work with just the one **inventory** collection we created:



Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
inventory	0	-	0.0 B	1	4.1 KB	

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.

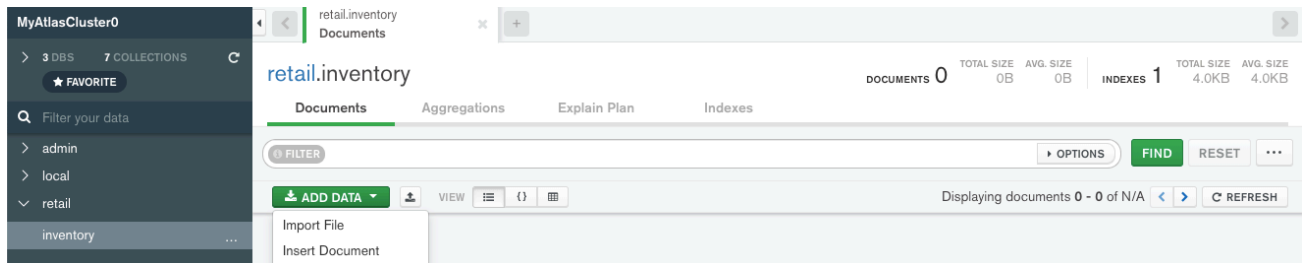


Hack(H)er413 2020

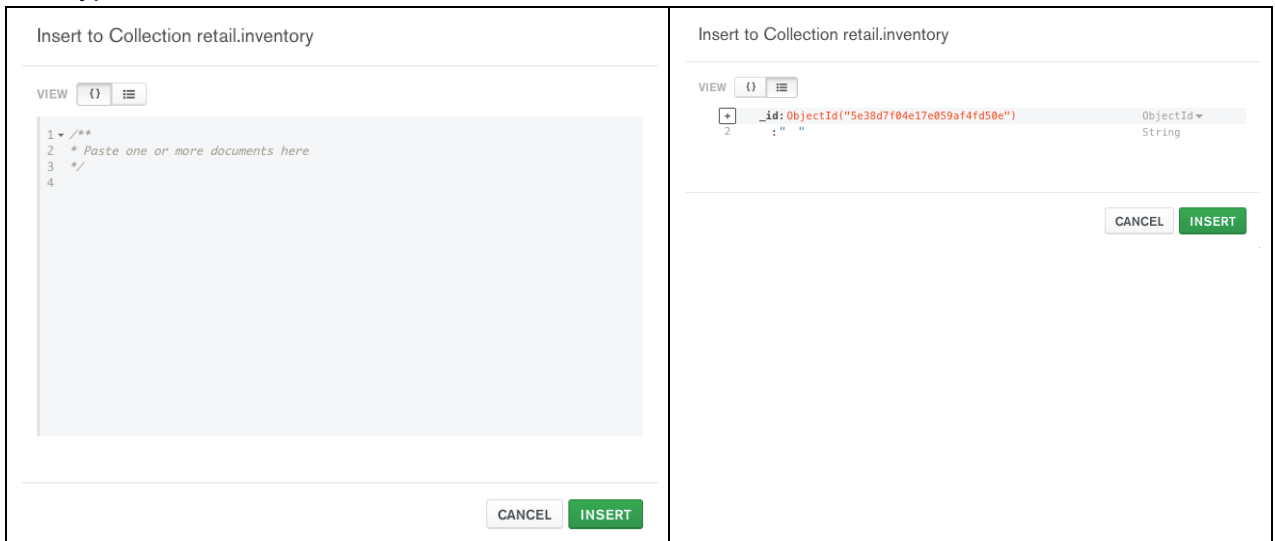
MongoDB Atlas Workshop – Presentation & Lab

Part 2: Insert a Document

1. In the left navigation panel, select the **inventory** collection. Select the **ADD DATA** dropdown and then select **Insert Document**:



2. Notice next to **VIEW** that the **{ }** button is selected (1st picture). This allows for data in JSON format to be pasted. You can also select the other button (2nd picture), which is the **List** button, and it will provide a wizard for adding fields (clicking the plus sign) and selecting data types:



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.




Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

3. Since it is much faster to add data using JSON, click the `{ }` button and delete the code that was added when we selected to the **List** button (it should look like this again):

Insert to Collection retail.inventory

VIEW `{ }` 

```
1 /**
2  * Paste one or more documents here
3  */
4
```

CANCEL INSERT

4. Let's take a look at what attributes our document will contain:
 - **item** : an attribute of type String
 - **qty** : an attribute of type Integer
 - **status** : an attribute of type String
 - **size** : an Object that has 3 attributes:
 - **h** and **w** : both are attributes of type Integer
 - **uom** : an attribute of type String
 - **tags** : an Array which contains a single value (future documents will contain additional values in the tags array)

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

5. Paste the following code into the **Insert to Collection** window and select the **INSERT** button.

```
{
  "item": "canvas", "qty": 100, "status": "stock",
  "size": { "h": 28, "w": 35.5, "uom": "cm"},
  "tags": [ "cotton" ]
}
```

Insert to Collection retail.inventory

VIEW

```
1- /**
2  * Paste one or more documents here
3  */
4- {
5   "item": "canvas", "qty": 100, "status": "stock",
6   "size": { "h": 28, "w": 35.5, "uom": "cm"},
7   "tags": [ "cotton" ]
8 }
9 |
```

6. Click the little arrows next to **size** and **tags** to expand them. Your object should look like this. Notice an **_id** attribute was added. This acts as a primary key. **MongoDB Compass** automatically adds this attribute, if it is not included, along with a generated value. You can set your own value, but it must remain unique.

```
item: "canvas"
qty: 100
status: "stock"
size: Object
  h: 28
  w: 35.5
  uom: "cm"
tags: Array
  0: "cotton"
_id: ObjectId("5e38de353db11359af3407a5")
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.

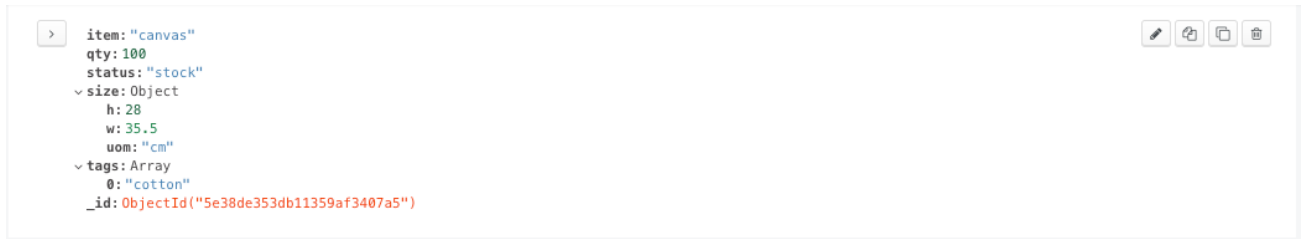


Hack(H)er413 2020

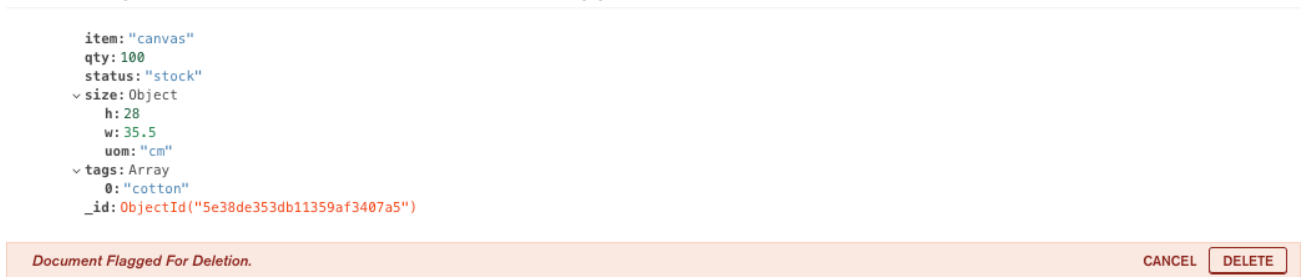
MongoDB Atlas Workshop – Presentation & Lab

Part 3: Delete a Document

1. To delete a document, place your mouse inside the box where the document is, and you will see 4 icons appear on the right-hand side. These icons allow you to **edit**, **copy**, **clone** and **delete** the document. Go ahead and select the **delete** button (last one on the right that looks like a trash can):



2. A message appears that the document is flagged for deletion. Select the **DELETE** button:



3. The document you added is now gone and there is no data in the collection. But, not to worry, in the next part we are going to add several documents.

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.

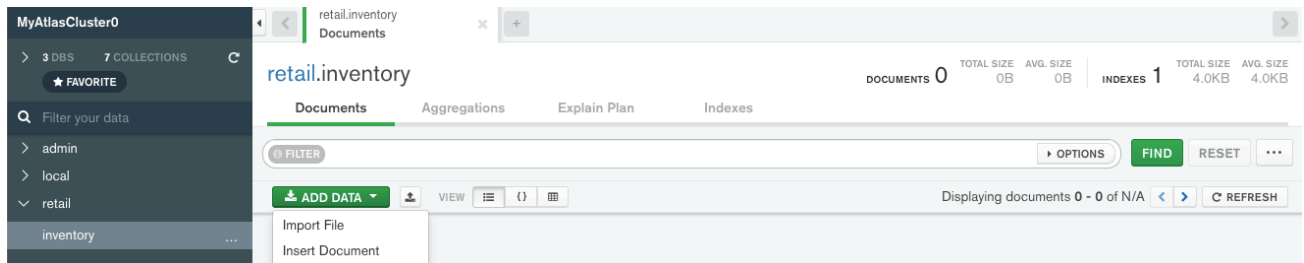


Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

Part 4: Insert Multiple Documents

1. In the left navigation panel, the **inventory** collection should still be selected. If not, go ahead and select it. Select the **ADD DATA** dropdown and then select **Insert Document**:



2. This time we will add 6 documents to our **inventory** collection. While they have a similar schema, there are differences:
 - 4th item (**paper**) doesn't have a **status** attribute
 - 6th item (**postcard**) doesn't have a **size** object
 - Several items (**journal**, **notebook**, **paper**, **planner**) have multiple values in the **tags** array
 - Paste the following code into the **Insert to Collection** window and select the **INSERT** button.

```
[
  { "item": "canvas", "qty": 100, "status": "stock",
    "size": { "h": 28, "w": 35.5, "uom": "cm"}, "tags": [ "cotton" ] },
  { "item": "journal", "qty": 25, "status": "backorder",
    "size": { "h": 14, "w": 21, "uom": "cm"}, "tags": [ "blank", "red" ] },
  { "item": "notebook", "qty": 50, "status": "stock",
    "size": { "h": 8.5, "w": 11, "uom": "in" }, "tags": [ "red", "blank" ] },
  { "item": "paper", "qty": 100,
    "size": { "h": 8.5, "w": 11, "uom": "in" }, "tags": [ "red", "blank", "plain" ] },
  { "item": "planner", "qty": 75, "status": "stock",
    "size": { "h": 22.85, "w": 30, "uom": "cm" }, "tags": [ "blank", "red" ] },
  { "item": "postcard", "qty": 45, "status": "backorder", "tags": [ "blue" ] }
]
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

Insert to Collection retail.inventory

VIEW

```
1 /**
2  * Paste one or more documents here
3  */
4 [
5 { "item": "canvas", "qty": 100, "status": "stock",
6   "size": { "h": 28, "w": 35.5, "uom": "cm"}, "tags": [ "cotton" ] },
7 { "item": "journal", "qty": 25, "status": "backorder",
8   "size": { "h": 14, "w": 21, "uom": "cm"}, "tags": [ "blank", "red" ] },
9 { "item": "notebook", "qty": 50, "status": "stock",
10  "size": { "h": 8.5, "w": 11, "uom": "in" }, "tags": [ "red", "blank" ] },
11 { "item": "paper", "qty": 100,
12   "size": { "h": 8.5, "w": 11, "uom": "in" }, "tags": [ "red", "blank", "plai
13 { "item": "planner", "qty": 75, "status": "stock",
14   "size": { "h": 22.85, "w": 30, "uom": "cm" }, "tags": [ "blank", "red" ] },
15 { "item": "postcard", "qty": 45, "status": "backorder", "tags": [ "blue" ] }
16 ]
17
```

CANCEL

INSERT

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



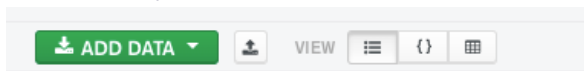
Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

3. Six documents have been added to the **inventory** collection. You can click the little arrows next to **size** and **tags** to expand them if you want to see that the additional data they contain:

<pre>_id: ObjectId("5e38e9f53db11359af3407a8") item: "canvas" qty: 100 status: "stock" > size: Object > tags: Array</pre>
<pre>_id: ObjectId("5e38e9f53db11359af3407a9") item: "journal" qty: 25 status: "backorder" > size: Object > tags: Array</pre>
<pre>_id: ObjectId("5e38e9f53db11359af3407aa") item: "notebook" qty: 50 status: "stock" > size: Object > tags: Array</pre>
<pre>_id: ObjectId("5e38e9f53db11359af3407ab") item: "paper" qty: 100 > size: Object > tags: Array</pre>
<pre>_id: ObjectId("5e38e9f53db11359af3407ac") item: "planner" qty: 75 status: "stock" > size: Object > tags: Array</pre>
<pre>_id: ObjectId("5e38e9f53db11359af3407ad") item: "postcard" qty: 45 status: "backorder" > tags: Array</pre>

4. Next to the **ADD DATA** button you will notice 3 **VIEW** buttons. The first one is displaying them in list format. The second button will display the data in JSON format. The third button will display the data in table format. You can play with each of these and decide which view you like best.



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

Part 5: Query Documents

1. In the left navigation panel, the **inventory** collection should still be selected. If not, go ahead and select it. You will notice that all your data shows below the **ADD DATA** button:

retail.inventory

DOCUMENTS 6 TOTAL SIZE 793B AVG. SIZE 132B INDEXES 1 TOTAL SIZE 24.0KB AVG. SIZE 24.0KB

Documents Aggregations Explain Plan Indexes

FILTER OPTIONS FIND RESET ...

ADD DATA VIEW {}

Displaying documents 1 - 6 of 6 REFRESH

```
> { "_id": ObjectId("5e38e9f53db11359af3407a8"),
  "item": "canvas",
  "qty": 100,
  "status": "stock",
  "size": Object,
  "tags": Array
}
```

```
{ "_id": ObjectId("5e38e9f53db11359af3407a9"),
  "item": "journal",
  "qty": 25,
  "status": "backorder",
  "size": Object,
  "tags": Array
}
```

```
{ "_id": ObjectId("5e38e9f53db11359af3407aa"),
  "item": "notebook",
  "qty": 50,
  "status": "stock",
  "size": Object,
  "tags": Array
}
```

```
{ "_id": ObjectId("5e38e9f53db11359af3407ab"),
  "item": "paper",
  "qty": 100,
  "size": Object,
  "tags": Array
}
```

```
{ "_id": ObjectId("5e38e9f53db11359af3407ac"),
  "item": "planner",
  "qty": 75,
  "status": "stock"
}
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

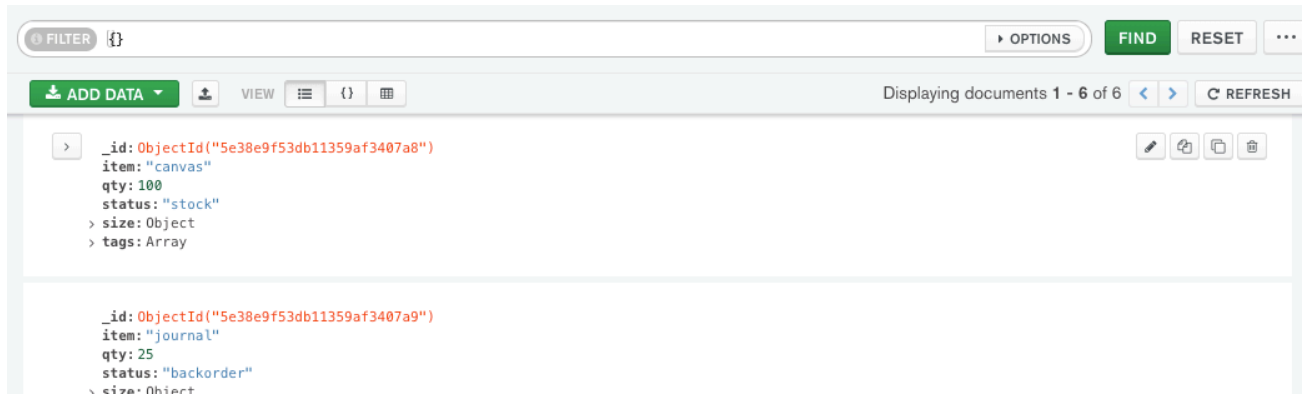
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

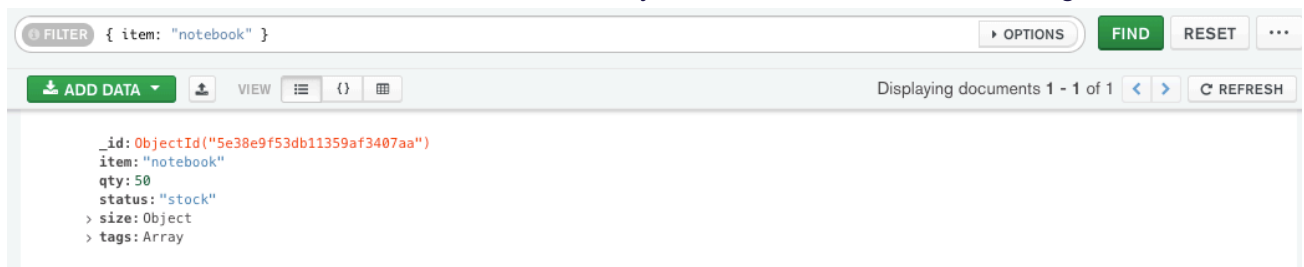
- The command bar (where you see **FILTER**) is where you can enter commands to query your data. Enter and open and closed curly brace like this: `{ }` and select the green **FIND** button and notice that this also displays all your data:



- Next, let's see if we can find the specific document where the **item** field has a value of **notebook**. To do this, type the following in the command bar:

{ item: "notebook" }

Then select the **FIND** button. You should not only the document with the matching item:



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

4. If you recall, our **inventory** documents have an embedded document for **size**. Let's search for documents matching a field that is part of the **size** embedded document. We will use **uom**. The following command is going to look for **size.uom** with a value of **in**. Enter the following in the command bar and select the **FIND** button. Expand **size** on each document to validate:

```
{ "size.uom": "in" }
```

The screenshot shows the MongoDB Atlas web interface. At the top, there's a filter bar with the query `{ "size.uom": "in" }` and buttons for **ADD DATA**, **VIEW**, and **FIND**. Below the filter bar, there's a table of documents. The first document is a **notebook** with a **size** of 50 and **uom** of "in". The second document is a **paper** with a **size** of 100 and **uom** of "in".

Because **size** is an embedded document, we need to pass in "**size.uom**" so that it knows to look for the **uom** field within the **size** embedded document. If we just passed in **uom**, it would not find anything since **uom** is not a field in **inventory**. You can give it a try if you want.

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.

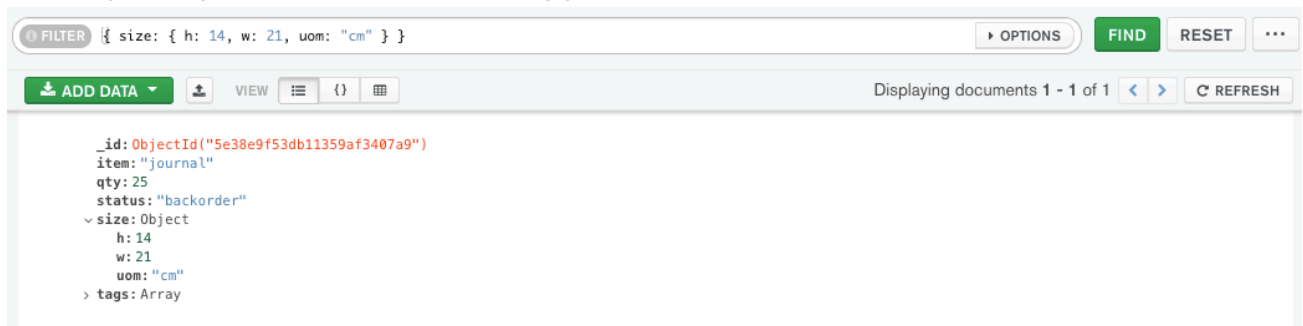


Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

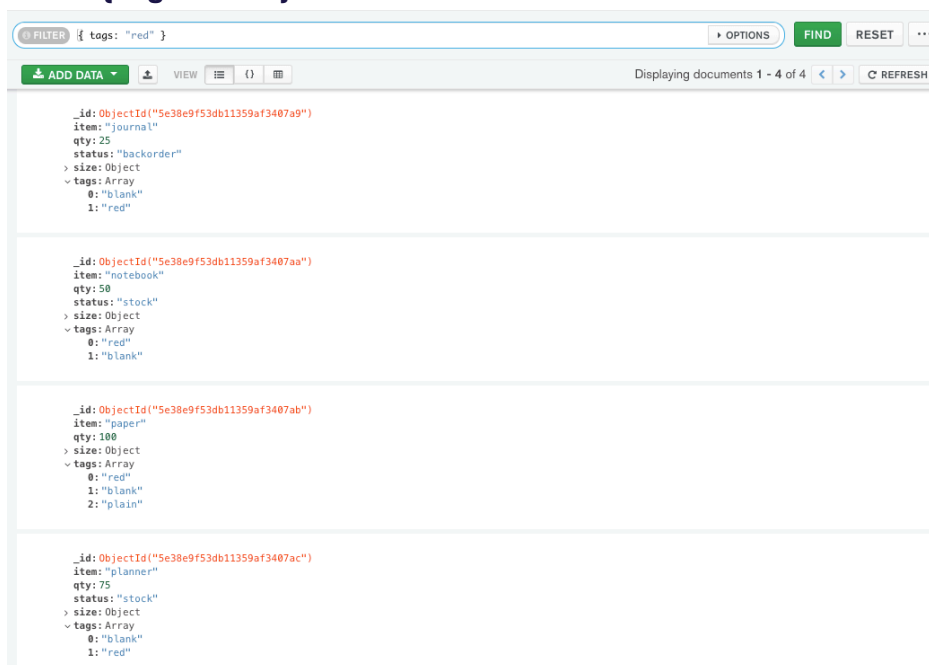
5. Another query we may wish to perform is to find a document that has a matching embedded document (i.e. all fields in the embedded document match the query). Enter the following in the command bar and select the **FIND** button. Expand **size** to validate:

{ size: { h: 14, w: 21, uom: "cm" } }



6. Our **inventory** document has a **tags** field which was defined as an **Array**. Let's search for a document that has a matching value in the **tags** array. Enter the following in the command bar and select the **FIND** button. Expand **tags** on each document to validate:

{ tags: "red" }



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.

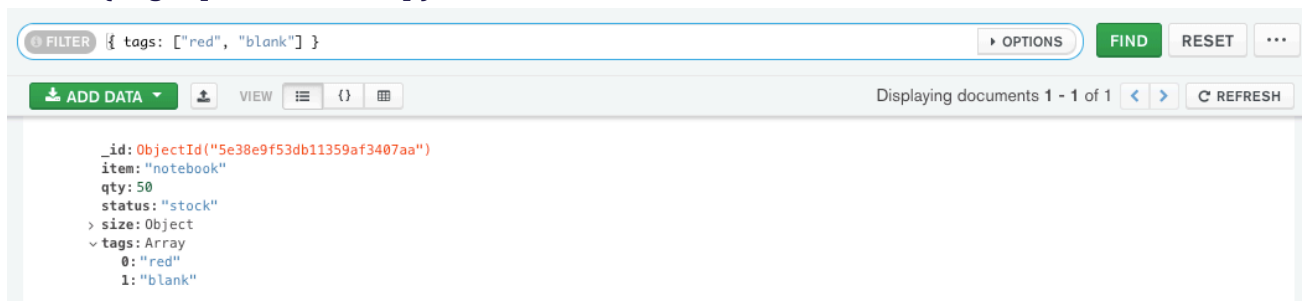


Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

7. This time let's run a query to match the values in the array exactly. Enter the following in the command bar and select the **FIND** button. This returns only 1 document. Expand **tags** to validate the first value equals **red** and the second value equals **blank**. If you noticed above, there were 3 documents that had **tags** with values of **red** and **blank**. However, they were in a different order or had more than those 2 tags.

{ tags: ["red", "blank"] }



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

Part 6: Update a Document

1. You should still have a document displayed from the last step above. To update, or edit, a document, place your mouse inside the box where the document is, and you will see 4 icons appear on the right-hand side. These icons allow you to **edit**, **copy**, **clone** and **delete** the document. We used the **delete** button in Part 3 above. Go ahead and select the **edit** button (first one that looks like a pencil).
2. The document will open in the **list** format and select the little arrows next to **size** and **tags** to expand them:

The screenshot shows the MongoDB Atlas interface with a document in list format. The document is expanded to show the 'size' and 'tags' fields. The 'size' field is an object with 'h' (8.5), 'w' (11), and 'uom' ('in'). The 'tags' field is an array with two elements: 'red' and 'blank'. The 'blank' tag is highlighted with a blue box. The interface includes a top bar with 'ADD DATA', 'VIEW', and 'REFRESH' buttons, and a bottom bar with 'CANCEL' and 'UPDATE' buttons.

```
1  _id: ObjectId("5e38e9f53db11359af3407aa")
2  item: "notebook "
3  qty: 50
4  status: "stock "
5  size: Object
6    h: 8.5
7    w: 11
8    uom: "in "
9  tags: Array
10    0: "red "
11    1: "blank "
```

3. Click on **“blank”** and select **Add Array Element After 1**:

The screenshot shows the MongoDB Atlas interface with the 'tags' array expanded. The 'blank' tag is highlighted with a blue box. A context menu is open over the 'blank' tag, showing the 'Add Array Element After 1' button. The button is highlighted with a blue box.

```
9  tags: Array
10    0: "red "
11    1: "blank "
```

4. Enter **blue** in the field and then select the **UPDATE** button:

The screenshot shows the MongoDB Atlas interface with the 'tags' array expanded. The 'blue' tag has been added to the array, and the 'blank' tag has been removed. The 'blue' tag is highlighted with a blue box. The interface includes a top bar with 'ADD DATA', 'VIEW', and 'REFRESH' buttons, and a bottom bar with 'CANCEL' and 'UPDATE' buttons.

```
9  tags: Array
10    0: "red "
11    1: "blue "
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

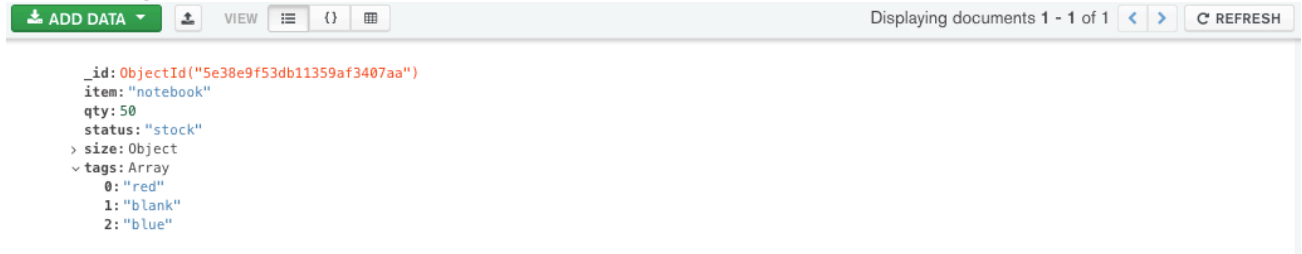
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

5. Expand **tags** and validate it now has **blue** as a value:



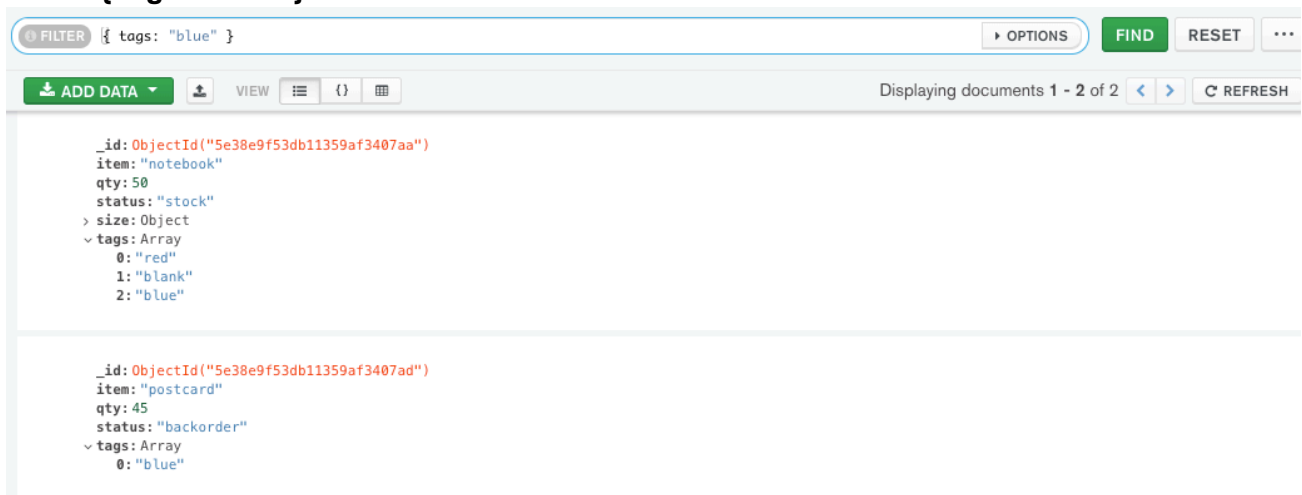
The screenshot shows the MongoDB Atlas document editor. At the top, there's a toolbar with 'ADD DATA', 'VIEW', and other icons. The status bar indicates 'Displaying documents 1 - 1 of 1'. The document content is as follows:

```
{
  "_id": ObjectId("5e38e9f53db11359af3407aa"),
  "item": "notebook",
  "qty": 50,
  "status": "stock",
  "size": Object,
  "tags": Array
}
```

The 'tags' array is expanded, showing:

```
0: "red"
1: "blank"
2: "blue"
```

6. Enter the following in the command bar and select the **FIND** button. You should get two results returned, which includes the one you just modified above. Expand **tags** to validate:
- { tags: "blue" }**



The screenshot shows the MongoDB Atlas search results. The filter bar at the top contains '{ tags: "blue" }'. The status bar indicates 'Displaying documents 1 - 2 of 2'. Two documents are returned:

```
{
  "_id": ObjectId("5e38e9f53db11359af3407aa"),
  "item": "notebook",
  "qty": 50,
  "status": "stock",
  "size": Object,
  "tags": Array
}
```

Expanded tags:

```
0: "red"
1: "blank"
2: "blue"
```



```
{
  "_id": ObjectId("5e38e9f53db11359af3407ad"),
  "item": "postcard",
  "qty": 45,
  "status": "backorder",
  "tags": Array
}
```

Expanded tags:

```
0: "blue"
```

7. After the lab, feel free to play with the other 2 buttons to **copy** and **clone** a document. You can also learn more about these [CRUD operations in MongoDB](https://docs.mongodb.com/manual/crud/) (<https://docs.mongodb.com/manual/crud/>).

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

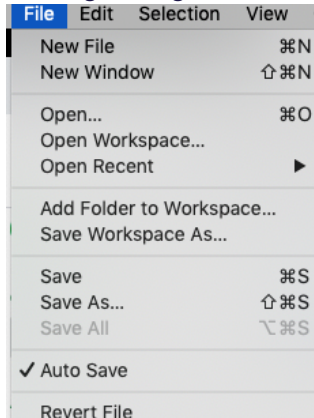
Part 7: Connect to MongoDB Atlas from a simple Node.js application

Most modern web applications need to be able to handle client requests via HTTP to consume and manipulate data. It is normal convention to pass JSON data between these requests. So, it makes sense to use a NoSQL document database where JSON is a common storage format. This eliminates the need to marshal data to new formats in every request.

Here we will create a RESTful API using [Node.js](https://nodejs.org/en/) (<https://nodejs.org/en/>) and [Express](https://expressjs.com/) (<https://expressjs.com/>) to demonstrate connecting to our MongoDB database and retrieving and adding data.

We will use [Visual Studio Code](https://code.visualstudio.com/) (<https://code.visualstudio.com/>) also referred to as VSC as our source-code editor / developer environment. You should have already installed VSC as part of the setup instructions.

1. Launch **Visual Studio Code** (VSC), which you installed as part of the setup instructions.
2. From the File Menu in VSC, select **File / Auto Save**. This will automatically save as you are making changes:



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

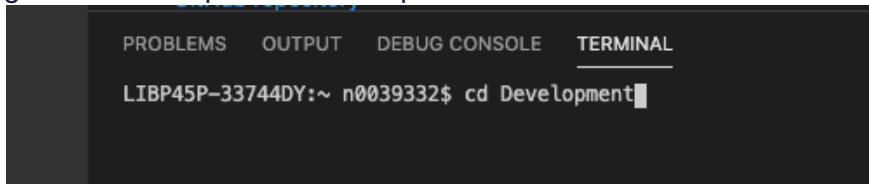
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

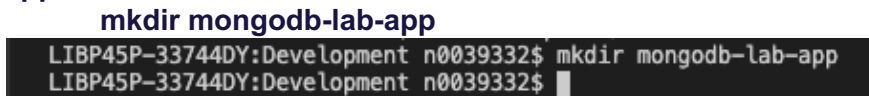
MongoDB Atlas Workshop – Presentation & Lab

- From the File Menu in VSC, select **View / Terminal**. This will open a command terminal in the bottom section of VSC. It should also default you to your user directory. In the terminal, type **cd Development** and then hit the **return/enter** key. This will change directory to the **Development** folder you created during the setup instructions. If you don't have this folder, go back to that part of the setup.



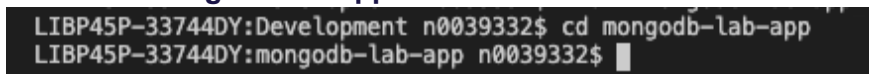
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
LIBP45P-33744DY:~ n0039332$ cd Development
```

- In the terminal, use the **mkdir** command to create a new directory named **mongodb-lab-app**.



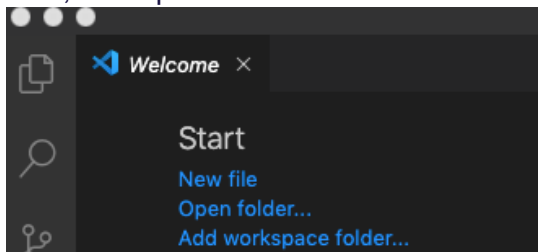
```
LIBP45P-33744DY:Development n0039332$ mkdir mongodb-lab-app
LIBP45P-33744DY:Development n0039332$
```

- Change directory (**cd**) into the **mongodb-lab-app** directory:

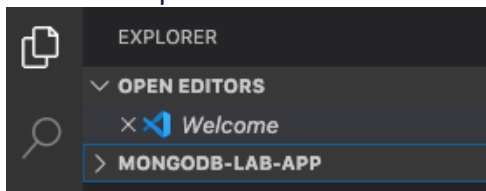


```
LIBP45P-33744DY:Development n0039332$ cd mongodb-lab-app
LIBP45P-33744DY:mongodb-lab-app n0039332$
```

- Now, let's open this folder in VSC. Under **Start** select the **Open folder...** link:



- Navigate to the location of your **Development** folder and select the **mongodb-lab-app** folder and open it. You will see **MONGODB-LAB-APP** in the left Explorer pane:



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

8. You may need to open the **Terminal** window again (follow Step 2 above). In the terminal window, type the following command to initialize our project:

npm init

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

LIBP45P-33744DY:mongodb-lab-app n0039332$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (mongodb-lab-app) █
```

9. Hit your **return/enter** key to accept the default package name:

```
Press ^C at any time to quit.
package name: (mongodb-lab-app)
version: (1.0.0) █
```

10. Hit your **return/enter** key again to accept the default version (1.0.0):

```
package name: (mongodb-lab-app)
version: (1.0.0)
description: █
```

11. Enter a description and hit your **return/enter** key:

```
version: (1.0.0)
description: App to connect to MongoDB
entry point: (index.js) █
```

12. Enter **server.js** as the entry point and hit the **return/enter** key:

```
entry point: (index.js) server.js
test command: █
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

13. You can just hit the **return/enter** key to bypass **test command**, **git repository**, and **keywords**. When you get to **author** enter your name and hit the **return/enter** key:

```
entry point: (index.js) server.js
test command:
git repository:
keywords:
author: Mary Michaud
license: (ISC)
```

14. Hit the **return/enter** key to bypass **license: (ISC)** and then type **yes** and hit the **return/enter** key:

```
About to write to /Users/n0039332/Development/mongodb-lab-app/package.json:
```

```
{
  "name": "mongodb-lab-app",
  "version": "1.0.0",
  "description": "App to connect to MongoDB",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Mary Michaud",
  "license": "ISC"
}
```

```
Is this OK? (yes) yes
```

```
LIBP45P-33744DY:mongodb-lab-app n0039332$
```

15. Next we are going to install some dependencies. In the **Terminal** window, type the following command followed by the **return/enter** key:

npm install express --save

```
LIBP45P-33744DY:mongodb-lab-app n0039332$ npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN mongodb-lab-app@1.0.0 No repository field.
```

```
+ express@4.17.1
added 50 packages from 37 contributors and audited 126 packages in 1.677s
found 0 vulnerabilities
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

16. In the terminal window, install these additional two dependencies, one at a time by typing the command and then hitting the **return/enter** key:

npm install body-parser --save

```
LIBP45P-33744DY:mongodb-lab-app n0039332$ npm install body-parser --save
npm WARN mongodb-lab-app@1.0.0 No repository field.

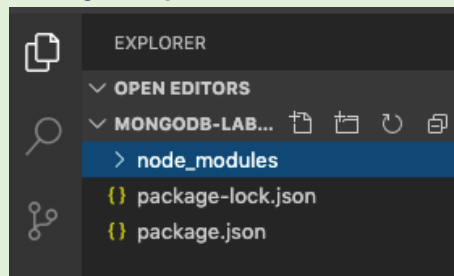
+ body-parser@1.19.0
updated 1 package and audited 158 packages in 0.781s
found 0 vulnerabilities
```

npm install mongodb --save

```
LIBP45P-33744DY:mongodb-lab-app n0039332$ npm install mongodb --save
npm WARN mongodb-lab-app@1.0.0 No repository field.

+ mongodb@3.5.2
added 16 packages from 10 contributors and audited 179 packages in 1.289s
found 0 vulnerabilities
```

The **express** package will be our framework. The **body-parser** package will allow us to handle request bodies with **Express**. Finally, the **mongodb** package is our database driver. Notice after running the **npm install** commands, we now have a **node_modules** folder in our project:



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

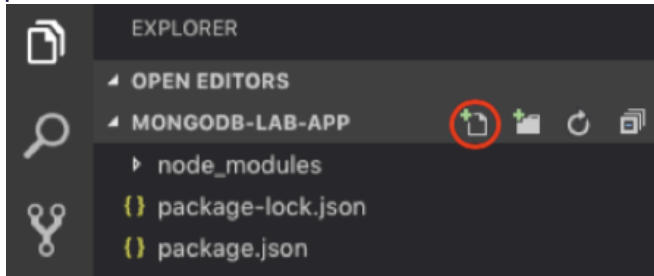
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

17. Now that we have all our dependencies installed, let's create a **server.js** file, which will be the code we run to startup our server. Hover over **MONGODB-LAB-APP** in the left Explorer pane and select the **New File** icon. Name the new file **server.js**:



18. Copy the following code below and paste into the **server.js** file:

```
server.js
const Express = require("express");
const BodyParser = require("body-parser");
const MongoClient = require("mongodb").MongoClient;
const ObjectId = require("mongodb").ObjectId;

const app = Express();
app.set('port', (process.env.PORT || 3000));
app.use(BodyParser.json());
app.use(BodyParser.urlencoded({ extended: true }));

const CONNECTION_URL = "copyConnectionStringHere";
//Add additional variables here

//Modify to connect to MongoDB Atlas
app.listen(app.get('port'), () =>
  console.log(`Listening on ${app.get('port')}`));

//Add GET endpoint here

//Add GET by id endpoint here

//Add POST endpoint here

module.exports = app;
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

```
JS server.js x
JS server.js > ...
1  const Express = require("express");
2  const bodyParser = require("body-parser");
3  const MongoClient = require("mongodb").MongoClient;
4  const ObjectId = require("mongodb").ObjectId;
5
6  const app = Express();
7  app.set('port', (process.env.PORT || 3000));
8  app.use(bodyParser.json());
9  app.use(bodyParser.urlencoded({ extended: true }));
10
11 const CONNECTION_URL = "copyConnectionStringHere";
12 //Add additional variables here
13
14
15 //Modify to connect to MongoDB Atlas
16 app.listen(app.get('port'), () =>
17   console.log(`Listening on ${app.get('port')}`));
18
19 //Add GET endpoint here
20
21
22 //Add GET by id endpoint here
23
24
25 //Add POST endpoint here
26
27
28 module.exports = app;
```

This code starts out by importing each of our dependencies. The **MongoClient** will allow us to establish a connection to our mongoDB database. The **ObjectId** will allow us to work with document ids, which we will do in a subsequent step.

Next we initialize **Express** framework and configure the **BodyParser** package. Finally, we configure our application to listen on port 3000.

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

19. **TEST:** Start the server by typing **npm start** in the terminal window. It should start without any errors and show it is listening on port 3000. To stop the server, you can use the **control/c** keys on your keyboard:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
LIBP45P-33744DY:mongodb-lab-app n0039332$ npm start

> mongodb-lab-app@1.0.0 start /Users/n0039332/Development/mongodb-lab-app
> node server.js

Listening on 3000
```

20. If you are still logged into **MongoDB Atlas**, go back to that page (or log in again – see Part 1, Step 1 above) and select **CONNECT**. This time select **Connect Your Application**:

×

Connect to Cluster0


✓ Setup connection security

Choose a connection method

Connect


Choose a connection method [View documentation](#)

See methods to add data and diagnostics in the [Command Line Tools](#) shortcut from within your cluster.




Connect with the Mongo Shell
Mongo Shell with TLS/SSL support is required

>



Connect Your Application
Get a connection string and view driver connection examples

>



Connect with MongoDB Compass
Download Compass to explore, visualize, and manipulate your data

>

Go Back

Close

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020


MongoDB Atlas Workshop – Presentation & Lab

21. **DRIVER** should default to **Node.js**. Select the **Copy** button to copy the connection string:

1 Choose your driver version

DRIVER	VERSION
Node.js	3.0 or later

2 Add your connection string into your application code

Connection String Only	Full Driver Example
<code>mongodb+srv://rwuser1:<password>@cluster0-3icku.mongodb.net/test?</code>	
 Copy	

Replace **<password>** with the password for the **rwuser1** user.
When entering your password, make sure that any special characters are [URL encoded](#).

22. In the **server.js** file, locate the **CONNECTION_URL** variable and replace the **copyConnectionStringHere** text with the connection string you just copied above. I put mine indented on the line below to make it more readable.

In the connection string you just pasted, replace **<password> with the password you created for the **rwuser1** id. Note you don't want to keep the **<>** surrounding the password and you must keep the quotes around the connection string:

```
11 const CONNECTION_URL =  
12     "mongodb+srv://rwuser1:<password>@cluster0-3icku.mongodb.net/test?retryWrites=true&w=majority";  
13 //Add additional variables here
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

23. Copy the additional variables from the code block below and paste them after the **//Add additional variables here** comment in the **server.js** file:

server.js

```
const DATABASE_NAME = "retail";
const COLLECTION = "inventory";
const client = new MongoClient(CONNECTION_URL, {
  useNewUrlParser: true,
  useUnifiedTopology: true } );
var database, collection;
```

```
13 //Add additional variables here
14 const DATABASE_NAME = "retail";
15 const COLLECTION = "inventory";
16 const client = new MongoClient(CONNECTION_URL, {
17   useNewUrlParser: true,
18   useUnifiedTopology: true } );
19 var database, collection;
```

24. In the **server.js** file, locate the **//Modify to connect to MongoDB Atlas** comment and replace the next 2 lines of code with the code from the code block below:

server.js

```
app.listen(app.get('port'), () => {
  console.log(`Listening on ${app.get('port')}`);
  client.connect(err => {
    if (err) throw err;
    database = client.db(DATABASE_NAME);
    collection = database.collection(COLLECTION);
    console.log("Connected to " + DATABASE_NAME);
  });
});
```

```
21 //Modify to connect to MongoDB Atlas
22 app.listen(app.get('port'), () => {
23   console.log(`Listening on ${app.get('port')}`);
24   client.connect(err => {
25     if (err) throw err;
26     database = client.db(DATABASE_NAME);
27     collection = database.collection(COLLECTION);
28     console.log("Connected to " + DATABASE_NAME);
29   });
30 });
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

25. **TEST:** Stop (**control/c**) and restart (**npm start**) your server to make sure it starts up without any errors and is listening on port 3000. You should also see a new message that it is **Connected to retail**:

```
LIBP45P-33744DY:mongodb-lab-app n0039332$ npm start  
  
> mongodb-lab-app@1.0.0 start /Users/n0039332/Development/mongodb-lab-app  
> node server.js  
  
Listening on 3000  
Connected to retail
```

Part 8: Add HTML GET and POST Endpoints

1. We will start by adding a GET endpoint that we can call on our application to interact with our MongoDB database. This endpoint will return all items in our **inventory** collection. Copy the code in the code block below and paste it under the **//Add GET endpoint here** comment in the **server.js** file:

```
server.js  
app.get("/inventory", (request, response) => {  
  collection.find({}).toArray((error, result) => {  
    if(error) {  
      return response.status(500).send(error);  
    }  
    response.send(result);  
  });  
});
```

```
32 //Add GET endpoint here  
33 app.get("/inventory", (request, response) => {  
34   collection.find({}).toArray((error, result) => {  
35     if(error) {  
36       return response.status(500).send(error);  
37     }  
38     response.send(result);  
39   });  
40 });
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

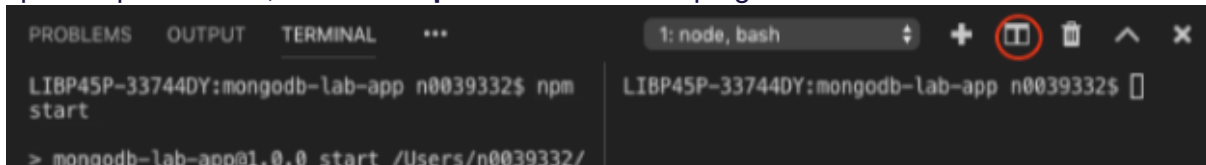
Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

2. **TEST:** Stop and Start your server and make sure it is listening and connected.
We will use curl to send a GET request to our server to see if we get back a list of all the items in our inventory collection in our MongoDB Atlas cluster. To do this, let's open another terminal in VSC. We can do a split terminal so we have access to start and stop our server in one terminal and the other terminal can be used to run our curl commands. To open a split terminal, select the **split icon** near the top right side of the terminal window:



cURL (<https://curl.haxx.se/>) is a computer software project providing a library and command-line tool for transferring data using various protocols. We are using it to make HTTP requests to our server.

3. **TEST:** Enter the following command from the code block below into the right-side terminal window. You should get back all the items in the inventory collection in JSON format. If you look closely you will see the **canvas, journal, notebook, paper, planner** and **postcard**.

curl

```
curl -X GET http://localhost:3000/inventory
```

```
LIBP45P-33744DY:mongodb-lab-app n0039332$ curl -X GET http://localhost:3000/inventory
[{"_id":"5e38e9f53db11359af3407a8","item":"canvas","qty":100,"status":"stock","size":{"h":28,"w":35.5,"uom":"cm"},"tags":["cotton"]}, {"_id":"5e38e9f53db11359af3407a9","item":"journal","qty":25,"status":"backorder","size":{"h":14,"w":21,"uom":"cm"},"tags":["blank","red"]}, {"_id":"5e38e9f53db11359af3407aa","item":"notebook","qty":50,"status":"stock","size":{"h":8.5,"w":11,"uom":"in"},"tags":["red","blank","blue"]}, {"_id":"5e38e9f53db11359af3407ab","item":"paper","qty":100,"size":{"h":8.5,"w":11,"uom":"in"},"tags":["red","blank","plain"]}, {"_id":"5e38e9f53db11359af3407ac","item":"planner","qty":75,"status":"stock","size":{"h":22.85,"w":30,"uom":"cm"},"tags":["blank","red"]}, {"_id":"5e38e9f53db11359af3407ad","item":"postcard","qty":45,"status":"backorder","tags":["blue"]}
LIBP45P-33744DY:mongodb-lab-app n0039332$
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

- Now, let's add a GET endpoint that will allow us to retrieve a document based on the `_id` Object ID in MongoDB. This is the primary key for a document record. Copy the code in the code block below and paste it under the `//Add GET by id endpoint here` comment in the `server.js` file:

server.js

```
app.get("/inventory/:id", (request, response) => {
  collection.findOne({ "_id": new ObjectId(request.params.id) }, (error, result) => {
    if(error) {
      return response.status(500).send(error);
    }
    response.send(result);
  });
});
```

```
43 //Add GET by id endpoint here
44 app.get("/inventory/:id", (request, response) => {
45   collection.findOne({ "_id": new ObjectId(request.params.id) }, (error, result) => {
46     if(error) {
47       return response.status(500).send(error);
48     }
49     response.send(result);
50   });
51 });
```

- TEST:** Stop and Start your server in the left-side terminal window and make sure it is listening and connected. Copy the curl command in the code block below and paste into the right-side terminal window, but don't run it yet. We need to replace `<id>` with an actual ID from one of our documents in the inventory collection. You should still have the results from the last curl command visible. Copy an `_id` value from one of the documents returned. It should look something like this: `5e38e9f53db11359af3407a9`. Replace `<id>` in the code you copied in the terminal window with the `_id` you copied from you're a document in your result set. Then run the command. You should get back a single document that matches that ID:

curl

```
curl -X GET http://localhost:3000/inventory/<id>
```

```
LIBP45P-33744DY:mongodb-lab-app n0039332$ curl -X GET http://localhost:3000/inventory/5e38e9f53db11359af3407a8
{"_id":"5e38e9f53db11359af3407a8","item":"canvas","qty":100,"status":"stock","size":{"h":28,"w":35.5,"uom":
LIBP45P-33744DY:mongodb-lab-app n0039332$
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

- Finally, let's create a POST endpoint to add a document to the inventory collection. Copy the code in the code block below and paste it under the **//Add POST endpoint here** comment in the **server.js** file:

```
server.js
app.post("/inventory", (request, response) => {
  collection.insertOne(request.body, (error, result) => {
    if(error) {
      return response.status(500).send(error);
    }
    response.send(result.result);
  });
});
```

```
53 //Add POST endpoint here
54 app.post("/inventory", (request, response) => {
55   collection.insertOne(request.body, (error, result) => {
56     if(error) {
57       return response.status(500).send(error);
58     }
59     response.send(result.result);
60   });
61 });
```

- TEST:** Stop and Start your server in the left-side terminal window and make sure it is listening and connected.
Copy the curl command in the code block below and paste into the right-side terminal and run it. If successful, this will add a pen item to the inventory collection. You can validate the add was successful by looking at the collection in MongoDB Compass Community or by running the first curl command above to get a list of all the documents in the inventory collection:

```
curl
curl -X POST \
  -H 'content-type:application/json' \
  -d '{" item":"pen","qty":10,"status":"A","tags":["black","blue","red"]}' \
  http://localhost:3000/inventory
```

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

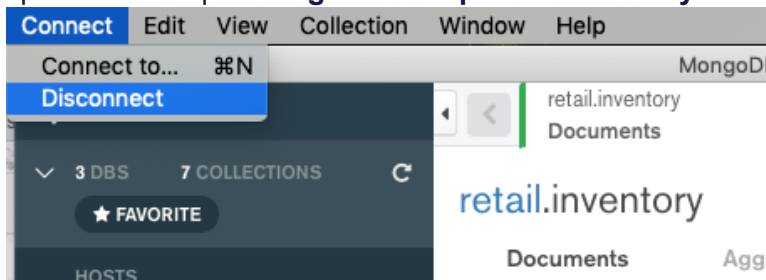
MongoDB Atlas Workshop – Presentation & Lab

```
LIBP45P-33744DY:mongodb-lab-app n0039332$ curl -X POST \
> -H 'content-type:application/json' \
> -d '{" item":"pen","qty":10,"status":"A","tags":["black","blue","red"]}' \
> http://localhost:3000/inventory
{"n":1,"opTime":{"ts":"6790194388459847681","t":3},"electionId":"7fffffff00000000
":{"clusterTime":"6790194388459847681","signature":{"hash":"fz5o7Q+MQVcM8b5TfmzfS
```

```
> {
  _id: ObjectId("5e3b9dd2495cd2524928ecb3")
  item: "pen"
  qty: 10
  status: "A"
  tags: Array
    0: "black"
    1: "blue"
    2: "red"
}
```

Notice we didn't include the embedded **size** document when we added the pen item to our **inventory**. This is okay and one of the great things about using a NoSQL database. Each document record can have different data, as needed.

8. Stop your server in the left-side terminal window and exit or quit **Visual Studio Code**.
9. Disconnect **MongoDB Compass Community** using the **Connect / Disconnect** menu option. Exit or quit **MongoDB Compass Community**.



For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.



Hack(H)er413 2020

MongoDB Atlas Workshop – Presentation & Lab

Conclusion

Congratulations!! You can now work with a MongoDB database. We encourage you to visit the Resources below to continue learning about MongoDB.

Resources

Some additional resources for you to consider:

1. MongoDB Docs: <https://docs.mongodb.com/>
 - Getting Started: <https://docs.mongodb.com/manual/tutorial/getting-started/>
 - MongoDB CRUD Operations: <https://docs.mongodb.com/manual/crud/>
 - The mongo Shell: <https://docs.mongodb.com/manual/mongo/>
2. MongoDB Tutorials from Tutorials Point: <https://www.tutorialspoint.com/mongodb/index.htm>
3. Mongo DB Compass: <https://www.mongodb.com/products/compass>
4. JSON: <https://www.json.org/json-en.html>
5. Node.js: <https://nodejs.org/en>
6. Express: <https://expressjs.com/>

For more information please visit:

lmi.co/undergraduate

   /WorkAtLiberty  /Liberty Mutual Insurance

Liberty Mutual Insurance is an equal opportunity employer. We acknowledge, welcome, and accept the differences that each employee brings to the workplace.

